17-20 sept 2013

MULTICONFERENCIA
CAEPIA'13

Madrid

- XV Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA)
- IX Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB)
- IV Simposio sobre Lógica Fuzzy y Soft Computing (LFSC)
- VI Simposio de Teoría y Aplicaciones de Minería de Datos (TAMIDA)
- Fusión de la Información y Ensembles (FINO)
- Agentes y Sistemas Multi-Agente: de la Teoría a la Práctica (ASMas)
- Key Works
- Doctoral Consortium

Editores del volumen

Amparo Alonso-Betanzos	Presidente del Comité Organizador de la Multi-Conferencia
Concha Bielza	Presidente del Comité de Programa de CAEPIA
Antonio Salmerón	Co-Presidente del Comité de Programa de CAEPIA
Abraham Duarte	Co-Presidente del Comité de Programa de MAEB
J. Ignacio Hidalgo	Co-Presidente del Comité de Programa de MAEB
Luis Martínez	Presidente del Comité de Programa de LFSC
Edurne Barrenechea	Co-Presidente del Comité de Programa de LFSC
Alicia Troncoso	Presidente del Comité de Programa de TAMIDA
Emilio Corchado	Presidente del Comité de Programa de FINO
Juan M. Corchado	Presidente del Comité de Programa de ASMas
Francisco Herrera	Co-Presidente de los Comités de Programa de Doctoral Consortium y Key Works
José C. Riquelme	Co-Presidente de los Comités de Programa de Doctoral Consortium y Key Works

ISBN: 978-84-695-8348-7

Prefacio

Este volumen contiene los artículos que fueron seleccionados para su presentación en la XV Multiconferencia CAEPIA (Conferencia de la Asociación Española para la Inteligencia Artificial) 2013, celebrada en Madrid del 17 al 20 de Septiembre de 2013. Esta Multiconferencia se celebra dentro del Congreso Español de Informática (CEDI 2013), y en ella CAEPIA se ha coordinado con otros congresos con entidad propia, como son: IX Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB 2013), IV Simposio sobre Lógica Fuzzy y Soft Computing (LFSC), VI Simposio de Teoría y Aplicaciones de Minería de Datos (TAMIDA 2013), Fusión de la INfOrmación (FINO 2013) y Agentes y Sistemas Multiagente: De la Teoría a la Práctica (ASMas). Este volumen contiene los 15 artículos seleccionados por CAEPIA, los 78 seleccionados por MAEB, los 39 seleccionados por LFSC (incluyendo en estos dos últimos congresos los artículos de sesiones especiales), los 16 seleccionados por TAMIDA, los 3 de FINO y los 4 de ASMas. El objetivo de todas estas conferencias es proporcionar a los investigadores en Inteligencia Artificial un foro en el que intercambiar ideas y opiniones, y avanzar en la construcción de una comunidad de Inteligencia Artificial en España amplia, plural y abierta.

Con el fin de promover la participación de estudiantes de doctorado en la Multiconferencia, y su interacción con investigadores senior de los distintos campos involucrados, se realizó una sección de la Multiconferencia denominada Doctoral Consortium, transversal a todas las conferencias participantes. Esta actividad resultó particularmente exitosa, recibiendo un importante número de contribuciones (29). Los trabajos predoctorales presentados serán valorados por un Comité de 10 expertos, que junto con los dos Presidentes del Comité de Programa, aportarán sus opiniones, y concederán 6 becas de inscripción en la Multiconferencia CAEPIA 2013. Los tres proyectos de tesis mejor valorados recibirán además un certificado con dicho reconocimiento. El proyecto de tesis mejor valorado recibirá el premio GENIL del Campus de Excelencia BioTICs de la Universidad de Granada, el segundo proyecto mejor valorado recibirá un premio concedido por la Asociación Española para la Inteligencia Artificial (AEPIA), que asimismo otorga un diploma acreditativo al tercer proyecto de tesis mejor valorado.

También de forma transversal, la Multiconferencia incluye por primera vez una sesión de trabajos publicados recientemente en revistas y foros de reconocido prestigio, que se denomina Key Works, y consta de 14 trabajos recientes publicados durante el período 2011-2013. Estos trabajos, seleccionados por un Comité formado por 3 expertos, se presentarán en varias sesiones, organizadas en varias temáticas concordantes con las áreas específicas de trabajo de las conferencias integrantes de la Multiconferencia.

Los editores de este volumen desean manifestar su agradecimiento a todos aquellos que contribuyeron al éxito de la Multi-Conferencia CAEPIA 2013, en especial a los organizadores de todos los congresos que integran la Multiconfe-

rencia, así como los organizadores de las sesiones de Doctoral Consortium y Key Works por su colaboración y disponibilidad; a los miembros de los Comités Científicos y los revisores adicionales, los autores de los trabajos, y los conferenciantes invitados de todas las conferencias federadas.

Nada hubiera sido igual sin la ayuda de los comités Organizadores de las distintas conferencias, (incluido por supuesto el Comité Organizador del CEDI), de la Universidad Complutense de Madrid, Universidad Politécnica de Madrid , el Campus de Excelencia BioTICs de la Universidad de Granada y la Asociación Española de Inteligencia Artificial (AEPIA).

Por último una mención especial a Diego Fernández Francos, que compiló y ordenó todos los artículos de este volumen.

Amparo Alonso-Betanzos

Índice general

CAEPIA

Prefacio 3

Organización 6

A Dynamic Bayesian Network for Breast Boundary Extraction 9
Hamed Habibi Aghdam, Domenec Puig, Jaime Melendez, and Agusti Solanas

CTC: Competitive in an Analysis of Genetic based Algorithms for Rule Induction in Imbalanced Datasets 19
Olatz Arbelaitz, Ibai Gurrutxaga, Javier Infante, Javier Muguerza, and Jesús M. Pérez

Clustering based navigation profiling in a tourism website using Cluster Validity Indices to select the best partition 29
Olatz Arbelaitz, Ibai Gurrutxaga, Aizea Lojo, Javier Muguerza, Jesús M. Pérez, and Iñigo Perona

Distributed Column Formation in Swarm Robotics for Simple Agents 39
Pilar Arques, Fidel Aznar, and Mireia Sempere

Aplicación del Algoritmo SPEA2 en el Problema de Gestión de Áreas de Localización en Redes Móviles 49
Víctor Berrocal-Plaza, Miguel A. Vega-Rodríguez, and Juan M. Sánchez-Pérez

Application of Bayesian Neural Networks for Indoor Temperature Time Series Forecasting 59
Paloma Botella-Rocamora, Pablo Romeu, Francisco Zamora-Martínez, and Juan Pardo

Content-based Recommendation: Experimentation and Evaluation in a Case Study 69
Ángel Castellanos, Juan M. Cigarrán, and Ana García-Serrano

Estudio de la Carga Cognitiva en Interfaces Web de Sistemas de Búsqueda Colaborativa 79
Sergio Cleger, Juan M. Fernández-Luna, Humberto Rodríguez-Ávila, and Julio C. Rodríguez-Cano

Integrating a Conversational Recommender System within a 3D Collaborative Environment 89
David Contreras, Maria Salamó, Anna Puig, and Inmaculada Rodríguez

Using Apriori Algorithm + Standard Deviation to Improve the Scalability of TSK-0 Rule Learning Algorithms 99
Javier Cózar, Luis delaOssa, and José A. Gámez

Aplicando Métodos de Aprendizaje Sensible al Coste para Mejorar Problemas de *Big Data* Extremadamente Desbalanceados Usando *Random Forest* 109
Sara del Río, Victoria López, José Manuel Benítez, and Francisco Herrera

Policy Reuse in a General Learning Framework 119
Fernando Martínez-Plumed, Cèsar Ferri, José Hernández-Orallo, and M. José Ramírez-Quintana

Improving the Scalability of a Distributed Learning Algorithm Based on Frontier Vector Quantization and Information Theory 129
Diego Peteiro-Barral and Bertha Guijarro-Berdiñas

PQ-learning: Aprendizaje por refuerzo multiobjetivo 139
Manuela Ruiz-Montiel, Lawrence Mandow, and José Luis Pérez-de-la-Cruz

Tri-Training con sobremuestreo para aprendizaje semi-supervisado 149
Isaac Triguero, Salvador García, and Francisco Herrera

MAEB

Prefacio 161

Organización 163

Obituario 166

I Adaptive Memory Programming

GRASP con Path Relinking para el Problema del Orienteering 169
Vicente Campos and Rafael Martí

A Tabu Search Approach for a Territory Design Problem with Stochastic Demands 179
Mónica Elizondo-Amaya, Roger Z. Ríos-Mercado, David P. Morton, and Erhan Kutanoglu

GRASP + *Path Relinking* aplicado al problema del *Cyclic Bandwidth* ... 189
Alfonso Fernández and David Candelas

Un Algoritmo Voraz Iterativo con Memoria de Destrucciones Tabú para el Problema de las Múltiples Mochilas Cuadráticas 199
Carlos García-Martínez, Francisco J. Rodriguez, and Manuel Lozano

II Algoritmos Evolutivos y Creatividad

Foto-renderización con Inteligencia de Enjambre 209
Carlos M. Fernandes, Antonio M. Mora, Juan Julián Merelo, and Agostinho C. Rosa

EvoSpace-Interactivo: Una herramienta para el Arte y Diseño Interactivo y Colaborativo 219
Francisco Fernández de Vega, Mario García-Valdez, Leonardo Trujillo, Juan Julián Merelo, and Gustavo Olague

Evolutionary Elaboration of Daily News as a Poetic Stanza 229
Pablo Gervás

III Algoritmos Multiobjetivo

Software *Auto-tuning* con Metaheurísticas Multi-Objetivo 239
Juan J. Durillo and Thomas Fahringer

Algoritmos evolutivos multi-objetivo para el diseño de plantas fotovoltaicas con seguimiento solar 249
Daniel Gómez-Lorente, Isaac Triguero, Consolación Gil, and Antonio Espín-Estrella

Problemática γ de Ordenación con Algoritmos Evolutivos Multi-Objetivo y Programación Compromiso 259
Máximo Méndez, Blas Galván, Mariano Frutos, and Antonio Luis Álamo

Un nuevo Algoritmo Evolutivo de Optimización Multiobjetivo basado en Preferencias: WASF-GA 268
Rubén Saborido, Ana Belén Ruiz, and Mariano Luque

IV Algoritmos Paralelos

Paralelización de nuevas meta-heurísticas para resolver problemas de enrutado de vehículos 278
Raúl Baños, Julio Ortega, and Consolación Gil

Programación Automática con Colonias de Hormigas Multi-Objetivo en GPUs 288
Alberto Cano, Juan Luis Olmo, and Sebastián Ventura

Un Modelo Formal para Analizar Propiedades de Algoritmos Genéticos Paralelos 298
Natalia López, Pablo Rabanal, Ismael Rodríguez, and Fernando Rubio

Experimentación con algoritmos distribuidos usando herramientas libres y gratuitas 308
Juan Julián Merelo, Maribel García Arenas, and Pedro A. Castillo

Relación entre la Estructura de un Problema y la Topología de Islas en Algoritmos Evolutivos Paralelos 318
Ignacio Arnaldo, Iván Contreras, David Millán-Ruiz, Javier Nobel, J. Ignacio Hidalgo, and Natalio Krasnogor

Obtención de los parámetros de un modelo de redes mediante un AGP con intercambio de individuos en Dropbox 328
J. Gabriel García-Caro, Javier Villanueva-Oller, J. Ignacio Hidalgo, and Rafael J. Villanueva

Mejorando el Comportamiento de las Metaheurísticas Cooperativas basadas en Trayectoria mediante Path Relinking 338
Gabriel Luque and Enrique Alba

Búsqueda dispersa cooperativa para problemas a gran escala en el contexto de biología de sistemas 348
Alejandro F. Villaverde, Jose A. Egea, and Julio R. Banga

V Búsqueda en Vecindad Variable

Búsqueda de Vecindad Variable secuencial y paralela: una aplicación al problema de la maximización del corte 358
David Concha, David Fernández-López, Jesús Sánchez-Oro, Antonio S. Montemayor, Juan José Pantrigo, and Abraham Duarte

Diversificación de la VNS para el problema del árbol generador de etiquetado mínimo 368
Sergio Consoli, Nenad Mladenovic, and José A. Moreno-Pérez

Cálculo de separadores de grafos utilizando búsqueda de vecindad variable reducida 378
David Fernández-López, Jesús Sánchez-Oro, and Abraham Duarte

Búsqueda de Formulación Variable aplicada al problema de la Minimización de la Anchura de Corte en ordenaciones lineales 388
Eduardo G. Pardo, Juan José Pantrigo, Abraham Duarte, and Nenad Mladenovic

VI Metaheurísticas Aplicadas a la Gestión de la Cadena de Suministro en Entornos Realistas

Algoritmo de Búsqueda Tabú para el HVRP con ventanas de tiempo y criterios de sostenibilidad 398
Ignacio Eguía, José Carlos Molina, Jesús Racero, and Fernando Guerrero

Local search method for a real case assembly problem with single machines on production stage 408
Sara Hatami and Carlos Andrés-Romano

Variable Neighborhood Search with Memory: Application to a Routing Problem with Flexibility in Delivery Dates 414
Joaquín Pacheco, Irma García, and Ada Álvarez

A Multi Ant Colony Optimization algorithm for a Mixed Car Assembly Line 424
Raúl Pulido, Álvaro García-Sánchez, Francisco J. Diego, and Carlos Andres-Romano

VII Metaheurísticas en Empresas y Producción

Pallet building and truck loading strategies for an inter-depot transportation problem 433
Maria Teresa Alonso, Ramon Álvarez-Valdés, Joaquim Gromicho, Francisco Parreño, and Gerhard Post

Algoritmos GRASP para solucionar el problema *Blocking Flow Shop* 443
Joaquín Bautista, Alberto Cano, Rocío Alfaro, and Cristina Batalla

A multiobjective genetic algorithm and robustness information function for time and space assembly line balancing 453
Manuel Chica, Oscar Cordón, Sergio Damas, and Joaquín Bautista

MOEAs para la Secuenciación de Operaciones con Restricciones de Disponibilidad de Máquinas y Capacidad del Buffer 463
Mariano Frutos, Máximo Méndez Babey, Fernando Tohmé, Diego Broz

VIII Metaheurísticas en Logística Portuaria y Problemas Afines

Híbrido basado en VNS para la Planificación Real de Rutas de Vehículos con Ventanas de Tiempo 473
Jesica de Armas, Belén Melián-Batista, Julio Brito, and José A. Moreno-Pérez

PSO Discreto para el VRPTW orientado a la calidad del servicio 483
Airam Expósito, Julio Brito, and José A. Moreno-Pérez

Un Algoritmo de Estimación de Distribución para el Problema de Planificación de Grúas en una Terminal Marítima de Contenedores 493
Christopher Expósito-Izquierdo, Belén Melián-Batista, and J. Marcos Moreno-Vega

Estudio de la Planificación de Atraque en una Terminal de Contenedores . 503
Eduardo Lalla-Ruiz, Belén Melián-Batista, and J. Marcos Moreno-Vega

MOAMP-Tabu Search and NSGA-II for a Bi-objective Urban Waste Collection Problem: A Head to Head Comparison 513
Joaquín Pacheco, José Rubén Gómez, and Hernán Gonzalo-Orden

IX Metaheurísticas en Routing & Scheduling

Aplicación de retraso en tierra y secuenciación de vuelos para la gestión de incidencias en aterrizaje 523
Alba Agustín, Antonio Alonso-Ayuso, Angel Juan, and Javier Faulin

Hybrid Approach combining Insertion Heuristic and Biased Random Sampling for the Vehicle Routing Problem with Time Windows 533
José Cáceres-Cruz, Daniel Riera, Angel Juan, and Silvia Padrón

Un Algoritmo Heurístico para el Uncapacitated r-Allocation p-Hub Median Problem 541
Juanjo Peiró, Ángel Corberán, and Rafael Martí

Meta-Heuristic Algorithm based on Particular Swarm Optimization (PSO) Algorithm and Project Scheduling Problem (PSP) for the Traveling Salesman Problem 547
Alejandro Fuentes-Penna, Jorge A. Ruiz-Vanoye, Ocotlán Diaz-Parra, Joel A. Domínguez-Narvaez, and Guillermo A. Alonso-López

Resolución del problema EDP para Grafos No Dirigidos utilizando algoritmos MSGA y GRASP 557
Bernardo Martín, Abraham Duarte, and Ángel Sánchez

Sistema de Autoadaptación de los Parámetros del Algoritmo Clearing para Problemas Combinatorios: Aplicación al *job shop* 567
Elena Pérez, Marta Posada, and Francisco Herrera

A simple heuristic approach for generating real-time solutions to real-life crew pairing problems 577
Jesus M. Puentes, Christine A. Oketch, and Angel A. Juan

X Metaheurísticas, Algoritmos Evolutivos y Bioinspirados en Bioinformática

Biclustering Evolutivo mediante Identificación y Explotación de Regiones Prometedoras 584
Fco. Javier Gil-Cumbreras, Raúl Giráldez, Federico Divina, and Jesús S. Aguilar-Ruiz

Optimización basada en Malla Variable para funciones multimodales 594
Daniel Molina, Amilkar Puris, Rafael Bello, and Francisco Herrera

Detección de puntos de comienzo y fin de la traducción génica mediante la combinación evolutiva de modelos basados en máquinas de vectores soporte 604
Javier Pérez-Rodríguez and Nicolás García-Pedrajas

Una Aproximación Multiobjetivo Basada en el Comportamiento de las Luciérnagas para la Inferencia Filogenética 614
Sergio Santander-Jiménez and Miguel A. Vega-Rodríguez

XI Metodologías y Herramientas Software para la Investigación sobre Metaheurísticas

Adaptando algoritmos evolutivos paralelos al lenguaje funcional Erlang . . 624
José Albert-Cruz, Liesner Acevedo-Martínez, Pedro A. Castillo, Maribel García Arenas, Juan Julián Merelo

Desarrollo de servicios para una Arquitectura Orientada a Servicios para Algoritmos Evolutivos 633
Pablo García-Sánchez, Maribel García Arenas, Antonio M. Mora, Pedro A. Castillo, Carlos Fernandes, Jesús González, Gustavo Romero, Juan Julián Merelo, and Paloma de Las Cuevas

VisualJCLEC: un entorno de trabajo gráfico para computación evolutiva . 643
Juan Ignacio Jaén, José Raúl Romero, and Sebastián Ventura

Achieving Replicability: Is there life for our experiments after publication? 653
Jose Antonio Parejo, Sergio Segura, and Antonio Ruiz-Cortés

XII Problemas Dinámicos de Optimización

Problema Dinámico de Rutas de Vehículos con Ventanas de Tiempo 663
Alondra De Santiago, Belén Melián-Batista, and Ada Álvarez

Aprendizaje en problemas de optimización dinámicos: un análisis sobre el algoritmo AHMA 673
Jenny Fajardo, Antonio D. Masegosa, Alejandro Rosete, and David A. Pelta

Algoritmo Genético Empotrado en Dispositivos Móviles para Control de Conectividad en Redes Inalámbricas Heterogéneas 683
María D. Jaraíz-Simon, Juan A. Gómez-Pulido, and Miguel A. Vega-Rodríguez

Sobre el rol de la auto-adaptación en ambientes dinámicos 693
Pavel Novoa-Hernández, Carlos Cruz, and David A. Pelta

XIII Sesión General

Automatic Design of Electronic Amplifiers using Grammatical Evolution . 703
Federico Castejón and Enrique J. Carmona

Auto-Ajuste del Foco de Atención para la Mejora de un Sistema de Detección de Punto Láser 713
Francisco Chávez, Eddie Clemente, León Dozal, Francisco Fernández de Vega, and Gustavo Olague

Coevolución para el Diseño Automático de Máquinas de Estados Finitos . 723
J. Manuel Colmenar, Alfredo Cuesta-Infante, José L. Risco-Martín, and J. Ignacio Hidalgo

Sumador evolutivo de 2 bits tolerante a fallos 733
Juan Lanchares, Oscar Garnica, J. Ignacio Hidalgo, and José L. Risco-Martín

Un Nuevo Enfoque para el Posicionamiento de Routers Sobre Redes de Sensores Inalámbricos 743
Jose M. Lanza-Gutierrez, Juan A. Gomez-Pulido, and Miguel A. Vega-Rodríguez

Extracción de reglas de asociación frecuentes en bases de datos relacionales 753
J. M. Luna, José Raúl Romero, Cristóbal Romero, and Sebastián Ventura

Aplicación de Algoritmos Meméticos Híbridos de Doble Fase en la Optimización de Tableros de Puentes de Vigas Artesa de Hormigones No Convencionales 763
José Vicente Martí, Víctor Yepes, and Tatiana García

Una comparación del método Tabu y GRASP para el Problema de Clasificación con Capacidades 773
Anna Martínez-Gavara, Vicente Campos, Micael Gallego, and Rafael Martí

Predicción de tráfico mediante co-evolución de Redes Neuronales de Funciones de Base Radial y selección de variables de entrada 782
Victor Manuel Rivas, Elisabeth Parras-Gutierrez, María G. Arenas, Pedro A. Castillo, Pablo García Sánchez, Juan Julián Merelo, and Pedro García-Fernández

XIV Finalistas del premio GENIL al mejor trabajo de un estudiante de doctorado

Sistemas de trading en la recesión económica: gramáticas evolutivas, relación fitness-objetivos, y sectores empresariales 792
Iván Contreras, J. Ignacio Hidalgo, Laura Núñez-Letamendia, and F. Javier Diego

Aprendizaje de reglas difusas TSK de orden 1 mediante búsqueda local .. 802
Javier Cózar, Luis delaOssa, and José A. Gámez

Validación Inteligente para la Sincronización de Semáforos Basada en Feature Models 812
Javier Ferrer, José García-Nieto, Enrique Alba, and Francisco Chicano

Generador de Instancias de Problemas de Optimización Combinatoria Basados en Permutaciones 822
Leticia Hernando, Alexander Mendiburu, and Jose A. Lozano

Estudiando la Creatividad mediante Algoritmos Evolutivos Desconectados 832
Lilian Navarro, Francisco Fernández de Vega, Cayetano Cruz, and Francisco Chávez

Scatter Search aplicado aplicada al problema de la minimización del *profile* de matrices y grafos 842
Jesús Sánchez-Oro, Abraham Duarte, Manuel Laguna, and Rafael Martí

XV Finalistas del premio al mejor trabajo metodológico/aplicado

Optimización del Problema de Planificación de Áreas de Localización con el Algoritmo NSGAII 852
Víctor Berrocal-Plaza, Miguel A. Vega-Rodríguez, and Juan M. Sánchez-Pérez

Metaheurísticas para Smart Mobility: Reducción de Emisiones y Consumo de Carburantes en el Tráfico Urbano 862
José García-Nieto and Enrique Alba

Estudio comparativo de distintos métodos de umbral en regresión ordinal 872
Pedro Antonio Gutiérrez, María Pérez-Ortiz, Javier Sánchez-Monedero, and César Hervás-Martínez

MOPNAR: Algoritmo Evolutivo Multi-Objetivo para Extraer Reglas de Asociación Cuantitativas Positivas y Negativas 882
Diana Martín, Alejandro Rosete, Jesús Alcalá-Fdez, and Francisco Herrera

Algoritmo de programación evolutiva para identificación de arquitecturas software 892
Aurora Ramírez, José Raúl Romero, and Sebastián Ventura

Un algoritmo voraz iterativo para el problema de la planificación sobre máquinas paralelas no relacionadas 902
Francisco J. Rodríguez, Carlos García-Martínez, Christian Blum, and Manuel Lozano

LFSC

Prefacio 915

Organización 916

I Aplicaciones del Soft Computing

BioSeq: Una librería R para el análisis de secuencias de datos 919
Jorge Cordero, Jorge Martínez, Oscar Sánchez, Victoria López, and Beatriz González

Modelado de Ludus Manager con Lógica Fuzzy 929
Antonio Dominguez, Juan Íñigo Monjas, Miguel Morata, Victoria López, and Rubén Fuentes-Fernández

Método para la Detección de Intrusos basado en la Sinergia de Técnicas de Inteligencia Artificial 939
César Guevara, Matilde Santos, and José Antonio Martín H

Diseño de un controlador borroso para el sistema de exposición de una cámara de fotos semi-automática 949
Jorge Martínez and Matilde Santos

Sistema fuzzy de decisión para la gestión de capital en inversión en bolsa . 959
Rodrigo Naranjo, Javier Arroyo, Albert Meco, and Matilde Santos

Sistema de recomendación colaborativo de la aplicación Hábitat Madrid . 969
Inmaculada Pardines, Victoria López, Antonio Sanmartin, Mar Octavio de Toledo, and Carlos Fernández

Aplicación de la Soft Computing al Problema de Ruteo de Camiones y Remolques 979
Isis Torres, Alejandro Rosete, Carlos Alberto Cruz, and José Luis Verdegay

II Aplicaciones del Soft Computing en la Ingeniería y el Control

Control Inteligente de un Robot Manipulador de Tres Grados de Libertad 989
Ildefonso Ayala de Diego and Matilde Santos

Utilización de Técnicas de Soft Computing para el Control de un Robot Móvil Autónomo de 2 Ruedas 999
Joseba Menoyo and Matilde Santos

ANMV: Augmented Navigation and Virtual Mapping 1009
Ricardo Pragnell, Miguel Gutierrez, Victor Ortiz, Alejandro Peñalver, and Victoria López

Sistema de Control Difuso con Reglas Sintonizadas mediante Algoritmos Genéticos para un Proceso de Fermentación Alcohólica 1018
Rafael Rendón and Matilde Santos

III Lógica Difusa y Procesamiento de Imagen

Estudio preliminar sobre medidas de similitud para la detección de puntos singulares en imágenes de huellas dactilares 1028
Juan Cerrón, Mikel Galar, Carlos Lopez-Molina, José Sanz, and Miguel Pagola

Fuzzy Sets for Texture Contrast Modelling Applied to Pattern Recognition1038
Jesús Chamorro-Martínez, Pedro Martínez-Jiménez, and J.M. Soto-Hidalgo

The WAT Algorithm, an Algorithm to see both Woods and Trees 1048
Gabriel Mattioli

Medidas de similitud difusas para comparar imágenes segmentadas en color 1058
Daniel Paternain, Mikel Galar, Aranzazu Jurio, and Edurne Barrenechea

IV Métodos Borrosos para Clasificación y Toma de Decisiones

Fuzzy Delphi method and fuzzy Spearman´s rank correlation coefficient . 1068
Raquel Caro, Luis Garmendia, and Ramón González del Campo

Recursive Measures of Specificity 1077
Luis Garmendia, Ramón González del Campo, and Ronald R. Yager

Clasificación borrosa basada en disimilitud para la valoración inicial de desastres 1085
J. Tinguaro Rodríguez, Begoña Vitoriano, and Javier Montero

Análisis Morfológico difuso lingüístico en el ámbito de la Planificación de Escenarios 1095
Pablo J. Villacorta, Antonio D. Masegosa, and María T. Lamata

Un Algoritmo Genético para aprendizaje de Sistemas de Clasificación Basados en Reglas Difusas dentro del paradigma *one-class* 1105
Pedro Villar, Ana María Sánchez, Rosana Montes, and Francisco Herrera

A Hierarchical Image Segmentation Algorithm 1115
Edwin Zarrazola, Javier Yáñez, Daniel Gómez, and Javier Montero

V Funciones de Agregación y Conectivos Lógicos

Interval-Valued Overlap Functions 1125
Benjamin Bedregal, Humberto Bustince, Javier Fernandez, Aranzazu Jurio, María José Asiain, and Laura De Miguel

A measure of k-specificity of fuzzy sets and the use of maximum k-specificity 1134
Ramón González del Campo, Luis Garmendia, and Ronald R. Yager

Constructing (multi) distances from t-norms 1144
Javier Martín and Gaspar Mayor

Sobre uninormas y nulnormas (α, T)-migrativas 1153
M. Mas, M. Monserrat, D. Ruiz-Aguilera, and J. Torrens

Caracterización de las (h, e)-implicaciones 1163
S. Massanet and J. Torrens

VI Soft Computing y Recuperación de Información en Internet

Una Aplicación de la Similitud Semántica en la mejora de la selección de Información Clínica 1173
Israel Alonso, Francisco P. Romero, David Contreras, and Jose A. Olivas

Evaluating the Library 2.0 Services via Computing with Words 1182
Francisco J. Cabrerizo, Rubén Heradio, David Fernández-Amorós, Manuel Herrera, and Enrique Herrera-Viedma

Sistema Borroso de Clasificación de Notas Clínicas de Texto Libre en base a Calidad de Datos 1192
Francisco P. Romero, Mauro J. Ourezabal, Jose A. Olivas, Jesús Serrano-Guerrero, and Ismael Caballero

Uso de OpenCyc y un Corpus de Ontologías para el Aprendizaje de Ontologías a partir de Mapas Conceptuales 1200
Amhed Suárez-Rodríguez, Alfredo Simón-Cuevas, and José A. Olivas

Una Aplicación de los Prototipos Deformables Borrosos para predecir el rendimiento académico de nuevos estudiantes a partir de la información generada por Sistemas Tutores 1210
Mª Rosario Vázquez, Francisco P. Romero, and Jose A. Olivas

VII Track General

Un sistema difuso evolutivo para la detección de excepciones en descubrimiento de subgrupos 1220
Cristobal J. Carmona, Pedro González, and M. Jose Del Jesus

Modelo Lingüístico de Evaluación de QoS de Red basado en Distancias .. 1230
M. Espinilla and S. Gramajo

A Fuzzy Approach to the Weighted Overlap Dominance Model 1240
Camilo Franco, Jens Leth Hougaard, and Kurt Nielsen

Nuevas Operaciones Binarias sobre los Conjuntos Borrosos de Tipo 2 1250
Pablo Hernández, Susana Cubillo, and Carmen Torres

Sistema de Apoyo al Consenso para Problemas de Toma de Decisión en Grandes Grupos basado en una Plataforma Multi-Agente 1260
Iván Palomares, Francisco J. Quesada, and Luis Martínez

Estudio comparativo de modelos de un vehículo aéreo obtenidos mediante técnicas analíticas y basadas en redes neuronales 1270
J. Enrique Sierra and Matilde Santos

Sobre las interacciones: distancias e integrales 1280
Vicenç Torra

A new framework in R to support consensus processes in group decision making 1286
Raquel Ureña and Enrique Herrera-Viedma

TAMIDA

Prefacio 1299

Organización 1301

Regla k_M-vecinos más cercanos en Bases de Datos de Baja Calidad 1303
José M. Cadenas, M. Carmen Garrido, Raquel Martínez, and Alejandro Martínez

Análisis del virus de la gripe A mediante descubrimientos de subgrupos difusos 1313
C. J. Carmona, C. Chrysostomou, H. Seker, and M. Jose Del Jesus

Minería de datos para la predicción de retenciones de tráfico 1323
Oscar Castaño, Francisco Martínez-Álvarez, Alicia Troncoso, and José C. Riquelme

Una debilidad de la estrategia Uno-contra-Uno en clasificación: Potenciando las clases difíciles 1333
Mikel Galar, Alberto Fernández, Edurne Barrenechea, and Francisco Herrera

Asignación de órganos en trasplante hepático mediante regresión ordinal . 1343
César Hervás-Martínez, María Pérez-Ortiz, Manuel Cruz-Ramírez, and Pedro Antonio Gutiérrez

EDM para la detección precoz del fracaso escolar en secundaria 1353
M. Jiménez, J. M. Luna, and S. Ventura

Optimización evolutiva de la votación para los k vecinos más cercanos . . . 1363
Daniel Mateos-García, José C. Riquelme, and Jorge García-Gutiérrez

Análisis de la Expresión Génica en Tratamientos con Vitamina D mediante Biclustering 1373
Sara Martín-Villanueva, Raúl Giráldez, and Beatriz Pontes

Accelerating learning by reusing abstract policies in gErl 1383
Fernando Martínez-Plumed, Cèsar Ferri, José Hernández-Orallo, and María José Ramírez-Quintana

A methodology to improve single-experiment k-fold Cross-Validation by reducing the impact of related dataset shift 1393
Jose G. Moreno-Torres, José A. Sáez, and Francisco Herrera

Time Series Database Characterization and Similarity Measure Selection . 1403
Usue Mori, Alexander Mendiburu, and Jose A. Lozano

Large scale part-to-part video matching by a likelihood function using featured based video representation 1413
Silvia Espinosa, Joaquín Ordieres-Meré, and Antonio Bello-García

Selección de características en la predicción a medio plazo del precio del aceite de oliva virgen extra 1423
Antonio J. Rivera, Pedro Pérez-Recuerda, M. Dolores Pérez-Godoy, M. Jose Del Jesus, Pilar Frías, and Manuel Parras

Improving Ensemble Methods for Imbalanced Datasets with Disturbing Neighbors 1433
Juan J. Rodríguez, José F. Díez-Pastor, Jesús Maudes, and César García-Osorio

Meta-aprendizaje para la recomendación de algoritmos de clasificación con conjuntos de datos educacionales 1444
Cristóbal Romero, Juan Luis Olmo, and Sebastián Ventura

Clustering and Interpretation on Real Nutritional Data 1454
Beatriz Sevilla Villanueva, Karina Gibert, and Miquel Sànchez-Marrè

FINO

Prefacio 1467

Organización 1468

Selección dinámica de clasificadores para la estrategia Uno-contra-Uno: Evitando los clasificadores no competentes 1469
Mikel Galar, Alberto Fernández, Edurne Barrenechea, Humberto Bustince, and Francisco Herrera

Analyzing the structure of ensembles based-on evolving classifiers 1479
José Antonio Iglesias, Agapito Ledezma, and Araceli Sanchis

Dental milling process optimization by the use of ensemble classification . 1489
Santiago Porras, Hector Quintian, Bruno Baruque, Beatriz Hernando, Ignacio Aliaga, Vicente Vera, and Emilio Corchado

ASMas

Prefacio 1501

Organización 1502

Service-oriented Multi-Agent Systems for Ubiquitous Domains 1503
Jorge Agüero, Miguel Rebollo, Carlos Carrascosa, and Vicente Julián

A multi-agent architecture to combine heterogeneous inputs in multimodal interaction systems 1513
David Griol, Jesús García-Herrero, and José Manuel Molina

Asignación de roles automática en organizaciones virtuales de agentes . . . 1523
Gabriel Villarrubia, Javier Bajo, Juan F. De Paz, Carolina Zato, and Juan M. Corchado

An adaptive multi-agent architecture for workplace integration 1533
Carolina Zato, Fernando De la Prieta, Sara Rodríguez, Yves Demazeau, and Juan M. Corchado

Key Works

Prefacio 1545

Organización 1546

La búsqueda de la distribución de clases óptima: una aproximación para mejorar la efectividad del aprendizaje mediante métodos de remuestreo para bases de datos desbalanceadas 1547
Iñaki Albisua, Olatz Arbelaitz, Ibai Gurrutxaga, Aritz Lasarguren, Javier Muguerza, and Jesús M. Pérez

Un amplio estudio comparativo de índices de validación de agrupamiento 1549
Olatz Arbelaitz, Ibai Gurrutxaga, Javier Muguerza, Jesús M. Pérez, and Iñigo Perona

On the study of nearest neighbor algorithms for prevalence estimation in binary problems 1551
Jose Barranquero, Pablo González, Jorge Díez, and Juan José del Coz

A Distance-based Ranking Model Estimation of Distribution Algorithm for the Flowshop Scheduling Problem 1553
Josu Ceberio, Ekhine Irurozki, Alexander Mendiburu, and Jose A. Lozano

A Methodology to Find the Elementary Landscape Decomposition of Combinatorial Optimization Problems 1555
Francisco Chicano, L. Darrell Whitley, and Enrique Alba

Optimal Cycle Program of Traffic Lights with Particle Swarm Optimization1557
José García-Nieto, Ana C. Olivera, and Enrique Alba

Avoiding Redundant Messages in BnB-ADOPT 1559
Patricia Gutierrez and Pedro Meseguer

Learning Bayesian Network Classifiers from Label Proportions 1561
Jerónimo Hernández-González, Iñaki Inza, and Jose A. Lozano

An Evaluation of Methods for Estimating the Number of Local Optima in Combinatorial Optimization Problems 1563
Leticia Hernando, Alexander Mendiburu, and Jose A. Lozano

Fuzzy numbers from raw discrete data using linear regression 1565
Juan Moreno-Garcia, Luis Jimenez-Linares, Luis Rodriguez-Benitez, and Ester Del Castillo

Multilabel Classifiers with a Probabilistic Thresholding Strategy 1567
José Ramón Quevedo, Oscar Luaces, and Antonio Bahamonde

Hesitant Fuzzy Linguistic Term Sets for Decision Making 1569
Rosa M. Rodríguez

Data Mining and User Centred Design to support the Recommendations Elicitation Process in Social Online Learning Environments 1571
Olga C. Santos and Jesus G. Boticario

A service oriented architecture to provide data mining services for non-expert data miners 1573
Marta Zorrilla and Diego García-Saiz

Doctoral Consortium

Prefacio 1577

Organización 1578

A Computational Model of Maspin Effects on Cell Migration Using Artificial Neural Network 1579
Mohammad Abdullah Al-Mamun

Vesta: Análisis y detección de contenido agraviante en imágenes 1585
Borja Alonso-Isla

Mejora de la Asistencia en los Accidentes de Tráfico combinando Redes Vehiculares e Inteligencia Artificial 1591
Javier Barrachina

Detección de tráfico de control de *botnets* mediante el modelado del flujo de paquetes 1597
Félix Brezo

Parallel Data Mining Algorithms on GPUs 1603
Alberto Cano

Studying Permutation-based Combinatorial Optimization Problems and Proposing Efficient Estimation of Distribution Algorithms 1607
Josu Ceberio

Tratamiento de la dimensionalidad, desbalanceo e información de dependencia en clasificación multietiqueta 1612
Francisco Charte

Moderación Automática de Comentarios y Reputación de Usuarios en Sitios Web Sociales 1618
Jorge de la Peña

Sistema multiagente para la asignación de recursos en entornos distribuidos1624
Fernando De la Prieta

Energy Efficient Resource Usage in Data Centers: GreenDoop 1630
Damián Fernández

eJason: una Plataforma para el Desarrollo de Sistemas Multiagente Distribuidos Eficientes y Robustos 1636
Álvaro Fernández Díaz

Linear programming with Interval Type-2 fuzzy constraints 1642
Juan C. Figueroa-García

Hacia un sistema de marketing dirigido más eficaz y personalizado en redes sociales 1648
Patxi Galán-García

A Distributed Dynamic Norm Monitoring Framework for Evolving Institutions 1653
Ignasi Gómez-Sebastià

Gene clustering with $\mathcal{K}$-Formal Concept Analysis 1659
José María González-Calabozo

Contributions to weakly supervised classification problems 1665
Jerónimo Hernández-González

Optimización de Redes de Sensores Inalámbricos mediante Computación Inteligente 1671
Jose M. Lanza-Gutierrez

Metaheuristic Strategies for Scheduling Problems with Uncertainty 1677
Juan José Palacios

Detección de emociones expresadas con la voz mediante caracterización biométrica 1683
Daniel Palacios-Alonso

Sistema Multi-agente para modelar Procesos de Consenso en Toma de Decisión en Grupo a Gran Escala usando Técnicas de Soft Computing ... 1689
Iván Palomares

Modelo para la generación de cursos virtuales en sistemas de gestión de aprendizaje usando tecnologías de la web semántica ... 1695
Danilo Pástor-Ramírez

Aprendizaje de redes bayesianas híbridas con mixturas de funciones base truncadas ... 1701
Inmaculada Pérez-Bernabé

Scalable and Efficient Algorithms in Machine Learning for High-Dimensional Data ... 1706
Diego Peteiro-Barral

Optimización de la gestión energética en vehículos eléctricos mediante Sistemas Inteligentes Híbridos ... 1712
Hector Quintian

A Recommendation Framework Based on the Analytic Network Process and its Application in the Semantic Technology Domain ... 1718
Filip Radulovic

A multi-modal and data mining based emotion detection approach to build an affective open standards-based user model to support adaptive learning environments ... 1724
Sergio Salmeron-Majadas

Improving recent episodic memory through location-based serious games . 1729
Jesús A. Sánchez

Bio-inspired Computing and Smart Mobility ... 1734
Daniel H. Stolfi

Computación Natural en Redes Vehiculares ... 1740
Jamal Toutouh

Author Index ... 1746

17-20 sept 2013
MULTICONFERENCIA
CAEPIA'13
Madrid

XV Conferencia de la Asociación Española para la Inteligencia Artificial [CAEPIA 2013]

Editores asociados:

Concha Bielza
Antonio Salmerón

Simposio organizado por

Asociación Española para la Inteligencia Artificial (AEPIA)

Prefacio

Este volumen contiene una selección de artículos aceptados para su presentación oral en la XV Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA 2013), celebrada en Madrid del 17 al 20 de Septiembre de 2013. Ésta es la décimo quinta conferencia bienal de la serie CAEPIA, que comenzó en 1985. Las ediciones previas se celebraron en Madrid, Alicante, Málaga, Murcia, Gijón, Donostia, Santiago de Compostela, Salamanca, Sevilla y La Laguna. Esta edición de CAEPIA se coordinó con otras conferencias independientes: IX Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB 2013), IV Simposio sobre Lógica Fuzzy y Soft Computing (LFSC), VII Simposio de Teoría y Aplicaciones de Minería de Datos (TAMIDA 2013), Fusión de la INfOrmación (FINO 2013) y Agentes y Sistemas Multiagente: De la Teoría a la Práctica (ASMas). Por ello esta vez la conferencia se denomina Multi-Conferencia CAEPIA, como signo del fuerte vínculo existente entre todos los investigadores de Inteligencia Artificial españoles. Más aún, este año CAEPIA se celebró dentro del IV Congreso Español de Informática (CEDI 2013).

CAEPIA 2013 es un foro abierto a investigadores de todo el mundo para presentar y discutir sus últimos avances científicos y tecnológicos en Inteligencia Artificial (IA). Sus principales objetivos son facilitar la diseminación de nuevas ideas y experiencias, fortalecer los lazos entre los distintos grupos de investigacion y ayudar a difundir los nuevos desarrollos hacia la sociedad. Todos los puntos de vista –teoría, metodología y aplicaciones– son bienvenidos.

Aparte de la presentación de artículos técnicos, el comité científico de CAEPIA 2013 incluyó dos conferencias invitadas. Conjuntamente con las otras conferencias, se celebraron sesiones de trabajos predoctorales (Doctoral Consortium) y, por primera vez, una sesión especial de trabajos publicados recientemente (Key Works) en revistas o foros de reconocido prestigio.

Con el objetivo permanente de hacer que CAEPIA sea una conferencia de alta calidad, y siguiendo el modelo de las exigentes conferencias de IA actuales, los Presidentes del Comité de Programa de CAEPIA organizaron el proceso de revisión como se describe a continuación. El Comité Científico se estructuró en dos niveles. En el primer nivel el conocimiento de la IA se distribuyó en 11 áreas y se designó un Responsable de Área para cada una. Todos ellos son miembros reconocidos de la comunidad de IA afiliados a universidades y centros de investigación españoles. En el segundo nivel se organizó un Comité de Programa con 116 miembros (37 instituciones eran de fuera de España). Todos los trabajos fueron cuidadosamente revisados por este comité (revisión por pares), algunas veces con la ayuda de revisores adicionales. Hubo 19 revisores adicionales (8 instituciones eran de fuera de España). Los revisores juzgaron la calidad global de los trabajos, junto con su originalidad y novedad, precisión

técnica, conocimiento de literatura relacionada y calidad de la presentación. Los revisores indicaron su confianza en el área temática de cada trabajo además de proporcionar comentarios detallados. Cada trabajo se asignó al menos a tres miembros del Comité de Programa (siguiendo el modelo de doble ciego) para ser revisado, y a un Responsable de Área para supervisar las revisiones. Éstos recomendaron las aceptaciones o rechazos de los trabajos a los Presidentes del Comité de Programa de CAEPIA, quienes tomaron las decisiones finales. También éstos escogieron el Mejor Trabajo CAEPIA y el Mejor Revisor, ambos premios otorgados por AEPIA.

CAEPIA recibió 66 trabajos. Los autores de los mismos debían marcar si su trabajo quería optar a ser publicado en el volumen Lecture Notes in Artificial Intelligence (LNAI). 48 trabajos de los 66 enviados optaron a dicha publicación. Después del proceso de revisión, 27 de ellos fueron aceptados en LNAI. También hubo envíos para LNAI de las otras conferencias federadas con CAEPIA: 17 de MAEB 2013 (6 aceptados), 10 de LFSC (4 aceptados), 8 de TAMIDA 2013 (2 aceptados), 1 de FINO 2013 (0 aceptados) y 2 de ASMas (1 aceptado). Estos trabajos se revisaron con el mismo proceso y Comité Científico que los trabajos de CAEPIA. Los Presidentes del Comité de Programa de estas conferencias y los Responsables del área más relacionada contribuyeron también al proceso de revisión.

Este volumen contiene los trabajos seleccionados para CAEPIA 2013 que no fueron publicados en LNAI. Es un total de 15 trabajos: 8 proceden de artículos que inicialmente marcaron ser candidatos a incluirse en el LNAI (los 48 mencionados) pero que finalmente no se seleccionaron y 7 son artículos que no tenían dicha marca (de los 18 del total de 66) y que fueron aceptados. En realidad hubo 4 artículos más aceptados, pero los autores decidieron retirarlos por diferentes motivos. Todos los trabajos aceptados se presentaron oralmente durante la conferencia en sesiones de 20 minutos, organizadas por temáticas afines.

Los dos conferenciantes invitados fueron Francisco Herrera (ECCAI Fellow, Universidad de Granada) y Tom Heskes (Editor-en-Jefe de la revista científica *Neurocomputing*, Radboud University Nijmegen, Holanda). Francisco Herrera presentó los aspectos más importantes de los llamados *big data*, las características de las librerías actuales y los retos en el desarrollo de nuevos algoritmos escalables. Tom Heskes mostró cómo las técnicas de aprendizaje automático Bayesiano pueden conducir a un nuevo paradigma de interfaces cerebro-ordenador, la reconstrucción de imágenes fMRI basadas en la activación y el análisis de redes a partir de imágenes de tensores de difusión.

Los editores de este volumen desean manifestar su agradecimiento a todos los que contribuyeron a la Multi-Conferencia CAEPIA 2013: los autores de los trabajos, los miembros del Comité Científico juntos con los revisores adicionales, los conferenciantes invitados, y los organizadores de las sesiones de Doctoral Consortium y Trabajos Destacados. Gracias también a Luis Guerra, que diseñó la página web de la conferencia CAEPIA, gestionó el sistema EasyChair (http://www.easychair.org/) y compiló este volumen y el LNAI. Por último,

gracias al Comité Organizador (incluido el del CEDI), las universidades Complutense y Politécnica de Madrid y AEPIA por su apoyo.

Junio 2013

Concha Bielza, Presidente del Comité de Programa de CAEPIA
Antonio Salmerón, Co-Presidente del Comité de Programa de CAEPIA

Organización

Comité ejecutivo

Concha Bielza	Universidad Politécnica de Madrid
Antonio Salmerón	Universidad de Almería
Amparo Alonso-Betanzos	Universidad de A Coruña

Comité de Programa Seniors

Enrique Alba, España
Ismael García Varea, España
Francisco Herrera, España
Pedro Larrañaga, España
Serafín Moral, España
Filiberto Plá, España
Emilio Corchado, España
Lluis Godó, España
Juan Huete, España
Pedro Meseguer, España
Sascha Ossowski, España

Comité de Programa

Enrique Alegre, España
Alessandro Antonucci, Italia
Carlos Beltran, Italia
Christian Blum, España
Daniel Borrajo, España
Vicent Botti, España
Humberto Bustince, España
Fidel Cacheda, España
Borja Calvo, España
Andrés Cano, España
Enric Cervera, España
Francisco Chicano, España
Carlos Coello, México
Oscar Cordón, España
Inés Couso, España
Sergio Damas, España
Arjen de Vries, Holanda
S. Irene Díaz, España
Richard Duro, España
Juan M. Fernández-Luna, España
João Gama, Portugal
Cesare Alippi, Italia
Rubén Armañanzas, España
Alexandre Bernardino, Portugal
Blai Bonet, Venezuela
Juan Botía, España
Alberto Bugarín, España
Pedro Cabalar, España
José M. Cadenas, España
José L. Calvo Rolle, España
Pablo Castell, España
Carlos Chesñevar, Argentina
Francisco Chiclana, Reino Unido
Juan M. Corchado, España
Carlos Cotta, España
Fabio Crestani, Italia
Andre C.P.L.F. de Carvalho, Brasil
María J. del Jesús, España
Bernabé Dorronsoro, Francia
Marcelo Federico, Italia
Francesc Ferri, España
José A. Gámez, España

Ismael García Varea, España
Karina Gibert, España
Manuel Gómez-Olmedo, España
Manuel Graña, España
Luis Guerra, España
José Hernández-Orallo, España
César Hervás, España
Iñaki Inza, España
Vicente Julian, España
Ryszard Kowalczyk, Australia
Beatriz López, España
David Losada, España
Francisco Luna, España
Andrew MacFarlane, Reino Unido
Lawrence Mandow, España
Francesco Marcelloni, Italia
Carlos Martínez, España
Jesús Medina, España
Pedro Meseguer, España
José M. Molina, España
Eduardo Morales, México
Thomas Nielsen, Dinamarca
Eugenio Oliveira, Portugal
Xosé Pardo, España
José M. Peña, España
Hector Pomares, España
José M. Puerta, España
Ramón Rizo, España
Richard Rose, Canadá
Rafael Rumi, España
Angel Sappa, España
Dragan Simic, Serbia
Thomas Stützle, Bélgica
Marco Tomassini, Suiza
Javier Traver, España
Miguel A. Vega, España
José R. Villar, España
Hector Geffner, España
Jorge Gómez, España
Juan M. Górriz, España
Sergio Guadarrama, Estados Unidos
José J. Guerrero, España
Enrique Herrera-Viedma, España
Juan Huete, España
Pedro Isasi, España
Georges Katsirelos, Francia
Daniel Le Berre, Francia
Maite López, España
José A. Lozano, España
Gabriel J. Luque, España
Nicolas Madrid, República Checa
Felip Manya, España
Lluis Márquez, España
Andrés Masegosa, España
Belén Melián, España
Majid Mirmehdi, Reino Unido
Raúl Montoliu, España
Herman Ney, Alemania
Manuel Ojeda-Aciego, España
Daniel Ortiz, España
Juan Pavón, España
Rafael Peñaloza, Alemania
Mike Preuss, Alemania
Héctor Quintián, España
Luis Rodríguez Ruiz, España
Michael Rovatsos, Escocia
Yvan Saeys, Bélgica
Carles Sierra, España
Vaclav Snasel, República Checa
Enrique Sucar, México
María I. Torres, España
Alicia Troncoso, España
Sebastián Ventura, España
Michal Wozniak, Polonia

Revisores adicionales

Carlos Agüero, España
Diego Fernández, España
Shima Gerani, Italia
Giacomo Inches, Italia
Henrique Lopes, Portugal
Francisco Martín, España
Luis Matías, Portugal
Jonathan Ortigosa-Hernández, España
Josep Puyol-Gruart, España
Ana Paula Rocha, Portugal
Hanen Borchani, España
Vreixo Formoso, España
Daniel Gutiérrez, España
Monica Landoni, Italia
Parvaz Mahdabi, Italia
Vicente Matellán, España
Manuel Mucientes, España
Antonio Pereira, Portugal
Alejandro Rituerto, España

Sponsors

- Asociación Española para la Inteligencia Artificial (AEPIA)

A Dynamic Bayesian Network for Breast Boundary Extraction

Hamed Habibi Aghdam [1], Domenec Puig [2],
Jaime Melendez [3], and Agusti Solanas [2]

[1]Computer Engineering and Information Technology Department, Amirkabir University of Technology
[2] Department of Computer Engineering and Mathematics, Rovira i Virgili University
[3] Department of Radiology, Radboud University Nijmegen Medical Centre

Abstract. Determining the breast boundary is one of the most important tasks in breast tissue classification for computer-aided diagnosis. Inaccurate boundary estimation would affect the ratio of dense to fatty region provided by computational methods. Since breast image is taken by x-ray imaging, its boundary can be characterized by textured regions. In this paper, we propose a fully probabilistic approach based on Dynamic Bayesian Networks for extracting the breast boundary. Experimental results show that the proposed method is able to produce accurate boundaries.

Key words: computer-aided diagnosis, breast boundary, x-ray mammogram, Dynamic Bayesian Network.

1 Introduction

Breast cancer is a very serious disease for women, since eight to twelve percent will develop breast cancer in their life [1]. Early detection is critical to increase chances of successful treatment. However, its detection is very difficult due to the fact that there are no symptoms during the first stages of the disease development. Fortunately, x-ray mammography can reveal small changes in breast tissue due to their density [2]. Those density differences translate into different brightness levels in the final image.

Analyzing mammograms is not a simple task even for experts. Hence, there is a considerable demand to Computer Aided Diagnosis (CAD) systems to help radiologists to detect and diagnose new cases. Generally, there are two main kinds of analyses for a mammogram, namely *segmentation* and *classification*.

The goal of the *classification* is to label a typical pixel inside breast region as *fatty* or *dense*. The ratio of dense to fatty regions indicates the risk of cancer in the mammogram under investigation [3]. The research community has paid wide attention to classification tasks and good results have been reported in last years [3–5]. Although some authors use these terms interchangeably, the goal of *segmentation* is to divide mammograms into their comprising parts. A typical mammogram has four different parts, namely background, artifacts (*e.g.* labels,

numbers and noise), breast and pectoral muscle. Generally, we are only interested in breast and pectoral muscle regions. In addition, the pectoral muscle region is always inside the breast region. Hence, finding the breast region provides a priori knowledge about thepectoral muscle location.

Finding these two regions is important since they affect the ratio of fatty to dense region. In early studies by Karssemeijer [6], segmentation of the breast region was carried out by thresholding and morphological operations. In mammograms, the number of dark pixels is usually greater than the number of bright pixels and, frequently, bright pixels are distributed uniformly. This produces a histogram with peaks at dark pixels and near uniform shape at bright pixels. For this reason, finding a proper threshold is a tough task and it gets even harder in presence of noise. The determination of inappropriate thresholds causes the wrong identification of larger or smaller areas. Notwithstanding, threshold values can be computed using interactive methods in which the user defines a threshold value and evaluates the result. This procedure is repeated until the user finds an accurate result [7]. In addition to threshold-based methods, Saha *et al.* [8] proposed a method based on scale-based fuzzy connectivity. This approach is not fully automatic since the values of some parameters must be determined empirically. As another threshold-based method, Raba *et al.* [1] computed N different threshold values and found values related to largest and smallest regions. Then, the final threshold value was computed by statistically evaluating those regions. Finally, the extraction of the pectoral muscle region was performed by using a statistical region-growing algorithm. Again, due to texture variation in pectoral regions this algorithm fails in some mammograms.

From a different perspective, a research carried out by Martí *et al.* [10] is based on tracing the boundary from an initial seed point. The concept resembles a snake movement and uses gradient information to find the breast boundary but, in most of mammograms, the gradient information is very noisy and unreliable.

In this article, we propose a method for tracing the breast boundary by using a Dynamic Bayesian Network (DBN). This approach is fully probabilistic and it is completely based on learning from training data. Most importantly, our method does not use unreliable gradient information. In contrast, it uses texture information to find boundary pixels and, at the same time, it tries to generate smooth boundaries. We have test our proposal on the mini-MIAS database, and the obtained results show that our method is able to reliably find breast boundaries. The reminder of this article is organized as follows: in the next section the need for using texture information instead of gradient vectors is justified. Then, the proposed DBN-based method is presented. Also, the probability models used are mentioned. Section 3 shows some experiments and provides a discussion about the obtained experimental results. Finally, the article concludes in Section 4.

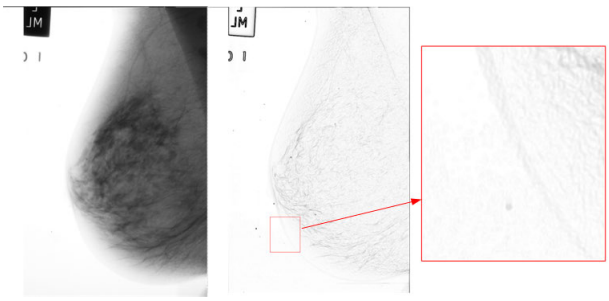


Fig. 1. Left: Breast image, Middle: Edge map, Right: A small portion of edge map

2 Proposed Method

Detecting the breast boundary in mammograms is a difficult task. It is not only due to artifacts such as labels and noise, but also because of the boundary itself. In mammography, the breast is compressed to a thickness of approximately $6\,cm$ and x-ray image is taken perpendicular to the plane of compression [2]. The boundary of the compressed breast has a lower thickness in comparison to its inner part but its texture remains as the tissue attenuates the x-rays to a lower degree and produces dark areas in the image.

Due to the fact that body tissues are not uniform in the boundary of the compressed breast, it will be characterized by using textured regions of the final image. Hence, simple edge detection methods are not able to extract the boundary in breast images because it is not defined in terms of gray level differences between neighbor pixels. As shown in Fig.1, breast images have specific textures in their boundary. However, making a decision about edge pixels is not simple since their neighbor pixels do not have uniform or zero gradient. In addition to its texture, the breast boundary has another important characteristic: Geometrically, it is a smooth curve. Hence, the set of pixels that is extracted as boundary pixels must indicate a smooth curve. For the sake of simplicity, in the reminder of the article we refer to the breast boundary as *boundary*.

Let's assume that we know the location of the initial point of the boundary on the image which is called P. There is a set $C = \{P_c^1, P_c^2, \ldots, P_c^N\}$ of N points near to P which are considered as candidates for the next boundary point. There is only a single point within C more likely to be a boundary point. As shown in Fig.2 (left), the point P_c^i in C is assumed to be more likely if its underlying texture has a high probability to be boundary and in addition it keeps the boundary smooth.

The reason for selecting P_c^{best} as the most likely point is that it satisfies both the above criteria. In other words, there may be some points with high texture likelihood in set C. In this situation, the point that keeps the boundary smoother will be selected. From a probabilistic view point, we are looking for a

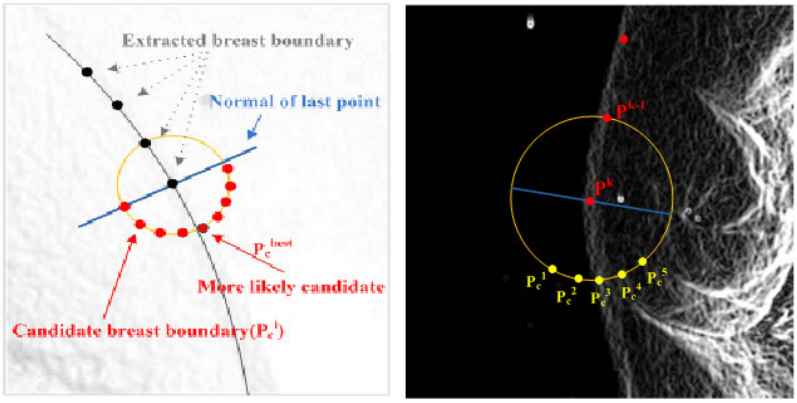


Fig. 2. (left) Candidates to be boundary points; (right) Need for prior knowledge

point within C which maximizes the following joint probability density function.

$$P_c^{best} = max_{P_c^i \in C} P(T, P_c^i, P^k, ..., P^1) \tag{1}$$

In this equation, T accounts for the observed texture feature and $P^k ... P^1$ accounts for the boundary smoothness. Also, P_c^i is the current candidate pixel. If we could find a model for the joint probability distribution described above, we could also easily find the most likely candidates. However, determining such s probability model is a hard problem.

Considering causal relationships between arguments of this joint probability, there is no relationship between T and P^k. In addition, we can make this model simpler if we assume that the smoothness is a second order Markov process. Hence, based on causal relationships between the texture, the current point and the previous points of the boundary and assuming that smoothness is a second order Markov process, we propose a Dynamic Bayesian Network (DBN) for tracing the breast boundary. According to this network, we can describe the joint probability of (1) as follow:

$$\begin{aligned} P(T, P_c^i, P^k, P^{k-1}) = P(P^{k-1}) \times P(P^k) \times P(P_c^i) \\ \times P(T|P_c^i) \times P(P_c^i|P^k, P^{k-1}) \end{aligned} \tag{2}$$

$P(T|P_c^i)$ is the probability of the textured region surrounding point P_c^i and $P(P_c^i|P^k, P^{k-1})$ accounts for the smoothness and computes the probability of the location of the candidate point P_c^i given the locations of the two last points on the boundary. In addition, $P(P_c^i)$ is the *a priori* probability of selecting the candidate point P_c^i and, $P(P^k)$ and $P(P^{k-1})$ indicate the *a priori* location probabilities of the two last points of the boundary. Without loss of generality, we can ignore the two last terms in Equation (2) since they are constant and their values are not affected by the value of P_c^i. Hence, the final joint probability

can be defined as follows:

$$P(T, P_c^i, P^k, P^{k-1}) = P(P_c^i) \times P(T|P_c^i) \times P(P_c^i|P^k, P^{k-1}) \tag{3}$$

In the next sections we show how to determine those probabilities.

2.1 Texture probability model

$P(T|P_c^i)$ is the probability of the underlying texture for point P_c^i. To find such probability model, we need some training images of breasts and their corresponding boundaries. Then, for each point P_c^i on the boundary, the texture feature of its surrounding region is extracted. After this procedure is repeated for all boundary points and for all training breast images, we have a collection F of feature vectors for all boundary points. Finally, we just need to find a probability model for the collection F.

Texture feature. To extract features from patches, we propose to divide them into four equal sub-regions in the neighborhood of the candidate point P_c^i. Then, the statistical features (i.e minimum, maximum, mean, variance, skew, kurtosis and entropy) of each sub-region are extracted and concatenated into a single vector to build the feature vector of the underlying region of pixel P_c^i. Considering that each sub-region has a 7-dimensional vector, the feature vector of the texture region will be a 28-dimensional vector.

Probability model. After collecting the feature vectors of the boundary pixels, we use kernel density estimation with a Gaussian kernel to model the distribution of feature vectors. Hence, $P(T|P_c^i)$ can be computed using Equation (4)

$$P(T|P_c^i) = \frac{1}{Nh} \sum_{i=1}^{N} e^{-\frac{(\|T-T_i\|^2)^2}{2h^2}} \tag{4}$$

where T is the texture feature of the surrounding region of Pc^i, T_i is the i^{th} training texture feature, h is the bandwidth of the Gaussian kernel and N is the total number of training feature vectors. Also, $\|.\|$ indicates the norm of the vector.

2.2 Smoothness Probability

To calculate $P(P_c^i|P^k, P^{k-1})$, we collected the following information from each training boundary:

$$\begin{aligned} x_k &= \left[P_x^k - P_x^{k-1}, P_x^{k-1} - P_x^{k-2}\right] \\ y_k &= \left[P_y^k - P_y^{k-1}, P_y^{k-1} - P_y^{k-2}\right] \end{aligned} \tag{5}$$

The vector that is constructed by concatenating x_k and y_k indicates the relative difference between the current point and the two last points of the boundary.

After these vectors are collected for all of training boundaries, a 3 component Gaussian mixture model is trained to model their distribution. Hence, we can compute $P(P_c^i|P^k, P^{k-1})$ using Equation (6):

$$P(P_c^i|P^k, P^{k-1} = \sum_{i=1}^{3} \pi_i \times (x_k, y_k; \mu_i, \Sigma_i) \tag{6}$$

Where, π_i is the mixing coefficient, μ_i is the mean vector and Σ_i is the covariance matrix of the i^{th} component.

2.3 Prior probability of the location

To show the importance of prior probability, we refer to Fig.2(right), that shows a small portion of a breast image. Assume that the algorithm has traced the boundary and reached to point P^k. Now, it proceeds by finding the best next candidate. Among different candidates in radius r of point P^k, five nearest points have been shown in Fig.2(right), (marked as $P_c^1 ... P_c^5$). Each of these candidates has a probability to be a breast boundary pixel. However, among them, P_c^2, P_c^3 and P_c^4 have higher likelihood according to the texture probability model and obviously P_c^3 has the highest likelihood.

Considering the line segments between (P^{k-1}, P^k, P_c^2) and those between (P^{k-1}, P^k, P_c^3), we realize that by selecting P_c^2 the boundary will be smoother than by selecting point P_c^3. In addition, if we consider all training boundary segments, the probability of occurrence of segment (P^{k-1}, P^k, P_c^3) will be lower than (P^{k-1}, P^k, P_c^2). Hence, multiplying the texture probability of P_c^2 and (P^{k-1}, P^k, P_c^2) it will be greater than multiplying the texture probability of P_c^3 and (P^{k-1}, P^k, P_c^3). As a result, P_c^2 will be selected as the next boundary pixel.

To solve this problem, we need a prior probability to penalize points such as P_c^2. The main difference between P_c^2 and P_c^3 is their texture probability. Putting these together, we can model the prior probability of a candidate point using a Gaussian or Laplace distribution. In this study we have used a Gaussian distribution with $\mu = 1$ and the value of σ^2 is computed using the training data.

$$P(P_c^i) = Gaussian(x; \mu, \sigma) = e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{7}$$

In Equation (7), x is equal to $P(T|P_c^i)$. The interpretation of this equation is that if the texture probability is high, it is much more likely to be a boundary point. If we apply this prior probability on candidate points of Fig.2(right), point P_c^3 will be selected instead of P_c^2.

2.4 Initializing Algorithm

To initialize the algorithm, an input image is scanned from top to bottom and from left to right. For each pixel, its surrounding region is selected and its texture feature is extracted. Then, its class is calculated using the kernel density

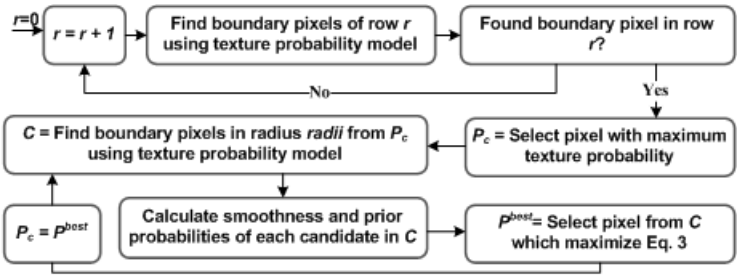


Fig. 3. Overall algorithm to trace boundary of breast

estimation model for $P(T|P_c^i)$. At the same time, the probability of the feature vector is computed. Assume that the algorithm finds a pixel in row r, which is classified as a boundary pixel. In this stage, all the other pixels in row r are classified. Assume Q is the set of all pixels in row r that are classified as boundary pixels. Then, the pixel with maximum probability in Q is selected as the starting point of the boundary.

2.5 Tracing Boundary

We apply the previously described DBN to extract the boundary in mammograms. To this end, we first need to collect some training data in order to train the probability models of DBN for tracing breast boundaries. In this way, we will have a set including trained textures, smoothness and prior probability models. Given this set, we can apply the algorithm shown in Fig.3 to trace the boundary. To this end, we need to determine the values of some parameters including the size of the texture patch and a radius parameter named *radii*. The size of the texture patch specifies the size of the surrounding region of each candidate point in which textured features are extracted. As Fig.2(left) shows, candidate points lay on the surface of a circle. The radius of this circle equals to *radii*, which is determined empirically.

3 Experiments

We have implemented the proposed method in MATALB 2010. The parameters of the Gaussian mixture models were obtained using MATLAB statistical toolbox. The training of the Gaussian model was repeated 100 times with a random starting point and the model with largest likelihood was selected. In the following experiments, the texture features of each pixel were extracted by using 50×50 pixels windows and candidate points were computed in radius of 20 pixels from the current pixel. Also, a value of 0.01, computed empirically, was assigned to the bandwidth of the Gaussian kernel.

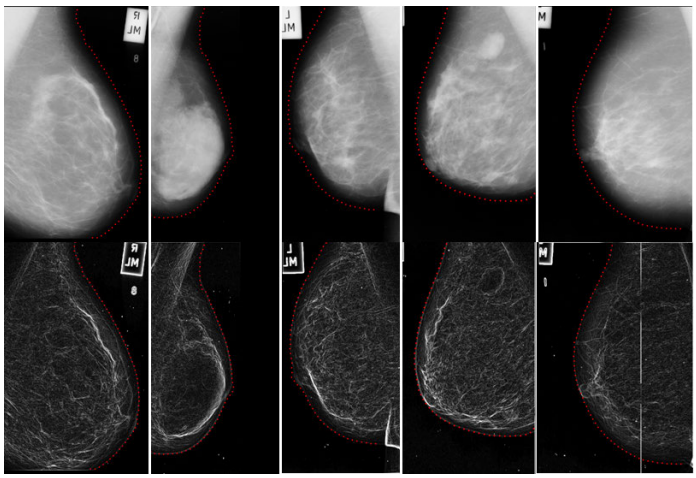


Fig. 4. Overall performance of algorithm to trace breast boundary

3.1 Database

To test our method, the mini-MIAS database has been used. In order to manually extract boundaries (ground-truth) and prepare training data, the gradient of sample images was processed using image processing tools and the boundary was manually determined using a combination of magnetic and polygonal lasso tools. We selected 41 images and extracted their boundaries manually. In this way, 27346 positive and 27518 negative texture patches were selected respectively to train the model distribution of texture features.

3.2 Experiment 1

In this experiment we investigate the overall performance of the proposed method. To this end, the algorithm is applied on test images that have not been used in the modeling of probability terms of DBN. Fig.4 shows the obtained results.

In this Figure, the images show the original mammogram blended with extracted boundaries. To be more tractable and evaluate the performance visually, we have blended those boundaries with the edge map of the same mammograms. We have applied a power low transformation on the edge maps to increase their dynamic range. It is clear from Fig.4 that the proposed method is able to extract boundaries of breast and pectoral muscle accurately. In addition, the extracted boundary is smooth and there are no sharp changes. However, we should notice that there are situations in which this method make wrong decisions. For example, Fig.5(left) shows one of these situations.

There are a few locations on the boundary which are selected incorrectly by the algorithm. This problem has two major causes: First, it can happen due to insufficient training data. By inspecting those locations we realized that there are noisy patches around that pixels. If the number of noisy patches in the training

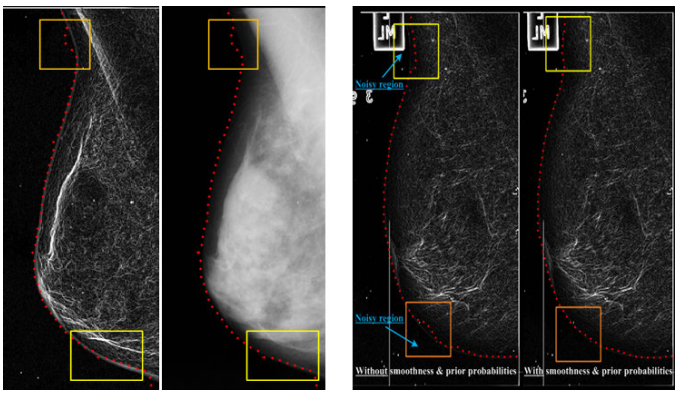


Fig. 5. (left) Incorrect decisions made by algorithm; (right) Effect of smoothness and prior probabilities

data is not sufficient, the probability of occurrence of such vectors will be low and consequently wrong choices may happen. The second reason is related to the size of the texture region. If it is small, it will not be able to encode important features of texture. On the other hand, if it is large there might be situations in which the probability of occurrence of the feature vector decreases.

Fortunately, there are simple solutions to both problems. To address the first problem, we can add boundaries of noisy mammograms into training dataset. For the second problem, we can calculate the texture region in several windows with different size.

3.3 Experiment 2

In this experiment we try to show how smoothness and prior probabilities affect the boundary extraction process. To achieve this goal, we selected a noisy mammogram and extracted its boundary using the proposed method. Once again, the algorithm was applied on the same mammogram after omitting smoothness and prior probabilities. The results of this experiment are shown in Fig.5(right).

There are two significant differences in the extracted boundaries which are indicated by two squares with different colors. Considering the texture region inside each of them, we realize that their common characteristic is the existence of noise. In the bottom square, the noise is produced during the imaging process but the noise in the above square is due to the label of the mammogram. In both cases, the feature vector of the texture region is affected by noise and consequently, the probability of feature vectors is not computed correctly. For this reason, whenever the shape of the boundary is ignored, the decision making of the algorithm is done using noisy texture measures only. Therefore, sharp changes occur during the extraction process.

Smoothness and prior probabilities penalize those points which can cause sharp changes in the boundary. Hence, even if texture features are noisy and

their probability is not computed accurately, applying smoothness and prior probabilities prevent the algorithm from selecting points which are not smooth.

4 Conclusion

In this article, we proposed a probabilistic framework for extracting breast and pectoral muscle boundaries in mammograms. To this end, a Dynamic Bayesian Network which utilizes texture and shape information of boundaries is used. In this method, we used statistical texture features with a kernel density estimation method to calculate the probability of a typical texture region. In addition, to encode shape information, we used the second order Markov model which computes the probability of candidate points by taking into account the last two points on the boundary and using a Gaussian mixture model.

Acknowledgments. This work was partly supported by the Spanish Government through project TIN2012-37171-C02-02.

References

1. D. Raba, A. Oliver, J. Martí, M. Peracaula, J. Espuny, *Breast segmentation with pectoral muscle suppression on digital mammograms*, Lecture Notes in Computer Science Volume 3523, pp 471-478, 2005
2. J. Raundahl, *Mammographic Pattern Recognition*, PhD Dissertation, The Image Group, Department of Computer Science, University of Copenhagen, 2007
3. A. Oliver et al., *A Novel Breast Tissue Density Classification Methodology*, IEEE Trans. on Information Technology in Biomedicine, Vol. 12, No. 1, 2008
4. J. Freixenet et al., *Eigendetection of masses considering false positive reduction and breast density information*, Medical Physics, 35(5):1840-53, 2008
5. M. Kallenberg, M. Lokate, C. van Gils, N. Karssemeijer, *Automatic breast density segmentation:an integration of different approaches*, Physics in Medicine and Biology, Vol.56, No.9, 2011
6. N. Karssemeijer, *Automated classification of parenchymal patterns in mammograms*, Physics in Medicine and Biology, Vol.43, No.2, 1998
7. V.A. McCormack, I. Dos Santos Silva, *Breast density and parenchymal patterns as markers of breast cancer risk: a meta-analysis*, Cancer Epidemiol. Biomarkers Prev. 15, 1159-1169, 2006
8. P.K. Saha, J.K. Udupa, E.F. Conant, P. Chakraborty, D. Sullivan, *Breast Tissue Density Quantification Via Digitized Mammograms*, IEEE Transactions on Medical Imaging, Vol.20, No.8, 2001
9. S. Man Kwok, R. Chandrasekhar, Y. Attikiouzel, M.T. Rickard, *Automatic pectoral muscle segmentation on mediolateral oblique view mammograms*, IEEE Transactions on Medical Imaging, Vol.23, No.9, 2004
10. R. Martí, A. Oliver, D. Raba, J. Freixenet, *Breast Skin-Line Segmentation Using Contour Growing*, Lecture Notes in Computer Science Vol.4478, pp 564-571, 2007
11. A. Oliver et al., *Automatic Classification of Breast Tissue*, 2nd Iberian Conf. on Pattern Recognition and Image Analysis, pp.431-438, 2005

CTC: Competitive in an Analysis of Genetic based Algorithms for Rule Induction in Imbalanced Datasets

Olatz Arbelaitz, Ibai Gurrutxaga, Javier Infante, Javier Muguerza, Jesús M. Pérez

Computer Science Faculty, University of the Basque Country (UPV/EHU),
Manuel Lardizabal 1, 20018 Donostia, Spain

{olatz.arbelaitz, i.gurrutxaga}@ehu.es,
jinfante001@ikasle.ehu.es, {j.muguerza, txus.perez}@ehu.es
http://www.sc.ehu.es/aldapa

Abstract. The class imbalance problem is one of the main challenges of machine learning today. It appears when the number of instances in one class significantly outnumbers the number of instances in the other classes. The CTC algorithm was designed to address a car insurance fraud detection problem with class imbalance. The aim of this work is to compare CTC to a set of results available in bibliography for 16 genetic-based machine learning (GBML) algorithms for rule induction and 6 classical non-evolutionary algorithms, such as decision trees, using the same training/test partitions of a 5-fold cross validation the authors used in 33 imbalanced datasets. Our study follows the same experimental methodology used in the original work; it includes hierarchical comparisons using the conclusions obtained by the Shaffer statistical test. Results show that CTC outperforms most of the algorithms, obtaining the second position in the average ranking among all algorithms, but without statistically significant differences with the first algorithm, Ripper. Thus, we can state that the CTC algorithm achieves competitive results when is used to deal small and imbalanced learning problems.

Keywords: Class imbalance problems, CTC algorithm, consolidated trees, decision trees, genetics-based machine learning, rule induction.

1 Introduction

The class imbalance classification problem has been considered as a current challenge of the Data Mining community [9, 11, 17]. It occurs when the number of examples of one class is much lower than the examples of the other ones. This fact makes the learning process much more difficult because most of the learning algorithms are designed to minimize the global measure of classification error. However, in the class imbalance context, this causes a bias towards the majority class and, as a conse-

quence, a loss in the sensitivity on the minority class examples which are the most important in many applications.

In this context, in [12] the authors proposed the CTC algorithm to deal with a class imbalance problem: a fraud detection problem. This algorithm is based on the well–known C4.5 decision tree construction algorithm, but it extracts knowledge from a set of samples to build a single decision tree and, so, it maintains the explaining capacity. The aim was to generate samples containing the proportion of fraud cases that occurred in the reality (minority class) according to estimation of the experts in the assurance company.

Therefore, the CTC algorithm allows extracting knowledge from a set of samples with a class distribution different from the one existing in the original sample. The CTC has been used in different contexts with this aim. On one hand, Weiss and Provost [15] suggested that each domain has an optimal class distribution to be used for training, whatever the original class distribution may be. Based on this conclusion, in [13] CTC was used with samples generated using the optimal class distribution Weiss and Provost proposed for 10 of the datasets used in [15]. Although the results were, in average, better, the authors suspected that the optimal class distribution for CTC could be different in contrast to the assertion of Weiss and Provost's work. This motivated the work [2] where the aim was to find the optimal class distribution for CTC and C4.5 algorithms in a wide experiment with 30 datasets and 14 different class distributions, from 2% to 98%. The results show that the CTC is able to draw together the information enclosed in a set of these samples to achieve higher discriminating capacity than C4.5 classifiers generated with samples of modified class distribution and even than C4.5 trees built with the complete training sample.

On the other hand, in [1] the behaviour of CTC was analyzed in a class imbalance context, where 14 of the most imbalanced datasets from the previous work were used. The CT trees built with balanced samples were compared with some different intelligent resampling methods used in bibliography to deal with this kind of problems (the well-known SMOTE [5] and some variants), obtaining competitive results.

Focusing on the class imbalance problem, in this paper we have analyzed the performance of the CTC algorithm with more datasets, being these smaller and more imbalanced than the ones used in previous works. These characteristics make these domains to be more difficult for any learning process. With this purpose, we have taken advantage of the analysis focused in imbalanced datasets carried out in the paper "Genetics-Based Machine Learning for Rule Induction: State of the Art, Taxonomy and Comparative Study" [7] where the authors used 33 datasets from the UCI repository [4] and made a comparative study of the results of 16 Genetics-Based Machine Learning approaches for rule induction and 6 classical non-evolutionary algorithms. Fortunately, all the results of this work, and the datasets (included the training/test partitions of a 5-fold cross validation) are available in the web site of the KEEL Project [3] (http://sci2s.ugr.es/keel/index.php)[1] and the authors invite the researchers to download them to analyze the behaviour of other learning methods.

[1] Also the authors have a external website associated to this paper: http://sci2s.ugr.es/gbml/

Therefore, the main objective of this paper is to position the CTC algorithm in this study in the context of class imbalance.

The study carried out follows the same experimental methodology used in the referenced work [7], this is, we run a 5 fold cross validation 5 times and the performance measure used for the classifiers obtained with CTC was the geometric mean (GM) of the accuracy rates per class. Finally, the significance of the results is supported by the kind of statistical tests lately suggested in literature [6, 10].

The rest of paper is organized as follows. Section 2 discusses works related to this issue. The experimental methodology is described in Section 3. We present a summary of the experimental results and the major outcomes in Section 4. Finally, Section 5 is devoted to conclusions and further work.

2 Related Work

2.1 The CTC Algorithm

The consolidated tree construction (CTC) algorithm was designed to solve a class imbalance problem in the car insurance fraud detection domain, where the class distribution needed to be changed and an explanation of the classification done was required. The aim was to build a single tree but using a set of samples (with modified class distribution) instead of a single one as C4.5 does. This method is based on the standard C4.5 classification algorithm [14] by means of building the corresponding decision tree associated to each sample but using a voting process to decide the variable that will split the node of the consolidated tree at each step of the tree's building process.

The main idea behind this method is to extract samples with a bigger proportion of minority class examples (by simply eliminating majority class ones), in order to facilitate the learning process, and to build a classifier with a set of samples, so that most of the examples of the original sample are taken into account.

The C4.5 algorithm was proposed by Quinlan [14] to build decision trees using a divide-and-conquer strategy. This supervised learning paradigm uses the gain ratio as split criteria to divide the data based on the most discriminant attribute according to the class. It has achieved high quality results as single classifier [8] and it was identified as one of the top 10 algorithms in data mining in the IEEE International Conference in Data Mining held in 2006 [16]. Besides, it is one of the most used learning algorithms specially as base classifier in multiple classifier systems (MCS) and as the classifier to be combined with intelligent resampling methods to solve class imbalance problems.

Although the idea of CTC is wider, the CTC has been shown in the bibliography as the consolidation process of C4.5 and, as a consequence, C4.5 has been its main competitor in the experiments carried out [1-2, 12-13]. The authors have recently made available an implementation of CTC for WEKA in the Web[2].

[2] http://www.sc.ehu.es/aldapa/weka-ctc

2.2 The referred work

In this work [7] the authors provide a state-of-the-art in Genetics-Based Machine Learning (GBML) approaches for rule induction and, then, they propose a taxonomy of this kind of methods. They also include a comparative study of the studied algorithms with some well-known non-evolutionary algorithms such as CART, AQ, CN2, C4.5, C4.5-Rules and Ripper, all of them with explaining capacities. To address this, they followed a methodology based on hierarchical comparisons where, first they identify the algorithms with the best performance inside each category and then, they select them as representatives of each category for the cross-family comparison.

Although the hierarchical methodology is applied for both: the standard classification and the classification with imbalanced datasets, this paper focuses only on the latter issue. The study is divided in two analyses: the first one uses the original imbalanced datasets to determine which algorithms are best adapted to this framework, and then, it applies a well-known data pre-processing technique, SMOTE [5], to rebalance the distribution of examples of the different classes in the training samples. In this case the aim is to analyse whether intelligent class rebalancing leads to performance improvements or not.

The table below summarizes the proposed taxonomy of the 16 GBML algorithms studied. It distinguishes among 3 different families based on the chromosome codification (the first family is further divided into 3 categories or subfamilies).

Table 1. Taxonomy of the 16 GBML algorithms studied

1. Chromosome = Rule:
   (a) Michigan: XCS and UCS
   (b) IRL (Iterative Rule Learning): SIA and HIDER
   (c) GCCL (Genetic Cooperative Competitive Learning): CORE, OCEC and COGIN
2. Chromosome = Rule Set:
   (a) Pittsburgh: GIL, Pitts-GIRLA, DMEL, GASSIT, OIGA and ILGA
3. Chromosome = Decision Tree or Tree Rule
   (a) HEDT (Hybrid Evolutionary Decision Tree): DT-GA, Oblique-DT and TARGET

3 Experimental Methodology

We performed experiments, only for CTC algorithm, with the 33 imbalanced datasets used in the referred work [7]. The characteristics of the databases vary within a wide range of values. Due to space limitation, we have not repeated here the values of the characteristics for all datasets but, instead, Table 2 summarizes the minimum and maximum values, as well as, the average (and median) values for the main characteristics.

In order to estimate the performance of the built CT trees we followed the same methodology used by Fernández et al in [7], a 5-iterated 5-fold cross-validation. We used the same training and test partitions they have available in the web site.

Table 2. Summary description of imbalanced datasets

	#Examples	*#Attributes*	*% Minority Class*	*#Examples Min.Class*
Min	150	3	0.77%	10
Max	5,472	19	35.51%	560
Mean	919.94	9.39	17.61%	119.97
Median	482.00	8.00	15.48%	52.01

From each training sample, we have to generate a set of samples for the construction of the consolidated trees. In order to determine what kind of set of samples to use, we have taken into account the results of previous work around the CTC algorithm [2]. Although the obtained optimal value (for AUC as performance measure) depended on the concrete dataset, a set of five balanced samples (minority class distribution equal to 50%) was in general a good configuration (with statistically significant differences comparing with C4.5). Besides, the size of the used subsamples was fixed to the number of examples of the minority class in the training sample, as Weiss and Provost did in [15]. This makes it possible to generate any class distribution for this size, without replicating any examples. This implies that the samples used for CTC were very small; the average size of the generated samples coincides with the average of the minority class distribution, that is, 17.61% for the datasets of this paper (see Table 2). In the case of abalone19 dataset, an extreme case, the subsamples' size would be only of 26 examples, when the training sample's size is 3,339 (80% of the complete sample). Using 5 subsamples of this type, if all of examples would be different, it would take into account only 3.89% of the examples of the training sample.

Because of this, we decided to use also another size for the subsamples and to increment the number of subsamples to be used for the most imbalanced datasets. In concrete, on the one hand, we used two different values to determine the size of the subsamples to be generated: the size of the minority class in the training sample (the mentioned one) and the biggest possible size so that the minority class examples are the 50% of the subsample, without using replacement, that is, simply erasing the adequate number of majority class examples. We marked both options as 'Tam=SizeOfMinClass' and 'Tam=Max', respectively. The latter sample's size was also used in the previously mentioned work [13].

On the other hand, we proposed a rule to determine the number of subsamples to be generated for a given dataset according to its original minority class distribution. In this way, the number of subsamples will be equal to the number of samples required to involve the training sample size in a determined percentage (we named this value 'Cover%'), taking into account the size of the subsamples, that is, minority class size. Anyway, we think it is advisable to ensure a number of at least 5 samples. Expression (1) computes the number of samples to be generated.

$$NumberSamples = \min\left(5, \frac{SampleSize \times Cover\%}{MinorityClassSize}\right) \quad (1)$$

In concrete, for each dataset, we have used three different values for the number samples (N_S) parameter: a fixed value of 5 samples and the corresponding values of Expression (1) to cover 120% and 200% ('Cover%' value) of the examples in the

training sample. The table below summarizes the values obtained for the extreme values (Min, Max), as well as, the average (and median) value for all datasets.

Table 3. Summary of the main values of number samples used for all datasets

	Min	*Max*	*Mean*	*Median*
N_S=5	5	5	5	5
N_S 120%	5	156	17.48	8.00
N_S 200%	6	260	28.48	13.00

Therefore, we built 6 different consolidated trees for each one of the 25 folds of the cross-validation: 2 types of subsample size ('Tam=SizeOfMinClass' and 'Tam=Max') and 3 values for number samples ('N_S=5', 'N_S 120%' and 'N_S 200%').

We pruned all the consolidated trees, using the default pruning method proposed by Quinlan for C4.5 algorithm [14]. However, when the pruning process returned just the root node of the tree (the trivial classifier), we maintained the unpruned tree.

We tested the built trees with the test sample available in each fold and we measured their performance with the geometric mean (GM) such as Fernández et al did in [7], in order to obtain comparable results.

Finally, we used the same non-parametric statistical tests used by Fernández et al and proposed by Demšar in [6] and Garcia et al in [8] and [10] to evaluate the statistical significance of the results.

4 Experimental Results

As described in Experimental Methodology Section, we performed extensive experiments to compare the performance of the CTC algorithm with the results of 22 algorithms (6 classical methods and 16 GBML approaches), already analyzed in the referred work [7]. All of these 23 algorithms were evaluated on 33 datasets of the class imbalance context.

As Fernández et al did in [7], we used a hierarchical methodology for the comparison. First, we analyze the 6 different strategies to build the consolidated trees in order to select the most competitive one in this context. And then, we compare the selected one with the remaining algorithms of the comparison.

Table 4 shows the average results in test for all datasets using GM performance measure (± standard deviation) obtained by CTC algorithm. The average rank (and the rank position) for each of the 6 resampling strategies of CTC are also included. The best global result is stressed in bold-face. Results for each dataset can be downloaded from a web page associated to this work[3].

As it can be observed, the best results are obtained when the biggest sample size and the highest number of samples is used. In general, better results are obtained with 'Tam=Max' samples than with 'Tam=SizeOfMinClass' samples, and, in each type of sample size, the performance improves as N_S increases. This confirms our suspicion that it is important to cover most of the examples of the training sample with the set of

[3] http://www.sc.ehu.es/aldapa/2013/CTCvsGBML

samples generated for the consolidation process. This is vital in the most imbalanced datasets.

Table 4. Average results for the 6 resampling strategies for CTC algorithm

Resampling strategies		*GM*	*Avg. Rank*
Tam=	N_S=5	71.22 ± 5.18	4.41 (6)
SizeOf	N_S 120%	76.18 ± 4.02	3.77 (5)
MinClass	N_S 200%	77.34 ± 3.56	3.71 (4)
Tam=	N_S=5	75.65 ± 3.98	2.92 (2)
Max	N_S 120%	78.68 ± 3.05	3.30 (3)
	N_S 200%	**78.87 ± 2.61**	2.88 (1)

We want to notice that the second best strategy was N_S=5 with Tam=Max, instead of N_S 120% as we could expect. In a more in-depth analysis of the rankings obtained by each strategy in each dataset, it is observed that N_S=5 strategy achieved the first ranking in the most balanced datasets and N_S 200% strategy achieved the first ranking in the most imbalanced datasets. However, N_S 120% is able to achieve a good average value of GM measure, but not as good average rank.

In order to analyze the statistically significant differences between the obtained results, as explained before, we applied the same tests used by Fernández et al in [7]. First, we applied the Iman-Davenport test to detect differences among the 6 strategies. The null hypothesis can be rejected, since the p-value was 0.0055 (Iman-Davenport statistic was 3.4476), which was lower than the set α-value (0.05). So, we can say there are significant differences among the different strategies.

The next step is to carry out a powerful post-hoc procedure to discover between which pairs of algorithms (strategies in this case) the significant differences appeared. Table 5 shows the results associated to a Shaffer test where the adjusted p-value (APV) related to each comparison appears in brackets. Besides, a "+" implies that the strategy in the row is statistically better than the one in the column; whereas "-" implies the contrary; and "=" means that there are not significant differences.

The results in this table show that, in most of the cases, no significant differences were found by the test. However, differences were found between the two strategies with better results (N_S=5 and N_S %200 for Tam=Max) and the worst one (N_S=5 for Tam=SizeOfMinClass). We stressed these cases in the table with a gray background.

Table 5. Shaffer test for the 6 resampling strategies for CTC algorithm

		Tam=SizeOfMinClass			Tam=Max		
		N_S=5	N_S 120%	N_S 200%	N_S=5	N_S 120%	N_S 200%
Tam=	N_S =5	x	= (1.0)	= (0.9114)	- (0.0134)	= (0.1633)	- (0.0134)
SizeOf	N_S 120%	= (1.0)	x	= (1.0)	= (0.6544)	= (1.0)	= (0.5226)
ClassMin	N_S 200%	= (0.9114)	= (1.0)	x	= (0.7039)	= (1.0)	= (0.7039)
Tam=	N_S =5	+ (0.0134)	= (0.6544)	= (0.7039)	x	= (1.0)	= (1.0)
Max	N_S 120%	= (0.1633)	= (1.0)	= (1.0)	= (1.0)	x	= (1.0)
	N_S 200%	+ (0.0134)	= (0.5226)	= (0.7039)	= (1.0)	= (1.0)	x

After the latter analysis, we selected Tam=Max and N_S 200% strategy as representative of CTC algorithm. Then, we aimed at comparing its classification ability in imbalanced datasets with the main GBML approaches and some state-of-the art non-evolutionary rule induction methods. We used the same representative method found within each family of the GBML algorithms in the referred work [7]. For each method, average results calculated using the geometric mean, together with the associated ranking, are shown in Table 6. To carry out the statistical study, Fernández et al left out of the analysis the three worst methods within the family of non-evolutionary approaches (CART, AQ and CN2). The average rankings obtained for the remaining methods can also be found in the last column, Avg. Rank(*).

Table 6. Average results for the studied methods

	GM	*Avg. Rank*	*Avg. Rank(*)*
UCS	64.92 ± 16.55	6.76 (8)	5.85 (8)
SIA	69.62 ± 9.49	6.55 (7)	5.83 (7)
OCEC	70.88 ± 10.38	7.09 (10)	6.24 (9)
GASSIT	67.58 ± 9.69	6.24 (6)	5.60 (6)
Oblique-DT	75.81 ± 8.55	5.38 (5)	4.86 (5)
CART	69.72 ± 11.51	6.98 (9)	---
AQ	59.81 ± 8.72	9.58 (11)	---
CN2	45.97 ± 12.09	10.79 (12)	---
C4.5	73.28 ± 10.82	5.35 (4)	4.71 (4)
C4.5-Rules	75.21 ± 10.23	5.12 (3)	4.52 (3)
Ripper	**79.34 ± 8.98**	3.47 (1)	3.14 (1)
CTC	78.87 ± 2.61	4.70 (2)	4.24 (2)

As it can be observed, CTC is the second best algorithm, after Ripper, and followed by those based on C4.5. Checking for significant differences using Iman-Davenport, as in [7], the p-value is near to zero, and so, we can proceed with a Shaffer test. Comparing the results with the ones obtained in the referred work, the conclusions did not change qualitatively, that is, Ripper is statistically better than UCS, SIA, OCEC and GASSIT but it has not significant differences with those methods based on decision trees, neither for evolutionary ones nor for non-evolutionary ones, included CTC. Due to space limitation, Table 7 shows the results of the multiple comparison test performed on all algorithms but only for Ripper and CTC.

Table 7. Shaffer test for Ripper and CTC algorithms

	UCS	SIA	OCEC	GASSIT	Oblique-DT	C4.5	C4.5-Rules	Ripper	CTC
Ripper	+ (0.0016)	+ (0.0017)	+ (0.0001)	+ (0.0069)	= (0.2914)	= (0.5121)	= (0.8986)	x	= (1.0)
CTC	= (0.4819)	= (0.5121)	= (0.0843)	= (0.9485)	= (1.0)	= (1.0)	= (1.0)	= (1.0)	x

Since CTC with N_S=5 (and Tam=Max) also achieved competitive results, we wanted to make the same statistical analysis between this option and the remaining algorithms. From the point of view of the significant differences analysis, the conclu-

sions are qualitatively the same among all the algorithms, although CTC obtained the forth ranking in this case, after C4.5 and C4.5-Rules.

Finally, we wanted to corroborate whether CTC (without pre-processing) is a competitive algorithm when compared to the remaining algorithms with the application of the SMOTE pre-processing technique (the second part of the analysis of the GBML algorithms focused in class imbalance problems in the referred work [7]), such as it is suggested in [1]. In this case, CTC obtained the last position (ranking 9) with significant differences in favour of the three first algorithms: XCS, Gassit and C4.5. So, when working with smaller and more imbalanced datasets, CTC without SMOTE does not seem a competitive algorithm compared with the remaining algorithms with SMOTE pre-processing. To apply SMOTE before using CTC algorithm is a line we already started working: we experimented only with five subsamples and CTC obtained the forth ranking and without significant differences.

5 Conclusions and further work

In this work we have analyzed the performance of the CTC algorithm addressing more difficult classification problems, in concrete, smaller and more imbalanced datasets. For this, we used the results already published in a work done by Fernández et al in [7] where 22 different (16 evolutionary and 6 classical) algorithms for rule induction are compared on 33 imbalanced datasets.

The results show that CTC performs better than most of the algorithms, obtaining the second position in the average ranking among all algorithms, and besides, based on a Shaffer test, without statistically significant differences with the first algorithm, Ripper. However, significant differences appear with the first three algorithms when SMOTE is applied to the remaining algorithms (but not to CTC) in order to balance the training sample before learning process.

According to the use of SMOTE before using CTC, we expect that with a better tuning of the number samples parameter (also taking into account the class distribution), more competitive results can be obtained. Precisely, we would want to make a deeper analysis about the relation between the representation of the examples of both classes among samples and the performance of consolidated trees. Finally, we would like applying the consolidation process to other learning paradigms, as Ripper.

Acknowledgments. This work was funded by the University of the Basque Country, general funding for research groups (Aldapa, GIU10/02), by the Science and Education Department of the Spanish Government (ModelAccess project, TIN2010-15549), by the Basque Government's SAIOTEK program (Datacc2 project, S-PE12UN064) and by the Diputación Foral de Gipuzkoa. We would like to thank Francisco Herrera for his advise and his research group, SCI^{2}S, for making available the data partitions and results for this work, as well as the tools for analysis of statistically significant differences.

6 References

1. Albisua I., Arbelaitz O., Gurrutxaga I., Muguerza J., Perez J.M.: C4.5 consolidation process: An alternative to intelligent oversampling methods in class imbalance problems. In: Lozano, Jose A.; Gámez, José A.; Moreno-Pérez, José A. (Eds.) Advances in Artificial Intelligence, CAEPIA'11, LNAI, Vol. 7023, Springer, pp. 74–83 (2011)
2. Albisua I., Arbelaitz O., Gurrutxaga I., Lasarguren A., Muguerza J., Perez J.M.: Analysis of the effect of changes in class distribution in C4.5 and consolidated C4.5 tree learners. Technical Report. EHU-KAT-IK-01-12, University of the Basque Country (UPV-EHU) (2012).http://www.sc.ehu.es/aldapa/Argitalpenak/12.EHU-KAT-IK-01-2.pdf
3. Alcalá-Fdez J., Fernandez A., Luengo J., Derrac J., García S., Sánchez L., Herrera F.: KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. Journal of Multiple-Valued Logic and Soft Computing 17:2-3, 255-287 (2011)
4. Bache, K., Lichman, M. UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science (2013)
5. Chawla N. V., Bowyer K. W., Hall L. O., Kegelmeyer W. P.: SMOTE: Synthetic Minority Over-sampling Technique, Journal of Artificial Intelligence Research, 16, 321-357 (2002)
6. Demšar J.: Statistical Comparisons of Classifiers over Multiple Data Sets. Journal of Machine Learning Research, 7, 1-30 (2006)
7. Fernández A., García S., Luengo J., Bernadó-Mansilla E., Herrera F.: Genetics-Based Machine Learning for Rule Induction: State of the Art, Taxonomy and Comparative Study. IEEE Transactions on Evolutionary Computation, 14, 913-941, (2010)
8. García S., Herrera F.: An Extension on "Statistical Comparisons of Classifiers over Multiple Data Sets" for all Pairwise Comparisons. Journal of Machine Learning Reserarch, 9, 2677-2694 (2008)
9. García S., Fernández A., Herrera F.: Enhancing the effectiveness and interpretability of decision tree and rule induction classifiers with evolutionary training set selection over imbalanced problems. Applied Soft Computing, 9, 1304–1314 (2009)
10. García S., Fernández A., Luengo J., Herrera F.: Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental Analysis of Power. Information Sciences, 180, 2044–2064 (2010)
11. He H., Garcia E.: Learning from imbalanced data. IEEE Trans. Knowl. Data Eng. 21(9), 1263–1284 (2009)
12. Pérez J.M., Muguerza J., Arbelaitz O., Gurrutxaga I., Martín J.I.: Consolidated Tree Classifier in a Car Insurance Fraud Detection Domain with Class Imbalance. In: S. Singh et al. Pattern Recognition and Data Mining. LNCS, Vol. 3686, Springer, pp. 381-389, (2005)
13. Pérez J.M., Muguerza J., Arbelaitz O., Gurrutxaga I., Martín J.I.: Combining multiple class distribution modified subsamples in a single tree. Pattern Recognition Letters 28, 4, 414-422 (2007)
14. Quinlan J.R.: C4.5 "Programs for Machine Learning". Morgan Kaufmann Publishers Inc.(eds), San Mateo, California (1993)
15. Weiss G.M., Provost F.: Learning when Training Data are Costly: The Effect of Class Distribution on Tree Induction. Journal of Artificial Intelligence Research 19, 315-354 (2003)
16. Wu X., Kumar V., Quinlan J. R., Ghosh J., Yang Q., Motoda·H., McLachlan G. J., Ng A., Liu B., Yu P. S., Zhou Z.-H., Steinbach M., Hand D. J. Steinberg D.: Top 10 algorithms in data mining. Knowledge and Information Systems 14, 1–37 (2008)
17. Yang Q., Wu X.: 10 challenging problems in data mining research. International Journal of Information Technology & Decision Making. 5(4), 597–604 (2006)

Clustering based navigation profiling in a tourism website using Cluster Validity Indices to select the best partition

O. Arbelaitz, I. Gurrutxaga, A. Lojo, J. Muguerza, J. M. Pérez, and I. Perona
{olatz.arbelaitz,i.gurrutxaga,aizea.lojo,j.muguerza,txus.perez,
inigo.perona}@ehu.es

Dept. of Computer architecture and Technology
University of the Basque Country
M. Lardizabal, 1, 20018 Donostia, Spain

Abstract. The huge increase of information on the web makes web personalisation essential. The work presented in this paper builds a system that profiles the users browsing the Bidasoa Turismo website and uses link prediction to adapt it. The process uses clustering techniques combined with frequent pattern mining techniques to build user profiles that will then be used to propose the adaptations to new users. Precision and recall values show that the system is successful. This work also faces the problem of selecting the best partition in a clustering process. It includes an analysis of the problem that arises to Cluster Validity Indices (CVIs) in databases with a large number of clusters and, proposes a procedure to select a pseudo optimal partition based on some of those CVIs. Values of the calculated performance metrics confirm the adequateness of the procedure.

Keywords: Web personalisation, Web Usage Mining, User profiles, Link prediction, Clustering, Cluster Validty Index

1 Introduction

In recent decades, information on the web has increased dramatically and this introduces a need of adaptation to the preferences and needs of the users; i.e., web personalization [1] becomes essential. Web personalization can be defined as the set of actions that are useful to dynamically adapt the presentation, the navigation scheme and content of the Web based on preferences, abilities, or user requirements.

The navigation scheme can be adapted based on the navigation experience of previous users extracted from the logs stored in web servers. That is, user profiles generated based on log information can be used to predict the navigation of new users, propose or emphasize links according to the prediction and thus improve their browsing experience.

Tourism websites are not away from this adaptation requirement. The tourism industry experienced a shift from offline to online travellers. Experts underlined many years ago that the Internet is the main source of information in the tourist domain [2]. An increasing number of travellers are no longer dependent on travel agencies to look for information for their trips, they have replaced using agencies by the use of the Internet. As e-Destinations serve as platforms where consumers can be inspired, get information about the desired destination and eventually book the holiday, the presence of destinations in the web is crucial [3]. Moreover, the web facilities provided to tourists affect their loyalty [4].

Clustering algorithms are often used to group into the same segment users with similar navigation characteristics. One of the main problems of most of the unsupervised classification algorithms is the selection of the optimal number of clusters or parameter K so that the partition that best matches the data is obtained. The process of estimating how well a partition fits the structure underlying the data is known as cluster validation [5]. In the case of navigation log based user profiling, there is not prior knowledge of the structure of the data what makes compulsory the validation based on the partitioned data; i.e., using *Cluster Validity Indices* (CVI).

Previous works found in literature compare some CVIs with the aim of obtaining some general conclusions about their behaviour [6, 7]. Nevertheless, all of them have performed the evaluation in databases with few clusters (up to 6 or 8 in general); being 15 the greatest number of clusters explored [7] and just for a single real database. To our knowledge there is no study on how CvIs deal with large databases where the data require to be partitioned in more clusters but they seem to have some problems.

The work presented in this paper uses clustering techniques combined with frequent pattern mining techniques to build user profiles from the logs in a tourism website and then, based on the obtained profiles, propose adaptations to new users navigating in the website. Moreover, the work includes an analysis of the behaviour of a set of CVIs in a database that requires a large number of clusters and, proposes a procedure based on some of those CVIs to select a good quality partition.

The next section describes the data environment where the system was built. Section 3 describes the profiling system, Section 4 describes the tools used generally to select the best partition. In Section 5 we show the experiments and main results of the work and, finally, we draw some conclusions and suggest some possible future works in Section 6.

2 Application Environment

The application environment was a tourism website from our environment: Bidasoa Txingudi bay, which is located at the western tip of the Pyrenees mountains and, straddling two countries, France and Spain, links the Basque provinces of Gipuzkoa and Lapurdi. The area offers the opportunity of a wide range of tourism activities and, Bidasoa Turismo website (BTw), *www.bidasoaturismo.com*, in-

cludes all sorts of practical tourist information to visit the area: thematic tourism, professional tourism, gourmet tourism, agenda, suggestions, etc.

Nearly ten months of usage data for BTw were used in the experiments: from January 2012 to October 2012. The tourism agents provided us with a database containing a total of 3,636,233 user requests.

3 The system

When the aim is to find use or interaction patterns in the web that allow to model the users' behaviour so that adequate recommendations or adaptations can be done, the process is called web usage mining [8]. The work presented in this paper is a web usage mining application and it can be divided into two main steps: data acquisition and preprocessing and pattern discovery and analysis.

3.1 Data acquisition and preprocessing

The data contained in this database was obtained from BTw and belongs to web server logs of user requests. The contained information is in the standardized text file format, i.e. Common Log Format which is the minimum information saved on a web server.

We removed erroneous requests and the indirect requests of images, videos, style or functionality sent to the browser due to user clicks to work with the requests directly related with the user activity. We then removed requests related to the web administration activity which could be detected by analysing the URLs' subfields. We thereafter selected and normalized the format of the parameters of the URLs so that only those that really influence the web pages' final appearance were taken into account. We completed user identification based on IP addresses and fixed the expire time of each session to 10 minutes of inactivity [9]. We selected the most relevant sessions (the ones with higher activity level: 3 or more clicks) and removed the longest sequences (the ones with more than 86 requests: out of 98% percentile), with the assumption that long sequences are outliers and might be caused by robots, such as, crawlers or spiders (googleboot appears as main user agent). We represented each session as a sequence of URLs where a session of length l could be represented as:

$$S_i = u_i^1 u_i^2 ... u_i^l \tag{1}$$

Where u_i^l corresponds to the lth URL in the sequence i.

We focused our work in the stable part of the site and with this aim, we deleted from the click sequences the URLs related to more dynamically changing parts such as cultural programming and news. We further eliminated accesses to *home* URL because we consider that it does not provide any information about the user navigation preferences and the proposal of *home* URL would not make easier the navigation of new users. After these preprocessing stages we obtained a database of 17,891 sessions with an average length of 7.7 clicks per session (median = 5 and stdv=7.7).

3.2 Pattern Discovery and Analysis

Before starting with the pattern discovery and analysis phase we performed a preliminary analysis to decide whether or not any structure exists in the data. We used Edit Distance [10] normalized with the length of the longest compared sequence as a metric to compare sequences, obtained the distance matrix of the complete database and plotted it using heat map graphical representation of data where the cases in the heat map were ordered using hclust hierarchical clustering algorithm based on single linkage. We concluded that the database is disperse; many of the sequences are not similar to any other sequence since their distance is very big. This suggests that it would be very difficult to find structure in the complete database; clustering algorithms generally partition the data in a concrete number of clusters what makes very probable to be obliging to group together patterns that are very far away from each other.

The results of the described analysis lead us to divide the database in two parts. The one used for profiling contains the closer sessions (12,238) added for the first time to a cluster at a distance smaller or equal to 0.5 and the one with more distant examples, containing the remaining sessions (5,653).

We built user profiles using PAM (Partitioning Around Medoids) [11] clustering algorithm with Edit Distance [10, 12] as metric to compare sequences. The aim was to group into the same segment users that show similar navigation patterns. PAM, as many other clustering algorithms, requires the K parameter to be estimated.

For profiling each cluster of group of users, we used SPADE (Sequential PAttern Discovery using Equivalence classes) [13], an efficient algorithm for mining frequent sequences, able to extract the most common click sequences in the cluster. The application of SPADE provides for each cluster a set of URLs that are likely to be visited for the sessions belonging to it. The number of URLs proposed depends on SPADE parameters such as minimum support and maximum allowed number of proposals per cluster. In our system we fixed minimum support to a small value (0.2) so that enough URLs were proposed to profile a cluster. This allowed us to experiment with two values for the maximum allowed number of proposals per cluster: 3 and 4.

4 Selection of the best partition

When no insight of the structure of the data exists, CVIs are used to select the best partition. The selection of the CVI to be used is not obvious and therefore, most of the studies for CVIs have been done for partitions with few clusters.

The most extensive CVI comparison we found in bibliography [7] compares the performance of 30 CVIs in many synthetic databases and 20 real databases.

From the CVIs evaluated in [7] we selected the ones we found implementations that could be directly applied to distance matrices (summarized in Table 1) and we had to adapted some of them replacing the centroids by medoids (centroid cannot be calculated in sequential representation). Note that some of the

indices require data points for their calculations and having the web navigation sessions represented as URL sequences does not allow this possibility.

CVIs tend to measure the compactness and separation of the clusters based generally in inter-cluster and intra-cluster distances and estimating a trade-off between them. Depending on the combination, the index can be of one of two types: a lower value of the index means a "better" partition (marked with a down arrow) or a higher value of the index means a "better" partition (marked with an up arrow).

Table 1. List of explored CVIs.

Index		Short description
DunnMST (D)	↑	Index proposed in 1997 by Pal and Biswas.
Gamma (G)	↓	Index proposed in 1975 by Baker and Hubert.
C-Index (CI)	↓	Index proposed in 1976 by Hubert and Levin.
DaviesMST	↓	Index proposed in 1997 by Pal and Biswas.
Davies*	↓	Index proposed in 2005 by Kim and Ramakrishna.
Silhouette	↑	Index proposed in 1987 by Rousseeuw.
CS	↓	Index proposed in 2004 by Chou, Su and Lai.
Score function	↑	Index proposed in 2007 by Saitta, Raphael and Smith.
COP	↓	Index proposed in 2010 by Gurrutxaga et al.
SV	↑	Index proposed in 2011 by Žalik and Žalik.
OS	↑	Index proposed in 2011 by Žalik and Žalik.

5 Experimental results and analysis

We applied a hold-out strategy and divided the database obtained after the cleaning and preprocessing phase in three parts: training data (8,567 examples, 70%), validation data (2,447 examples, 20%) and test data (1,224 examples, 10%). We used the training data to generate the data partitions and corresponding user profiles, the validation set to tune the models and the test data to estimate their prediction capacity respectively.

5.1 Selection of the best partition

The first aim of our study was to determine the best partition and to analyse the behaviour of the CVIs in this context. We thus applied PAM to the training data, generated partitions with 10, 15, 20, 25, 50, 75, 100, 150 and 200 clusters and finally, calculated the values of the 11 CVIs for each of the partitions.

It is a matter of fact that the CVI evaluations we found in bibliography are for partitions with few clusters (generally up to 8 or at most 15) and the increase of the number of groups could affect or even invalidate the effect of one or both components, intra-cluster distance or inter-cluster distance, taken into account

in the indices; i.e. the increase in the number of clusters might affect as much as the quality of the partition to the values obtained for the CVIs.

Based on the trends of the values obtained by the CVIs in this experiment we could divide them into two main groups:

- A group of 6 CVIs that fluctuates and, as a consequence, does not show any coherent behaviour or does not give any insight about the characteristics of the best partition.
- A group of 5 CVIs that show a trend and as a consequence, makes possible the analysis of the quality of the partition based on CVIs. The CVIs in this group are DunnMST, C-Index, OS, COP and Gamma.

The CVIs in the latter group, that is, the ones showing a trend, improve their value as K increases, what would suggest us to generate new partitions with larger number of clusters. This could be considered a bias of the indices but a further analysis showed that there is a point where the variation of the index value (increment or decrement depending on the type of index), suffers a change in slope which could be considered to be the point obtaining a trade-off between the quality of the partition and K. Thus, it could separate the values of K that make the number of clusters become dominant in the value of the index. We therefore hypothesise that the improvement in the value of the CVI before this value is due to the quality of the partition and after this value it is due to the bias of the index (the effect of the large number of clusters in the inter-cluster and/or intra-cluster components and their combination is bigger than the effect of the quality of the partition). Moreover, the maximum value of K explored, 200, is bigger than the usual exploration limit for the number of clusters: $\sqrt{n}$ (being n the number of examples).

In order to find the trade-off point we propose to analyse the derivative of the values obtained for each of the CVIs in the second group and to use as stop criteria the change of magnitude order of the decrement or increment on the obtained index value. When the change happens, the previous K will be considered the one obtaining the best partition. Table 2 shows the values of the derivatives for the 5 indices, where the values selected according to the stop criteria are marked in bold.

Table 2. Derivatives of the values obtained for the CVIs showing a trend.

Clusters (K)	D↑	CI↓	OS↑	COP↓	G↓
15	0.000000	0.008569	-0.007116	-39.469200	0.004325
20	0.000000	-0.005739	-0.002989	-25.098500	-0.007791
25	0.000000	-0.005433	-0.003357	**-21.061840**	-0.006066
50	**0.002103**	**-0.003017**	-0.001871	-6.977660	**-0.003889**
75	0.000667	-0.000280	**-0.001148**	-2.978823	-0.000534
100	0.000000	0.000365	-0.000586	-1.666815	-0.000170
150	0.000000	-0.000326	-0.000447	-0.850714	-0.000424
200	-0.000462	-0.000054	-0.000303	-0.442405	-0.000212

Based on the proposed rule, 3 of the CVIs, DunnMST, C-Index and Gamma, propose as best partition the one with 50 clusters whereas OS proposes the one with 75 clusters and COP the one with 25 clusters. We decided to use a majority vote strategy which would suggest that the best number of clusters is near 50.

5.2 Evaluation of the generated models and comparison to the CVI based decision

The aim of this section is to evaluate the generated system and to confirm the outcome of the previous subsection according to the obtained results.

With this aim, we evaluated profiles generated for each of the partitions generated with the training sample (10, 15, 20, 25, 50, 75, 100, 150 and 200 clusters), according to the test sample which would be equivalent to new users navigating in the website. The system was evaluated selecting a profile for the new session which will then be compared to its click sequence. We simulated a real situation using the first 25% (more or less 2 links if the click sequence has around 7.7 links) of the new sequences to select the profile of the new user based on the built model.

Previous works [14] showed that the characteristics of the new users might be a mixture of some of the profiles discovered with the training set; their profile can have similarities with more than one profile and thus, the diversification makes sense; it is better to build the profiles of the new users dynamically according to some of their nearest profiles. We propose the use of k-Nearest Neighbour supervised learning approach to calculate the distance of the click sequence (Edit Distance to the medoid) of the new users to the clusters generated in the profiling phase. We selected $k = 5$ because if the examples in a cluster are not similar enough, SPADE might not able to extract a profile for them because no URL exceeds the required support value. We then combined the profiles of the two nearest clusters with defined profiles, using their distance to fix URL selection probabilities; they contain the URLs with higher support in each of the clusters. We experimented with a maximum number of 3 and 4 URLs. Both options obtained similar values for the three performance metrics but we selected 3 as best option according to the validation sample.

We computed performance metrics based on the results obtained for each one of the new users. We compared the number of proposed links which are really used in the test or validation examples (hits) and the number of proposals that are not used (misses) and calculated precision, recall and and F0.5-measure. We measured two characteristics through these performance metrics: the concordance of the generated profiles with the profiles of the new users navigating in the website (using the whole sequences in the calculation) and how good the generated profiles would be for link prediction when new users are navigating (using the rest of the sequence, 75%, in the calculation).

Table 3 shows average results (precision (Pr), recall (Re) and F0.5-measure (F0.5)) obtained with the test set for different values of K. The second column, N URLs, shows the average number of links proposed in each profile and it shows that, although the number of proposed URLs was fixed to 3, for some

of the cases fewer links are proposed. The remaining columns are divided into two parts, the ones that show values for the evaluation of the concordance of the profiles (Profiles) and the ones related to the evaluation of the system for link prediction (Prediction). The solutions obtained with the partitions selected according to the CVIs appear shaded whereas the best solution according to the values of the performance metrics appear underlined in the table.

Table 3. Precision, Recall and F0.5 measure values obtained when evaluating profile concordance (left) and link prediction capacity (right).

		Profiles			Prediction		
Clusters (K)	N URLs	Pr	Re	F0.5	Pr	Re	F0.5
10	2.9	0.341	0.277	0.326	0.232	0.237	0.233
15	3.0	0.434	0.378	0.421	0.300	0.323	0.305
20	2.8	0.460	0.380	0.441	0.321	0.325	0.322
25	2.8	0.460	0.381	0.441	0.317	0.321	0.318
50	3.0	0.551	0.496	0.539	0.383	0.416	0.389
75	2.9	0.523	0.460	0.509	0.340	0.359	0.343
100	2.9	0.545	0.487	0.532	0.353	0.376	0.357
150	2.9	0.573	0.511	0.559	0.363	0.385	0.367
200	2.9	0.554	0.494	0.541	0.346	0.366	0.349

The results in Table 3 show that the profiles generated by the system have a considerable concordance with the profile of new users the values of the performance metrics are greater than in other state of the art publications [15]. The solution selected using the CVI based procedure (K = 50) is the break even point in this experiment; more than 50% of the proposed links are hits, and, moreover, nearly 50% of the places navigated by the new users are guessed by the system. Although with slightly lower values, the system shows similar behaviour when used for link prediction, so, it will help new users to achieve their objectives faster. We can therefore state that CVI based rule we proposed seems adequate because the best results for link prediction are obtained with the selected partitions and in the case of the concordance of profiles, the difference of those partitions with the ones obtaining the best values is not more than 0.02.

The results suggest that the CVI based procedure defined in the previous section is adequate to find the best partition, or at least a good quality one, in a clustering problem with a large number of clusters.

5.3 Behaviour of the system for distant examples

Although the profiles were built according to the closer examples, we also evaluated their concordance and usability for link prediction in the case of distant examples. Table 4 shows the results obtained for these examples by the system selected according to the CVI based rule (K=50).

Table 4. Precision (Pr), recall (Re) and F0.5-measure (F0.5) values obtained when evaluating profile concordance (left) and link prediction capacity (right) for distant examples.

		Profiles			Prediction		
Clusters (K)	N URLs	Pr	Re	F0.5	Pr	Re	F0.5
50	2.9	0.512	0.171	0.366	0.228	0.089	0.174

Although generally the values of the performance metrics worsen, precision values are maintained even for distant examples. This means that even in the cases the system was used for distant users the proposed links would still be useful for them. Recall values however drop meaning that there was another part of their navigation that our system was not able to model. When analysing these results we have to take into account that the sessions grouped in the distant sample are usually longer (average length of 11.3 clicks) and we still propose at most 3 URLs per profile, what might explain the drop in recall values.

6 Conclusions and further work

The work presented in this paper builds a profiling and link prediction system for the users of a tourism website. After an analysis of the structure of the provided data and a division of the database, clustering techniques were combined with frequent pattern mining techniques to build user profiles and then, propose adaptations to the new users navigating the website. The work also includes an analysis of the behaviour of a set of CVIs in a database with elevate number of clusters and proposes a procedure to select the best partition based on some of these CVIs.

The experiments show that the system is able to generate profiles that are concordant with the profile of the new users and useful for link prediction. Furthermore the obtained profiles seem to represent part of the structure of a set of distant examples that have not been used in the modelling process. Furthermore, the obtained results suggest that the CVI based procedure we defined to find the best partition seems to be adequate in a clustering problem with a large number of clusters.

As this work treats several problems, it opens the door to further analysis in many directions. Related to the database, a deeper analysis and comparison of the navigation areas for the profiled sessions and the distant ones could probably help to improve the performance of the system. In the case of distant examples, simply proposing more links would probably help to improve recall values. An analysis of the effect of K in the inter-cluster and intra-cluster distance components used to calculate the different CVIs would also help to refine the procedure proposed to select the best partition.

Acknowledgements. This work was funded by the University of the Basque Country, general funding for research groups (Aldapa, GIU10/02), by the Science and Education Department of the Spanish Government (ModelAccess project, TIN2010-15549), by the Basque Governments SAIOTEK program (Datacc2 project, S-PE12UN064) and by the Diputacin Foral de Gipuzkoa.

References

1. Pierrakos, D., Paliouras, G., Papatheodorou, C., Spyropoulos, C.D.: Web usage mining as a tool for personalization: A survey. User Modeling and User-Adapted Interaction **13**(4) (November 2003) 311–372
2. Gretzel, U., Yuan, Y., Fesenmaier, D.: Preparing for the new economy: Advertising strategies and change in destination marketing organizations. Journal of Travel Research **39**(2) (2000) 146–156
3. Pan, B., Fesenmaier, D.R.: Travel information search on the internet: a preliminary analysis. In Proceedings of the International Conference in Helsinki, Finland. (2003) 242–251
4. Hsu, C.I., Shih, M.L., Huang, B.W., Lin, B.Y., Lin, C.N.: Predicting tourism loyalty using an integrated bayesian network mechanism. Expert Systems with Applications **36**(9) (2009) 11760 – 11763
5. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: On clustering validation techniques. Journal of Intelligent Information Systems **17** (2001) 107–145
6. Milligan, G.W., Cooper, M.C.: An examination of procedures for determining the number of clusters in a data set. Psychometrika **50**(2) (June 1985) 159–179
7. Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J.M., Perona, I.: An extensive comparative study of cluster validity indices. Pattern Recognition **46**(1) (2013) 243 – 256
8. Mobasher, B.B.: 12 web usage mining. Encyclopedia of Data Warehousing and Data Mining Idea Group Publishing (2006) 449–483
9. He, D., Gker, A.: Detecting session boundaries from web user logs. In: Proceedingsof of the BCS-IRSG 22nd Annual Colloquium on Information Retrieval Research. (2000) 57–66
10. Gusfield, D.: Algorithms on Strings, Trees, and Sequences - Computer Science and Computational Biology. Cambridge University Press (1997)
11. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data: An Introduction to Cluster Analysis. 9th edn. Wiley-Interscience (March 1990)
12. Chordia, B., Adhiya, K.: Grouping web access sequences using sequence alignment method. Indian Journal of Computer Science and Engineering (IJCSE) **2**(3) (June 2011) 308–314
13. Zaki, M.J.: Spade: An efficient algorithm for mining frequent sequences. Machine Learning **42**(1/2) (2001) 31–60
14. Arbelaitz, O., Gurrutxaga, I., Lojo, A., Muguerza, J., Prez, J., Perona, I.: Adaptation of the user navigation scheme using clustering and frequent pattern mining techniques for profiling. Proceedings of the 4th International Conference on Knowledge Discovery and Information Retrieval (KDIR) (2012) 187–192
15. Senkul, P., Salin, S.: Improving pattern quality in web usage mining by using semantic information. Knowledge and Information Systems **30**(30) (February 2012) 527–541

Distributed Column Formation in Swarm Robotics for Simple Agents

Pilar Arques, Fidel Aznar, Mireia Sempere

Departamento de Ciencia de la Computación e Inteligencia Artificial.
Universidad de Alicante
{arques, fidel, mireia}@dccia.ua.es

Abstract. This paper presents a flocking algorithm inspired by the movement of ants forming columns when gathering food and return to the nest. In our approach, all agents have the same behaviour using a distributed algorithm so there is no leader to follow. Moreover, the only information they use is their infrared sensors. This flocking algorithm is based on four basic behaviours: wander, follow an agent, yield and stop to avoid collisions. In this way, this algorithm could be implemented in very cheap hardware platforms (with few computational requirements) using a large number of agents.
In a distributed behaviour all agents are equally important. If a particular agent is the head of a column, it is not more important than the rest of the swarm and the loss of this agent causes no change in the execution of the macroscopic behaviour. This is very important to ensure the fault tolerance of the system.
We have tested it successfully with both, real robots (using 16 Khepera III) and with simulation with up to 1000 agents. We have developed both, the percentage of agents that are in columns formation and the scalability of this algorithm.

1 Introduction

In swarm robotics, flocking is a widely studied problem. Numerous papers have been written on the topic [1–8].

In some of these works, communication between agents is needed [1, 8] and in others a virtual leader is defined [7]. In some works where communication between agents in not required [3], robots have a totally emergent behaviour, and scalability is not demonstrated. In [4] a flocking algorithm based on sensor readings is defined. This algorithm uses attraction-repulsion measures between robots to achieve a homogeneous movement. In this study real robots are used, but they need a compass and some communication between them in certain steps of the algorithm.

In flocking algorithms, rules defined by Reinolds are widely used [6].These rules are also known as *cohesion*, *separation* and *alignment*.

In this paper we propose a new fully distributed flocking algorithm, in which agents will be able to form columns, without any communication between them,

and without the definition of any leader. Our main purpose is to propose a basic simple behaviour, inspired by ants, with low computing requirements. This behaviour would be easy to implement in systems with low computing capability and it could be the basis for more complex behaviours.

In this case, individuals take into account only the information perceived by their infrared sensors. Using a finite state machine, each robot will perform one of the following basic movements depending on the data perceived: wander, follow, yield or stop. It is important to highlight that a robot never knows what happens behind itself, neither if it is the leader nor if it wander alone. This way, a high fault tolerance algorithm is obtained (there is not defined a leader or a robot with a specific behaviour to be the head of a column).

Based on Reinolds rules, our algorithm implements the following behaviours: **Cohesion:** follow the robot in front. **Separation:** stop and yield to avoid collision. **Alignment:** try to establish the velocity to avoid that a robot wanders alone in front of the others.

In this algorithm, robots are totally independent without knowledge about the environment and without any capability of communication with other robots. This simplicity makes the algorithm suitable for working with a large number of agents and with a very economical and computational cost. Therefore, a swarm robotic algorithm, totally distributed, with fault tolerance, simple and scalable is defined.

In section 2, the flocking algorithm will be presented. In experimental sections, the performance of the algorithm with real robots and with a simulator with large number of robots will be analyzed. Finally, conclusions and future lines will be presented.

2 Algorithm description

In this work robots are located in a quasi-random way [1]. The main objective of robots is to move forward forming columns (one robot behind the other), like ants when gathering food or return to the nest, but in this case, without using pheromones neither chemical nor virtual. Robots are circular, equipped with infrared sensors located in a ring around it, that measure distances to obstacles. The main movement of each robot will be to move forward, either because it has no obstacles or because it is following another robot.

When a robot detects something by one of its sensors, the robot will follow the object detected by its middle sensor, although a closer robot is detected by another sensor. It is important to have into account that a robot will stop to avoid collisions. Additionally, when a robot follows another robot, it will adapt its velocity to avoid a collision.

If two robots are facing one in front the other, they can be blocked. In order to avoid this possible limitation of the algorithm, it is established that if a robot is paused during an interval of time, it must take a random turn to unblock.

[1] Robots are randomly located and oriented but only in the left upper quadrant of the environment.

Initially all robots are in wander(W) behaviour. They remain in the same behaviour while there is no change in its sensors readings. In general terms, the proposed flocking algorithm is defined by figure 1, transitions between states are explained below:

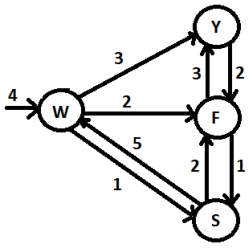


Fig. 1. Automaton showing the behaviours. (W) Wander, (F) Follow, (S) stop and (Y) yield.

Formally, α corresponds to an agent of the system. For each agent: S_c is the set of central sensors, S_i is the set of left sensors and S_d the set of right sensors. $Vel(\alpha)$ current robot speed and $Dir(\alpha)$ its direction angle. We also define D_{col} as the safety distance of the robot to avoid collisions and D_{max} as the maximum distance at which sensors detect an obstacle. Figure 1 shows an automaton with the following behaviours:

1. **Avoid collision.** Before performing any movement, the robot must ensure that it is a safe movement and it will not collide with another robot.
   `if` $(S_c \leq D_{col}$ `or` $S_d \leq D_{col}$ `or` $S_i \leq D_{col})$ `then`
   $Vel(\alpha) = 0$
2. **Follow.** When a robot detects another robot, the robot should try to follow it. α_i and α_j are two agents.
   `if` $(S_c > D_{col})$ `then`
   $Vel(\alpha_i) = Vel(\alpha_j)$
3. **Correct its trajectory.** If an agent detects another agent only by its side's sensors, the agent must correct its trajectory in order to follow it. There are two possible situations depending on if the agent detects the other agent by its left or right sensors. This is very important in the algorithm. If two robots detect each other (one on the right, and the other on the left), both rotate 90° creating a blockage. In order to avoid this possible situation, if a robot detects another agent on the right, it must give way to the other agent, wait until the other agent stand in front of it and then follow it. And if an agent detects other agent on the left, it must correct its trajectory.
   `if` $(S_d > D_{col}$ `or` $S_i > D_{col})$ `then`
   if $(S_d > D_{col})$ `then`

$Vel(\alpha) = 0$ // Yield
if $(S_i > D_{col})$ then
$Dir(\alpha_i) = Dir(\alpha_j)$ // Correct its trajectory

4. **Wander.** If no obstacle is detected, the robot will move forward doing a small zig-zag in its movement.
   if $(S_c > D_{max}$ and $S_i > D_{max}$ and $S_d > D_{max})$ then
   $\Delta Vel(\alpha) = k * Vel(\alpha)$
   $\Delta Dir(\alpha) = k * Dir(\alpha)$
   k is a random value in $-\pi/4$ .. $\pi/4$ range.
5. **Unblocked.** If a robot is blocked, it executes a randomly rotation.
   if $(S_c \leq D_{col}$ and $S_d \leq D_{col}$ and $S_i \leq D_{col})$ then
   $Dir(\alpha) = k * Dir(\alpha)$

2.1 Behaviour of each independent robot

For any agent i, the selection of each behaviour is determined by following predicates:

$$FOLLOW(i) \Leftrightarrow (\exists\, t)_{\leq n}\, [(D_{col} \leq S_t < D_{max}) \wedge (S_t \in S_c)]$$

$$WANDER(i) \Leftrightarrow (\forall\, t)_{\leq n}\, (S_t > D_{max})$$

$$CORRECT(i) \Leftrightarrow (\exists\, t)_{\leq n}\, [(D_{col} \leq S_t < D_{max}) \wedge (S_t \in S_d) \wedge (\forall j)_{\leq n} (j = t) \vee (S_j > S_{max})]$$

$$YIELD(i) \Leftrightarrow (\exists\, t)_{\leq n}\, [(D_{col} \leq S_t < D_{max}/2) \wedge (S_t \in S_d)]$$

$$STOP(i) \Leftrightarrow (\exists\, t)_{\leq n}\, (S_t \leq D_{col})$$

Vel_{max} will be default speed. The speed of each agent i will be estimated by the following function:

$$Vel(i) = \begin{cases} Vel_{max} & \text{if } FOLLOW(i) \\ k * Vel_{max} & \text{if } WANDER(i) \\ 0 & \text{if } STOP(i) \vee YIELD(i) \vee CORRECT(i) \end{cases}$$

Wander speed is multiplied by a reduction coefficient ($k = 0.75$), so that, robots can find each other and will not advance in isolated way.

In order to calculate the robot steering angle, the angle for an instant in time $Dir(i)^{t+1}$ must be calculated. Previous direction $Dir(i)^t$ and the angles $\theta(S_t)$ at which each robot sensor is situated are taken into account.

$$\beta(i) = \begin{cases} max(\theta(S_t)) \text{ if} |max(\theta(S_t))| = max\,(\theta(S_t)) \\ min(\theta(S_t)) \text{ in other case} \end{cases}$$

$$Dir(i)^{t+1} = \begin{cases} Dir(i)^t & \text{if } FOLLOW(i) \vee WANDER(i) \vee STOP(i) \vee YIELD(i) \\ \beta(i) & \text{if } CORRECT(i) \end{cases}$$

Two robots can be blocked if both are facing one in front the other. In order to resolve this potential conflict situation, we decided to perform random turns that vary the direction of the robot between $-\pi/4$ and $\pi/4$ radians. The system

need to know the time T that a robot remains in a given state to detect this situation:

$$BLOCK(i) \Leftrightarrow (\exists t)_{\leq n}(\exists j)_{\leq n}(t \neq j) \wedge (S_t \in S_d) \wedge (S_j \in S_d) \wedge (T > 2s)$$

In this case, two random numbers, $rand_1$ and $rand_2$, are generated, range from 0 to $\pi/4$. Robot speed will be expressed by the speed of each of the wheels:

$$Vel(i) = \begin{cases} Vel_1 = rand_1 * Vel_{max} \\ Vel_2 = -rand_2 * Vel_{max} \end{cases}$$

With this defined movement, robot will take a random direction where usually find another robot to follow, though it may possible that it remains isolated.

3 Experimentation with real robots, Khepera III

In order to determine the correctness of the algorithm, real robot tests, with Khepera III robots, were carried out. These robots are circular and are equipped with 9 infrared sensors located in a ring around it, thus the robot is able to detect any object around it (360°), as shown in Khepera III user manual[2]. We have taken into account Khepera III features to determine the initial parameters of the algorithm: maximum speed, avoid collision distance and minimum distance to detect objects.

In Khepera III robots, a new sensor (S_i) measurement is taken every $33\,ms$ and $20\,cm$ (D_{max}) is the maximum distance at which an object is detected. Khepera III has 9 sensors around it. In this paper, only 6 (n) sensors located on the front and side of the robot are used. Thus, we define the following sets: $S_i = \{S_1, S_2\}$; $S_c = \{S_3, S_4\}$ y $S_d = \{S_5, S_6\}$. Sensors numbered as 0, 7 and 8 are not used in this algorithm because they are located at the back of the robot, and robots do not need information from those sensors.

Initially, we started working with V-REP[3] simulator and LUA scripts because the detection and correction of errors was easier than in real robots and at the same time, we obtained a code that was directly interpretable by Khepera III. To execute LUA scripts in robots, it is only need to delete simulator headers from the scripts. In our case, we also took the following determinations: reduce maximum speed, defining $\pi/4\,rad/seg$ (Vel_{max}) and maximize collision distance $5cm$ (D_{col}), in both cases in order to avoid any damage in real robots.

The results of the execution of four tests with 16 Khepera III robots, with different initial configurations are showed in table 1. These results have been obtained empirically. For each test, following data is shown: Percentage of robots at **wander** state at the end of the simulation. That is, the robot is isolated or it is the head of a formation; percentage of robots at **follow** state at the end of the simulation.That is, the robot is part of a column; number of **columns** at the end of simulation; and number of **isolated** robots at the end of simulation.

[2] http://ftp.k-team.com/KheperaIII/UserManual/Kh3.Robot.UserManual.pdf
[3] Virtual Robot Experimentation Platform. http://www.coppeliarobotics.com

Trial	Wander	Follow	Columns	Isolated
1	31%	69%	4	1
2	25%	75%	3	1
3	31%	69%	5	0
4	12.5%	87.5%	2	0

Table 1. Summary of results obtained after 4 algorithm executions, using 16 Khepera III robots

In tests, almost 70% of the robots performed the follow behaviour, as can be seen in table 1. In two cases, the algorithm performance is totally correct, since no robot is isolated, that is, all robots are part of a column (or in the head, or behind other robot).

The image sequence 2 shows the evolution of the algorithm using, in this example, 6 Khepera III robots. In the initial image robots are grouped (aggregation is not a subject of this work). Following images show the evolution of the algorithm. After each sensor reading, agents perform a single action: advance, correct its trajectory or stop. The robot that is on the head develops a wander behaviour, and the other robots arrange in the line.

Fig. 2. Image sequence that shows the evolution of algorithm with 6 Khepera III

4 Experimentation with simulator

We have also used MASON [9] simulator to study the scalability of the algorithm. We work with up 1000 agents, obtaining satisfactory results. Simulated robots have a diameter of 1 unit and are equipped with sensors that allow the robot to detect any other agent located less than 3 units.

In experimental tests, all agents are initially randomly positioned in the upper left corner of the environment and the simulation ends when robots reach any edge of the environment. Environment size varies depending on the number of agents in the swarm. For groups from 100 to 300 agents, a 100×100 environment is used, for less than 900 agents a 200×200 environment and groups up to 1000 agents a 300×300 environment. Different sizes of environment are used to ensure the isolation, and initial safety distance between agents.

The sequence of images (Fig. 3) shows the evolution of the behaviour for 400 agents from its initial position up to 1250 iterations. Initial positions are random,

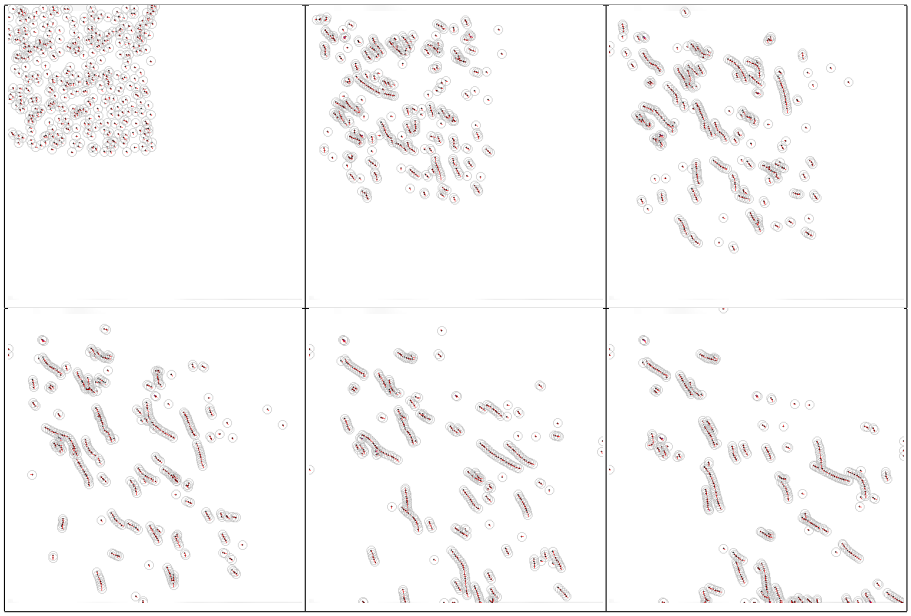


Fig. 3. Evolution of swarm behaviour. Example with 400 robots. From initial position until 1250 iterations, displayed every 250 iterations.

having into account a minimum safe distance to avoid robots overlapping in the initial state. Initial angles are random as well within a certain range $-\pi/4..\pi/4$, so that all formations tend to address the bottom right corner.

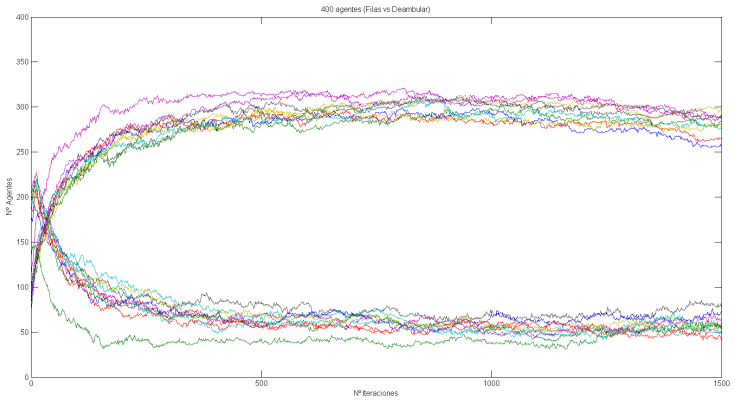


Fig. 4. Number of agents on follow behaviour (upper curves) vs. wander behaviour (bottom curve), regards number of iterations. 10 different initial configurations. 400 agents. 1500 iterations.

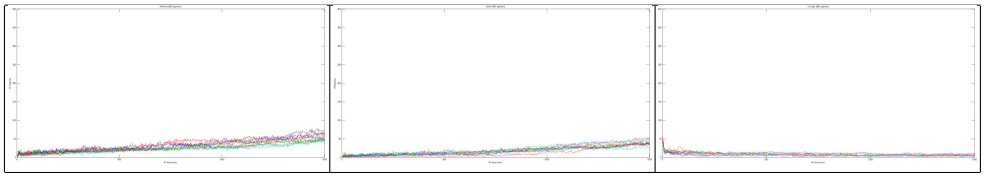


Fig. 5. Number of agents in each behaviour: stop (1), yield (2) and correct (3) regards number of iterations. 10 different configurations. 400 agents. 1500 iterations.

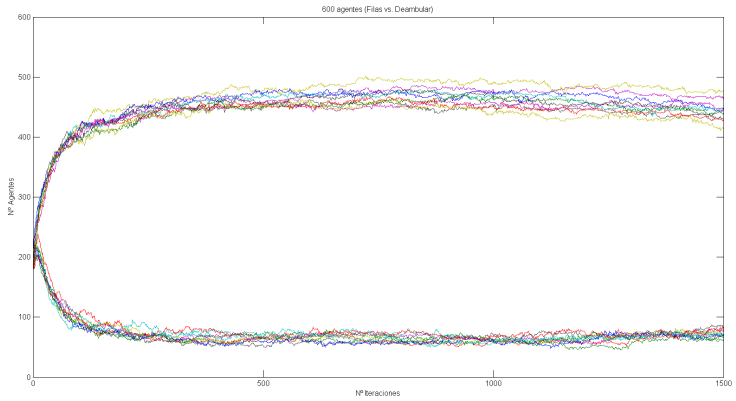


Fig. 6. Number of agents on follow behaviour (upper curves) vs. wander behaviour (bottom curve), regards number of iterations. 10 different initial configurations. 600 agents. 1500 iterations.

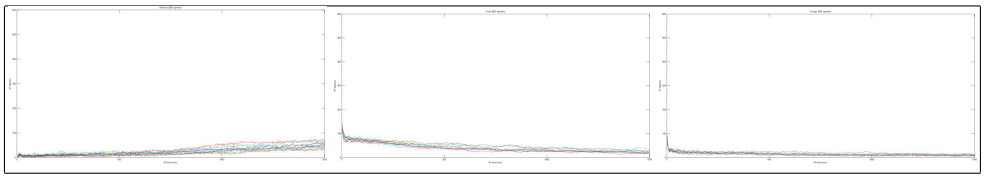


Fig. 7. Number of agents in each behaviour: stop (1), yield (2) and correct (3) regards number of iterations. 10 different configurations. 600 agents. 1500 iterations.

The evolution of the different basic behaviours of robots over 1500 iterations, is shown in figures 4, 5, 6, 7, 8, 9. The algorithm is been tested with different number of agents, from 100 to 1000 robots. For each group, experimental tests are repeated 10 times. As the initial position and orientation of robots is random, the results vary slightly on the evolution of behaviours. The first graph from

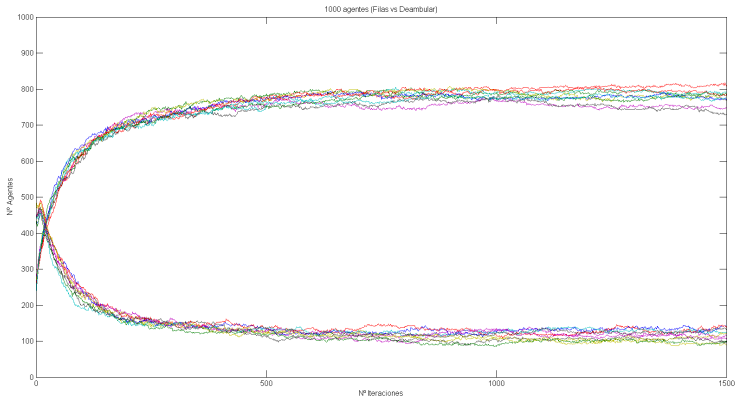


Fig. 8. Number of agents on follow behaviour (upper curves) vs. wander behaviour (bottom curve), regards number of iterations. 10 different initial configurations. 1000 agents. 1500 iterations.

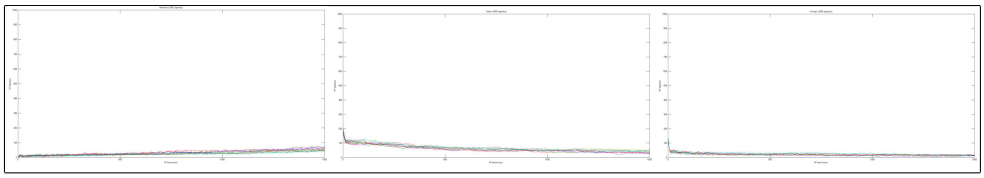


Fig. 9. Number of agents in each behaviour: stop (1), yield (2) and correct (3) regards number of iterations. 10 different configurations. 1000 agents. 1500 iterations.

each block, 4, 6 and 8, shows a comparison between follow behaviour, those robots that detect something in front sensors, and wander behaviour, robots with no obstacles in front. In graph, upper curves show the follow behaviour and lower curves the wander behaviour. Figures 5, 7 and 9 show the other possible behaviours: stop, yield and correct the trajectory

In the sequence of images of figure 3 can be seen that from the beginning robots tend to join forming columns. Charts 4, 6 and 8 show that more than 70% of robots are in follow behaviour during all the simulation.

These charts show the scalability of the system, and the possibility of working with large groups of robots. The proportion of robots in each behaviour is similar for different sizes of swarm. The majority of robots are in follow state since the beginning of simulation, and the rest of robots are divided, more or less, equally between the other behaviours.

5 Conclusions

In this paper we presented an algorithm inspired by the movement of ants and applied to swarm robotics. We proposed a fully decentralized model, fault-tolerant and scalable. We demonstrated its correct performance from 6 to 1000 robots.

In our proposal, robots, starting from quasi-random positions, are able to form columns like ants or worms.

A fully distributed fault-tolerant algorithm has been defined. In this algorithm agents move forward forming columns. Several tests that verify the correct performance of the algorithm with Khepera III robots, as can be seen in figure 2 and table 1, and with a simulator have been presented. In both cases, a high percentage of robots that are part of the columns is obtained. We also demonstrated that the loss of a member of the swarm is not crucial to the achievement of the global objective, as verified, in the execution sequence 3.

We have shown that the algorithm is scalable to a very large swarms. As shown in graphs 8, 9 with 1000 robots. Graphs, 4, 6, 8 show that, in all instances, more than 70% of the robots are aligned in a column, and less than 20% of the robots are wandering (part of this 20% are those that go at the head of the formations and also form columns). And figures 5, 7, 9, show that over 10% of robots are in other behaviours. In all these charts can be appreciated that curves are similar, even when working with different size of swarms, demonstrating the scalability of our algorithm.

Our future work will focus on adding to robots more complex behaviours. For example, target points. So, we will also work with sensors located under the robot.

Acknowledgements. This work has been supported by Ministerio de Ciencia e Innovación, project TIN 2009-10581.

References

1. Hauert, S., Leven, S., Varga, M., Ruini, F. Reynolds flocking in relity with fixed-wing robots: communication range vs. maximun turning rate. I. C. on Intelligent Robots and Systems, (2011) 5015-5020
2. Liang, X., Li, X. A New Decentralized Planning Strategy for Flocking of Swarm Robots. Journal of Computers. Vol. 5. Number 6 (2010) 914-921
3. Moeslinger, C., Schmickl, T., Crailsheim, K. A Minimalist Flocking Algorithm for Swarm Robots. E. C. on Advances in Artificial Life. (2011) 375-382
4. Navarro, I., Gutierrez, A., Matía, F., Monasterio, F.: An Approach to Flocking of Robots Using Minimal Local Sensing and Common Orientation. Hybrid Artificial Intelligent Systems (2008) 616-624
5. Olfati-Saber, R. Flocking for Multi-Agent Dynamic Systems: Algorithms and Theory. IEEE Transactions on Automatic Control, Vol. 51, Number 3, (2006) 401-420
6. Reynolds, C. W. Flocks, Herds, and Schools: A Distributes Behavioural Model. Computer Graphics, Volume 21, Number 4, (1987) 25-34
7. Housheng S., Xiaofan W., Guanrong C. (2009): A connectivity-preserving flocking algorithm for multi-agent systems based only on position measurements, I. J. of Control, 82:7, 1334-1343
8. Wen, G., Duan, Z., Li, Z., Chen, G. Flocking of multi-agent dynamical systems with intermittent nonlinear velocity measurements I.J.R.N.C. **22** (2012) 1790-1805
9. Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K. A new multi-agent simulation toolkit. Proceedings of the 2004 SwarmFest Workshop (2004)

Aplicación del Algoritmo SPEA2 en el Problema de Gestión de Áreas de Localización en Redes Móviles

Víctor Berrocal-Plaza, Miguel A. Vega-Rodríguez, Juan M. Sánchez-Pérez

Dept. de Tecnología de los Computadores y de las Comunicaciones
Universidad de Extremadura, Campus Universitario S/N, 10003
Cáceres, España
{vicberpla, mavega, sanperez}@unex.es

Resumen. Este artículo presenta nuestra adaptación del algoritmo Strength Pareto Evolutionary Algorithm 2 (SPEA2) para optimizar las tareas de uno de los sistemas de gestión más importantes en las redes móviles terrestres: el Sistema de Gestión de Localización. Las funciones de cualquier sistema de gestión de localización son el rastreo de los terminales móviles y la localización del abonado destinatario de la llamada. De entre las técnicas de gestión de localización destaca la estrategia de Áreas de Localización por ser una de las más utilizadas en las redes actuales. Esta estrategia define un problema de optimización con dos funciones objetivo en conflicto: minimizar el coste de actualización de localización y minimizar el coste de paginación. En trabajos anteriores se propuso optimizar una única función objetivo que fuese combinación lineal de las dos funciones de coste anteriores. Sin embargo, en este trabajo se propone el uso de un algoritmo de optimización multi-objetivo para evitar los inconvenientes de la combinación lineal de funciones objetivo y para obtener un conjunto de soluciones entre las cuales el operador de red puede seleccionar aquella que mejor se adapte al estado real de la red de señalización. Los resultados experimentales muestran que nuestra adaptación de SPEA2 obtiene mejores soluciones que los algoritmos propuestos en otros trabajos, así como las ventajas de la optimización multi-objetivo.

Palabras clave: Áreas de Localización, Gestión de Localización Móvil, Optimización Multi-objetivo, Strength Pareto Evolutionary Algorithm 2.

1 Introducción

Las redes de telefonía móvil pública se caracterizan por prestar en todo momento servicios de comunicaciones móviles independientemente de la localización en la que se encuentren sus abonados. Además, estas redes deben atender una alta demanda de tráfico con un número de frecuencias muy limitado. Para ello, la zona de cobertura deseada se divide en zonas de menor tamaño conocidas como células o celdas, entre las cuales las distintas frecuencias de las que dispone el operador de red son distribuidas y reutilizadas. Estas celdas pueden ser de diferente tamaño en función de las tolerancias de los emplazamientos y de la densidad de tráfico de la zona cubierta, requi-

riéndose celdas de menor tamaño en aquellas zonas con mayor densidad de tráfico (es decir, el coeficiente de reutilización de frecuencias aumenta con la densidad de tráfico) [1]. La división celular de la zona de cobertura requiere de un sistema de gestión que controle el movimiento de los abonados y localice la celda exacta en la que se encuentra cada usuario destinatario de llamada, este sistema es el Sistema de Gestión de Localización [2]. El seguimiento de los abonados por la red móvil se controla mediante la gestión de la actualización de la localización. Este procedimiento es iniciado por el terminal móvil de acuerdo a un método preestablecido, por ejemplo: cada vez que se cambia de celda (actualizar siempre), cada cierto número de celdas visitadas (actualización basada en distancia), cada cierto número de cruce de celdas (actualización basada en movimiento), cada vez que se cambia de área (actualización basada en áreas de localización), cada vez que se entra en una celda determinada (actualización basada en celdas de notificación), o no actualizar nunca. Por otro lado, la paginación o radiobúsqueda es iniciada por la red para conocer la celda exacta en la que se encuentra el usuario destinatario de la llamada. Este procedimiento consiste en el envío de mensajes de búsqueda entorno a la última localización conocida, ya sea de forma secuencial (con un máximo de 3 pasos) o de forma simultánea.

En este trabajo se estudia la estrategia de gestión basada en Áreas de Localización (AL) con un procedimiento de radiobúsqueda en el que todas las celdas a paginar se encuestan simultáneamente, técnica ampliamente utilizada en las redes móviles actuales. Este problema es conocido como Problema de Planificación de Áreas de Localización (PPAL) y pertenece al grupo de problemas NP-Completos debido al gran tamaño del espacio de objetivo [3]. Esto último, unido al incremento exponencial en el número de abonados, hace que sea realmente interesante aplicar metaheurísticas para encontrar configuraciones de Áreas de Localización que minimicen la carga del Sistema de Gestión de Localización. P. R. L Gondim aplicó en [3] un Algoritmo Genético (GA) para encontrar configuraciones quasi-óptimas de Áreas de Localización. P. Demestichas et al. aplicaron tres algoritmos (Tabu Search, Simulated Annealing, y Algoritmo Genético) para estudiar el problema de planificación de Áreas de Localización en diferentes entornos [4]. Posteriormente, J. Taheri y A. Y. Zomaya desarrollaron y estudiaron un conjunto de redes de prueba en las que los patrones de movilidad y llamada de usuario son similares a los que podemos encontrar en redes actuales. En sus trabajos, J. Taheri y A. Y. Zomaya implementaron los siguientes algoritmos: red neural de Hopfield [5], Simulated Annealing [6], Algoritmo Genético [7] y varias combinaciones de la red neural de Hopfield con el Algoritmo Genético [8]. Estas mismas redes fueron estudiadas por S. M. Almeida-Luz et al. en [9] mediante el algoritmo Differential Evolution, y serán las que estudiaremos en este artículo.

Cabe destacar que en todos los algoritmos definidos anteriormente se utiliza la combinación lineal de las funciones objetivo, técnica que permite el uso de métodos de optimización con menos complejidad computacional pero que tiene asociada varios inconvenientes. En primer lugar, la combinación lineal de funciones objetivo requiere de un conocimiento muy preciso del problema para poder asignar correctamente los valores de los coeficientes de ponderación (que pueden ser valores reales). En segundo lugar, un algoritmo de optimización mono-objetivo sólo proporciona una única solución, aquella que mejor satisface su función objetivo. Y por último, la com-

binación lineal no tiene en cuenta que la carga de la red es variable con el tiempo, y por tanto se requeriría de un valor diferente de los coeficientes de ponderación para cada estado de la red. Es por eso que, en este artículo, se propone el uso de un algoritmo de optimización multi-objetivo: Strength Pareto Evolutionary Algorithm 2, (SPEA2), el cual proporciona un conjunto de soluciones, cada una de ellas relacionada con un compromiso específico entre objetivos y entre las cuales el operador de red puede seleccionar aquella que mejor se adapte al estado real de la red en cada momento.

La Sección 2 de este documento explica detalladamente el Problema de Planificación de Áreas de Localización. La Sección 3 muestra nuestra adaptación del algoritmo SPEA2 para resolver este problema. Los resultados experimentales y la comparativa con otros autores se detallan en la Sección 4. Finalmente, las conclusiones y posibles trabajos futuros se discuten en la Sección 5.

2 Áreas de Localización

La estrategia basada en Áreas de Localización (AL) es una de las más utilizadas en las redes móviles actuales para gestionar automáticamente la actualización de la localización de los abonados. En este esquema, las celdas de red se agrupan en áreas lógicas (o Áreas de Localización: grupo continuo y no solapado de celdas) de tal forma que el abonado sólo actualiza su posición (celda en la que se encuentra) cuando se mueve a una celda que tiene asociada un área de localización diferente a la última celda actualizada. Además, como la posición de los abonados es conocida a nivel de áreas de localización, sólo se paginan las celdas pertenecientes a la última área de localización actualizada. Por tanto, el principal reto de esta estrategia en encontrar la configuración de Áreas de Localización que minimice el coste de actualización de localización al mismo tiempo que se minimiza el coste de paginación. Debe tenerse en cuenta que estos dos costes son conflictivos, ya que para reducir al mínimo el coste de actualización todas las celdas deberían pertenecer al mismo Área de Localización (es decir, cuando no existe actualización de localización), dando lugar a un coste de paginación máximo puesto que todas las celdas de la red deben ser encuestadas cada vez que un abonado recibe una llamada. Y viceversa, para reducir al mínimo el coste de paginación, cada celda de red debería pertenecer a un Área de Localización diferente (es decir, cuando la localización de los abonados es conocida a nivel de celda), lo que provocaría un coste de actualización máximo debido a que se realizaría una actualización de localización cada vez que un usuario cambia de celda. Así, esta estrategia define un problema de optimización multi-objetivo cuyas funciones objetivo son:

$$\boldsymbol{f}_1 = \min\left\{ \mathrm{LU}_{\mathrm{cost}} = \sum_{i=1}^{\mathrm{NCell}} \sum_{j=1}^{\mathrm{NCell}} \rho_{i,j} \times \mathrm{N}_{i,j} \right\}, \tag{1}$$

$$\boldsymbol{f}_2 = \min\left\{ \mathrm{PA}_{\mathrm{cost}} = \sum_{k=1}^{\mathrm{NArea}} \sum_{i \in \mathrm{CA}_k} \mathrm{NIC}_i \times \mathrm{NA}_k \right\}, \tag{2}$$

bajo la siguiente restricción:

$$\sum_{k=1}^{\mathrm{NArea}} \mu_{i,k} = 1, \ \forall i \in [1, \ \mathrm{NCell}]. \tag{3}$$

Donde $\boldsymbol{f}_1$ es la primera función objetivo: minimizar el coste de actualización de localización ($\mathrm{LU}_{\mathrm{cost}}$). En esta ecuación $\rho_{i,j}$ es una variable binaria que es igual a 1 cuando la celda i y la celda j pertenecen a áreas de localización diferentes, en caso contrario su valor es 0. $N_{i,j}$ es el número de abonados que se mueven desde la celda i hacia la celda j. Y NCell es el número de celdas de red. Por otro lado, $\boldsymbol{f}_2$ es la segunda función objetivo: minimizar el coste de paginación ($\mathrm{PA}_{\mathrm{cost}}$). En esta última ecuación NIC_i es el número de llamadas entrantes en la celda i, NA_k es el número de celdas que pertenecen al área de localización k. Y NArea es el número de áreas de localización presentes en la red. En la ecuación (3), $\mu_{i,k}$ es una variable binaria que es igual a 1 cuando la celda i pertenece al área de localización k, lo que implica que una celda sólo puede pertenecer a una única área de localización.

En trabajos anteriores, estas dos funciones objetivo se combinaron linealmente para formar una única función objetivo a optimizar, véase la ecuación (4) donde β es un coeficiente de ponderación definido para tener en cuenta que el coste de actualización de localización podría ser mayor al coste de paginación. Comúnmente, este coeficiente toma el valor de 10. Los algoritmos con los que comparamos utilizan la ecuación (4) como función objetivo [5-9].

$$\boldsymbol{f}_3(\beta) = \beta \times \boldsymbol{f}_1 + \boldsymbol{f}_2. \tag{4}$$

3 Strength Pareto Evolutionary Algorithm 2

El algoritmo Strength Pareto Evolutionary Algorithm 2 (SPEA2) es una metaheurística multi-objetivo basada en población (donde cada individuo de la población codifica una posible solución del problema) en la que se utilizan los operadores evolutivos típicos de sistemas biológicos (cruce o recombinación de padres, mutación de la descendencia y selección natural) para mejorar iterativamente un conjunto de soluciones. E. Zitzler et al. propusieron en [10] este algoritmo como una versión mejorada de su predecesor, SPEA [11]. Básicamente, SPEA2 es un algoritmo genético elitista con un archivo en el que se almacenan las mejores soluciones encontradas y cuya función de fitness se muestra en la ecuación (5). Esta ecuación está compuesta por dos términos principales: la estimación de la densidad y el Raw fitness. La estimación de la densidad se calcula a partir de la distancia euclidiana entre el vector objetivo $\mathbf{z}^i$ y su k-ésimo vector objetivo más cercano ($d(\mathbf{z}^i, \mathbf{z}^k)$), donde k es la raíz cuadrada de la suma del tamaño de la población y del tamaño del archivo. Y el Raw fitness se define como el sumatorio de "fuerzas" de las soluciones que dominan a la solución procesada ($\mathbf{z}^j \prec$

$\mathbf{z}^i$), donde la "fuerza" de una solución o individuo ($\boldsymbol{S}(\mathbf{z}^j)$) es el número de soluciones dominadas por esa solución. Por definición, una solución o individuo se dice que domina a otro cuando el primero mejora al segundo en al menos un objetivo, y ambos son iguales en el resto de objetivos. Además, se entiende por vector objetivo $\mathbf{z}^i$ al vector que almacena el valor de las funciones objetivo (ecuaciones (1) y (2)) de la solución o individuo i. Cabe destacar que los mejores individuos serán aquellos que minimicen esta función de fitness, la cual se ha definido para valores normalizados de los vectores objetivo. El motivo de la normalización es garantizar que ambos objetivos (ecuaciones (1) y (2)) tienen la misma influencia en la función de fitness.

$$\boldsymbol{f}_{\text{fitness}}^{\text{SPEA2}}\left(\overline{\mathbf{z}^i}\right) = \frac{1}{2+\text{d}\left(\overline{\mathbf{z}^i},\overline{\mathbf{z}^k}\right)} + \sum_{\text{j}\in \text{P}_t+\text{P}_t^{\text{arch}},\overline{\mathbf{z}^j}\prec\overline{\mathbf{z}^i}} \boldsymbol{S}\left(\overline{\mathbf{z}^j}\right) \tag{5}$$

El pseudo-código de nuestra adaptación de SPEA2 se muestra a continuación. Este pseudo-código define un método iterativo que se ejecuta hasta que no se alcanza la condición de parada. En este trabajo se utiliza la misma condición de parada que la utilizada en los trabajos con los que se compara: número máximo de generaciones. El primer paso consiste en inicializar y evaluar la población. Para ello se generan, reparan y evalúan N_{POP} soluciones aleatorias, véase Sección 3.1. A continuación, se crea un archivo vacío de tamaño N_{ARCH} en el que se irán almacenando las mejores soluciones encontradas, y se inicia el bucle principal del algoritmo. La primera operación dentro del bucle principal consiste en evaluar la función fitness de todas las soluciones o individuos disponibles (almacenadas tanto en la población como en el archivo)

Algoritmo 1. Pseudo-código de nuestra adaptación de SPEA2

```
1:    %Inicializar la población
2:    Indv ← Inicializacion (N_POP);
3:    %Evaluar la población: funciones objetivo
4:    Indv ← EvaluacionFO(Indv);
5:    %Crear archivo vacío
6:    Arch ← CrearArchivo(N_ARCH);
7:    %Bucle principal
8:    while (condicion de parada ≠ TRUE) {
9:       %Evaluar la población: función de fitness
10:      [Indv,Arch]← EvaluacionFF(Indv,Arch);
11:      %Copiar los mejores individuos en el archivo
12:      Arch ← Copiar(Arch, Indv);
13:      %Aplicar el operador de cruce
14:      Indv ← CruceElitista (Arch, P_C);
15:      %Aplicar el operador de mutación
16:      Indv ← Mutacion (Indv, P_M);
17:      %Evaluar la población de descendientes
18:      Indv ← EvaluacionFO(Indv);}
```

y copiar los N_{ARCH} mejores individuos (aquellos que minimizan la función de fitness, ecuación (5)) en el archivo, reemplazando a los previamente almacenados. Posteriormente, estos N_{ARCH} mejores individuos se utilizan para generar la población de descendientes mediante la operación de cruce, véase Sección 3.2. A continuación, se aplica la operación de mutación a la población de descendientes obtenida en el paso anterior, véase Sección 3.3. Finalmente, se evalúan las funciones objetivo para cada uno de los individuos de la población de descendientes y se comienza con la siguiente iteración. Tras cada operador evolutivo (cruce y mutación), se aplica una función de reparación cromosómica para transformar soluciones inviables en soluciones factibles.

3.1 Representación del Individuo e Inicialización de la Población

Cada individuo de la población codifica una solución factible del problema. En este trabajo se utiliza una codificación vectorial en la que se almacena el Área de Localización asociada con cada celda de red. Con el objetivo de crear la primera población de individuos, cada vector se llena con un patrón aleatorio de 0s y 1s, el cual se utilizará para determinar la configuración de Áreas de Localización correspondiente. Para ello, cada Área de Localización estará formada por grupos continuos y no solapados de celdas con el mismo valor de vector. La Figura 1 muestra un ejemplo de inicialización de individuo, donde el número central representa el Área de Localización asignada y número situado en la parte superior representa el identificador de dicha celda.

3.2 Operación de Cruce

En este trabajo se ha implementado una operación de cruce elitista que se realiza con probabilidad P_C. Para ello, cuatro individuos seleccionados al azar se agrupan en parejas, siendo cada padre el mejor individuo de cada grupo. Posteriormente, se fraccionan y mezclan ambos padres. Esta operación proporciona dos hijos pero sólo el mejor de ellos se almacenará en la población de descendientes. El número de puntos de cruce así como sus posiciones se determinan aleatoriamente en los intervalos [1, 4] y [0, NCell-1] respectivamente. La Figura 2 muestra la operación de cruce seguida en este trabajo.

3.3 Operación de Mutación

La operación de mutación se aplica con probabilidad P_M sobre la población de descendientes para modificar levemente el genotipo de los individuos. En este trabajo se han definido dos operaciones de mutación: mutación a nivel de gen y mutación a nivel de Área de Localización (AL). La mutación a nivel de gen consiste en incorporar una celda frontera entre dos o más Áreas de Localización a su Área de Localización vecina de menor tamaño (en términos de número de celdas). Un ejemplo de mutación a nivel de gen se muestra en la Figura 3. Por otro lado, la mutación a nivel de Área de Localización consiste en fusionar el Área de Localización con menor número

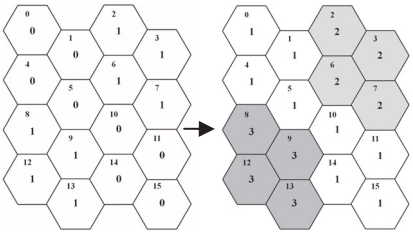


Fig. 1. Inicialización de individuo

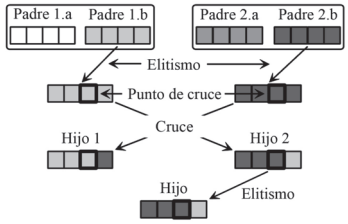


Fig. 2. Operación de cruce

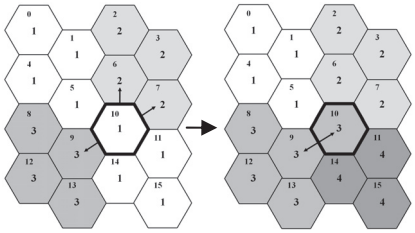


Fig. 3. Mutación a nivel de gen

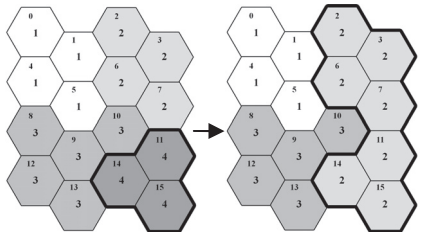


Fig. 4. Mutación a nivel de AL

de celdas con su Área de Localización vecina de menor tamaño. La Figura 4 muestra un ejemplo de mutación a nivel de Área de Localización.

4 Resultados Experimentales

Con el propósito de conocer la calidad de nuestro algoritmo, y por ende de nuestras soluciones, se han estudiado cuatro redes de prueba de diferente complejidad y se han comparado nuestros resultados con los obtenidos por otros autores sobre las mismas instancias. Todas estas redes han sido desarrolladas en los trabajos de J. Taheri y A. Y. Zomaya y tienen la particularidad de que presentan patrones de movilidad y de llamada de usuario similares a los que se pueden encontrar en las redes móviles actuales [5-8]. Estas redes reciben el nombre de LA25 (red regular de 5x5 celdas), LA35 (red regular de 5x7 celdas), LA49 (red regular de 7x7 celdas) y LA63 (red regular de 7x9 celdas). En trabajos anteriores, estas redes han sido ampliamente estudiadas con diferentes tipos de metaheurísticas (red neural de Hopfield (HNN) [5], Simulated Annealing (SA) [6], Algoritmo Genético (GA) [7], combinaciones de la red neural de Hopfield con el Algoritmo Genético (GA-HNN) [8] y Evolución Diferencial (DE) [9]), todas ellas pertenecientes al campo de la optimización mono-objetivo. Debido a que no es posible una comparación con otras aproximaciones multi-objetivo, en este trabajo se compara nuestra adaptación de SPEA2 con los algoritmos desarrollados en los trabajos [5-9]. Para ello, debemos buscar en nuestro conjunto de soluciones factibles y no dominadas (también conocido como frente de Pareto) la solución que mejor satisface la función objetivo utilizada en estos trabajos, que es la ecuación (4) con β igual a 10. Tras esto, se comparará esta solución con la mejor solución encontrada por cada algoritmo de optimización mono-objetivo. Además, se utiliza el Hipervolumen (HV) para conocer la calidad de nuestro algoritmo en el contexto multi-objetivo [12].

Tabla 1. Comparativa con otros trabajos ($f_3(10)$)

Red	Algoritmo							
	SPEA2	HNN[5]	SA[6]	GA[7]	GA - HNN1[8]	GA - HNN2[8]	GA - HNN3[8]	DE[9]
LA25	26990	27249	26990	28299	26990	26990	26990	26990
LA35	39832	39832	42750	40085	40117	39832	39832	39859
LA49	60849	63516	60694	61938	62916	62253	60696	61037
LA63	89085	92493	90506	90318	92659	91916	91819	89973

Table 2. Datos estadísticos del Hipervolumen

Red	$HV_{medio}(\%)$	$HV_{\sigma}(\%)$
LA25	74.937	0.126
LA35	79.434	1.034
LA49	81.163	1.820
LA63	81.225	1.746

El Hipervolumen es un indicador multi-objetivo que, en el caso de dos objetivos, mide el área del espacio objetivo factible cubierta por el frente de Pareto. Para ello, el área del espacio objetivo factible se acota mediante los puntos de referencia. En el problema de planificación de Áreas de Localización, estos puntos de referencia se calculan a partir de las dos configuraciones extremas: cuando todas las celdas pertenecen al mismo Área de Localización (coste de actualización mínimo y coste de paginación máximo) y cuando cada celda de red pertenece a un Área de Localización diferente (coste de actualización máximo y coste de paginación mínimo).

Para realizar una comparativa justa, nuestra adaptación de SPEA2 y los algoritmos con los que comparamos utilizan el mismo tamaño de población ($N_{POP} = 250$) y el mismo número de generaciones (5000). El resto de parámetros (tamaño del archivo o N_{ARCH}, probabilidad de cruce o P_C y probabilidad de mutación o P_M) se han configurado por medio de un estudio paramétrico de 30 ejecuciones independientes por experimento. La configuración elegida es aquella que maximiza el HV en las cuatro redes de prueba estudiadas, esta configuración es: $N_{ARCH} = 600$, $P_C = 0.90$ y $P_M = 0.26$.

En la Tabla 1 se muestra la comparativa de nuestro algoritmo con los algoritmos desarrollados en [5-9]. Esta tabla revela que nuestra adaptación de SPEA2 mejora a los algoritmos propuestos por otros autores, siendo esta mejora mayor a medida que aumenta la complejidad de la red. Tarea nada trivial pues nuestro algoritmo, especializado en encontrar un amplio conjunto de soluciones, compite contra algoritmos especializados en encontrar una única solución. En la Tabla 2 se muestran los datos estadísticos del HV (media y desviación estándar) de 30 ejecuciones independientes en cada una de las instancias. Y en la Figura 5 se observan los frentes de Pareto con HV más cercano al HV medio. Esta figura pone de manifiesto que la densidad de puntos en cada frente de Pareto decrece a medida que aumenta el coste de paginación, esto es debido a que el número de soluciones factibles decrece con el incremento en el tamaño de las Áreas de Localización. Además queda patente que el coste de pagina-

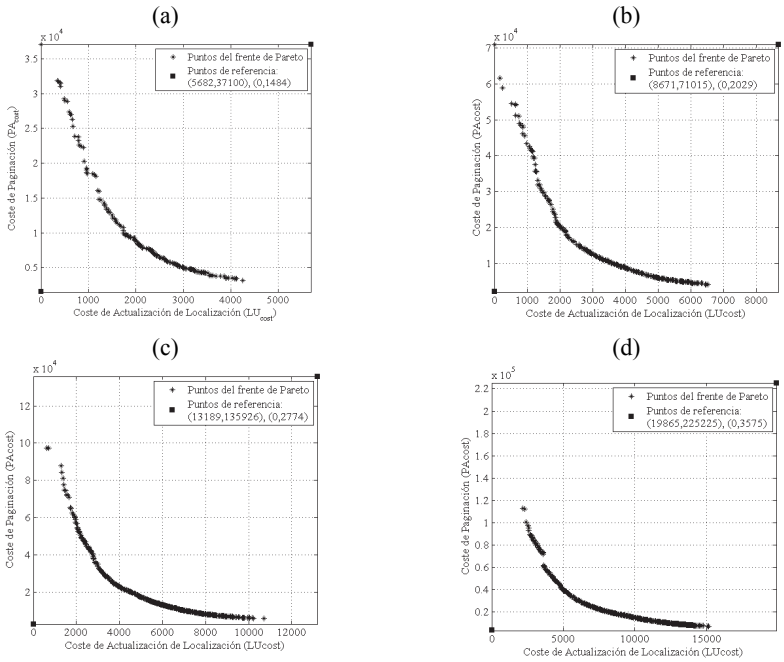


Fig. 5. Frente de Pareto obtenido: (a) LA25, (b) LA35, (c) LA49, (d) LA63

ción crece más rápido que el coste de actualización, siendo el factor que limita el tamaño máximo de las Áreas de Localización.

5 Conclusiones y Trabajo Futuro

Este artículo presenta nuestra adaptación del algoritmo Strength Pareto Evolutionary Algorithm 2 para optimizar el problema de planificación de Áreas de Localización, problema que caracteriza las tareas de uno de los sistemas de gestión más importantes en cualquier red de comunicaciones móviles: el Sistema de Gestión de Localización. En este trabajo, a diferencia de trabajos anteriores, se ha optado por una aproximación multi-objetivo para resolver este problema. El uso de algoritmos de optimización multi-objetivo nos permite evitar los inconvenientes derivados de la combinación lineal, y al mismo tiempo proporcionar al operador de red un conjunto de soluciones que minimizan simultáneamente ambos costes y entre las cuales se podría seleccionar aquella que mejor se adapta al estado real de la red de señalización. Los resultados experimentales muestran que nuestra adaptación del algoritmo SPEA2 obtiene mejores resultados que los algoritmos desarrollados por otros autores en otros trabajos [5-9] a medida que aumenta la complejidad de la instancia, tarea nada trivial pues debemos competir con algoritmos especializados en encontrar una única solución. Como trabajo futuro se propone mejorar los resultados presentados en este artículo, tal vez con nuevos operadores evolutivos que nos permitan explorar las regiones del espacio

objetivo factible con valores extremos de ambas funciones objetivo. Además, sería interesante estudiar otras redes e implementar otros algoritmos de optimización.

Agradecimientos. Este trabajo ha sido parcialmente subvencionado por el Ministerio de Economía y Competitividad y por el Fondo Europeo de Desarrollo Regional (FEDER) bajo el contrato TIN2012-30685 (proyecto BIO). Víctor Berrocal-Plaza disfruta del apoyo de la beca de investigación FPU-AP2010-5841, proporcionada por Ministerio de Educación.

Referencias

1. Agrawal D. P., Zeng Q.A.: Introduction to Wireless and Mobile Systems, 3th ed. Cengage Learning, Stamdford (2010)
2. Kyamakya K., Jobmann K.: Location management in cellular networks: classification of the most important paradigms, realistic simulation framework, and relative performance analysis. IEEE Transactions on Vehicular Technology 54(2), 687–708 (2005)
3. Gondim P. R. L.: Genetic algorithms and the location area partitioning problem in cellular networks. In: Proceedings of IEEE 46th Vehicular Technology Conference, 'Mobile Technology for the Human Race', pp. 1835–1838 (1996)
4. Demestichas P., Georgantas N., Tzifa E., Demesticha V., Striki M., Kilanioti M., Theologou M.: Computationally efficient algorithms for location area planning in future cellular systems, Computer Communications 23 (13), 1263–1280 (2000)
5. Taheri J., Zomaya A. Y.: The use of a Hopfield neural network in solving the mobility management problem. In: Proceedings of The IEEE/ACS International Conference of Pervasive Services, pp. 141–150 (2004)
6. Taheri J., Zomaya A. Y.: A simulated annealing approach for mobile location management. In: Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium, pp. 194–201 (2005)
7. Taheri J., Zomaya A. Y.: A genetic algorithm for finding optimal location area configurations for mobility management. In: Proceedings of The IEEE Conference on Local Computer Networks 30th Anniversary, pp. 568–577 (2005)
8. Taheri J., Zomaya A. Y.: A combined genetic-neural algorithm for mobility management. Journal of Mathematical Modeling and Algorithms 6(3), 481–507 (2007)
9. Almeida-Luz S. M., Vega-Rodríguez M. A., Gómez-Pulido J. A., Sánchez-Pérez J. M.: Differential evolution for solving the mobile location management. Applied Soft Computing 11(1), 410–427 (2011)
10. Zitzler E., Laumanns M., Thiele L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In: Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems, pp. 95–100 (2002)
11. Zitzler E., Thiele L.: An Evolutionary Algorithm for Multiobjective Optimization: The Strength Pareto Approach. Technical report 43, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH), Zürich (1998)
12. Coello C. A., Lamont G. L., Veldhuizen D. A.: Evolutionary Algorithms for Solving Multi-Objective Problems, 2th ed. Springer, Berlin (2007)

Application of Bayesian Neural Networks for Indoor Temperature Time Series Forecasting

Paloma Botella-Rocamora, Pablo Romeu,
Francisco Zamora-Martínez, and Juan Pardo

Embedded Systems and Artificial Intelligence group
Escuela Superior de Enseñanzas Técnicas
Universidad CEU Cardenal Herrera
C/ San Bartolomé 46115 Alfara del Patriarca, Valencia, Spain
{paloma.botella,pablo.romeu,francisco.zamora,juan.pardo}@uch.ceu.es
http://www.uch.ceu.es

Abstract Nowadays, the need to design and implement new systems at home to produce and use energy efficiently are being considered. Recent studies show that a large percentage of the consumed energy at homes belong to heating, ventilation and air conditioning systems, which are highly related to indoor temperature. Therefore, an accurate forecasting of indoor temperature could yield to an energy-efficient control. This paper presents an application of Bayesian Neural Networks in this field, applied to a problem of indoor-temperature time-series forecasting. Results are compared to a standard Artificial Neural Network with one hidden layer showing how the bayesian estimation approach can provide very useful insights from a practical point of view in the field of energy efficiency.

Keywords: Bayesian Neural Networks, time series forecasting, energy efficiency

1 Introduction

Energy is a scarce resource in nature, which has an important cost, which is finite and must be shared. Recent studies in European Union demonstrate that primary energy consumption in buildings represents about 40% of the total energy consumption. Moreover, half of this energy is spent for indoor climate conditioning by HVAC (Heating, Ventilation and Air Conditioning) systems [1,2]. Hence there is a need to design and implement new systems at home, which must be able to use energy efficiently achieving a balance between consumption and streamlined comfort. Indoor temperature is a variable directly related with the notion of comfort but also with energy consumption. Its prediction could be translated into actions, for a near future, in order to optimize the activation/deactivation of HVAC systems. If a home control system is able to acquire this knowledge, it is presumable that it will be able to suggest the inhabitants to perform some actions. This is the aim of the present research, developed under

the framework of the 2013 Solar Decathlon Competition, where the university participated with a solar house. Data used in this paper is real data obtained from the monitoring system of such facility.

Artificial Neural Networks (ANNs) models have enjoyed considerable popularity in recent years. There are several authors who applied them for indoor and outdoor temperature prediction [3,2]. ANNs have been applied to many nonlinear regression problems and have been used extensively for time series prediction [4,5,6,7]. ANNs have demonstrated to be one of the most useful tools for time series forecasting. Different proposals of ANNs are available. However, multilayer perceptron (MLP) with back-propagation learning are the one of the most employed ANNs in time series studies.

Many papers compare ANNs and other standard statistical methods for time series forecasting [8,9,10,11,12]. Most of these papers have found ANNs superior to standard methods such as ARIMA models for various time series problems. A known problem of ANNs models is the lack of tools for analyzing the results (confidence intervals like 10% and 90% quantiles, etc.). However, Bayesian analysis for these models yields posterior predictive distributions for any variables of interest, making the computation of confidence intervals feasible.

In this work, the power of Bayesian treatment of neural networks in this context will be shown. For neural networks the Bayesian approach was introduced in [13,14] and reviewed in [15,16,17]. Some works apply also this kind of neural networks for time series analysis [18,19]. ANNs parameters are treated as random variables whose posterior distribution is obtained from the data. In this approach, predictions are obtained by averaging over all possible models and parameter values according to their posterior distributions.

An ANN with one hidden layer is used to compare between a Bayesian and non-Bayesian estimation of ANN parameters in an indoor temperature time series forecasting task. It is showed how the Bayesian estimation approach can provide very useful tools, from a practical point of view, in the field of energy efficiency.

This paper is organized as follows: section 2 provides a brief description of the database; sections 3 and 4 presents the model and its estimation; finally in the sections 5 and 6 the results and discussion and conclusions are presented.

2 Time series dataset description

The data used in this work are temperature signal values read by a temperature sensor $s_1, s_2, \ldots, s_m$ sampled each minute [7]. These values are preprocessed using a low-pass filter which computes the mean value of the current plus last 14 samples. Hence, $s'_1, s'_2, \ldots, s'_n$ are computed, where

$$s'_i = (s_i + s_{i-1} + s_{i-2} + s_{i-3} + \ldots + s_{i-14})/15$$

In a second step, differences are calculated between each two adjacent elements, to estimate the variation of temperature within 15 minutes. Then $s''_1, s''_2 \ldots s''_{n-1}$ are calculated, being $s''_i = s'_i - s'_{i+1}$.

Data is divided in three partitions: training (2 016 training patterns, 21 days), validation (672 validation patterns, 7 days), and the last one for testing (672 test patterns, 7 days). The validation partition is sequential with the training partition, but the test partition is one week ahead from them [7].

This work presents preliminary results for validation and test partitions. Data Figure 1 shows plots for this two datasets.

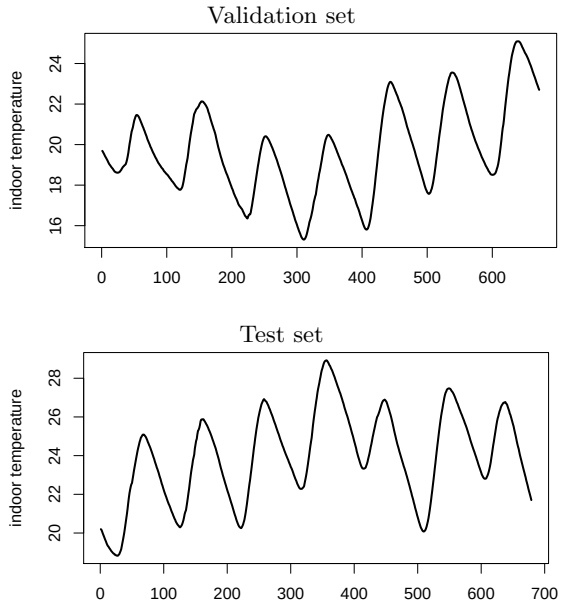


Figure 1. Plots of indoor temperature (°C) for validation and test sets.

3 Model definition

Univariate and detrended time series data $s''_1, s''_2 \dots s''_{n-1}$ (indoor temperature) has been considered. The goal is to forecast, for each next point in time t, the next p values in the series $y_t = (s''_t, ..., s''_{t+p-1})$ based on q the past values $x_t = (s''_{t-1}, s''_{t-2}, ..., s''_{t-q})$. Each element is modelled by a nonlinear autoregression of q past values plus a normally distributed error term:

$$y_{t(i)} = f(x_t) + \epsilon_{t(i)}\,;$$

$$\epsilon_{t(i)} \sim N(0, \sigma^2)\,, \quad i = t, ..., t + p - 1\,;$$

where $y_{t(i)}$ is the i-th element of y_t, so can be written as

$$y_{t(i)}|x_t \sim N(f(x_t), \sigma^2)\,, \quad i = t, ..., t+p-1\,.$$

An ANN model has been proposed (as shown in Figure 2) with q input nodes (q previous values in the serie), one hidden layer with k hidden nodes, p output nodes (p next values in the serie), and activation function ϕ:

$$\hat{y}_{t(i)}|x_t = \beta_{0(i)} + \sum_{j=1}^{k} \beta_{j(i)} \phi(t(x_t)\gamma_j) \quad i = t, ..., t+p-1\,,$$

where $t(x_t)$ denotes the transposed of x_t and where $\beta_{0(i)}$ and $\beta_{j(i)} \in \Re$ and $\gamma_j = \{\gamma_{j(1)}, ..., \gamma_{j(q)}\} \in \Re^q$. Sigmoidal activation function $\phi(h) = \frac{exp(h)}{1+exp(h)}$ is widely used in literature, but hyperbolic tangent function seems more accurate because the ANN input domain and output range are positive/negative sequences of scalar numbers.

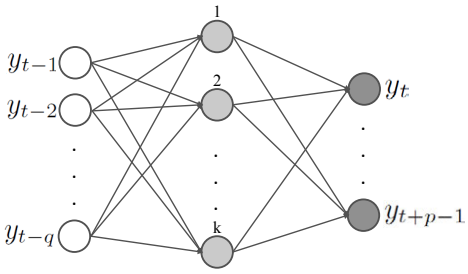


Figure 2. Scheme of the ANN architecture employed.

3.1 Full Bayesian model specification

In order to perform Bayesian inference for the described model, prior distributions for the ANNs parameters β, γ and σ^2 must be defined. In ANN research community is widely accepted that little prior information is needed, and a well selected and trained model could fit any real function [16]. Therefore, relatively diffuse prior distributions are assumed in Bayesian inference. The prior distributions in this works are based on the work of Paige and Butler [20], assuming a conditional normal prior for $\beta_{j(i)}$ and $\gamma_{j(h)}$ given σ^2. A conjugate prior is used for σ^2 which is the Inverse Gamma distribution:

$$\beta_{0(i)}|\sigma^2 \sim N(0, \frac{\sigma^2}{\lambda}), \qquad i = t, ..., t+p-1;$$

$$\beta_{j(i)}|\sigma^2 \sim N(0, \frac{\sigma^2}{\lambda}), \qquad j = 1, ..., k \text{ and } i = t, ..., t+p-1;$$

$$\gamma_j|\sigma^2 \sim N^q(0, \frac{\sigma^2}{\lambda} I_q), \qquad j = 1, ..., k;$$

$$\sigma^2 \sim \Gamma^{-1}(\frac{\nu}{2}, \frac{\varphi}{2});$$

where $\lambda = \frac{10^{-3}}{4}$, N^q denotes the Multivariate Normal Distribution with dimension q, I_q represents the q-dimensional identity matrix, and parameters ν and φ are fixed to $\nu = 3$ and $\varphi = 4\lambda$.

Aiming to avoid the identifiability problem of parameters (β, γ), as is suggested in [21,20], the constraints $\gamma_{11} > \gamma_{21} > ... > \gamma_{k1}$ have been considered.

4 Model estimation

4.1 Bayesian ANNs estimation

Given the prior structure, a closed form for the posterior parameter distribution is not available. In these situations it is common to apply Markov Chain Monte Carlo (MCMC) algorithms to generate an approximate sample from the posterior distribution (see [22] for a deeper review). For different ANN models different MCMC algorithms have been proposed for sampling (see [23] for a good review). In this case, however, due to practical nature of this work, we choose to use the Bayesian software WinBUGS [24], which implements a Gibbs sampling inference algorithm. Other works have applied this tool for bayesian analysis of ANN models (as for example [25]). This tool performs an automatic sampling inference procedure which is based on hierarchical structure of the model, the training data, and a set of initial random values for the MCMC sampler.

To estimate the model in WinBUGS, three chains of a MCMC sampler were computed for each parameter of interest (weights and bias). A simulation of 30 000 iterations was done, discarding the first 20 000 in each sequence.

The posterior distribution samples for the model parameters were used to estimate the predictive distribution for the new inputs. For each iteration r, the Bayesian inference algorithm computes a set of ANN parameters $(\beta, \gamma)^r$, and provides one value $(\hat{y}_{t(i)})^r$ from the posterior distribution for $\hat{y}_{t(i)}$ (for $i = t, ..., t+p-1$).

So, the posterior mean for $\hat{y}^{t(i)}$ can be calculated as:

$$\overline{\hat{y}_{t(i)}} = \frac{1}{M} \sum_{r=1}^{M} (\hat{y}_{t(i)})^r,$$

where M is the number of samples of the posterior distribution. But, in addition to this estimate, the posterior sample values provide an information source on the uncertainty of the prediction.

4.2 Standard ANNs training

Standard ANNs were trained using error back-propagation algorithm [16] implemented at `April-ANN` toolkit [26] in order to validate Bayesian estimation approach. For comparison purposes, the topology of these ANNs is the same as the ones used in Bayesian estimation. Learning rate, momentum and weight decay hyper-parameters were estimated following a random search optimization algorithm [27]. The activation function of hidden layers is the hyperbolic tangent instead of translated sigmoidal. The loss function used to compute error gradients is the Mean Square Error function.

5 Results

Different error measures are available at literature for time series forecasting tasks. In this work, due to indoor temperature characteristics, the Mean Absolute Error (MAE) provides more interpretable information, and is computed following next equation:

$$MAE = \frac{1}{p}\sum_{i=1}^{p} |\hat{y}_i - y_i| \,,$$

where $\hat{y}_i$ is the predicted i-th value and y_i its ground truth. Similar results in both types of model estimation (standard and bayesian approach) are found. Table 1 shows MAE measures obtained with each model for validation and test sets. The input window was set to 7 values in all the experiments. Models varying hidden layer size and future window horizon were compared ($k = 2, 4, 6, 8$ and $p = 1, 4, 12$).

Table 1. MAE results obtained by models with different combinations of hidden layer size and future window horizon estimated following Bayesian ANNs (BNNs) and Standard ANNs (SNNs) approaches.

Model $q.k.p$	BNNs MAE (Validation)	SNNs MAE (Validation)	BNNs MAE (Test)	SNNs MAE (Test)
7.2.1	**0.0172**	0.0177	**0.0178**	0.0184
7.2.4	0.0483	**0.0477**	**0.0474**	0.0488
7.2.12	0.1941	**0.1719**	0.1868	**0.1677**
7.4.1	**0.0170**	0.0178	**0.0177**	0.0183
7.4.4	**0.0477**	0.0483	**0.0461**	0.0496
7.4.12	0.1877	**0.1752**	0.1804	**0.1666**
7.6.1	**0.0170**	0.0178	**0.0178**	0.0186
7.6.4	**0.0472**	0.0486	**0.0452**	0.0507
7.6.12	0.1875	**0.1685**	0.1773	**0.1613**
7.8.1	**0.0171**	0.0179	**0.0177**	0.0186
7.8.4	**0.0476**	0.0483	**0.0456**	0.0499
7.8.12	0.1878	**0.1837**	**0.1809**	0.1896

It should be noted that in those models with larger future window horizons ($p = 12$) standard approach has worked better than bayesian approach, probably due to an insufficient number of simulations in the MCMC algorithm. In any case, the results from both approaches provide equivalent results with similar errors and non relevant differences for our aims. A deeper study of this concern has been done computing the individual MAE of each future prediction for the model 7.6.12 estimated following the two proposed approaches (see Figure 3). It is observed a clear trend to worst prediction performance of Bayesian approach while the future horizon increases (it is clear from sixth future horizon), totally correlated with the increment of task difficulty. This result maybe proof the lack of simulation iterations hypothesis, needed to improve performance. Computational cost of $p = 12$ models training needs to be reduced in order to increase the number of iterations.

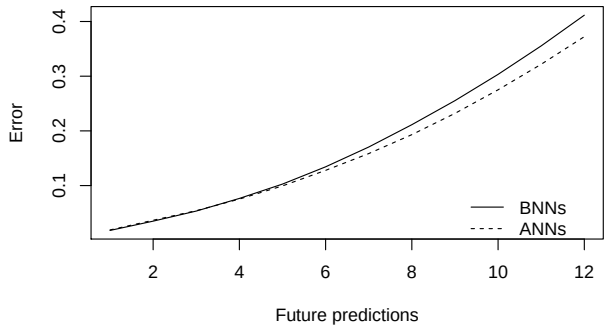


Figure 3. MAE measure for each future predicted horizon.

However, the Bayesian approach provides more useful results from practical standpoint, a probability distribution on the value of each predicted point instead of a unique forecasted value. Figure 4 shows the posterior distribution at time $t = 460$ sampled from model 7.6.4. From this posterior distribution, Bayesian credible intervals are easily calculated.

The main contribution of the Bayesian approach is the ability to answer questions like: *What is the probability that the next hour the temperature exceeds 24 degrees?*. These questions may help decisions in the field of energy efficiency (for instance, on the operation of HVAC systems). Figure 5 shows the answer of this question for every time step. Highlighted values are probabilities greater than 0.80.

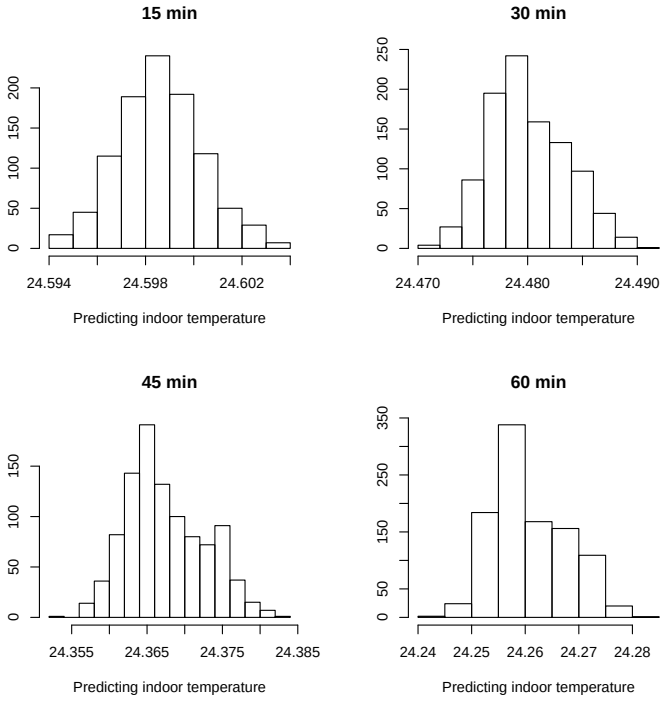


Figure 4. Indoor temperature: posterior distribution for the prediction in $t = 460$ for the four next values of the model 7.6.4.

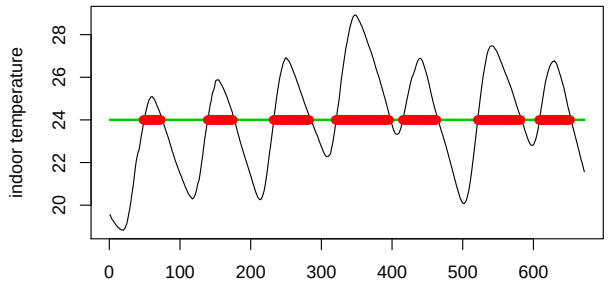


Figure 5. Indoor temperature: probability to exceed 24 degrees in the next hour. Highlighted values have probability greater than 0.80.

6 Discussion and conclusions

The present study has applied Bayesian approach compared to backpropagation approach for ANN estimation in a real indoor temperature forecasting task. Both approaches achieve similar results. However, a deeper analysis about number and size of hidden layers, input window size, forecast output size, ... needs to be done in order to improve the performance of both systems. Besides, a larger study examining the behavior of each algorithm in more complex tasks is necessary (for instance, larger input windows). However, this paper shows how Bayesian approach provides a framework which is particularly useful in the context of decision algorithms based on time series forecasting for energy efficiency environments. Many issues remain unexplored for future work: more complex tasks where estimation of backpropagation ANNs is difficult; the exploration of other prior distributions for model parameters; the consideration of hidden layer size k as a parameter in the Bayesian model; and comparison with traditional statistical models and other learning algorithms. Bayesian inference algorithms, as Reversible Jump MCMC (not implemented in WinBUGS tool) will be needed in order to achieve some of this future goals.

References

1. Álvarez, J., Redondo, J., Camponogara, E., Normey-Rico, J., Berenguel, M., Ortigosa, P.: Optimizing building comfort temperature regulation via model predictive control. Energy and Buildings (0) (2012)
2. Ferreira, P., Ruano, A., Silva, S., Conceição, E.: Neural networks based predictive control for thermal comfort and energy savings in public buildings. Energy and Buildings **55**(0) (2012) 238–251 Cool Roofs, Cool Pavements, Cool Cities, and Cool World.
3. G.Mustafaraj, G.Lowry, J.Chen: Prediction of room temperature and relative humidity by autoregressive linear and nonlinear neural network models for an open office. Energy and Buildings **43**(6) (2011) 1452–1460
4. Zhang, G., Patuwo, B.E., Hu, M.Y.: Forecasting with artificial neural networks: The state of the art. International Journal of Forecasting **14**(1) (1998) 35–62
5. Ruano, A., Crispim, E., Conceição, E., Lúcio, M.: Prediction of building's temperature using neural networks models. Energy and Buildings **38**(6) (2006) 682–694
6. Ben Taieb, S., Bontempi, G., Atiya, A., Sorjamaa, A.: A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. Expert Systems with Applications (2012) Preprint.
7. Zamora-Martínez, F., Romeu, P., Pardo, J., Tormo, D.: Some empirical evaluations of a temperature forecasting module based on Artificial Neural Networks for a domotic home environment. In: IC3K – KDIR. (2012)
8. Z.Tang, C.Almeida, P.A.Fishwick: Time series forecasting using neural networks vs. box- jenkins methodology. Simulation **57**(5) (1991) 303–310
9. W.R.Foster, F.Collopy, L.H.Ungar: Neural network forecasting of short, noisy time series. Computers & Chemical Engineering **16**(4) (1991) 293–297
10. H.Stern: Neural networks in applied statistics (with discussion). Technometrics **38** (1996) 293–297

11. T.Hill, M.OĆonnor, W.Remus: Neural network models for time serie forecasts. Mangmnt Sci. **42** (1996) 1082–1092
12. J.Faraway, C.Chatfield: Time series forecasting with neural networks: a comparative study using the airline data. Journal of the Royal Statistical Society **47** (1998) 231–250
13. W.L.Buntine, A.S.Weigend: Bayesian back-propagation. Complex systems **5**(6) (1995) 603–643
14. D.J.C.Mackay: A practical bayesian framework for backpropagation networks. Neural Computation **4** (1998) 448–472
15. D.J.C.Mackay: Probable networks and plausible predictions - a review of practical bayesian methods for supervised neural networks. Network: Computation in Neural Systems **6**(3) (1995) 469–505
16. Bishop, C.M.: Neural networks for pattern recognition. Oxford University Press (1995)
17. R.M.Neal: Bayesian learning for neural networks. Springer-Verlag, New York (1996)
18. Zhang, T., Fukushige, A.: Forecasting time series by bayesian neural networks. In: Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on. Volume 1. (2002) 382–387
19. Liang, F.: Bayesian neural networks for nonlinear time series forecasting. Statistics and Computing **15**(1) (January 2005) 13–29
20. Paige, R., Butler, R.: Bayesian inference in neural netwoks. Biometrika **88**(3) (2001) 623–641
21. A.Menchero, R.Montes-Diez, Insua, D., P.Muller: Bayesian analysis of non-linear autoregression models based on neural networks. Neural Computation **17**(17) (2001) 453–485
22. W.R.Gilks, S.Richardson, D.J.Spiegelhalter: Bayesian learning for neural networks. Chapman and Hall, London (1996)
23. H.Lee: Bayesian Nonparametrics via Neural Networks. SIAM (2004)
24. D.J.Lunn, A.Thomas, N.Best, D.Spiegelhalter: Winbugs-a bayesian modelling framework: Concepts, structure, and extensibility. Statistics and Computing **10** (2000) 325–337
25. Wiper, M., Palacios, A., Marın, J.: Bayesian software reliability prediction using software metrics information. Quality Technology & Quantitative Management **9** (2012) 35–44
26. Zamora-Martínez, F., et al.: April-ANN toolkit, A Pattern Recognizer In Lua, Artificial Neural Networks module (2013) https://github.com/pakozm/april-ann.
27. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. J. Mach. Learn. Res. **13** (March 2012) 281–305

Content-based Recommendation: Experimentation and Evaluation in a Case Study

A. Castellanos, J. Cigarrán, A. García-Serrano

Universidad Nacional de Educación a Distancia, UNED

{acastellanos,juanci,agarcia}@lsi.uned.es

Abstract. Recommendation focuses on discovering new relevant contents to users. The main approaches for that are: Collaborative Filtering, based on finding similarities between users, and Content-Based Recommendation, based on the recommendation of contents similar to those already seen by the user; that is, to find similarities between contents. Given that, the content-based recommendation task can be seen as an Information Retrieval task. In this paper it is presented a content and user modelling approach using a terminology extraction technique based on language divergences (KLD); instead of TF-IDF, almost the standard in recommendation. This intends to improve the recommendation step through more representative models. To experiment with this proposal, an experimentation prototype and evaluation framework has been developed. Details about the developed prototype and its evaluation using an user-centred approach are included. Results confirm that the proposed recommendations improve the user satisfaction, in terms of performance and utility.

Keywords: Recommender Systems, Content-based Recommendation, Language Divergences, User-centred evaluation, IR-Based Recommendation.

1 Motivation

Recommender systems (RS) seeks to identify interesting items to users based on: (a) items already seen in the past or, (b) items already seen by other users with similar interests. These two types have been traditionally used to differentiate the recommenders in two classes [1]: Content-Based and Collaborative Filtering. Our proposal is focused on content-based recommendation, trying to avoid some of the drawbacks of the collaborative filtering approach like the so-called "cold start" problem (what to recommend when a new user arrives). Content-based systems can be viewed as Information Retrieval (IR) systems: systems for finding relevant contents (recommendations) given an user query (model of the user). Other works have also related both fields; for example: the work in [2] tries to adapt the recommendation task for applying some IR algorithms; in [6] it is proposed the utilization of folksonomies for the IR process, applied to a RS; while in [11] both fields are applied together by using techniques coming from IR field to diversify the recommendations.

Concerning to the modelling step, our proposal is based on the application of a language divergence-based technique, the Kullback Leibler Divergence (KLD), for modelling the contents and user profiles according to their divergences, instead of their similarities. This modelling pretends to improve the performance of the recommendation step in relation to problems such as the overspecialization problem [12]. This approach has been already evaluated, improving the performance with respect to other common modelling techniques in terms of representativeness [2].

In the state of the art of RS evaluation two methods have been commonly used: laboratory-based, based on a comparison between the generated recommendations and a gold standard [4]; and a user-centred, based on the opinions of a set of users [7], [9]. This latter is particularly useful because the performance of RS is specially related with the improvement of the user experience. Given that, for the evaluation of the recommendation step, an user-centred evaluation has been conducted in this work.

After a short description of the developed recommendation prototype, currently running at http://albali.lsi.uned.es/Recomendador, some experiments performed with the prototype and the results of the evaluation are described.

2 Work Proposal

The collection used in this work has been developed by the authors inside the Buscamedia Project (partners Tecnalia, ISID, Daedalus[1], and UNED) [5]. It contains 6000 news from 14 on-line newspapers, classified on categories and sub-categories with the Daedalus tool: Stilus Class[2].

To model the news reports, categories and subcategories, (contents of the set of news belonging to each one) as well as the user profiles, it has been followed a model language technique based on language divergences: KLD. The core idea is to identify those terms that better shows the divergence between two bags of terms: the longer a term appears in some document and less in other documents, the more relevant is. The terms are gathered and stored by using the Vector Space Model [8] weighted with the KLD. The KLD-based weight is calculated as follows:

$$KLD_{pD,pC} = pD(t) \cdot \ln\left(\frac{pD(t)}{pC(t)}\right) \quad (1)$$

where $pD(t)$ is the probability of each term within a document D (frequency of t divided by the whole of terms in the document D) and $pC(t)$is the probability of the same term t within the collection C (frequency of t divided by the number of terms in the collection C).

For news modelling, it has been compared every individual news with the rest of the news on the collection (the same for the categories and subcategories). For user profiles modelling, the content of the news read by the user has been gathered and compared with the content of the rest of the news (those that has not been read by the user). As the KLD is based in the "differences" between contents, it facilitates to ob-

[1] http://www.tecnalia.com, http://www.isid.es/, http://www.daedalus.es/

[2] http://api.daedalus.es/stilusclass-info

tain a more diverse recommendation: the recommended items are different between themselves. The user modelling presented in [10] is similar to our proposal: they also use a divergence-based model. Main difference is the term weighting approach: they compare each user model with the rest, and in this way any user model can be modified if someone else, even without any relation to the user, sees a new content. By contrast we compare user profiles with the unread documents in the collection.

The recommendation step calculates the similarities between user models and contents: If some content is similar (according to some criteria) it will be recommended. In our work, we decide to apply an IR-based approach to find similarities between the textual contents (where the user models are used as query to the index with the documents). Since in IR field this similarity issue has been broadly studied in deep, we expect to improve the recommendation step by applying this know-how, derived from IR.

We decided to use SOLR as the search engine[3]. SOLR bases their operation on the Apache Lucene implementation of Cosine function as similarity measure. Cosine function normalizes the dot product of the vectors *(V(query)•V(document))* by dividing it by the Euclidean norm of these vectors. But Lucene uses different document length normalization for vector V (document). Instead of calculating the Euclidean norm for each document, with a high computational cost, Lucene uses the document length. More details are included in [2]

3 Experimentation

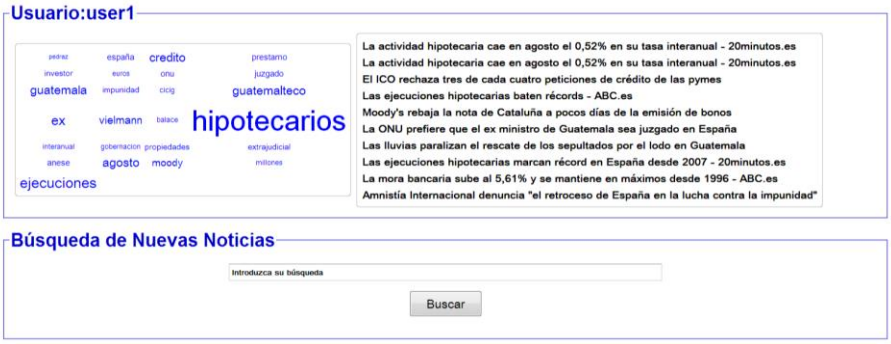
In the following it is presented the recommendation cases addressed in the experimentation, trying to cover some user information needs.

3.1 Recommendation based on the user previous interactions

This recommendation is useful when an user accesses to the system. Due to that there isn't any information related with the current user information need; previous information about the user is used to offer new recommendations. An example can be seen in the Fig. 1: the system offers the recommendations in two formats: a tag cloud (on the left) and a list of related documents or news (on the right). The tag cloud contains the terms in the user model, sized according KLD weight. These terms are used to obtain the list of related documents, by querying with the terms pondered according to their relevance.

[3] http://lucene.apache.org/solr/

Fig. 1. User Model based Recommendation



3.2 Recommendation based on new user activity

In this case the target user has already interacted with the system in the ongoing session. These interactions extend the user model with new data, related with the current user information need. There are two cases where this kind of data could be useful to extend user profile:

The user is searching new contents. In this case, the relevance of the results list depends on the specific user preferences, contained in the user model; for example:

If some user who is interested in sports news, and more specifically in football news, carries out the query: MADRID, then he/she is probably interested in football news about the teams of Madrid (Real Madrid, Atlético de Madrid…) and not in news only about the city of Madrid.

Given an user query, our prototype uses the information in the results list, obtained from the query, and the information in the user model to recommend new contents. The system offers two types of recommendations: a set of terms (shown in the tag cloud in the Fig. 2), obtained by modelling the terms contained in the result list and the terms in the user profile. Then, a set of documents (news) related to these terms (shown below of the tag cloud in the Fig. 2) is obtained by querying with the terms to the SOLR index that contains the news, in the same way that in the previous approach.

Fig. 2. Search-based Recommendation

The user accesses to a specific content. In this case the user is viewing a concrete document; so the system recommends other documents related with this one. An example of this case:

> *The user is consulting news about a Real Madrid match; the news-based recommendation offer terms directly related like: Milan, the rival in the match; Cristiano and Ronaldo, a Real Madrid player who scores two goals in the match; or Seedorf, a Milan player. On the other hand, in the subcategory-based recommendation, the system offers more general terms like Atlético, Barcelona, Bayern or Arsenal, rivals in other matches of the Real Madrid; Champions, competition played by Real Madrid. In the category-based recommendation the system offers terms related with the news even more generals, like: balón (ball), árbitro (referee), or anoto (scored).*

The system uses the information of the visited document at three levels: the content of the document, the sub-category of the document and the category of the document. In the Fig. 3 the three types of information (in the three tag clouds) are shown; the recommendations will be at these three levels. While the recommendation based on the news model tries to offer other news related to it, the recommendation based on categories and subcategories pretends to offer news in the context of the displayed document.

Fig. 3. Specific-content recommendation

From the terms of the news-based recommendation and the terms in the user model, the system also offers a new list of documents (see Fig. 4). This recommendation tries to offer other related news, taking into account the previous information about the user.

Fig. 4. Specific-content recommendation (II)

4 Evaluation

The presented approach tries to solve some of the problems in the recommendation task through the improvement of the modeling step. The enhancement of the recom-

mendation step is going to influence the user experience. This evaluation tries to measure how the recommendation proposal can help during the searching process. For that, a recommendation prototype was developed that allows searching, consulting and rating news. For the evaluation, some real users (non-expert) have been selected and they were required to perform ten searching tasks. In order to have a baseline for comparison, the users also have the option to complete the tasks with a classical IR system (from here denoted as Searcher), besides of the recommender system (see Fig. 5).

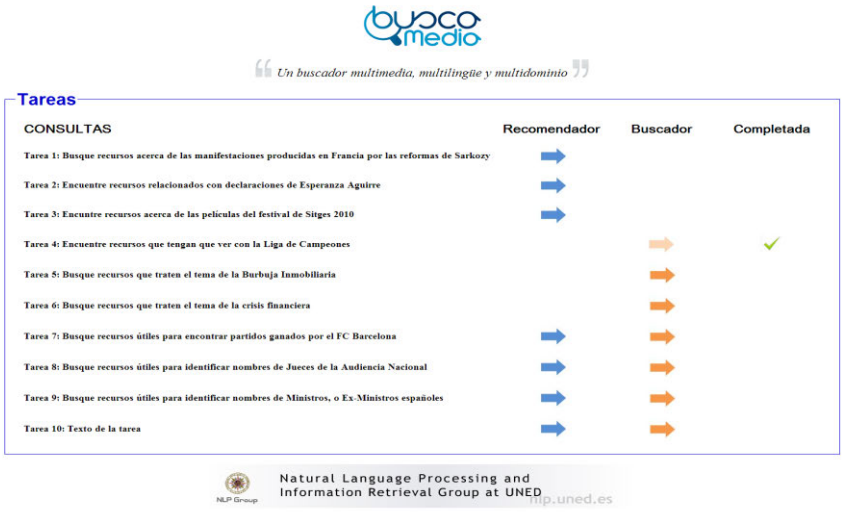
4.1 Evaluation Setup

The tasks to be completed by the users were divided in three types, trying to cover different situations of information searching in the context of news report:

- General information: *Find information related with the FIFA World 2010.*
- Related events: *Find information about the matches of Spain in the World Cup.*
- Imprecise information: *Important goals of Iniesta in his professional career.*

The users had to use the prototype (Recommender System or Searcher) to find information related with each of the queries. They had 5 minutes to complete each task and they could consider the task as completed when they considered that they have enough information (as if they were looking for information about a topic by reading a set of news). No further test was presented to the user because the concept of "enough information" is very different depending on the user: some users briefly read the news while others perform deeper reading of the news. Once the users finished all the tasks, a questionnaire was offered to them to know their opinion about the system and recommendation approaches.

Fig. 5. Evaluation tasks



4.2 Results

After the evaluation period, there were a total of 23 users that had completed all the tasks. The analysis of the results is focused on checking the user interaction with the system. We expect to find behavioural patterns that reveal how the recommendations affect to the process of information searching and if the tasks had been easier to complete with the recommendations. For that, we analyse the interactions with the recommender system compared with the baseline system. In the Table 1, it is shown some of the data obtained from the user logs. Note that the number of interactions with each system depends on the number of task completed with it. Thus, for allowing the comparison between systems, the data are normalized according to the number of task completed with them.

The Searcher has been used 90 times; while the Recommender has been used 140 times. These results show the preference of the users for the Recommender. Attending to the normalized data, a general comment is that users had to do fewer interactions with the Recommender than with the Searcher (10.34 vs. 12.02 interactions/task). This can be due to that the users can complete the tasks using the recommendations easier than with the Searcher.

Focusing in interactions, the differences are even bigger: the number of searches (1.42 searches/task vs. 1.89 searches/task) or the number of navigations (0.31 navs/task vs. 1.4). Once again, these data can be attributed to the utility of the recommendations. Attending to the ratings, with the Searcher the users rated more documents (4.65 vs. 2.7 ratings/task), because they had to carry more interactions with the Searcher. But looking at detail, with the Recommender the 53% of ratings are 5 and the 83% of ratings are at least 4, while with the Searcher the percentage is reduced to 37% of 5s and 63% of 4s or 5s. These results suggest that the documents offered by the Recommender are considered as more relevant by users. To sum up, as the recommendations helps users to complete the task and the ratings are higher with the recommender, that suggest that recommendations offer a valuable help.

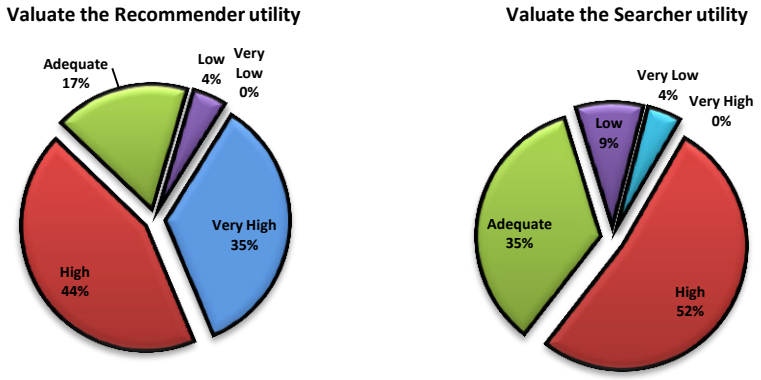
Table 1 –User logs data

	Recommender	**Searcher**
# of tasks completed with the system	**140**	**90**
# of interactions	**10.34**	**12.02**
# of search	1.42	1.89
# of accesses to a document	1.62	2.07
# of accesses to documents after recommended	0.76	**X**
# of navigations performed	0.31	1.4
# of access through the tags	1.14	**X**
# of ratings	**2.7**	**4.65**
# of rates equal to 1	0.27	0.51
# of rates equal to 2	0.11	0.41
# of rates equal to 3	0.31	0.78
# of rates equal to 4	0.56	1.22
# of rates equal to 5	1.44	1.72

Fig 6 shows some of the results obtained from the questionnaires about the Searcher (right) and the Recommender (left). For the Searcher, 0% of the users have set the utility of the system as Very High versus the 35% for the Recommender; For

the Searcher, 13% of the users have set the utility as Low or Very low, and only 4 % for the Recommender. As with the implicit information, the information explicitly collected shows that the satisfaction of the users is higher with the Recommender.

Fig. 6. Questionnaire Results



5 Remarks and Future Work

This work aimed to take advantage of the work done in the IR field and apply them in the field of recommendation. The recommendation has been proposed as a Searching process: An automatic query is constructed taking the background information about users and the information about their current information needs.

The content modelling (news/categories/subcategories and user profiles) has been carried out by applying a divergence-based technique: the Kullback Leibler Divergence; because a divergence-based modelling will allow the recommendation of contents with higher diversity, in the later recommendation step.

For the experimentation, a recommender prototype has been developed (see Fig. 5), which offers a platform to search, consult and rating the news. While the user interacts with the system, it provides recommendations according different approaches. The evaluation of the proposal has been conducted by applying a user-centred method, in which a set of users was required to complete ten tasks. To have a baseline the users had the possibility of complete the task using an IR System.

The results of the user-based evaluation prove that the combination of KLD modeling and the IR-based approach are suitable to offer unexpected but relevant recommendations to the user, in terms of user satisfaction (the users prefer the results offered by the recommender and they can easily complete the task with the recommender). Finally, the results of the questionnaire show that the users are highly satisfied by the operation of the recommender, in terms of utility.

Once that the proposed approach has proven to be promised, some more work has to be done in order to include temporal aspects (long-term versus short-term) or contextual information. Also the enrichment of the user models with social information is a promising line to work, in order to have more information about the user prefer-

ences. Focusing on the evaluation the comparison between our proposal and other recommendation proposals, based on common modelling techniques such as TF-IDF could also be done.

Acknowledgments. This work has been partially supported by the Regional Government of Madrid under Research Network MA2VIRMR (S2009/TIC-1542), and by Spanish Government by the projects BUSCAMEDIA (CEN-20091026) and HOLOPEDIA (TIN 2010-21128-C02).

6 References

1. Candillier, L., Jack, K., Fessant, F., Meyer, F.: State-of-the-art recommender systems. In: Chevalier, M., Julien, C., Soule-Dupuy, C. (eds.) Collaborative and Social Information Retrieval and Access: Techniques for Improved User Modeling, pp. 1-22. USA (2009).
2. Castellanos A: Recomendación de contenidos digitales basada en modelos del lenguajes: Diseño, experimentación y evaluación. Master Thesis, UNED (2013).
3. Costa, A., Roda, F.: Recommender systems by means of information retrieval. In: Proceedings of the International Conference on Web Intelligence, Mining and Semantics (WIMS '11), Article 57, 5 pages. ACM, New York, NY, USA (2010).
4. Gunawardana, J., Shani, G.: A Survey of Accuracy Evaluation Metrics of Recommendation Tasks. In: Journal of Machine Learning Research, 10, 2935-2962 (2009).
5. Hernández Aranda, D., Fresno Fernández, V., Granados Muñoz, R., García Serrano, A.: Evaluando Modelos de Indexación Multilingüe en Recuperación de Imágenes Basada en Texto. In: Proc. of the Conf. of the Spanish Association for Artificial Intelligence. (2011).
6. Huemer, C., Setzer, T.: Information Retrieval and Folksonomies together for Recommender Systems. In: E-Commerce and Web Technologies, 85, 172-183 (2011).
7. Knijnenburg, B.P. Willemsen, M.C. Kobsa, A.: A Pragmatic Procedure to Support the User-Centric Evaluation of Recommender System. In Proceedings of the 5th ACM Conference on Recommender Systems (RecSys '11), pp. 321-324. ACM NY, USA (2011).
8. Salton, G., Wong, A., Yang, C.S.: A Vector Space Model for Automatic Indexing. In: Communications of the ACM 18, (11), 613-620 (1975).
9. Schröder, G., Thiele, M., Lehner, W.: Setting Goals and Choosing Metrics for Recommender System Evaluations. In: Proceedings of the Second Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI 2) (2011).
10. Shmueli-Scheuer, M., Roitman, H., Carmel, D., Mass, Y., and Konopnicki, D. 2010. Extracting User Profiles form Large Scale Data. In Proc. of the 2010 Workshop on Massive Data Analytics on the Cloud (MDAC 2010), USA, 2010, ACM NY, USA, 1-6.
11. Vargas, S., Castells, P., Vallet, D.: Inter-Oriented Diversity in Recommender Systems. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '11), pp. 1211-1212. ACM NY, USA (2011).
12. Zhang, Y. Callan, J. Minka, T.: Novelty and redundancy detection in adaptive filtering. In: Proceedings of the 25th annual international ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR '02), pp. 81-88. ACM NY, USA (2002).

Estudio de la Carga Cognitiva en Interfaces Web de Sistemas de Búsqueda Colaborativa

Sergio Cleger[1], Juan M. Fernández-Luna[2],
Humberto Rodríguez-Ávila[1], and Julio C. Rodríguez-Cano[1]

[1] Departamento de Ingeniería Informática. GISIDI–UHO.
Universidad de Holguín. Holguín, Cuba 80100
{sergio,hrodriguez,jcrcano}@facinf.uho.edu.cu
[2] Departamento de Ciencias de la Computación e Inteligencia Artificial.
ETSI Informática y de Telecomunicación. CITIC–UGR
Universidad de Granada. Granada, España 18071.
jmfluna@decsai.ugr.es

Abstract. En las redes de ordenadores actuales es frecuente que grupos de personas con intereses comunes, trabajen y realicen búsquedas en grupos para satisfacer necesidades compartidas de información. Este fenómeno se enmarca en lo que algunos investigadores denominan Búsqueda Colaborativa de Información. Sin embargo, los motores de búsquedas y navegadores Web tradicionales consideran la búsqueda como una acción individual. Así, resulta importante estudiar el diseño de nuevas interfaces multi-usuario que minimicen la carga cognitiva a la que se someten los usuarios de este nuevo tipo de sistemas, considerando que éste y muchos otros comportamientos sociales en la red, han propiciado una evolución hacia la Web 2.0. En este trabajo se presenta un conjunto de tácticas de usabilidad que contribuyen a disminuir la carga cognitiva durante sesiones de búsqueda colaborativa, las cuales son ejemplificadas a través de un caso de estudio.

Keywords: Búsqueda Colaborativa de Información, Carga Cognitiva, Dimensiones Cognitivas, Usabilidad

1 Introducción

La búsqueda en la Web es una de las actividades más comunes para satisfacer las necesidades profesionales y personales de los usuarios de Internet. Es frecuente que grupos de personas con intereses comunes, trabajen y realicen búsquedas en grupos para satisfacer necesidades compartidas de información. Este fenómeno se enmarca en lo que algunos investigadores denominan Búsqueda Colaborativa de Información (CIS, por sus siglas en inglés). Sin embargo, los motores de búsqueda y navegadores Web tradicionales consideran la búsqueda como una acción individual y proveen un limitado soporte para la colaboración. El surgimiento de la Web social ha facilitado el desarrollo de sistemas de búsqueda que facilitan una colaboración explícita entre grupos de usuarios con necesidades compartidas de información.

CIS se ha convertido en un nuevo y emergente campo dentro de la recuperación de información, por su aplicabilidad en diferentes contextos o dominios (planeación de viajes y eventos sociales, bibliotecas digitales, etc.). Sin embargo, una definición formal para este campo en ocasiones resulta difícil por la diversidad de definiciones encontradas en la literatura actual. Resulta común hallar trabajos bajo el nombre de búsqueda colaborativa, recuparación colaborativa de información, entre otros. En este trabajo se ha optado por la definición de Sha [9], quien se ha referido a CIS como un proceso de búsqueda de información *"que es definido explícitamente entre participantes, interactivo y de beneficio común"*.

Los sistemas CIS permiten una mayor interacción entre los usuarios y el sistema tanto de forma síncrona como asíncrona. Además, representan un gran desafío al tratar de extender los sistemas tradicionales de búsqueda de información (ej. Google, Yahoo). A pesar de los grandes beneficios que intentan proveer este tipo de sistemas, la adopción de estos a menudo es entorpecida por la carga cognitiva adicional que requieren los usuarios para adaptarse a sus nuevas interfaces. Estos sistemas necesitan ser diseñados de una manera consistente respecto a las habilidades cognitivas de sus usuarios, con el objetivo de reducir complejidad cognitiva de sus actividades. En este sentido, durante las últimas dos décadas se han propuestos varias técnicas para el análisis de artefactos de información complejos, entre las cuales se destaca las Dimensiones Cognitivas. Esta técnica ha sido diseñada para realizar un meta-análisis, que considera aspectos estructurales del sistema e identifica características que pueden impedir o permitir ciertas clases de interacciones con el sistema. Puede ser usada de manera formativa para evaluar sistemas existentes y conducir al diseño de nuevos sistemas [5]. Tomando como base el análisis cognitivo de interfaces de búsqueda propuesto por Golovchinsky [4], en este trabajo se aplica esta técnica para analizar interfaces Web de búsqueda colaborativa en el contexto de CIS.

En la presente investigación, se realizará un estudio de las infraestructuras que han dado soporte al desarrollo de los sistemas CIS hasta la actualidad. Se analizarán las principales ventajas y desventajas de estas infraestructuras, con el objetivo de determinar cuál de estas puede contribuir en mayor medida al desarrollo de sistemas CIS más usables. Se presentarán un conjunto de tácticas de usabilidad basadas en dimensiones cognitivas, para disminuir el esfuerzo de adaptación requerido por parte de los usuarios de este tipo de sistemas. Finalmente, se presentará el prototipo *Drakkar* que ejemplificará las tácticas de usabilidad propuestas.

La estructura organizativa del artículo será la siguiente: Sección 2, se abordan las principales infraestructuras que han dado soporte al desarrollo de las interfaces de los sistemas CIS. En la Sección 3, se realiza una introducción al conjunto de dimensiones cognitivas seleccionadas para el estudio a desarrollar. Luego se introducen las tácticas de usabilidad propuestas en la Sección 4, la evaluación y un caso de estudio para ejemplificar el uso las mismas son presentadas en las Secciones 5 y 6 respectivamente. Finalmente, se presentan las conclusiones en la Sección 7.

2 Trabajos Relacionados

En esta sección se realiza un breve resumen de las principales infraestructuras que han dado soporte a las interfaces de los sistemas CIS desarrollados hasta la actualidad.

Navegadores Web Embebidos. La mayoría de los sistemas desarrollados en el contexto CIS, han sido aplicaciones de escritorio clientes (SearchTogether [8], POSSESRC [3]) con un navegador Web embebido. Este tipo de sistema permite el desarrollo de interfaces de usuario enriquecidas para facilitar el desempeño de las tareas colaborativas de sus usuarios.

Las aplicaciones clientes de escritorio por naturaleza facilitan proteger al usuario de condiciones y situaciones de error no deseables de una forma familiar. Evitan al usuario cometer errores graves al reducir el riesgo de que estos presionen teclas/botones de forma errónea. En su gran mayoría proveen funciones de deshacer acciones y ventanas de confirmación que les permiten a los usuarios otra oportunidad para considerar sus acciones. Además, ofrecen la posibilidad de agregar atajos/aceleradores de acciones mediante combinaciones de teclas; estos aceleradores permiten a usuarios expertos, familiarizados con el sistema navegar por el mismo de una forma más rápida. Sin embargo, estos sistemas en su gran mayoría requieren un conjunto de acciones de configuración, las cuales pueden representar cierto grado de dificultad para algunos usuarios. Además, requieren ser descargados e instalados por el usuario y en algunos casos requieren software de terceras fuentes incrementando el grado de complejidad requirido para su uso.

Extensiones a Navegadores Web. La interfaz cliente de este tipo de sistemas CIS es implementada mediante el uso de tecnologías Web (ej. AJAX, JFS y PHP) y tecnologías de desarrollo de complementos/extensiones/plug-ins para navegadores. El progresivo avance en las tecnologías Web y la creciente popularidad de la Web social han permitido extender el desarrollo de sistemas con interfaces de usuarios enriquecidas en la Web (RIAs - Rich Internet Applications). Los sistemas CIS se han visto beneficiados de estos progresos implementando características soportadas anteriormente solo por las aplicaciones de escritorio. Algunos ejemplos de este tipo de aplicaciones son Coagmento [9] cuyo cliente fue desarrollado como un complemento para el navegador Mozilla Firefox y SearchTogether (versión Plug-in) en este caso desarrollado como un plug-in para el navegador Internet Explorer; basado en el diseño original de la aplicación de escritorio referenciada anteriormente. Estos sistemas tienen como limitantes el uso restringido a un navegador Web en particular. Además, necesitan ser instalados y configurados en el navegador Web por parte del usuario, quien en muchas ocasiones no dispone del conocimiento mínimo requerido para ejecutar dichas operaciones, lo cual influye negativamente en la adopción del uso de este tipo de sistema.

Aplicaciones Web 2.0. En esta categoría se enmarcan aquellos sistemas CIS cuyas interfaces clientes son implementadas completamente usando alguna tecnología Web o combinación de ellas, pero que no estén limitadas a ningún navegador Web. Group Unified History (GUH) [1] y SearchTeam [7] constituyen algunos ejemplos de este tipo de sistemas CIS. Este tipo de sistema toma ventaja de los mismos beneficios de los sistemas desarrollados como extensiones a navegadores Web. Sin embargo, superan a estos últimos por no requerir una sobrecarga adicional por cuestiones de instalación y configuración. Además, no están limitados a un navegador Web en particular, minimizan la carga cognitiva involucrada para su uso y facilitan una correcta convergencia de la búsqueda de información con otras partes de la colaboración. Por estas razones se considera a este tipo de aplicaciones, como el futuro de las interfaces Web de búsqueda en el contexto de CIS.

En esta sección se han expuesto algunas de las principales ventajas y desventajas de las principales infraestructuras en las cuales se han desarrollado las interfaces Web de búsqueda en el contexto CIS hasta la actualidad. Se mostraron algunas de las causas por las cuales este tipo de sistemas no han sido adoptados ampliamente en la práctica, a pesar de los grandes beneficios que pueden ofrecer a usuarios con necesidades compartidas de información. Entre las principales limitantes se encuentran: la carga cognitiva involucrada en el uso de estos sistemas; la elevada curva de aprendizaje, incluso para su uso inicial y la falta de una correcta convergencia de la búsqueda de información con otras partes de la colaboración[9]. Este trabajo se enfocará en cómo minimizar la carga cognitiva requerida para la aceptación de estos sistemas por parte de los usuarios.

3 Dimensiones Cognitivas

Las Dimensiones Cognitivas representan una herramienta analítica que se enfoca en el proceso de interacción, en lugar de análisis estáticos de artefactos de información; estos últimos representan los medios empleados para almacenar, manipular y visualizar la información [5]. Estas dimensiones difieren de otras pro-puestas analíticas tales como GOMS/KLM, en que no requieren de un análisis especializado y detallado sobre características específicas de la interfaz que consuman mucho tiempo. Por el contrario, éstas permiten la comparación y discusión de una interfaz con otras propuestas empleando términos más amplios, representados como diferentes dimensiones. Durante los últimos años el número de dimensiones identificadas ha ido en ascenso, pero solo algunas de ellas resultan útiles para analizar los sistemas CIS. Resulta importante resaltar que aunque las dimensiones seleccionadas reflejan diferentes aspectos de interacción, éstas no son completamente independientes.

- *Compromiso Anticipado*: refleja la secuencia de pasos que un usuario debe realizar para lograr un resultado específico [4].
- *Viscosidad*: evalúa la resistencia al cambio de un diseño.

- *Dependencias Ocultas*: evalúa la presencia de enlaces/vínculos ocultos entre los componentes del sistema CIS cuya existencia puede ser difícil de aprender por los usuarios.
- *Visibilidad*: refleja el grado de simplicidad en que la información es mostrada por el sistema CIS.
- *Consistencia*: evalúa el grado de similitud en el que un usuario puede lograr metas similares en diferentes partes de la interfaz del sistema.
- *Operaciones de Esfuerzo Mental*: se refiere a las operaciones que requieren la atención absoluta del usuario para su ejecución.
- *Expresividad del Rol*: refleja el grado de expresividad visual en el que los diferentes componentes de una interfaz reflejan su propósito y las operaciones disponibles en estos.
- *Evaluación Progresiva*: evalúa el grado de dificultad en que el usuario puede determinar en un momento dado del proceso de búsqueda, que documentos ha salvado o comentado, con el propósito de evaluar el progreso de la búsqueda.

4 Tácticas de Usabilidad

El principal objetivo de un diseño centrado en el usuario, consiste en el desarrollo de sistemas con un alto grado de usabilidad. Un sistema usable generalmente significa que el sistema es fácil de leer, tiene un uso efectivo y provee agradable experiencia al usuario. A partir de este principio, con el propósito de obtener un sistema CIS usable, se debe tomar en cuenta dónde y quién va a usar el sistema. Estos factores resultan de gran importancia pues, aunque muchos de los problemas para el desarrollo de sistemas CIS han sido resueltos, algunos problemas de usabilidad todavía no han sido eliminados. Tomando como base esta situación se proponen ocho tácticas de usabilidad basadas en las dimensiones cognitivas analizadas, bajo la percepción de que éstas pueden contribuir al desarrollo de sistemas CIS consistentes de acuerdo con las habilidades cognitivas de sus usuarios. A continuación se realiza una descripción de las tácticas propuestas, es válido aclarar que aunque éstas est 'an diseñadas para sistemas CIS pueden ser aplicables a sistemas tradicionales.

1. *Evitar errores y frustraciones*: la interfaz de usuario del sistema CIS debe ser intituitiva y de fácil uso por los miembros del grupo de colaboración. Las acciones colaborativas deberán ser implementadas seguiendo patrones establecidos por los sistemas de búsqueda de información tradicionales con el objetivo de facilitar su uso. Los componentes que representen acciones de awareness complejas, deben reflejar e intuir a los usuarios las operaciones que representan con el objetivo de tener una mayor visibilidad y expresividad de los roles que desempeñan.
2. *Distribución correcta de componentes*: los sistemas CIS deben tratar de extender los sistemas de búsqueda de información tradicionales, con el menor número de cambios referentes a la interfaz del usuario. Esta extensión deberá realizarse con mucho cuidado y con un alto nivel de consistencia para reducir la resistencia al cambio por parte de los usuarios.

3. *Reconocer en lugar de recordar*: un sistema CIS debe emplear componentes colaborativos que se conformen a partir de otros con los que el usuario se encuentre familiarizado y conozca como emplearlos. El uso de componentes, así como las metáforas visuales empleadas deben de tratar de mantener patrones ya establecidos por los sistemas tradicionales, para facilitar su uso y aceptabilidad por el usuario.
4. *Visibilidad de Información*: Los resultados de búsqueda de un sistema CIS deben proveer suficiente metainformación para facilitar al usuario evaluar y determinar la utilidad de un resultado específico. El usuario debe tener la posibilidad del evaluar el estado actual de su búsqueda y la de sus colaboradores en cualquier momento durante el proceso de búsqueda.
5. *Accesibilidad de acciones colaborativas*: un sistema CIS debe facilitar el uso de aceleradores de teclas que permitan a los usuarios ejecutar acciones colaborativas y manipular la información de una forma más eficiente. Acciones de anotaciones, evaluaciones y recomendaciones de resultados de búsquedas, deben poseer una buena accesibilidad por parte del usuario para facilitar el proceso de colaboración.
6. *Evitar operaciones de esfuerzo mental*: la información que se presente a los colaboradores debe ser legible y no requerir de grandes esfuerzos mentales para su interpretación durante el proceso colaboración.
7. *Colaboración simple*: un sistema CIS debe proveer al usuario un entorno de trabajo colaborativo transparente y sencillo que propicie la colaboración síncrona. El sistema debe proveer mecanismos que permitan a los miembros del grupo conocer en todo momento las acciones de sus colaboradores en el sistema, para facilitar la coordinación de sus actividades.
8. *Comunicación síncrona efectiva*: proveer mecanismos de comunicación afectiva donde los usuarios puedan transmitir sus estados emocionales (alegría, tristeza, enojo) y el grado de prioridad de sus mensajes (importante, urgente). Contribuirá a minimizar esfuerzo adicional requerido para coordinar las tareas del grupo ante la pérdida de los canales de comunicación (señas con la cabeza, gestos, sonidos) producto de una comunicación basada en texto.

5 Caso de Estudio

En esta sección se presenta un caso de estudio el cual ejemplifica las tácticas de usabilidad propuestas para disminuir la carga cognitiva durante sesiones de búsqueda colaborativas, las que serán referidas en este trabajo como CoSS (acrónimo de **Co**llaborative **S**earch **S**ession). Para reflejar la ejemplicación de las tácticas se utilizará la letra *T* seguida del número de la táctica, por ejemplo: *T1* - Evitar errores y frustraciones.

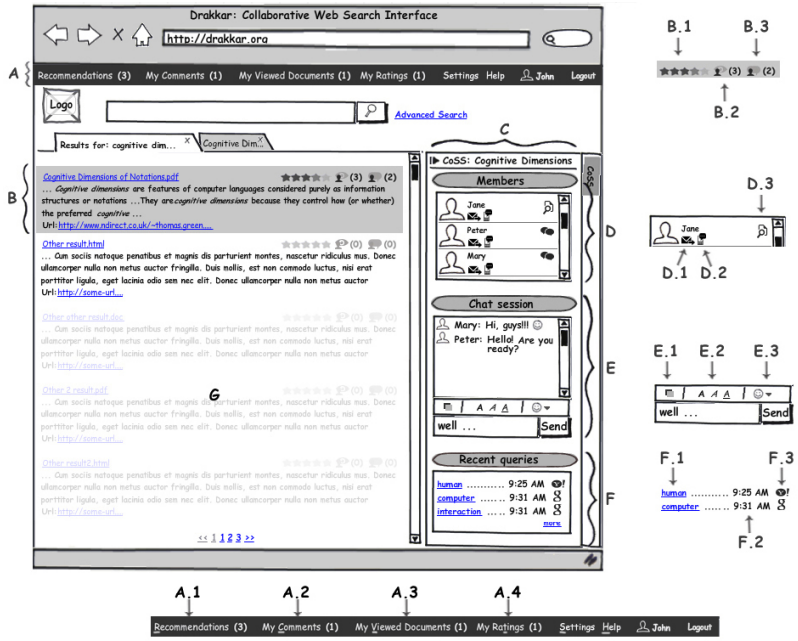


Fig. 1. Prototipo del cliente *Drakkar*. (a) panel del información y configuración, (b) metainformación de documentos, (c) panel de la CoSS, (d) miembros de la CoSS, (e) mensajería instantánea, (f) historial de consultas, (g) panel de resultados de búsquedas.

El proceso de evaluación y refinamiento de las tácticas de usabilidad propuestas se realizó a través conjunto de sesiones con estudiantes de postgrado de la disciplina de Interacción Persona-Ordenador (HCI, por sus siglas en inglés). Este proceso de evaluación tuvo lugar durante un curso postgrado sobre HCI desarrollado en Febrero del 2013 durante la segunda edición de la Escuela Internacional de Invierno de la Universidad de las Ciencias Informáticas de Cuba, e impartido por profesores del *Institute for Design and Assessment of Technology, Vienna University of Technology.*

Para el estudio de evaluación realizado se emplearon 27 personas, de las cuales 5 eran mujeres y 22 hombres, para un 18,52 % y 81,48 % respectivamente. La formación académica de este personal era la siguiente 74,07 % Ingenieros Informáticos, 22,22 % Licenciados en Ciencias de la Computación y 3,70 % Ingenieros en Telecomunicaciones. Resulta válido aclarar que a pesar de la diversidad académica de este personal, los mismos contaban con una gran experiencia profesional en actividades desarrollo de software y diseño de interfaces de usuario. Como objeto de estudio para la presente evaluación se desarrolló el prototipo CIS Drakkar. El proceso de evaluación se desarrolló durante 5 sesiones de intercambio entre los autores y los evaluadores. Este proceso inició con una charla donde se realizó una presentación a manera de introducción del trabajo que se estaba desarollando, resaltando aspectos claves CIS y HCI, acto

seguido comenzaron las sesiones donde fueron presentados diferentes prototipos de la interfaz de Drakkar. En estas sesiones el colectivo de autores intercambió con el personal seleccionado las diferentes propuestas de tácticas de usabilidad y como éstas se evidenciaban en el prototipo presentado. Las diferentes sesiones de intercambio facilitaron el proceso de retroalimentación entre los autores y los evaluadores, dando lugar a un proceso de rediseño iterativo de las tácticas de usabilidad y la interfaz final del prototipo.

Drakkar es un prototipo CIS diseñado para permitir una colaboración remota explícita entre sus usuarios de una forma síncrona y asíncrona durante sesiones de búsqueda. Este prototipo está basado en una arquitectura cliente-servidor, en la que el servidor es el encargado de la implementación de varias técnicas colaborativas como: conciencia colectiva (awareness), división del trabajo y persistencia. La interfaz de usuario (IU) del cliente de este prototipo es completamente Web (ver Sección 2.3). La selección de este escenario estuvo motivada por los beneficios que en la actualidad pueden brindar este tipo de aplicaciones, para la extensión de los sistemas tradicionales de búsqueda de información hacia sistemas con comportamientos más sociales. Drakkar dispone de una interfaz que le permite al usuario navegar por varias secciones y realizar un grupo de operaciones sin tener la necesidad de recargar o cambiar de página. La interfaz se encuentra dividida en tres secciones principales (*T2*) *A*, *G*, *C* como se puede apreciar en la Fig. 1, las cuales serán descritas con más detalle a continuación.

La sección *A* representa un panel de información donde el usuario puede observar el estado de su actividad de búsqueda en todo momento (*T4*). El usuario puede visualizar las recomendaciones de documentos recibidas por miembros de su CoSS (Fig. 1 A.1), sus comentarios sobre los resultados de búsqueda (Fig. 1 A.2), los documentos revisados (Fig. 1 A.3) y evaluados (Fig. 1 A.4) por él durante la sesión de búsqueda actual. El acceso a una información más detallada de estas categorías solo requiere por parte del usuario accionar el acelerador de teclas específico, dar clic sobre ésta, o simplemente posicionar el puntero del ratón (mouse) sobre ella (*T5*).

El panel de resultados (sección *G*) comprende el área de mayor tamaño en la interfaz del prototipo, debido a que la búsqueda de conjunto con la colaboración constituyen los principales objetivos perseguidos. En este sentido se ha tratado de mantener el estilo tradicional de representación de los resultados de búsqueda (*T1*) por los diferentes motores de búsqueda (ej. Google, Yahoo!, Bing), así como la convención de los navegadores Web (ej. Firefox, Safari, Opera) de habilitar varias pestañas para los resultados de búsqueda (*T3*). Además, de incluir un conjunto de metadatos característicos en muchos casos de los sistemas gestores de contenidos y sitios de comercio electrónico (ej. Amazon), que faciliten la evaluación del resultado por parte del usuario (*T4*). Se hace uso de diferentes tonos de color para indicar que un resultado de búsqueda ha sido revisado por un miembro de la CoSS (Fig. 1 B), así como las propias revisiones del usuario (*T6*). De forma general cada metáfora visual (icono) empleada en el prototipo cuenta

con una descripción o tooltip para facilitar el entendimiento de los usuarios (*T1*) de no identificar de un primer momento su significado (*T4*).

Cada resultado de búsqueda como se puede apreciar en la Fig. 1 B visualiza:

- promedio de evaluación (Fig. 1 B.1) recibida por los usuarios del sistema.
- número de revisiones del resultado (Fig. 1 B.2), así como la lista de los autores de dichas revisiones en caso de posicionar el puntero del mouse sobre el icono que lo representa.
- número de comentarios del resultado (Fig. 1 B.3), así como la lista de los autores de dichos comentarios en caso de posicionar el puntero del mouse sobre el icono que lo representa.

La sección *C* de la Fig. 1 encapsula gran parte de las acciones de awareness implementadas en Drakkar. Este panel provee las principales operaciones que pueden realizar los usuarios dentro de una CoSS, además ha sido diseñado para poder ser colapsado por el usuario y obtener un mayor aprovechamiento del espacio de la interfaz (*T2*).

En el apartado *D* de la Fig. 1 se muestra una lista de los miembros de la CoSS y se proporcionan algunos elementos adicionales aparte de la clásica imagen del usuario y su nombre. Por ejemplo, facilita la colaboración asíncrona entre los miembros de la CoSS mediante la posibilidad del envio de correos (Fig. 1 D.1) y mensajes de textos a los celulares (Fig. 1 D.2) de los usuarios (*T7,T8*), esta última operación es posible solo en caso que el usuario active este servicio. Adicionalmente, se incluye una metáfora visual que representa la operación en curso del usuario (Fig. 1 D.3), con el objetivo de reducir el esfuerzo requerido por los miembros de la CoSS para conocer la labor de otros y coordinar sus acciones (*T7,T8*). En la Fig. 1 se puede observar como el usuario *John* no requiere de ninguna comunicación e interacción para conocer las acciones que se encuentran realizando sus colaboradores (Jane - buscando, Mary y Peter - chateando).

Drakkar está diseñado para soportar una comunicación síncrona (Fig. 1 E) entre sus usuarios mediante un sistema de mensajería instantánea (IM, por sus siglas en inglés). Este sistema incluye características hasta ahora no provistas por los sistemas CIS desarrollados, aunque éstas han sido implementadas por sistemas IM tradicionales y por otros sistemas de mensajería asíncrona como el tradicional correo electrónico (*T1*). Con el propósito de reducir las carencias de la comunicación basada en texto [2] Drakkar incluye la posibilidad de transmitir el estado emocional del usuario (*T8*) durante la comunicación síncrona (Fig. 1 E.3). Permite además la posibilidad de resaltar la relevancia y el grado de urgencia de los mensajes (*T8*) a través de la aplicación de estilos de fuentes (Fig. 1 E.2) y colores (Fig. 1 E.1). Los usuarios pueden observar el historial de consultas personal, así como de la CoSS (*T4, T7*). De esta forma el usuario en todo momento pueda de conjunto con las funcionalidades del panel de información (Fig. 1 A.2 - A.4) realizar una evaluación del proceso de búsqueda actual. Por ejemplo, el usuario puede observar el término(s) de consulta realizada (Fig. 1 F.1), la fecha de ejecución de la misma(Fig. 1 F.2) y el motor de búsqueda empleado (Fig. 1 F.3).

6 Conclusiones

En este trabajo se han propuesto un conjunto de tácticas de usabilidad basadas en varias dimensiones cognitivas y fundamentos teóricos de los campos HCI & CSCW, para minimizar la carga cognitiva a la que se someten los usuarios de interfaces multi-usuarios del campo de CIS. Se realizó además un estudio de los principales escenarios en los que se han desarrollado los prototipos de sistemas CIS, considerando que las aplicaciones implementadas completamente con tecnologías Web no dependientes de los navegadores, representan el futuro de este tipo de software social. El proceso de evalución iterativo desarrollado sirvió como base para redefinir y mejorar el enfoque de las tácticas de usabilidad propuestas. Se presentó un caso de estudio el cual refleja como las tácticas de usabilidad propuestas pueden mejorar el desarrollo de sistemas CIS más usables y con una menor carga cognitiva requerida para su uso.

Acknowledgements. Este trabajo ha sido financiado por el Ministerio de Educación y Ciencia español y la Junta de Andalucía, por medio de los proyectos TIN2011-28538-C02-02 y de excelencia TIC-04526, respectivamente.

References

1. M. Aneiros and V. Estivill-Castro. Usability of real-time unconstrained www-co-browsing for educational settings. In Web Intelligence, 2005. Proceedings. The 2005 IEEE/WIC/ACM International Conference on, pages 105-111. IEEE, (2005).
2. A. Dix, J. Finlay, G. D. Abowd, and R. Beale. Human-Computer Interaction. (2004).
3. J. M. Fernández-Luna, J. F. Huete, R. Pérez-Vázquez, and J. C. Rodríguez-Cano. Improving Search-Driven Development with Collaborative Information Retrieval Techniques. In Proceedings of the Third Workshop on Human-Computer Interaction and Information Retrieval. Washington DC, USA, pages 42-45, (2009).
4. G. Golovchinsky. Cognitive Dimensions Analysis of Interfaces for Information Seeking. In Proceedings of the Third Workshop on Human-Computer Interaction and Information Retrieval, pages 50-53. ACM, (2009).
5. T. Green and A. Blackwell. Cognitive dimensions of information artefacts: a tutorial. In BCS HCI Conference, (1998).
6. J. Grudin. Groupware and social dynamics: eight challenges for developers. In Communications of the ACM, pages 92-105. ACM, (1994).
7. T. KuFmann, S. Elbeshausen, and T. M. C. Womser-Hacker. Discovering Ellis' Phases of Information Seeking Behavior in Collaborative Search Processes. (2010).
8. M. R. Morris and E. Horvitz. SearchTogether: an interface for collaborative web search. In Proceedings of the 20th annual ACM symposium on User interface software and technology, pages 3-12. ACM, (2007).
9. C. Shah. Collaborative Information Seeking: The Art and Science of Making The Whole Greater Than The Sum of All (The Information Retrieval Series)(p. 185), (2012).

Integrating a Conversational Recommender System within a 3D Collaborative Environment

David Contreras, Maria Salamó, Anna Puig and Inmaculada Rodríguez

Dept. Matemàtica Aplicada I Anàlisi, Universitat de Barcelona, Spain.

dcontrag7@alumnes.ub.edu, {maria,anna,inma}@maia.ub.es

Abstract. E-commerce on a virtual world environment allows user immersion in a content-based space. This environment provides both access to products information and interaction with other users. In real life a user involved in a buying process usually interacts with other users, which may help her for finding the best product that satisfies her preferences. Conversational recommender systems have shown to be useful on assisting users in the buying process. This paper is focused on incremental critiquing, a particular conversational recommender system, that uses critiquing as a feedback to efficiently personalize products. In this paper, we propose to integrate an incremental critiquing recommender system in a collaborative virtual world environment with the aim of giving the user an E-commerce space with similar features to the real world. Our results denote that the integration of a recommender system within a collaborative virtual world significantly improves the performance of conversational recommendation process.

Keywords: Conversational recommenders, Incremental Critiquing, Virtual world, Collaborative environment, User modeling.

1 Introduction

In just a few decades Internet has experienced a gradual transformation from document-oriented perspective towards a content-oriented perspective. Nowadays, this transformation is visible as users are finding for items to buy, text articles or news to read, among the infinite number of content that currently exists on the web. Additionally, this transformation has also stimulated users to provide content in order to collaborate and help other users in their findings. In real life, the social interaction has an important role in commerce and will definitely be an important factor in the future of E-Commerce [1, 2]. Hence, any technology or platform that facilitates the interaction and collaboration among users of E-Commerce could improve user's experience in the buying process. Currently, one type of collaborative environments that facilitates interaction between users and offers a content-oriented interface corresponds to virtual worlds [3, 4]. 3D virtual worlds are persistent, synthetic, three dimensional spaces where users (represented by avatars) interact each other and with the environment [3]. Virtual worlds can be used to perform social, educational and business activities in a

simulated and interactive environment where the user feels immersed, having a sense of place, a sense of being there [4].

A virtual world is an alternative solution to 2D interfaces implementing E-commerce applications. However, with the wide range of products and their specific characteristics, at times is difficult for a user to search and select the desired product within a large space of possibilities. Conversational recommender systems help users to navigate through a product space, alternatively making product suggestions and eliciting user feedback [5]. Thus, a typical session with a conversational recommender system takes the form of a series of *recommend-review-revise cycles*: in this cycle the user is presented with one or more item recommendations; she provides some form of feedback regarding the suitability of these items; and the recommender updates its user model according to this feedback and proceeds to the next recommendation cycle [6]. Critiquing [6,7] is a powerful form of user feedback, where users provide limited feedback at the feature-level by constraining a feature's value-space. For example, a user may request a cheaper product, thus critiquing the price feature. Incremental Critiquing [8] and some algorithms derived from it [9,10,11,12] are conversational recommenders that use this type of user feedback.

Relatively little work has been focused on the integration of recommender systems within virtual worlds [13,14]. Most of them only give attention to the interaction between shopping assistants and the users, without exploring the collaborative features in the virtual world, which we believe is the real advantage of the integration a recommender system and a virtual world. In this paper, we propose the integration of the recommender system within a virtual world environment, empowering the interaction and collaboration between similar users, making use of natural features from virtual world like chat, talk, gestures, and movements. To this end, we first describe the process of interaction among the virtual world and the conversational recommender. Accordingly to this interaction, we adapt the incremental critiquing algorithm and we update the user model. In order to analyze the benefits of the proposed integration, we simulate and evaluate the collaborative behavior in a simulated recommender. Our results denote that allowing a collaborative interaction among users significantly improves the performance of a conversational recommendation process.

The rest of the paper is organized as follows: Section 2 introduces the incremental critiquing; Section 3 describes in depth the proposal; Section 4 evaluates the effectiveness of the proposal; Finally, Section 5 concludes the paper.

2 Background on Incremental Critiquing

This section briefly describes the basis of the incremental critiquing (IC) algorithm. First of all, we introduce some basic definitions. Let $CB = \{p_1, \dots, p_n\}$ be the case base defined as a set of products or cases for recommendation, where p_i is the ith product, and $|CB|$ is the total number of products. Let $F = \{f1, \dots, fm\}$ be the set of features that describes each product, where $|F|$ is the total number of features. We denote a particular feature f_s of case p_i as $p_i^{f_s}$. For example $p_i^{f_1}$ as $p_i^{f_2}$ refer to the first and the second features in case p_i, respectively.

Let $U = \{U_1, \ldots, U_k\}$ be the user model defined as a set of critiques, where U_j is the th critique, and $|U|$ is the number of critiques. The user model is incrementally updated in each cycle of the conversational recommender. Thus, U shows the preferences of the user over time. Let $U^{f_s} \subseteq U$ be the set of critiques from U made for a particular feature f_s, and $|U^{f_s}|$ is the number of critiques in U that refers to feature f_s. The IC algorithm uses the satisfaction measure, described as $\delta(p_i, U_{cq}) = Satisfies(p_i, U_{cq})$. The satisfaction measure $\delta(p_i, U_{cq})$ shows for a particular product p_i and a critique U_{cq} if the critique is satisfied by this particular product. This measure returns 1 when the critique is satisfied, 0 otherwise.

The incremental critiquing implementation assumes a conversational recommender system in the style of Entrée [7]. The incremental critiquing process consists of four main steps: (1) a new product p_r is *recommended* to the user based on the current query q and previous critiques if there are any; (2) the user *reviews* the recommendation and applies a directional feature critique, U_{cq}; (3) the query q is *revised* for the next cycle; (4) the user model $U = \{U_1, \ldots, U_k\}$ is *renewed* by adding the last critique U_{cq} and pruning all the critiques that are inconsistent with it. Finally, the recommendation process terminates either when the user retrieves a suitable case or when this user explicitly finishes the recommendation process. This recommendation process is highly influenced by the user model U containing previous consistent critiques, which is incrementally updated in each cycle. In incremental critiquing, instead of ordering the filtered cases on the basis of their similarity to the recommend case, it also computes a *compatibility* score as follows:

$$ {}^{p_i}_{t}(U) = \frac{\sum_{j=1}^{|U|} \delta(p_i, Uj)}{|U|}, \tag{1}$$

where ${}^{p_i}_{t}(U)$ is the *compatibility* score of candidate case p_i at instant t given an user model U. The satisfaction function δ returns 1 if case p_i satisfies the critique U_j or 0 otherwise, and $|U|$ stands for the total number of critiques in the user model U. Thus, the compatibility score is essentially the percentage of critiques in the user model that case p_i satisfies. Then, the compatibility score and the similarity of a candidate case p_i to the current product recommendation q are combined to obtain an overall *quality* score Q:

$$Q(p_i, q, U) = \beta \cdot {}^{p_i}_{t}(U) + (1 - \beta) \cdot S(p_i, q) \tag{2}$$

where S is the similarity function based on an Euclidean distance, and β is set to 0.75 by default as denoted in [8]. The quality score Q is used to rank the filtered cases prior to the next cycle, and the case with the highest quality is then chosen as the new recommendation.

3 Proposal

Under the hypothesis that user collaboration is useful in the buying process, we propose the integration of a recommender system within a virtual world environment.

In this section, we firstly present the conceptual architecture of the proposal, the communication protocol for the users and their state in the conversational recommendation process. Secondly, we describe the conversational recommender algorithm that allows collaboration and interaction among users.

3.1 Virtual World Environment

The conceptual architecture of our virtual world is based on the classical client-server model. The virtual world server stores an asset inventory for the correct display of both the product and the information shown by the recommender system and mediate communication among users and between users and their corresponding recommender systems. On the other hand, on the virtual world client, each recommender system (see Figure 1 (a)) within the virtual world is represented through a 3D object. In the same layer, users are represented by avatars as depicted in Figure 1 (b). Users have available different functions to communicate using text and voice chat, to walk around the space and perform gestures. Users can employ these functions to provide feedback to the recommender system. As a result, each avatar in the virtual world has associated a user model that contains her preferences and that will be used in the conversational recommendation process.

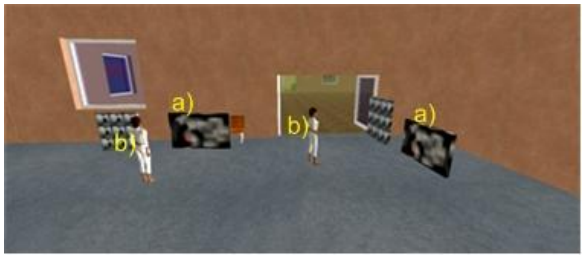


Fig. 1. Virtual world scene

Figure 1 displays several elements: (a) the image of the current recommended product is displayed with a 3D object and the features of this product are displayed as text on top the object; and (b) the users represented by avatars. The user starts their session in the client application (e.g., Second Life Viewer) and the user communicates with the recommender system by a chat.

Initially, a recommender system is in off state until the user starts the communication. Once the user has logged, it can interact with the recommender system (see Figure 2 (a)). Here, the state of the user only contains its identification. During the conversational recommender process, the user moves to different states. Once the user establishes a communication with a recommender system (see Figure 2 (b)), the user model is created and its state is defined as *individual critiquing*. At this state the user is able perform critiques by using the chat. For each critique, the recommender provides a new recommendation and it updates both the user model and state.

In this paper so far, we have argued that collaboration is a primarily factor in the buying process. Then, assuming that the user is interested for a product recommended

to another user, she must manifest it as (*like*), her user model is accordingly updated considering the latest recommended product of the other user, and the user state will be changed to *grupal critiquing* (see Figure 2 (c)).

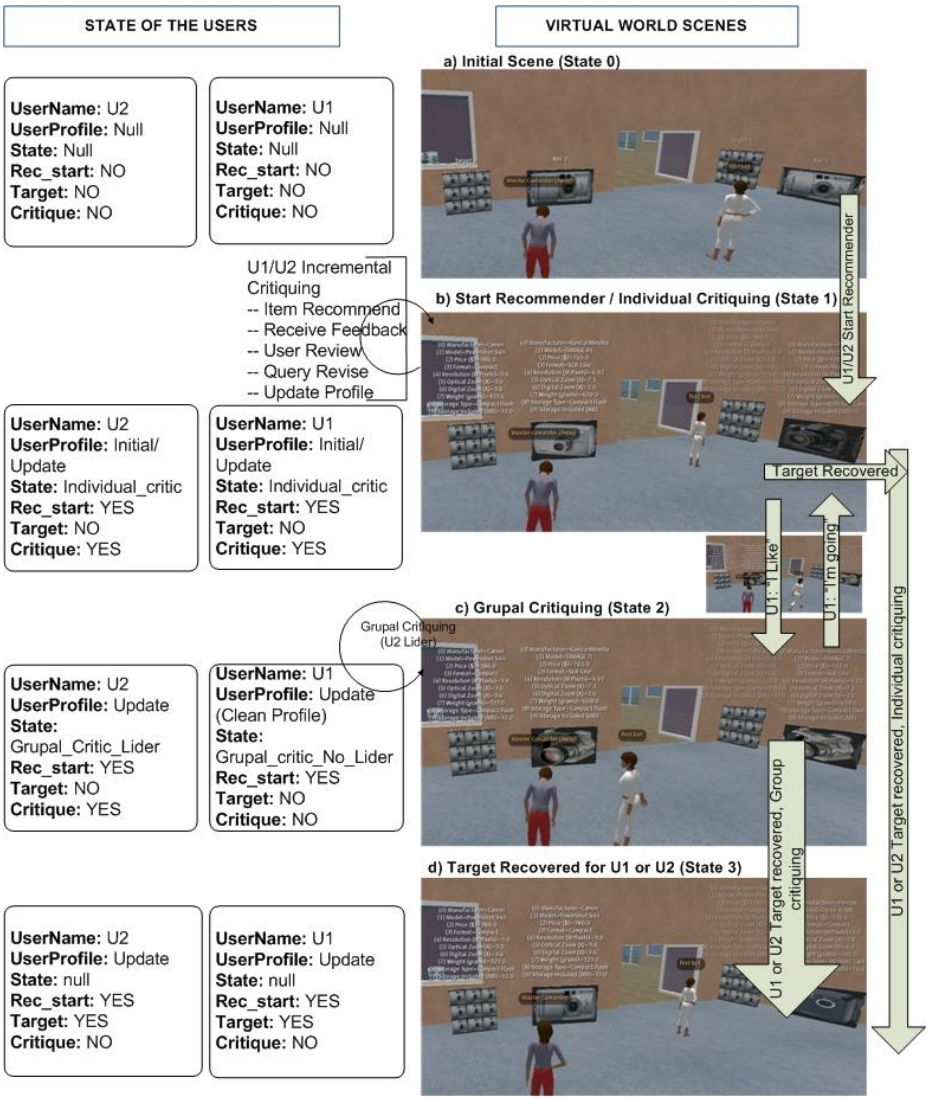


Fig. 2. States of the users: (a) initial scene, (b) start recommender/individual critiquing, (c) grupal critiquing, (d) target recovered.

In this state the user that joins to start collaboration is not able to make critiques (the user is in a state of consensus with another user). They can talk each other using text chat and voice chat, but only the one that receives the collaborator is able to make

critiques. At any time, the user is free to continue the process of individual critiques but it must manifest (*going*). In that case, she can continue individually to her own recommender system (her status is changed to *individual critiquing*). Finally, any user in state of *individual critiquing* or *grupal critiquing* may find the target product (see Figure 2 (d)). At this point, the user model is updated and the user will be unable to make new critiques as the conversational recommender process is finished.

3.2 Incremental Critiquing Virtual World

In real life users tend to collaborate in order to find the desired product. With the aim of emulating the collaborative behavior, we extend the incremental critiquing [8] algorithm for allowing -as a new feedback mechanism- the selection of product candidates from other users (called a *collaborative user*).

We propose a conversational recommender algorithm named *Incremental Critiquing Virtual World* (ICVW) that allows the interaction and collaboration among users (*grupal critiquing*, see Figure 2 (c)). The ICVW algorithm is depicted in Figure 3. As parameters, the ICVW algorithm receives an initial query, q, together with the Case Base, CB. Firstly, the algorithm requires some initializations that consist in assigning an empty critique U_{cq}, an empty user model U, an empty case base of recommendations, CB_r, and an initial feedback set up to "*Critiquing*". The rationale behind the feedback initialization is to let the recommender focus an initial point on the data space before the user starts the interaction with the system.

Lines 7 to 18 in the algorithm are devoted to the incremental critiquing virtual world process. At each iteration cycle, the system applies five main phases. First phase depends on user feedback and it may start the *ItemRecommend* or the *ItemSelecVirtualWorld* procedure. In the *ItemRecommend* (see lines 8-10 and 20-26), a new product p_r is recommended to the user based on the current query q and previous critiques if there any. Additionally, this procedure returns a collection of relevant products that can be shown to the user. A relevant product is a case that satisfies the last critique made by the user. Note that when the critique is null, all cases satisfy it. On the other hand, in the *ItemSelecVirtualWorld* procedure (see lines 11-12 and 27-32) the query is a selected product from another user present in the Scene. This procedure also returns a collection of products minus the selected one.

Second phase consist of receiving the feedback action the user wants to perform, see lines 14 and 33-37. Currently, two feedback actions are possible: Critiquing and Selection in the Virtual World. Next, ICVW algorithm implements the review phase through two main steps: (1) it gets a directional feature critique or the selection of a product from another collaborative user, see lines 40-46; (2) it removes current product recommendation p_r from the case base CB, see line 47. In fourth phase the query q is revised for next cycle. Finally, ICVW algorithm renew the user model by invoking contradict and refine procedures, see lines 7 and 55-63. The contradict procedure removes any critique that contradict current critique U_{cq} if there is any or that contradict the current product recommendation p_r. For example, in a camera recommendation process, if the user model has a critique $[manufacturer, \neq, Sony]$ and the current product recommendation is a Sony camera that has been selected by the user

from her collaborative user, the critique $[manufacturer, \neq, Sony]$ will be removed as it contradicts current recommendation. On the other hand, the refine procedure improves the critiques in the user model. For example, a user model with $[price, \leq, \$1500]$ and $[price, \geq, \$500]$ critiques that receives a selection whose price is $\$1000$, the first critique will be refined to $[price, \leq, \$1000]$. As in incremental critiquing, the recommendation process terminates when the user retrieves a suitable product or when she explicitly finishes the recommendation process.

```
Input: q:query, CB: case base, U_cq: current critique, p_r current product recommendation, U: usermodel
1  define ICVW(q,CB)
2  begin
3     Ucq:=null
4     U:=null
5     CBr:=null
6     feedback:="Critiquing"
7  repeat
8     if feedback is Critiquing then
9        pr, CB_r ←ItemRecommend(q, CB, U_cq, U)
10    else
11       //feedback is Selection from other user
12    p_r, CB_r ←ItemSelecVirtualWorld(CB, q)
13    end
14    feedback ← ReceiveFeedback()
15    U_cq, p_r, CB ←
      UserReview(p_r, CB, CB_r, feedback)
16    q ←QueryRevise(q, p_r)
17    U ←ModelRenew(U, p_r, feedback)
18 until UserAccepts(pr)
19 end
20 define ItemRecommend(q, CB, U_cq, U)
21 begin
22    CB' ← {p ∈ CB | δ(p, cq) }
23    CB' ← sort all p_i ∈ CB in
            decreasing Q(p_i, q, U) //see eq. 2
24    p_r ←most quality product in CB'
25    return p_r, CB'
26 end
27 define ItemSelecVirtualWorld(CB, q)
28 begin
29    CB' ← CB − q // Relevant Case of
collaborative user
30    p_r ← q
31    return p_r, CB'
32 end
33 define ReceiveFeedback()
34 begin
35    feedback ← user activates a feedback action
36    return feedback
37 end
38 define UserReview(p_r, CB, CB_r)
39 begin
40    if feedback is Critiquing then
41       U_cq ←user critique for some f ∈ p_r
42       p_new := null
43    else
44       p_new ←user selection of a product from other collaborative user
         p_new ∈ CB_r
45       U_cq := null
46    end
47    CB ← CB − p_r
48    return U_cq, p_new, CB
49 end
50 define QueryRevise(q, p_r)
51 begin
52    q ← p_r
53    return q
54 end
55 define ModelRenew(U, U_cq, p_r, feedback)
56 begin
57    U ← U −contradict(U, U_cq, p_r, feedback)
58    U ← U −refine(U, U_cq, p_r)
59    if feedback is Critiquing then
60    U ← U + (< U_cq, p_r >)
61    end
62    return U
63 end
```

Fig. 3. Incremental Critiquing Selection Virtual World

4 Evaluation

We are especially interested in analyzing how collaboration environment of virtual world should improve the performance of a conversational recommender system. To this end, we have developed a simulator that emulates the actions of the user within the virtual world for finding a suitable product with the conversational recommender system. This simulator applies both the standard incremental critiquing (IC) and the

incremental critiquing virtual world algorithm (ICVW). We focus our performance on average session length, which measures the number of cycles that a user must work through before being presented with her ideal target product.

4.1 Setup

The experiment is limited to 2 domain products[1]: digital cameras and personal computer (PC). The case-base consists of 210 and 120 products respectively. In the domain of digital cameras, each product is described in terms of 10 features (manufacturer, model, price, etc.). In the PC data set, each product is described in terms of 8 features (manufacturer, processor type, processor speed, etc.).

In our experiments each case (base) in the case-base is temporarily removed and used in two ways. First, it serves as basis for a set of queries by taking random subsets of its features. We focus on subsets of one, three, and five features to allow us to distinguish between *hard*, *moderate*, and *easy* queries respectively. The *hard* queries represent a user with little knowledge of the product that satisfies her preferences. On other hand, the *easy* queries represent a user with high knowledge of the target product. Second, we select the case that is most similar to the original base. These cases are the recommendation targets for the experiments. Thus, the base represents the ideal query for a user, the generated query is the initial query provided by the user, and the target is the best available case for the user. In the experiment, we work with 300 queries for the camera and 240 queries for the PC domain. Each query is a test problem for the recommender. We repeat each query one hundred times and the recommendation sessions terminate when the target case is returned.

Finally, we have implemented the selection virtual world feedback (see Figure 3, Line 33) in the simulator selecting a random case of the user's relevant cases. Assuming that the interaction in the virtual world is collaborative and takes place between users with similar preferences, we only consider a subset of relevant cases in the range of the proximity (called preference ratio) to the target case from 5% to 30%. In this way, a range of 5% in a cycle of recommendation means that the product to be selected is very close to the user's preferences. In other hand, a range of 30% means that the product to be selected is not very close to the user's preferences.

4.2 Analysis of results

Figure 4 presents for each data set a graph with the average session length achieved by IC and ICVW approaches (from 5% to 30%) for hard, moderate and easy queries. The results show that IC presents an average session length of 7.63 and 5.23 cycles for the camera and PC data sets, respectively. On the other hand, ICVW with a 5% preference ratio to the target shows an average session length of 4.01 and 2.87 for the hard queries. This relative session length reduction represents a benefit of 47.5% for the camera and a 45.2% for the PC when using ICVW proposal in front of IC. Moreover, Figure 4 shows that the reduction in the session length depends on the percent-

[1] The datasets are fully described in [12]. They are available on demand.

age of preference ratio considered when collaborating. As expected, the graph also depicts that hard queries present larger session lengths than easy queries.

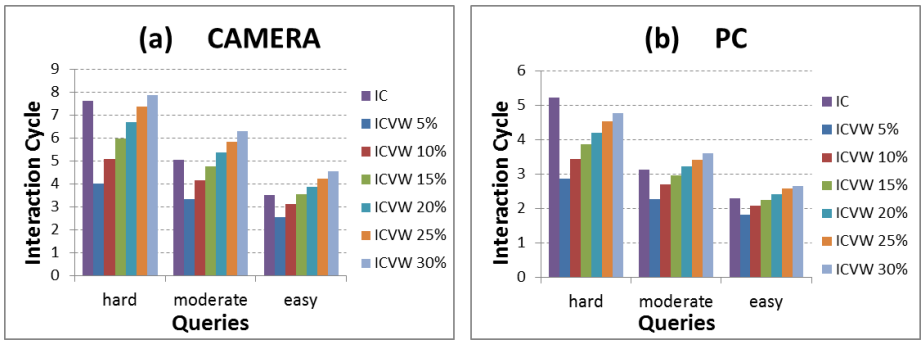


Fig. 4. Average session length for the camera and PC data sets

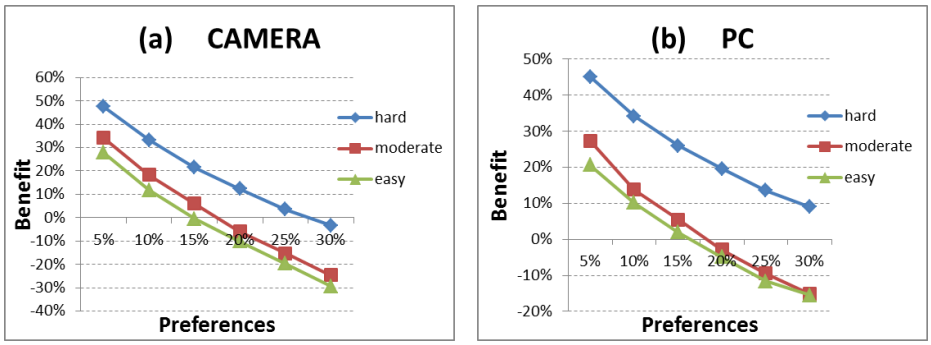


Fig. 5. Benefit of the ICVW proposal in front of the IC algorithm

Figure 5 illustrates the average benefit percentage over IC when using ICVW. We have averaged the results from 5% to 30% preference ratio to the target. Clearly the curve descends when the percentage of this preference ratio to the expected target is increased. These results suggest us that it is necessary to collaborate when a minimum preference over the product is reached.

Summarizing, ICVW had higher performance with hard queries. Our results simulate the level of collaboration between users in a virtual world. In this way, a user without much knowledge of what the required product could reduce the conversational recommendation process by selecting a product from another user with similar preferences. However, it has been proven that a minimum preference is needed.

5 Conclusions

This paper has addressed the integration of an incremental critiquing algorithm in a collaborative virtual world. We have described the integration of the recommender system in virtual world and have proposed the Incremental Critiquing Virtual World

algorithm (ICVW). Moreover, we have analyzed in depth the performance of the recommender system in this type of environment. To this end, we have simulated the user behavior in order to know the conditions that favor collaboration in a virtual environment. Our results depict that ICVW collaboration considerably reduces the average session length when the user interacts with a user with highly similar preferences and she is not really aware the product she is looking for (hard queries in our experiments). As part of our future work, we intend to analyze the performance with a live-user evaluation in the virtual world environment.

Acknowledgements. Work partially funded by the MICINN Grants TIN2009-14404-C02, TIN2012-38876-C02-02, AT (CONSOLIDER CSD2007-0022) and TIN2011-24220 Spanish research projects.

References

1. C. Thompson. Next-Generation Virtual Worlds. IEEE Internet Computing, 1089-7801,30-35.
2. S. Anton, C. Sierra, H. Berger. Implicit Training of Virtual Shopping. IADIS Press, 50-57, 2005.
3. J. Spence, Demographics of Virtual Worlds. Journal of Virtual Worlds Research. Vol. 1 Nº2, 2008.
4. N. Magnenat-Thalmann, H. Kim, A. Egges, S. Garchery. Believability and Interaction in Virtual Worlds. 11th International Multimedia Modelling Conference, 2-9, 2005.
5. D. Aha, L. Breslow, and H. Muñoz-Avila. Conversational case-based reasoning. Applied intelligence, 14:9-32, 2000.
6. J. Reilly, K. Mccarthy, L. Mcginty, B. Smyth. Dynamic critiquing. Advances in casebased reasoning, 3155:37-50, 2004.
7. R. Burke, K. Hammond, B. Young, The FindeMe approach to assisted browsing, IEEE Expert 4, 2:32-40, 2000.
8. J. Reilly, K. McCarthy, L. McGinty, B. Smyth. Incremental critiquing. In research and development in intelligent systems XXI, AI-2004:101-114, 2004.
9. M. Salamó, J.Reilly, L. McGinty, B. Smyth. Improving incremental critiquing. In Proceeding of: Irish Conference on Artificial Intelligence and Cognitive Science, 379-388, 2005.
10. M. Salamó, J.Reilly, L. McGinty, B. Smyth. Reducing critiquing repetition in conversational recommendation. In Proceeddings IX Workshop on multi-agent information retrieval and recommender systems at the Int. Joint Conf. on AI, 55-61, 2005.
11. D. McSherry, D. Aha. The ins and outs of critiquing. IJCAI'07 Proceedings of the 20th international joint conference on Artifical intelligence, 962-967, 2007.
12. M. Salamó, S. Escalera. Increasing Retrieval Quality in Conversational Recommenders, Knowlegde and Data Engineering, IEEE Transactions, 2:1876:1888, 2012.
13. B. Xu, Y. Yu. A Personalized Assistant in 3D Virtual Shopping Environment, Second International Conference on Intelligent Human-Machine Systems and Cybernetics, 266-269, 2010.
14. G. Guibing, E. Mohamed. A New Recommender System for 3D E-Commerce: An EEG Based Approach. Journal of Advanced Management Science, Vol.1, No.1, pp.61-65, March 2013.

Using Apriori Algorithm + Standard Deviation to Improve the Scalability of TSK-0 Rule Learning Algorithms

Javier Cózar, Luis delaOssa, and José A. Gámez

Computing Systems Department - I^3A
University of Castilla-La Mancha - Spain
{javier.cozar,luis.delaossa,jose.gamez}@uclm.es

Abstract. Some algorithms for learning Linguistic Fuzzy Logic-Based Systems part from the definition of the linguistic variables, and search a set of rules among those covering the input examples of the training data. Since the amount of candidate rules grows exponentially with the number of variables, sometimes they do not work with most datasets. This article proposes a method to make a pre-selection of candidate rules for the case of TSK-0 fuzzy rules. It bases on the combination of two criteria: the number of examples covered by each rule, i.e., its support, and the standard deviation of the output values for these examples. Experimental results show that the proposed method reduces the number of rules substantially, allowing the existing algorithms to deal with bigger datasets, while producing simpler models.

Keywords: Fuzzy Modeling, Evolutionary Fuzzy Systems, Apriori

1 Introduction

Fuzzy Rule Based Systems (FRBSs)[19] are generally used for prediction and modelling [12]. Their main advantage in relation with other paradigms, such as neural networks, is the interpretability. For this purpose, Linguistic FRBSs (LFRBSs) [13] use a database with the definition of the linguistic variables [20], and represent the fuzzy rules by means of linguistic labels.

There are many algorithms for LFRBSs learning [6, 2]. A specific type of them take the linguistic database as starting point, and only derive the rule base [18, 5]. In general, such algorithms generate candidate rules which cover the examples in the training dataset, and then select some of them and fix their consequents. Even with a number of input variables relatively small, some of these algorithms do not work. The reason for that is either that rules are too specific –when their number of antecedents is fixed to the number of variables–, or the huge amount of candidate rules (with different sizes) which can be generated from data.

The aim of this work is to provide a method to pre-select only those candidate rules which are potentially usefull for the case of *0-order Takagi-Sugeno-Kang* (TSK-0) rules with variable size.

Since one of the criteria related to the quality of a rule is its support, the Apriori [1] algorithm for mining association rules [16] can be adapted to obtain candidate fuzzy rules [10]. This has already been done in [14, 3], where this approach is used to learn fuzzy classifiers.

Even if the reduction achieved with Apriori is very significant, the search of the rule base over the resulting rule set may still not be possible. However, such a reduction makes it possible to evaluate each rule individually, which allows a second filtering stage. In this respect, as the evaluation criteria described in [5] for Mamdani-type rules is not valid for TSK-0 rules, we calculate the score of each rule as the standard deviation of the output values for the examples that it covers.

This paper is divided into 6 sections besides this introduction. First, sections 2 and 3 introduce some notation by describing TSK-0 FRBSs and the general process of rule learning. Then, Section 4 describes the adaptation of the Apriori algorithm for obtaining candidate rules, and Section 5 explains the use of standard deviation for the second filtering stage. Afterwards, Section 6 describes the experimental study of our proposal. Last, Section 7 presents some conclusions and future works.

2 TSK-0 Fuzzy Rule-Based Systems

Linguistic Fuzzy Rule-Based Systems (LFRBSs) [20] [13] are based on the use of a linguistic $DataBase$, which contains the linguistic variables and the specification of the linguistic terms. Thus, Linguistic Fuzzy Rules are composed of predicates as (X is A), where A is a linguistic label in the domain of X.

The use of linguistic variables makes LFRB models easier to interprete by human experts. However, the limitation in the number of fuzzy sets that can be used to build the fuzzy rules (only those defined in the $DataBase$) limits their precision. There are some variations of the original LFRBSs which aim to improve their accuracy with a small loss of interpretability. Takagi-Sugeno-Kang (TSK) rules [17] are the most important of these. In TSK systems, the consequent for each rule R_s is a polynomial function of the input variables $P_s(X_1, \ldots, X_n)$. Therefore, a rule is specified as:

$$R_s : \text{If } X_1 \text{ is } A_1^s \& \ldots \& X_n \text{ is } A_n^s \text{ then } Y = P_s(X_1, \ldots, X_n)$$

The order of a TSK FRBS refers to the degree of P_s. Hence, in TSK-0 systems the consequent is a constant value, and the rules are expressed as:

$$R_s : \text{If } X_1 \text{ is } A_1^s \& \ldots \& X_n \text{ is } A_n^s \text{ then } Y = b_s$$

Given an example $e_l = (x_1^l, \ldots, x_n^l, y^l)$, the output in a TSK system is obtained as the weighted average of the outputs produced by each individual rule R_s in the *RuleBase* $\mathcal{RB}_o$:

$$\hat{y}_o^l = \frac{\sum_{R_s \in \mathcal{RB}_o} h_s^l P_s(x_1^l, \ldots, x_n^l)}{\sum_{R_s \in \mathcal{RB}_o} h_s^l}$$

where $h_s^l(e_l) = T(A_1^s(x_1^l), \ldots, A_n^s(x_n^l))$ is the matching degree of the example e_l with R_s, and T is a T-norm. In TSK-0 rules, the expresion can be reduced to

$$\hat{y}_o^l = \frac{\sum_{R_s \in \mathcal{RB}_o} h_s^l b_s}{\sum_{R_s \in \mathcal{RB}_o} h_s^l} \tag{1}$$

3 Learning Linguistic Fuzzy Rules

Data-Driven methods which learn the *rule base* of a LFRBS [18, 5, 11, 4] take as input both a *data set* $\mathcal{E} = \{e_1, \ldots, e_l, \ldots, e_N\}$, and a *linguistic data base* which contains the definition of each (input) linguistic variable X_i: its domain, the fuzzy partition, and the linguistic labels $\mathcal{A}_i$.

This kind of algorithms generate a set of candidate rules from the labels in the database in such a way that every example $e_l \in \mathcal{E}$ matches at least one rule. Afterwards, the algorithms choose some of the candidate rules for the final rule base, and fix their consequents.

[4] defines a fuzzy subspace of the input domain $S_s = (A_1^s, \ldots, A_i^s, \ldots, A_n^s)$, with $A_i^s \in \mathcal{A}_i$, in such a way that $e_l \in S_s$ if $\mu_{A_1^s}(x_1^l) \cdot \ldots \cdot \mu_{A_n^s}(x_n^l) \neq 0$. Therefore, all examples in S_s match the rule:

$$R_s : \text{If } X_1 \text{ is } A_1^s \& \ldots \& X_n \text{ is } A_n^s \text{ then } Y = \ldots$$

In general, the methods proposed for deriving rules first determine a set of non-empty input subspaces, S^+. There are two different ways to do that.

In the *Example*-based approach, each example e_l enables at most one rule. Therefore,

$$S_s \in S^+ \text{ if } \exists e_l \in \mathcal{E} \,|\, \forall i \in \{1, \ldots, n\}, \forall A_i' \in \mathcal{A}_i, \mu_{A_i^s}(x_i^l) \geq \mu_{A_i'}(x_i^l)$$

i.e., a subspace $S_s \in S^+$ if there is an example e_l such that the rule with the highest matching degree with e_l is R_s. This is the case of Wang and Mendel [18].

The second approach generates S^+ from what is called a *FuzzyGrid*:

$$S_s \in S^+ \text{ if } \exists e_l \in \mathcal{E} \,|\, \mu_{A_1^s}(x_1^l) \cdot \ldots \cdot \mu_{A_n^s}(x_n^l) \neq 0,$$

Here, an input subspace S_s is considered to generate candidate rules if there is an example e_l in it.Therefore, each example e_l can generate more than one rule.

The search methods determine the consequent for each rule R_s such that $S_s \in S^+$. Consequents can be chosen individually [18, 11]. However, due to the nature of the fuzzy systems, algorithms such as COR [4, 7] or Least Squares [15], which use cooperation and evaluate the whole set of rules, perform better. In such cases, the problem converts into an optimitation one, and the algorithms search for the best combination of consequents for all candidate rules. There are several algorithms to do that, either based on populations[4] or based on local search [7]. In most of them, it is considered the consequent $\emptyset$ to represent the non inclusion of the rule.

One of the main problems with all these methods is scalability. Thus, given an input data set defined over n variables, a simple example could generate up to 2^n candidate rules if following the *FuzzyGrid* approach. On the other hand, when using the approach based on *Examples*, the rules generated become very specific due to the sparse distribution of data over the input domain, and the systems do not match many input examples when being tested.

One solution to deal with the problem, which has already been tested, consists of using rules with a variable number of antecedents [8]. Since smaller rules cover more examples, it is easier to cover the whole input domain while using few of them. This way, input subspaces are defined as $S_s = (A^s_{i_{1_s}}, \ldots, A^s_{i_{m_s}})$, with $1 \leq i_{1_s}, i_{m_s} \leq n$, and $i_{1_s} < i_{m_s}$. The rules generated from a subspace are then:

$$R_s : \text{If } X_{i_{1_s}} \text{ is } A^s_{i_{1_s}} \& \ldots \& X_{i_{m_s}} \text{ is } A^s_{i_{m_s}} \text{ then } Y = \ldots .$$

where m_s is the number of antecedents of the rule R_s.

However, when considering rules of variable size, one example generates a minimum of 2^n rules, and the total number of candidate rules can rise up to $\prod_{i=1}^{n} (|\mathcal{A}_i| + 1)$. The effect of that is so dramatical, that it is even impossible to evaluate each rule individually as a way for pre-selection, and it becomes necessary to provide some mechanism to limit the number of candidate rules.

4 Using Apriori to Find Candidate Rules

One of the goals when learning LFRBSs is producing a set of rules which is both accurate and small. Therefore, rules should not be too specific. Because of that, rules which do not cover a minimum number of examples, i.e., which do not reach a minimum support, must be discarded in advance. The algorithms presented in [10] for fuzzy association rule discovering, or in [14, 3] for fuzzy classification, can be adapted to find the candidate fuzzy rules in modeling tasks. Let $\mathcal{E}^s \subseteq \mathcal{E}$ the set of examples in S_s. The algorithm determines S^+ by finding those subspaces S_s such that $|\mathcal{E}^s| > minSupport$, i.e., when $support(R_s) > minSupport$.

Given two rules R_r and R_s, such that $m_s < m_r$ (R_s has fewer antecedents than R_r), it can be stated that, if $S_r \subseteq S_s$ then $support(R_s) \geq support(R_r)$ (since $\mathcal{E}^r \subseteq \mathcal{E}^s$). As the anti-monotone property of support holds, S^+ can be

determined by means of the Apriori algorithm for finding frequent itemsets. In order to do that, the procedure, shown in Algorithm 1, starts by including in S^+ those subspaces S_s defined by one label ($M = 1$) with support greater than *minSupport* (set $\mathcal{F}_M$). Then, it iteratively determines S^+ by following the apriori principle, i.e, if $S_r \subseteq S_s$, then S_r can be frequent if and only if S_s is frequent.

In practice, the *Combine* procedure (line 8 of Algorithm 1), takes each subspace $S_s = (A^s_{i_{1_s}}, \ldots, A^s_{i_{m_s}}) \in \mathcal{F}_{M-1}$ and generates $\mathcal{L}_M$ with non empty subspaces $S_r = (A^s_{i_{1_s}}, \ldots, A^s_{i_{m_s}}, A^r_i)$ by adding a label A^r_i to the definition of S_s, i.e., by adding an antecedent to R_s. This is done in such a way that every subset of labels from those defining S_r is also frequent. Then, those $S_r \in \mathcal{L}_M$ with a minimum support, are included in $\mathcal{F}_M$.

Algorithm 1 Finding frequent itemsets

```
1: procedure APRIORI(minSup, E)
2:     M ← 1
3:     L_M = {S^s | m_s = 1}
4:     F_M = {S^s ∈ L_M |  |E^s| > minSup}
5:     S^+ ← F_M
6:     while |F_M| > 1 do
7:         M ← M + 1
8:         L_M ← COMBINE(F_{M-1})
9:         F_M = {S^r ∈ L_M |  |E^r| > minSup}
10:        S^+ ← S^+ ∪ F_M
11:    end while
12:    return S^+
13: end procedure
```

4.1 Modifying the estimation of the suport

As mentioned above, we calculated $support(R_s)$ as $|\mathcal{E}^s|$. However, this method increases the support in rules, which can sometimes match examples with a low covering degree. In [14], the support of R_s is obtained by adding the covering degree of each example in $\mathcal{E}^s$. However, preliminary experiments show that results improve if considering only those examples whose covering degree (using the minimum as T-Norm) is greater than a particular threshold α when calculating the support of a rule.

Concretely, and due to the fact that we are using simmetrical partitions and rules of different sizes, we have used values $\alpha = 0.0, \alpha = 0.5$ and $\alpha = 0.75$, which still yields a large amount of candidate rules.

5 Filtering by Standard Deviation

Although the support threshold allows doing a pre-filtering of rules, the obtained sets are still so big, that some learning algorithms could not work even with small size problems. Therefore, it becomes necessary to eliminate some other rules.

Some works show that the individual evaluation of a rule can provide another efficient criteria for filtering [9]. However, they use the metrics proposed in [5] to evaluate Mamdani-type fuzzy rules, and cannot be applied for TSK-0 rules, where the consequents are real numbers. Nevertheless, there is a way to estimate the error produced by a rule R_s.

The individual error of the rule R_s is related to the standard deviation of the set $\mathcal{Y}^s = \{y^l \mid e_l \in \mathcal{E}^s\}$. Thus, if the standard deviation is 0, all $y^l \in \mathcal{Y}^s$ have the same value, and the error can be 0 when predicting such a value. On the other hand, the biggest possible error for R_s is given by the standard deviation of $\mathcal{Y}^s$. Because of that, we have fixed a threshold for the standard deviation as a percentage of the range of the output variable.

6 Experimental Study

In order to test the proposed methods, we have used four datasets obtained from KEEL repository[1]: *wankara* (9 attributes), *forestFires* (12 attributes), *mortgage* (15 attributes) and *baseball* (16 attributes). Although their number of attributes is not big, algorithms like COR [4, 7] perform poorly when dealing with them.

The fuzzy variables have been modeled through triangular symmetrical fuzzy partitions, considering 7 labels for each linguistic variable. We have fixed the minimum support as 5% of the instances of the dataset. This value allows discarding the most specific candidate rules, whereas it is not as high as to eliminate many potentially useful rules which do not cover enough examples. We have also tested the aforementioned values for the parameter α: 0, 0.5, and 0.75. Finally, we have used 20% and 30% of the range of the output variable as the threshold to filter by standard deviation. Results have been obtained by performing 30 independent executions over the dataset. On each run, the dataset was randomized using the same seed for all algorithms, and 20% of the examples were left for test.

Table 1 shows the number of candidate rules obtained with each configuration for the chosen datasets. As can be observed, using the standard estimation of support ($\alpha = 0$) results in large sets of candidate rules. Thus, we obtained more than 2.6 million rules in case of the problem baseball. However, 97.50% of them are discarded when $\alpha = 0.5$, and 99.83% when $\alpha = 0.75$.

The effect of filtering based on standard deviation depends on the algorithm (see Figure 1). Contrary to what it might be expected, the higher reduction is obtained when $\alpha \geq 0.5$. This happens because rules with higher support cover a greater number of examples, and therefore tend to have more standard deviation.

[1] http://sci2s.ugr.es/keel/

	wankara	forest	mortgage	baseball
minSupport 5%, $\alpha = 0$	26654.80	83310.47	2564670.17	2635952.53
minSupport 5%, $\alpha = 0$, $SD < 30\%$	20869.63	49.53	2287866.87	1543552.33
minSupport 5%, $\alpha = 0$, $SD < 20\%$	10686.20	10.53	1284806.53	1277736.50
minSupport 5%, $\alpha = 0.5$,	867.67	2762.23	59548.47	28855.20
minSupport 5%, $\alpha = 0.5$, $SD < 30\%$	337.27	4.67	53312.93	18507.87
minSupport 5%, $\alpha = 0.5$, $SD < 20\%$	71.70	4.67	16037.83	14081.57
minSupport 5%, $\alpha = 0.75$,	97.27	172.27	1069.10	1557.97
minSupport 5%, $\alpha = 0.75$, $SD < 30\%$	23.37	2.70	720.20	679.00
minSupport 5%, $\alpha = 0.75$, $SD < 20\%$	7.40	2.70	228.23	443.90

Table 1. Average number of candidate rules generated

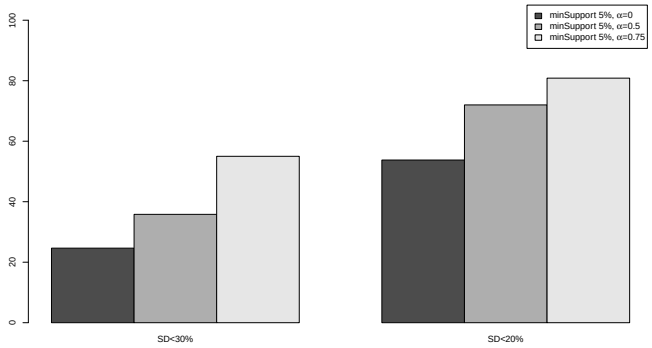


Fig. 1. Percentage of rules discarded by standard deviation

In order to evaluate the obtained sets, we have used those with less than 2000 candidate rules to learn a rule base (by fixing their consquents) with the Least Squares procedure described in [15] (see Table 2). For the remaining cases, we just show a line, as we consider that such systems are too big and, furthermore, Least Squares requires a high amount of runtime.

We can observe the results for the problems wankara and forestFire in order to gain some insight on the influence of α (for the rest of the problems we have only obtained less than 2000 candidate rules for $\alpha = 0.75$).

In the first case (wankara), the training error gets worse as α increases to 0.75. However, the test error is smaller when also filtering by standard deviation. Furthermore, in such a case the number of rules is reduced considerably. Thus, when $SD < 30\%$ the error is the smallest one, 3.38, and is obtained with only 23.37 rules on average. The trend is the same in the case of the problem forestFire. However, this case is very special, since we see that the number of rules is fairly small (2.70). Last, it seems that when $\alpha = 0.75$, the rules tend to have a smaller number of antecedents. This fact is also important, since interpretability is one of the goals when learning LFRBSs.

Problem: wankara

	Tra. err.	Test err.	#Rules	#Antec.	#Proc. Inst.
minSupport 5%, $\alpha = 0.0$	-	-	-	-	-
minSupport 5%, $SD < 30\%$	-	-	-	-	-
minSupport 5%, $SD < 20\%$	-	-	-	-	-
minSupport 5%, $\alpha = 0.5$	0.00	4.73	867.67	2.92	257.00
minSupport 5%, $\alpha = 0.5$, $SD < 30\%$	0.15	15.80	337.27	2.92	257.00
minSupport 5%, $\alpha = 0.5$, $SD < 20\%$	2.61	4.33	71.70	3.00	257.00
minSupport 5%, $\alpha = 0.75$	3.28	6.55	97.27	1.86	257.00
minSupport 5%, $\alpha = 0.75$, $SD < 30\%$	2.74	3.38	23.37	1.78	257.00
minSupport 5%, $\alpha = 0.75$, $SD < 20\%$	4.66	4.66	7.40	1.63	257.00

Problem: forestFire

	Tra. err.	Test err.	#Rules	#Antec.	#Proc. Inst.
minSupport 5%, $\alpha = 0.0$	-	-	-	-	-
minSupport 5%, $SD < 30\%$	60.35	58.73	49.53	4.88	413.00
minSupport 5%, $SD < 20\%$	60.59	58.22	10.53	3.81	413.00
minSupport 5%, $\alpha = 0.5$	-	-	-	-	-
minSupport 5%, $\alpha = 0.5$, $SD < 30\%$	60.62	58.09	4.67	2.01	413.00
minSupport 5%, $\alpha = 0.5$, $SD < 20\%$	60.62	58.09	4.67	2.01	413.00
minSupport 5%, $\alpha = 0.75$	56.55	65.29	172.27	2.11	413.00
minSupport 5%, $\alpha = 0.75$, $SD < 30\%$	60.74	57.86	2.70	1.14	413.00
minSupport 5%, $\alpha = 0.75$, $SD < 20\%$	60.74	57.86	2.70	1.14	413.00

Problem: mortgage

	Tra. err.	Test err.	#Rules	#Antec.	#Proc. Inst.
minSupport 5%, $\alpha = 0.0$	-	-	-	-	-
minSupport 5%, $SD < 30\%$	-	-	-	-	-
minSupport 5%, $SD < 20\%$	-	-	-	-	-
minSupport 5%, $\alpha = 0.5$	-	-	-	-	-
minSupport 5%, $\alpha = 0.5$, $SD < 30\%$	-	-	-	-	-
minSupport 5%, $\alpha = 0.5$, $SD < 20\%$	-	-	-	-	-
minSupport 5%, $\alpha = 0.75$	0.22	0.36	1069.10	3.50	839.00
minSupport 5%, $\alpha = 0.75$, $SD < 30\%$	0.89	1.00	720.20	3.91	839.00
minSupport 5%, $\alpha = 0.75$, $SD < 20\%$	1.21	1.21	228.23	4.31	839.00

Problem: baseball

	Tra. err.	Test err.	#Rules	#Antec.	#Proc. Inst.
minSupport 5%, $\alpha = 0.0$	-	-	-	-	-
minSupport 5%, $SD < 30\%$	-	-	-	-	-
minSupport 5%, $SD < 20\%$	-	-	-	-	-
minSupport 5%, $\alpha = 0.5$	-	-	-	-	-
minSupport 5%, $\alpha = 0.5$, $SD < 30\%$	-	-	-	-	-
minSupport 5%, $\alpha = 0.5$, $SD < 20\%$	-	-	-	-	-
minSupport 5%, $\alpha = 0.75$	5.07	1863.31	1557.97	3.53	269.00
minSupport 5%, $\alpha = 0.75$, $SD < 30\%$	520.55	1480.21	679.00	4.15	269.00
minSupport 5%, $\alpha = 0.75$, $SD < 20\%$	669.20	1088.01	443.90	4.41	269.00

Table 2. Average results for the systems learnt by Least Squares for those cases where the number of candidate rules is smaller than 2000

As for the use of standard deviation for filtering, the training error seems to increase as the threshold decreases. However, with the exception of the problem mortgage, the test error improves if compared with the algorithms which do not filter by standard deviation. The optimal threshold, according to the experiments, depends on the problem. However, it seems that using 20% of the range as the maximum standard deviation can make the error increase. Since some of the learning algoritms (not Least Squares) also delete rules, it could be suitable to fix the threshold to 30% and let the algorithms to eliminate rules.

7 Conclusions

This work uses a version of the Apriori algorithm to find candidate rules with high support in the case of modelling tasks. As it is shown by the experimental study, the number of rules obtained when fixing the minimum support as a 5% of the examples in the training dataset, is still very big to use the traditional rule derivation algorithms. Because of that, we have used a modified and more restricted notion of support. Furthermore, the standard deviation of the consequents in the examples covered by a rule can be used in the case of TSK-0 systems to perform a second stage of filtering. As it has been shown, in many cases this allows reducing substantially the amount of rules, whereas the results obtained with Least Squares after filtering show that most eliminated rules are not usefull for generalization.

As future work we plan to make a deeper study, learning the rule base by means of other algorithms, as those presented in [7]. In such a case, the algorithms may themselves discard rules during the search process. Therefore, the aim is to find a balance between the pre-selection made in this study and the subsequent search so as to improve the error while using the smallest possible number of rules.

Acknowledgements. This work has been partially funded by FEDER funds and the Spanish Government (MINECO) through project TIN2010-20900-C04-03. Javier Cózar is also funded by the MICINN grant FPU12/05102.

References

1. R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. *SIGMOD Rec.*, 22(2):207–216, June 1993.
2. R. Alcalá, M.J. Gacto, and F. Herrera. A fast and scalable multiobjective genetic fuzzy system for linguistic fuzzy modeling in high-dimensional regression problems. *IEEE T. Fuzzy Systems*, 19(4):666–681, 2011.
3. J. Alcala-Fdez, R. Alcala, and F. Herrera. A fuzzy association rule-based classification model for high-dimensional problems with genetic rule selection and lateral tuning. *Fuzzy Systems, IEEE Transactions on*, 19(5):857–872, 2011.

4. J. Casillas, O. Cordón, and F. Herrera. COR methodology: a simple way to obtain linguistic fuzzy models with good interpretability and accuracy. In J. Casillas, O. Cordón, F. Herrera, and L. Magdalena, editors, *Accuracy improvements in linguistic fuzzy modelling*, volume 129 of *Studies in Fuzziness and Soft Computing*, pages 27–45. Spinger Verlag, 2003.
5. O. Cordón and F. Herrera. A proposal for improving the accuracy of linguistic modeling. *IEEE Transactions on Fuzzy Systems*, 8(3):335–344, 2000.
6. O. Cordon, F. Herrera, and P. Villar. Generating the knowledge base of a fuzzy rule-based system by the genetic learning of the data base. *IEEE Transactions on Fuzzy Systems*, 9(4):667–674, 2001.
7. J. Cózar, L. de la Ossa, and J.M. Puerta. Learning cooperative tsk-0 fuzzy rules using fast local search algorithms. In *CAEPIA*, pages 353–362, 2011.
8. J. Cózar, L. delaOssa, and J.A. Gámez. Learning heterogeneus cooperative linguistic fuzzy rules using local search: Enhancing the cor search space. In *ISDA*, pages 475–480, 2011.
9. J. Cózar, L. delaOssa, and J.A. Gámez. Un algoritmo grasp para el aprendizaje de sistemas basados en reglas difusas lingsticas. In *Actas del VIII Congreso Espaol sobre Metaheursticas, Algoritmos Evolutivos y Bioinspirados (MAEB12)*, pages 611–618, 2012.
10. T. Hong, C. Kuo, and S. Chi. Trade-off between computation time and number of rules for fuzzy mining from quantitative data. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(5):587–604, 2001.
11. H. Ishibuchi, K. Nozaki, H. Tanaka, Y. Hosaka, and M. Matsudai. Empirical study on learning in fuzzy systems by rice taste analysis. *Fuzzy Sets Syst.*, 64(2):129–144, June 1994.
12. E.H. Mamdani. Applications of fuzzy algorithm for control a simple dynamic plant. In *Proceedings of the IEEE 121(12)*, pages 1585–1588, 1974.
13. E.H. Mamdani and S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, 7:1–13, 1975.
14. Y. Nojima, Y. Kaisho, and H. Ishibuchi. Accuracy improvement of genetic fuzzy rule selection with candidate rule addition and membership tuning. In *Fuzzy Systems (FUZZ), 2010 IEEE International Conference on*, pages 1–8, 2010.
15. K. Nozaki, H. Ishibuchi, and H. Tanaka. A simple but powerful heuristic method for generating fuzzy rules from numerical data. *Fuzzy Sets and Systems*, 86:251–270, 1997.
16. G. Piatetsky-Shapiro. Discovery, analysis, and presentation of strong rules. In *Knowledge Discovery in Databases*, pages 229–248. AAAI/MIT Press, 1991.
17. T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications for modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, 15(1):116–132, 1985.
18. L.X. Wang and J.M. Mendel. Generating fuzzy rules by learning from examples. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(6):1414–1427, 1992.
19. L.A. Zadeh. Outline of a new approach to the analysis of complex systems and decisionprocesses. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(1):28–44, 1973.
20. L.A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning. *Information Science*, 8:199–249, 1975.

Aplicando Métodos de Aprendizaje Sensible al Coste para Mejorar Problemas de *Big Data* Extremadamente Desbalanceados Usando *Random Forest*

Sara del Río, Victoria López, José Manuel Benítez, and Francisco Herrera

Departamento de Ciencias de la Computación e Inteligencia Artificial, CITIC-UGR (Centro de Investigación en Tecnologías de la Información y las Comunicaciones), Granada, España
{srio,vlopez,j.m.benitez,herrera}@decsai.ugr.es

Abstract. Debido a la explosión y proliferación de datos en los últimos años, surge la necesidad de aplicaciones que permitan procesar cantidades ingentes de información. Esto supone un desafío pues las técnicas de minería de datos no están bien preparadas para afrontar los requerimientos de tiempo y espacio. Por otra parte, en muchos dominios de aplicación los problemas de clasificación suelen presentar una distribución de datos compleja donde las clases aparecen desbalanceadas. En este trabajo proponemos el uso de técnicas de aprendizaje sensible al coste en un entorno *Hadoop* para el algoritmo *Random Forest* de la biblioteca *Mahout*. Se ha llevado a cabo un estudio experimental con datos extremadamente desbalanceados en el que los resultados obtenidos muestran una mejora de rendimiento tanto en precisión como en tiempos de respuesta.

Keywords: *Big Data*, *Mahout*, *Hadoop*, Datos no Balanceados, Aprendizaje Sensible al Coste, *Random Forest*

1 Introducción

En la actualidad, en multitud de sectores como el de las telecomunicaciones, el farmacéutico o el de la salud, se genera información constantemente y en cantidades astronómicas. La realización de análisis sobre grandes volúmenes de datos es de crucial importancia para la obtención y mejora del conocimiento. Debido a que las bases de datos han crecido a mayor velocidad que la capacidad de procesamiento y de análisis inteligente, los almacenes de datos y las tecnologías de minería de datos han quedado desbordados por no poder hacer frente a dichas cantidades masivas de datos. Estos datos se empiezan a conocer como *Big Data*.

Por otra parte, en la mayoría de aplicaciones los datos presentan una distribución de clases en los que una o más clases están representadas por un gran número de ejemplos, mientras que el resto se representan por unos pocos. Esta situación es conocida como clasificación sobre conjuntos de datos desbalanceados y su importancia reside en que las clases minoritarias son el foco de interés del problema [13, 20].

Las soluciones de *Big Data* representan un nuevo paradigma en la administración de grandes volúmenes de datos. Estas soluciones se han centrado en el procesamiento de los mismos de forma distribuida, escalable y confiable. Para lograr este objetivo, surgió un modelo de programación denominado *MapReduce* [8]. Dicho modelo de programación divide el conjunto de datos original en subconjuntos más fáciles de abordar para después combinar las soluciones parciales obtenidas. Sin embargo, esta distribución de los datos suele ocasionar un efecto negativo, especialmente en conjuntos de datos desbalanceados donde se agrava especialmente dicho problema [19]. Además, al realizar una partición adicional de los datos podemos inducir artificialmente un problema de cambio en la distribución de los datos [16].

Para tratar de solventar estos problemas, en este trabajo presentamos un nuevo método basado en el algoritmo *Random Forest* [5] para el manejo de datos extremadamente desbalanceados. Gracias a su naturaleza de *ensemble*, se facilita la creación de una versión paralela del algoritmo, donde el conjunto de datos original se distribuya entre los distintos nodos de un clúster [11]. Para su realización, proponemos el uso conjunto de *Hadoop* [22] y de técnicas sensibles al coste [7].

Con este propósito, se lleva a cabo un estudio experimental en el que nos centramos en conjuntos de datos extremadamente desbalanceados. Para evaluar el rendimiento de la propuesta haremos uso de la Media Geométrica [3].

Este trabajo se organiza de la siguiente forma. En primer lugar, en la sección 2, se presenta una breve introducción a *Big Data* y a los problemas de clasificación con conjuntos de datos desbalanceados. En la sección 3, se da una descripción de la propuesta. Los experimentos y resultados se presentan en la Sección 4. Por último, en la sección 5 se ofrecen algunas conclusiones.

2 Clasificación con *Big Data* y Conjuntos de Datos Desbalanceados

En esta sección proporcionamos una introducción a *Big Data* en problemas de clasificación (Sección 2.1) y, a continuación, una breve descripción de los problemas de clasificación con conjuntos de datos desbalanceados (Sección 2.2).

2.1 El Desafío de *Big Data* en Clasificación

El constante incremento en la velocidad a la que se generan los datos nos conduce a manejar cantidades masivas a través de los algoritmos tradicionales de aprendizaje automático. Esto es un problema pues dichos algoritmos no se encuentran adaptados para tratarlos. Por este motivo, y para un procesamiento eficiente de estas cantidades masivas de datos (*Big Data*), será necesario adaptar los algoritmos de aprendizaje automático adecuadamente.

Facebook y *Twitter* son ejemplos de aplicaciones reales en las que se generan tales cantidades de datos. Por ejemplo, en 2010, *Facebook* contaba con 21 Peta Bytes en su almacén de datos y su almacenamiento crecía a un ritmo aproximado de 12 Tera Bytes diariamente [21].

Entre las soluciones propuestas al problema, algunas de las tecnologías relacionadas se están convirtiendo en el estándar de facto de *Big Data*. En 2004, Google presentó *MapReduce* [8], un modelo de programación para el procesamiento de grandes conjuntos de datos. Este nuevo paradigma computacional consta de dos fases, conocidas como *Map* y *Reduce*. La fase *Map* consiste en distribuir y ejecutar en cada nodo del clúster pequeños programas conocidos como *mappers*. Una de las implementaciones más populares en estos momentos de *MapReduce* es *Hadoop* [1, 22], un *framework* de libre distribución escrito en Java que facilita la escritura de aplicaciones distribuidas. *Hadoop* usa para el almacenamiento el sistema de ficheros HDFS (*Hadoop Distributed File System*), que crea múltiples réplicas de los datos y los distribuye en los nodos de un clúster.

Mahout [2, 18] es una biblioteca de algoritmos de aprendizaje automático y de minería de datos que aprovecha *Hadoop* para proporcionar el almacenamiento de los datos y la implementación del paradigma *MapReduce*. La biblioteca *Mahout* incluye algoritmos de recomendación, agrupamiento y clasificación.

2.2 Clasificación con Conjuntos de Datos Desbalanceados

El problema de la clasificación con conjuntos de datos desbalanceados se produce cuando existe una notable diferencia entre el número de ejemplos que pertenecen a las distintas clases [20]. Este problema ha adquirido mucha importancia en los últimos años debido a su presencia en numerosas aplicaciones reales tales como diagnosis médica, finanzas o bioinformática. En estos problemas, el interés de los expertos se centra en la identificación de los casos minoritarios pues suelen ser los más importantes desde el punto de vista del aprendizaje, y conllevan altos costes cuando su identificación no se realiza adecuadamente [9].

Tradicionalmente, la relación de desequilibrio [17], que se define como la relación entre el número de instancias de la clase mayoritaria y la clase minoritaria, determina el nivel de dificultad asociado a un conjunto de datos determinado. Sin embargo, existen factores adicionales que influyen negativamente en la clasificación con conjuntos de datos desbalanceados como por ejemplo: el solapamiento de clases, la existencia de una muestra de pequeño tamaño, y las diferencias en la distribución de los datos entre los conjuntos de entrenamiento y test [16], principalmente.

Numerosas técnicas han sido usadas para abordar el problema de clasificación de datos desbalanceados [13, 20]. Podemos hacer una división en enfoques: a nivel de datos [4, 6] en los que las instancias del conjunto de entrenamiento se equilibran para obtener una distribución de clases equitativa, y a nivel de algorítmico [15], que suponen una modificación de los algoritmos de manera que intenten beneficiar la clasificación de la clase minoritaria. Los enfoques sensibles al coste combinan los enfoques anteriores para intentar minimizar el coste de cometer un error clasificando una instancia de la clase minoritaria como mayoritaria [9]. Las técnicas basadas en *ensembles* han demostrado un buen comportamiento al adaptarse a problemas de clasificación desbalanceada [10].

El acierto en clasificación no suele ser una métrica de rendimiento apropiada cuando la distribución de las clases es muy diferente, pues no considera el coste

de las clasificaciones incorrectas y no identifica la incorrecta clasificación de las instancias de las clases menos representadas. En este trabajo vamos a utilizar la Media Geométrica (MG) [3], que se define como $MG = \sqrt{VP_{tasa} \cdot VN_{tasa}}$, donde VP_{tasa} es el porcentaje de ejemplos de la clase positiva bien clasificados y VN_{tasa} es el porcentaje de ejemplos de la clase negativa bien clasificados. Esta medida trata de maximizar el acierto de cada una de las dos clases equiparando estos dos objetivos en una única medida.

3 *Random Forest* para Clasificación con Problemas de *Big Data* Extremadamente Desbalanceados

En esta sección describiremos nuestra propuesta para hacer frente a los problemas con conjuntos de datos desbalanceados mediante el uso del algoritmo *Random Forest* (RF). Para ello, primero presentamos diversas aproximaciones de RF que serán la base de nuestra aproximación final.

De esta manera, en la Sección 3.1, se presenta la versión de RF para *Big Data*. A lo largo de la Sección 3.2, se describe la versión del algortimo RF sensible al coste. Por último, en la Sección 3.3, se describe la propuesta final en base a las versiones anteriormente presentadas.

Random Forest [5], es uno de los *ensembles* de árboles de decisión más conocidos y utilizados en clasificación. Este algoritmo combina clasificadores basados en árboles de decisión aleatorios, de forma que la clase a la que pertenece una nueva instancia vendrá determinada por el voto mayoritario de cada uno de estos árboles.

3.1 *Random Forest* para Clasificación con *Big Data*

La implementación *Partial* de *Mahout* [11] es un algoritmo que construye múltiples árboles para diferentes porciones de los datos. Se trata de una implementación *MapReduce* donde cada *mapper* es responsable de construir un subconjunto del *ensemble* de árboles haciendo uso de los datos disponibles en su partición. Este algoritmo tiene dos fases:

- **Fase de construcción:** En esta fase se construye el *ensemble* de árboles. Para ello, se ejecuta en el clúster un *Job MapReduce* (Figura 1).
- **Fase de clasificación:** Se ejecuta un nuevo *Job MapReduce* para la estimación de las clases asociadas al conjunto de datos (Figura 2).

Cada una de las fases anteriores consta de tres etapas: inicial, *map* y final. La etapa inicial lleva a cabo una segmentación del conjunto de datos original en bloques HDFS independientes, que son distribuidos por los nodos de un clúster. En la etapa *map*, cada *mapper* procesa una porción de información y genera un fichero que contiene, en la primera fase, la información relativa a cada uno de los árboles y, en la segunda fase, un fichero con las predicciones. En la etapa final se combinan los ficheros de salida de cada uno de los *mappers* para construir, en la primera fase, el *ensemble* de árboles y, en la segunda fase, el fichero con todas las predicciones.

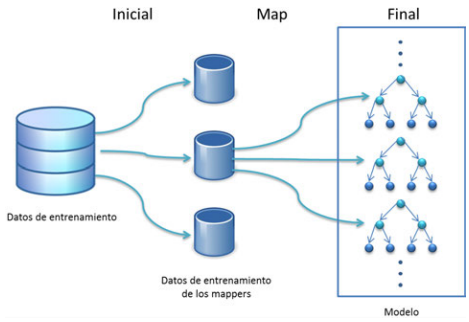


Fig. 1. Fase de construcción de un RF con la versión Partial de Mahout

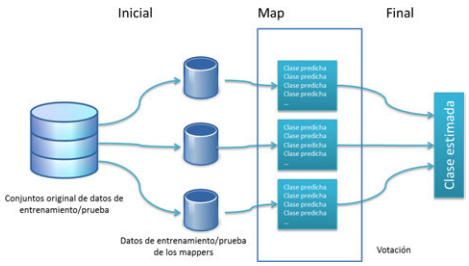


Fig. 2. Fase de clasificación de un RF con la versión Partial de Mahout

3.2 *Random Forest* para Clasificación con Conjuntos de Datos Desbalanceados

Weighted Random Forest [7] es una versión sensible al coste del algoritmo RF para tratar con conjuntos de datos desbalanceados. Esta aproximación modifica el algoritmo RF original en la etapa de construcción del árbol y en la etapa de clasificación. Durante la etapa de construcción del árbol, los costes son incorporados en todos los cálculos llevados a cabo internamente. Por ejemplo, el criterio utilizado para escoger una nueva división del árbol tiene en cuenta los costes asociados a cada clase en una instancia, en lugar de contabilizar las instancias en la misma proporción. Además, los pesos también se tienen en cuenta durante el cálculo de la clase asociada a la hoja del árbol. En la etapa de clasificación, modificamos el esquema de votación por mayoría a uno de votación ponderado. En este caso, calculamos el peso para cada hoja en cada árbol como una proporción de los pesos de las clases asociadas a cada una de las instancias clasificadas en la hoja en relación con el número de instancias clasificadas en dicha hoja.

Un problema de las técnicas sensibles al coste consiste en la propia estimación del coste, que no suelen proporcionar habitualmente los expertos. Para estimar el coste, se utiliza la estimación *out-of-bag* proporcionada por el algoritmo RF original [7].

3.3 RF-BigDataCS: *Random Forest* para Clasificación con Problemas de *Big Data* Extremadamente Desbalanceados

Inspirados en la implementación *Partial* del algoritmo RF de *Mahout*, hemos desarrollado un nuevo algoritmo que puede ser utilizado para problemas de clasificación de *Big Data* extremadamente desbalanceados: RF-BigDataCS. El hecho de que la implementación *Partial* del algoritmo RF de *Mahout* realice una segmentación del conjunto de datos original, dificulta aún más la clasificación con datos desbalanceados al reducir el tamaño de la muestra [19]. Por este motivo, debemos hacer uso de técnicas sensibles al coste.

Para adaptar el aprendizaje sensible al coste al entorno de *Mahout* es necesario incluir las modificaciones de la versión sensible al coste en serie en la implementación de la versión Partial de Mahout. En primer lugar, necesitamos estimar los pesos de cada una de las clases para que las clases minoritarias tengan un mayor peso. El peso de cada clase se calcula de la forma:

$$pesoClase_i = \frac{n_{claseMayoritaria}}{n_i}, \tag{1}$$

donde la clase mayoritaria es la clase con el mayor número de instancias y n_i es el número de instancias de la clase i. Esta estimación no puede llevarse a cabo teniendo en cuenta la estimación *out-of-bag* pues no es capaz de proporcionar pesos elevados para datos extremadamente desbalanceados.

Adicionalmente, se ha modificado el criterio utilizado para escoger una división al construir el árbol: en lugar de contabilizar las instancias de todas las clases en la misma proporción, consideramos el peso de cada clase asociado a la instancia. Además, los pesos también se tienen en cuenta para calcular cuál es la clase asociada a la hoja del árbol.

Una vez que cada *mapper* ha construido un subconjunto del *ensemble* de árboles, modificamos el método de razonamiento para clasificar nuevos ejemplos:

- Finalizada la etapa de construcción del modelo, el algoritmo calcula los pesos de las hojas de cada árbol. Para cada instancia del conjunto de datos, el algoritmo acumula en cada hoja el número de instancias clasificadas y el peso de la clase asociado a cada una de ellas. El peso de cada hoja es el peso acumulado dividido por el número de instancias clasificadas.
- Cuando se lleva a cabo la etapa de clasificación, para cada árbol que participa en la predicción de un ejemplo, el algoritmo acumula en la clase predicha por el mismo el peso asociado a la hoja. Para cada instancia y para todas las clases, se divide el peso acumulado anteriormente por el número de árboles que participan en la clasificación. Finalmente, se selecciona como clase asignada la que obtenga un mayor valor tras este proceso.

4 Estudio Experimental

En esta sección se describe el estudio experimental que se ha llevado a cabo para comprobar el rendimiento de nuestra propuesta RF-BigDataCS. En primer

Tabla 1. Resumen de conjuntos de datos no balanceados

Conjuntos de datos	#Ejemplos	#Atributos	Clase(may; min)	%Clase(may; min)	IR
kddcup_10_DOS_versus_normal	488736	41	(DOS; normal)	(80,10; 19,90)	4,02
kddcup_10_DOS_versus_PRB	395565	41	(DOS; PRB)	(98,96; 1,04)	95,31
kddcup_10_DOS_versus_R2L	392577	41	(DOS; R2L)	(99,71; 0,29)	349,83
kddcup_10_DOS_versus_U2R	391517	41	(DOS; U2R)	(99,98; 0,02)	6634,88
kddcup_10_normal_versus_PRB	101385	41	(normal; PRB)	(95,95; 4,05)	23,69
kddcup_10_normal_versus_R2L	98397	41	(normal; R2L)	(98,86; 1,14)	86,93
kddcup_10_normal_versus_U2R	97337	41	(normal; U2R)	(99,94; 0,06)	1648,78
kddcup_50_DOS_versus_normal	2428075	41	(DOS; normal)	(79,97; 20,03)	3,99
kddcup_50_DOS_versus_PRB	1962236	41	(DOS; PRB)	(98,95; 1,05)	94,48
kddcup_50_DOS_versus_R2L	1942248	41	(DOS; R2L)	(99,97; 0,03)	3448,82
kddcup_50_DOS_versus_U2R	1941711	41	(DOS; U2R)	(99,99; 0,01)	74680,19
kddcup_50_normal_versus_PRB	506941	41	(normal; PRB)	(95,95; 4,05)	23,67
kddcup_50_normal_versus_R2L	486953	41	(normal; R2L)	(99,88; 0,12)	863,93
kddcup_50_normal_versus_U2R	486416	41	(normal; U2R)	(99,99; 0,01)	18707,31
kddcup_full_DOS_versus_normal	4856151	41	(DOS; normal)	(79,97; 20,03)	3,99
kddcup_full_DOS_versus_PRB	3924472	41	(DOS; PRB)	(98,95; 1,05)	94,48
kddcup_full_DOS_versus_R2L	3884496	41	(DOS; R2L)	(99,97; 0,03)	3448,82
kddcup_full_DOS_versus_U2R	3883422	41	(DOS; U2R)	(99,99; 0,01)	74680,19
kddcup_full_normal_versus_PRB	1013883	41	(normal; PRB)	(95,95; 4,05)	23,67
kddcup_full_normal_versus_R2L	973907	41	(normal; R2L)	(99,88; 0,12)	863,93
kddcup_full_normal_versus_U2R	972833	41	(normal; U2R)	(99,99; 0,01)	18707,33

lugar, en la Sección 4.1 se presentan los problemas de clasificación utilizados en los experimentos. A continuación, la Sección 4.2 muestra los algoritmos y parámetros utilizados. Después, en la Sección 4.3 se presentan los resultados según la MG para cada una de las aproximaciones utilizadas. Finalmente, la Sección 4.4 muestra un análisis para evaluar la ganacia computacional (*speed-up*) por la utilización de técnicas para *Big Data*.

4.1 Conjuntos de Datos

Con el fin de analizar la calidad de nuestra propuesta se han seleccionado varios problemas de *Big Data* extremadamente desbalanceados derivados del conjunto de datos KDD Cup 1999 [14]. Puesto que este conjunto contiene múltiples clases, éstas se han agrupado para generar problemas binarios desbalanceados de gran tamaño. Concretamente se han creado nuevos conjuntos a partir de las clases *normal* y *DOS* como mayoritarias. Además, para llevar a cabo una prueba de escalabilidad se han generado versiones del 10% y del 50% del conjunto de datos original. Los datos utilizados se resumen en la tabla 1. Para el desarrollo de los experimentos se ha considerado un esquema de validación cruzada en 10 particiones.

4.2 Algoritmos y Parámetros

En este estudio, se ha comparado el rendimiento de nuestra propuesta, RF-BigDataCS, con respecto a las versiones básicas del algoritmo:

- ***Random Forest*** (RF) [5]: *Random Forest* original, *ensemble* de árboles de decisión disponible en la herramienta de minería de datos Weka [12].
- ***Random Forest* Sensible al Coste** (RF-CS) [7]: Versión adaptada de *Random Forest* que introduce aprendizaje sensible al coste.
- ***Random Forest* para *Big Data*** (RF-BigData) [11]: Versión Partial de *Random Forest* disponible en *Mahout* [18].

En la tabla 2 mostramos la configuración de parámetros para estos algoritmos, donde el parámetro *maxProfundidad* indica la profundidad de cada uno

de los árboles generados, *numCaracteristicas* es el número de atributos para construir los árboles aleatorios y *numArboles* representa el número de árboles que componen el *ensemble* de árboles. Además, podemos encontrar los atributos *estimacionCostes* y *numParticiones*. El primero ilustra cómo son calculados los costes de cada clase mientras que el segundo indica en cuántas partes se reparte el conjunto original para el procesamiento de los *mapper*.

Tabla 2. Parámetros para los algoritmos estudiados en la experimentación

Algoritmos	Parámetros
RF	maxProfundidad = ilimitada, numCaracteristicas = $log_2(N_{vars}) + 1$, numArboles = 100
RF-CS	maxProfundidad = ilimitada, numCaracteristicas = $log_2(N_{vars}) + 1$, numArboles = 100 estimacionCostes = basadaEnPesos
RF-BigData	maxProfundidad = ilimitada, numCaracteristicas = $log_2(N_{vars}) + 1$, numArboles = 20/10/5 numParticiones = 5/10/20 (respectivamente)
RF-BigDataCS	maxProfundidad = ilimitada, numCaracteristicas = $log_2(N_{vars}) + 1$, numArboles = 20/10/5 numParticiones = 5/10/20 (respectivamente), estimacionCostes = basadaEnPesos

4.3 Análisis de Rendimiento

Puesto que estamos interesados en problemas extremadamente no balaneados se han seleccionado las versiones *full* con mayor IR para analizar el rendimiento. En la tabla 3 se muestra la media de los resultados obtenidos para entrenamiento y test de todos los métodos estudiados en este trabajo. El símbolo NC (no computable) significa que la versión no fue capaz de completar el experimento. Esto es debido a que la implementación no está preparada para el uso de conjuntos de datos de este tamaño.

Podemos observar que los resultados de las versiones secuenciales, RF original y su modificación RF-CS, son ligeramente mejores con respecto a la adaptación a *Big Data*. Esta situación es común cuando se compara un enfoque secuencial con su paralelización pues parte de la precisión del modelo es sacrificada para la ganancia de velocidad. Además, se puede observar que RF-CS obtiene generalmente mejores resultados que RF original y que este comportamiento se extrapola a las versiones de RF-BigData.

También podemos comprobar el impacto que producen las distintas particiones sobre las versiones de RF-BigData pues cuantas más particiones se usan mayor es la pérdida de rendimiento del algoritmo. Nuestra propuesta, RF-BigDataCS, también se ve afectada por este problema, sin embargo, su variación no es tan dramática como en el caso de RF-BigData ya que mantiene un buen rendimiento tanto en precisión como en tiempos de respuesta, y por tanto, permite resolver el problema de clasificar grandes conjuntos de datos desbalanceados.

4.4 Estudio del *Speed-up*

Una de las principales cuestiones del uso de implementaciones distribuidas y paralelas es la obtención de mejores resultados en tiempo de ejecución. Por este motivo, es interesante evaluar el rendimiento de las versiones secuenciales frente a su adaptación distribuida. Para ello, haremos uso de la medida *Speed-up* o ganancia de velocidad, definida como $S_{up} = \frac{T_{sec}}{T_{par}}$, donde T_{sec} y T_{par} son los tiempos de ejecución de las versiones secuencial y paralela, respectivamente.

Tabla 3. Media de los resultados para las versiones de RF para conjuntos de datos extremadamente desbalanceados haciendo uso de la medida MG

Conjuntos de datos	kddcup_full_DOS_versus_U2R		kddcup_full_normal_versus_R2L		kddcup_full_normal_versus_U2R	
	MG_{entr}	MG_{tst}	MG_{entr}	MG_{tst}	MG_{entr}	MG_{tst}
Versiones secuenciales						
RF	NC	NC	1,0000	0,9832	0,9999	0,6836
RF-CS	NC	NC	0,9998	0,9976	0,9999	0,9813
Versiones para *Big Data*						
RF-BigData – 5 partes	0,7610	0,6731	0,9482	0,9376	0,3565	0,3333
RF-BigData-CS – 5 partes	0,8278	0,9608	0,9812	0,9835	0,9433	0,9960
RF-BigData - 10 partes	0,7045	0,6756	0,9274	0,9261	0,0000	0,0000
RF-BigData-CS – 10 partes	0,8032	0,9822	0,9793	0,9894	0,9381	0,9691
RF-BigData - 20 partes	0,0267	0,0000	0,8841	0,8767	0,0000	0,0000
RF-BigData-CS – 20 partes	0,9186	0,8853	0,9719	0,9657	0,9373	0,9593

La tabla 4 muestra los valores de *Speed-up* para cada uno de los conjuntos de datos considerados en este estudio. En ella podemos observar que las versiones de RF-BigData obtienen tiempos de respuesta mucho mejores. Además, podemos comprobar que el número de particiones influye en la ganancia en velocidad, sin embargo, no se mantiene una relación lineal, así como que la ganancia no es homogénea entre los conjuntos de datos considerados en este estudio.

Tabla 4. *Speed-up* para RF-BigData y RF-BigData-CS con 5, 10 y 20 particiones

Conjuntos de datos	RF-BigData - 5 partes			RF-BigData-CS - 5 partes			RF-BigData - 10 partes			RF-BigData-CS - 10 partes			RF-BigData - 20 partes			RF-BigData-CS - 20 partes		
	10%	50%	full	10%	50%	full	10%	50%	full	10%	50%	full	10%	50%	full	10%	50%	full
DOS_versus_normal	66,78	76,06	NC	39,88	50,81	NC	75,53	159,53	NC	40,83	76,82	NC	64,74	222,33	NC	29,07	66,04	NC
DOS_versus_PRB	48,74	57,18	NC	36,30	53,04	NC	58,14	115,74	NC	34,50	71,30	NC	48,25	154,94	NC	24,85	61,25	NC
DOS_versus_R2L	51,20	97,48	NC	35,96	73,35	NC	57,11	153,41	NC	34,03	91,79	NC	45,92	178,81	NC	23,24	70,85	NC
DOS_versus_U2R	47,52	102,80	NC	39,26	109,11	NC	49,99	187,68	NC	38,90	113,67	NC	41,39	184,88	NC	26,67	79,44	NC
normal_versus_PRB	5,39	77,07	NC	5,17	56,87	NC	6,09	98,55	NC	5,42	77,13	NC	4,99	103,65	NC	3,90	80,43	NC
normal_versus_R2L	4,01	66,29	7,32	3,37	62,44	7,37	4,68	80,90	35,60	3,54	78,49	16,39	3,96	122,39	60,01	2,55	78,49	15,31
normal_versus_U2R	3,70	75,96	10,54	3,91	68,81	8,15	3,84	92,03	40,10	3,56	32,66	16,47	3,18	92,03	62,28	2,59	23,67	14,19

5 Comentarios Finales

En este trabajo se ha presentado un algoritmo que basado en el algoritmo *Random Forest*, permite abordar problemas extremadamente desbalanceados con *Big Data*. Inspirado en las técnicas utilizadas para hacer frente a estas características de manera independiente, se han combinado ambas aproximaciones para construir un nuevo modelo que las integre y sea capaz de resolver ambos problemas de forma simultánea.

En la actualidad, los problemas de *Big Data* están ganando reconocimiento debido a las grandes cantidades de datos que se generan hoy en día. Los métodos de minería de datos tradicionales no son capaces de hacer frente a los nuevos requerimientos impuestos por *Big Data*. Por este motivo, hacemos uso del entorno *Hadoop*, que facilita el desarrollo de soluciones escalables y distribuidas. La biblioteca *Mahout*, construida sobre *Hadoop*, proporciona algoritmos de aprendizaje automático capaces de procesar grandes conjuntos de datos. El aprendizaje sensible al coste permite la transformación de los métodos de aprendizaje estándar en métodos que son capaces de hacer frente a conjuntos de datos desbalanceados en problemas de clasificación.

El rendimiento de nuestro modelo RF-BigData-CS ha sido contrastado en un estudio experimental que incluye varios problemas de *Big Data* extremadamente

desbalanceados. Los resultados obtenidos muestran que nuestra propuesta obtiene mejores resultados en cuanto a precisión y tiempo de respuesta. Además, señalan la necesidad de abordar conjuntamente problemas de *Big Data* y no balanceados, pues las distintas técnicas que por sí solas no son capaces de resolver el problema, al combinarse producen una buena sinergia.

Acknowledgements. Este trabajo ha sido parcialmente financiado por el Ministerio de Ciencia e Innovación bajo el proyecto TIN2011-28488 y los Planes Andaluces de Investigación P11-TIC-7765 y P10-TIC-6858. V.López posee una beca FPU del Ministerio de Educación, Cultura y Deporte.

Referencias

[1] Apache Hadoop Project: Apache Hadoop (2013), `http://hadoop.apache.org/`, [Online; consultado en junio 2013]
[2] Apache Mahout Project: Apache Mahout (2013), `http://mahout.apache.org/`, [Online; consultado en junio 2013]
[3] Barandela, R., Sánchez, J.S., García, V., Rangel, E.: Strategies for learning in class imbalance problems. Pattern Recognition 36(3), 849–851 (2003)
[4] Batista, G.E.A.P.A., Prati, R.C., Monard, M.C.: A study of the behaviour of several methods for balancing machine learning training data. SIGKDD Explorations 6(1), 20–29 (2004)
[5] Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
[6] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: Synthetic minority over–sampling technique. Journal of Artificial Intelligent Research 16, 321–357 (2002)
[7] Chen, C., Liaw, A., Breiman, L.: Using random forest to learn imbalanced data. Tech. Rep. 666, Statistics Department, University of California Berkeley (2004)
[8] Dean, J., Ghemawat, S.: Mapreduce: Simplified data processing on large clusters. Communications of the ACM 51(1), 107–113 (2008)
[9] Elkan, C.: The foundations of cost–sensitive learning. In: Proceedings of the 17th IEEE International Joint Conference on Artificial Intelligence (IJCAI'01). pp. 973–978 (2001)
[10] Galar, M., Fernández, A., Barrenechea, E., Bustince, H., Herrera, F.: A review on ensembles for class imbalance problem: Bagging, boosting and hybrid based approaches. IEEE Transactions on Systems, Man, and Cybernetics–part C: Applications and Reviews 42(4), 463–484 (2012)
[11] Hakim, D.A.: PartialData MapReduce Random Forests (2013), `http://cwiki.apache.org/confluence/display/MAHOUT/Partial+Implementation`, [Online; consultado en junio 2013]
[12] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: An update. SIGKDD Explorations 11(1), 10–18 (2009)
[13] He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering 21(9), 1263–1284 (2009)
[14] International Knowledge Discovery Data Mining Tools Competition: KDD Cup 1999 Data (2013), `http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`, [Online; consultado en junio 2013]
[15] López, V., Fernández, A., del Jesus, M.J., Herrera, F.: A hierarchical genetic fuzzy system based on genetic programming for addressing classification with highly imbalanced and borderline data-sets. Knowledge-Based Systems 38, 85–104 (2013)
[16] Moreno-Torres, J.G., Raeder, T., Aláiz-Rodríguez, R., Chawla, N.V., Herrera, F.: A unifying view on dataset shift in classification. Pattern Recognition 45(1), 521–530 (2012)
[17] Orriols-Puig, A., Bernadó-Mansilla, E.: Evolutionary rule–based systems for imbalanced datasets. Soft Computing 13(3), 213–225 (2009)
[18] Owen, S., Anil, R., Dunning, T., Friedman, E.: Mahout in Action. Manning Publications Co. (2012)
[19] Raudys, S.J., Jain, A.K.: Small sample size effects in statistical pattern recognition: Recommendations for practitioners. IEEE Transactions on Pattern Analysis and Machine Intelligence 13(3), 252–264 (1991)
[20] Sun, Y., Wong, A.K.C., Kamel, M.S.: Classification of imbalanced data: A review. International Journal of Pattern Recognition and Artificial Intelligence 23(4), 687–719 (2009)
[21] Thusoo, A., Shao, Z., Anthony, S., Borthakur, D., Jain, N., Sen Sarma, J., Murthy, R., Liu, H.: Data warehousing and analytics infrastructure at facebook. In: Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD 2010). pp. 1013–1020 (2010)
[22] White, T.: Hadoop, The Definitive Guide. O'Reilly Media, Inc. (2012)

Policy Reuse in a General Learning Framework

Fernando Martínez-Plumed, Cèsar Ferri, José Hernández-Orallo, and María José Ramírez-Quintana

DSIC, Universitat Politècnica de València, Camí de Vera s/n, 46022 València, Spain.
{fmartinez,cferri,jorallo,mramirez}@dsic.upv.es

Abstract. Policy reuse is a kind of transfer learning to improve a reinforcement learning system by reusing part of the information of the state-value function from previous problems to new problems. In this paper we overhaul this aproach in the context of a general learning framework for structured prediction using user-defined operators and a functional programming language for the representation of rules and operators. The redefinition of what policy reuse is in this context is motivated by the representation of states and actions as feature vectors, and the use of a Q matrix which is actually a table, from which a supervised model is learnt. This makes it possible to have a more flexible mapping between old and new problems, since we work with an abstraction of rules and actions. We do some experiments with the system gErl on some structured prediction problems (list patterns).
Keywords: policy reuse, reinforcement learning, inductive programming, transfer learning, complex data, heuristics, Erlang.

1 Introduction

The reuse of knowledge which has been acquired in previous learning processes in order to improve or accelerate the learning of future tasks is an appealing idea. Several solutions have been proposed to reuse knowledge, in areas such as meta-learning, incremental learning, employment of background knowledge or transfer learning. As the task and the learning system become more elaborated, this knowledge reuse becomes more important. In this paper we deal with a general rule-based learning setting using a functional programming representation, where operators can be defined and customised for each kind of problem. While one particular problem may require generalisation operators, another problem may require operators which add recursive transformations to explore the structure of the data. An appropriate choice of operators can embed transformations on the data but can also determine the way in which rules are generated and modified, so leading to (apparently) different learning systems. Making the user or the problem adapt its own operators is significantly different to the use of feature transformations or specific background knowledge. In fact, it is also significantly more difficult, since operators can be very complex things and usually embed the essence of a machine learning system. As a result, having a system which can work with different kinds of operators at the same time is a challenging proposal beyond the frontiers of the state of the art in machine learning.

Given the versatility of this system, the heuristics for exploring the search space must be more flexible. A reinforcement learning (RL) system determines which rules and operators are used and how they are combined. In this work, we will focus on how RL policies are reused between totally different tasks. In order to do that, we use an appropriate feature space for describing the kinds of rules and operators that are giving good solutions (and high rewards), so this history is reused for other problems, even when the task and operators are different.

The paper is organised as follows. Section 2 makes a short account of the related and previous work. Section 3 introduces briefly the gErl system. Section 4 describes the RL-based heuristics used to guide the learning process and section 5 discusses how to transfer knowledge between tasks. Section 6 includes an empirical study which illustrates the way in which operators are defined and solutions are reached. Section 7 closes the paper.

2 Previous work

The goal of this paper is to describe the policy reuse that we will use in our system to take advantage of previous learning episodes (problems). Consequently, we will focus on previous work on policy reuse in the areas of reinforcement learning (RL) and transfer learning (TL), see [12] for a survey.

There are three main families of TL methods (in the RL area) according to the description of states and actions. Firstly, the source and target tasks use the same state variables and set of actions. Here, transfer learning is performed by initializing the Q-values of a new episode with previously learned Q-values [6]. This family includes those TL methods which use multiple source tasks by leveraging all experienced source tasks when learning a novel target task [9] or by choosing a subset of previously experienced tasks [3]

The second family of methods are those which are able to transfer between tasks with different state variables and actions, so that no explicit mapping between the tasks is needed. Instead the agent reasons over abstractions of the Markov Decision Process that are invariant when the actions or state variables change. Some methods use macro-actions or *options* [11] to learn new action policies in Semi-Markov Decision Processes. *Relational Reinforcement Learning* [2] is another particularly attractive formulation in this context of transfer learning.

The last family are more flexible than those previously discussed as they allow the state variables and available actions to differ between source and target tasks using inter-task mappings. Namely, explicit mappings are needed in order to transfer between tasks with different actions and state representations [7].

Although TL in RL has made significant progress in recent years, there is a poor understanding about the reuse of knowledge for solving future different problems. This makes sense in our system since we can have a pool of user-defined operators to solve a problem, while only a subset is needed to solve it. Furthermore, in order to go beyond the previous approaches, we use an abstract representation of states and actions which facilitates the transfer of knowledge. The ultimate goal is to use old policy information to speed up the learning process of another different problem (which a different subset of operators).

3 The gErl system

As we have mentioned in Section 1, in this paper we use the gErl system [8], a general learning system which can be configured with different (possibly user-defined) operators. The system can be described as a flexible architecture (shown in Figure 1) which works with populations of rules (expressed as unconditional / conditional equations) and programs in the functional language Erlang, which *evolve* as in an evolutionary programming setting or a learning classifier system [5]. Operators are applied to rules and generate new rules, which are combined with existing or new programs. With appropriate operators, using some optimality criteria (based on coverage and simplicity) and using a reinforcement learning-based heuristic (where the application of an operator over a rule is seen as a decision problem fed by the optimality criteria) many complex problems can be solved. In the rest of this section we will introduce some notation and concepts of the system that will be required to understand how our policy reuse techniques are devised and implemented. We refer the reader to [8] for details.

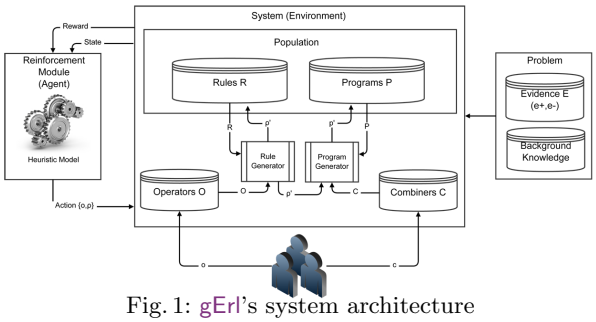


Fig. 1: gErl's system architecture

3.1 Data and model representation and learning problem statement

gErl can be considered and inductive (functional) programming system, as data and rules (and hence solutions) are represented in a functional programming language, Erlang. Let us see how these elements are represented and how the learning problem is stated.

Let Σ be a set of *function symbols* together with their arity and $\mathcal{X}$ a countably set of *variables*, then $\mathcal{T}(\Sigma, \mathcal{X})$ denotes the set of *terms* built from Σ and $\mathcal{X}$. The set of variables occurring in a term t is denoted *Var(t)*. A term t is a *ground term* if $Var(t) = \varnothing$. An equation is an expression of the form $l = r$ where l (the left hand side, *lhs*) and r (the right hand side, *rhs*) are terms. $\mathcal{R}$ denotes the space of all (conditional) functional rules ρ of the way l [when G] $\rightarrow T, r$ where l and r are the lhs and the rhs of ρ (respectively), $G = \{g_1, g_2, \ldots g_m \mid m \geq 0\})$ is a set of conditions or Boolean expressions called guards, and $T = b_1, \ldots, b_n$, the tail of ρ, is a sequence of equations. If $G = \varnothing$, then ρ is said to be an unconditional rule. Let $\mathcal{P} = 2^{\mathcal{R}}$ be the space of all possible functional programs formed by sets of rules $\rho \in \mathcal{R}$. Given a program $p \in \mathcal{P}$, we say that term t reduces to term s with respect to p, $t \rightarrow_p s$, if there exists a rule l [when G] $\rightarrow T, r \in p$ such that a subterm of t at occurrence u matches l with substitution θ, all conditions $g_i\theta$ hold, for each equation $b_{i_l} = b_{i_r} \in T$, $b_{i_l}\theta$ and $b_{i_r}\theta$ have the same normal form

(that is, $b_{i_l}\theta \rightarrow_p^* b$, and $b_{i_r}\theta \rightarrow_p^* b$ and b can not be further reduced) and s is obtained by replacing in t the subterm at occurrence u by $r\theta$.

An example e is a ground rule $l \rightarrow r$ (that is, without condition nor tail), being r in normal form. We say that e is covered by a program p (denoted by $p \models \{l \rightarrow r\}$) if l and r have the same normal form with respect to p. A program $p \in \mathcal{P}$ is a solution of a learning problem defined by a set of positive examples E^+, a (possibly empty) set of negative examples E^- and a background theory B if it covers all positive examples, $B \cup p \models E^+$ (posterior sufficiency or completeness), and does not cover any negative example, $B \cup p \not\models E^-$ (posterior satisfiability or consistency). Our system has the aim of obtaining complete solutions, but their consistency is not a mandatory property, so approximate solutions are allowed.

As we can see in Figure 1, our system works with two sets: a set of rules $R \subseteq \mathcal{R}$ and a set of programs $P \subseteq \mathcal{P}$, where each program $p \in P$ is composed by rules belonging to R. Initially, R is populated with E^+ and the set of programs P is populated with as many unitary programs as there are rules in R.

3.2 Operators and learning process

The definition of customised operators is one of the key concepts of our proposal (unlike others systems based on the use of different fixed operators [4]). The idea is to transform the set of rules R using a set of *operators* $O \subset \mathcal{O}$ (provided by the user or existing in the system).

An operator can be seen as a piece of code (as complex as the user may wants) which performs modifications over the *lhs* or *rhs* of a rule and which is written in the same functional language as the system (Erlang) to take advantage of its high-order and reflection capabilities. The main idea is that, when the user wants to deal with a new problem, he/she can define his/her own set of *operators*, especially suited for the data structures of the problem. This feature allows our system to adapt to the problem at hand.

As the process progresses, new rules and programs will be generated. First, the *Rule Generator* process (Figure 1) gets the operator o and the rule ρ returned as an action $a = \langle o, \rho \rangle$ by the *Reinforcement Learning Module* (policy). This process applies the operator over the rule obtaining a new rule ρ' (if the operator is not suitable for the rule selected, the process returns the same rule) which is added to R.

4 RL-based heuristics

In this section we describe the reinforcement learning approach followed by gErl in order to guide the learning process. Given that the users can define their own operators, the search heuristics have been conceived as decisions about the operator to be used at each particular state of the process. For this, a model-based reinforcement learning approach has been developed, where the application of an operator over a rule is seen as a decision problem, for which learning also takes place, guided by the optimality criteria which feed a rewarding module.

4.1 RL problem statement

To guide the learning process we need a picture of the system in each step of the process (before and after applying an action) in terms of the quality of the set

of rules and programs generated until now. Based on it, we model the decision process as a typical reinforcement learning task.

Formally, we define a state at each iteration t of the system as a tuple $\sigma_t = \langle R, P \rangle$ which represent the population of rules R and programs P in t. An action is a tuple $\langle o, \rho \rangle$ with $o \in \mathcal{O}$ and $\rho \in \mathcal{R}$ that represents the operator o to be applied to the rule ρ[1]. Our decision problem is a four-tuple $\langle \mathcal{S}, \mathcal{A}, \tau, \omega \rangle$ where: $\mathcal{S}$ is the state space; $\mathcal{A}$ is a finite actions space ($\mathcal{A} = \mathcal{O} \times \mathcal{R}$); $\tau : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ is a transition function between states and $\omega : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function. These components are defined below:

- **States.** As we want to find a good solution to the learning problem, we describe each state σ_t by a tuple of features $s_t = \langle \phi_1, \phi_2, \phi_3 \rangle$ from which to extract relevant information in t:
  1. *Global optimality* (ϕ_1): This feature shows the average optimality of all programs in P_t. In turn, the optimality of a program p is computed by weighting three simpler heuristics according to its importance:

$$Opt(p) = \beta_1 \cdot CRD(p) - \beta_2 \cdot OU(p) + \beta_3 \cdot RP(p) \tag{1}$$

     where CRD (coverage rate difference) is a normalised measure of coverage difference between positive and negative examples covered, OU (operator usage) is a normalised measure of how many operators have been used to derive the rule and RP is a measure of the expressiveness of the programs in terms of number of variables, functions and constants. As for the current implementation of gErl, the weights are $\beta_1 = 1$, $\beta_2 = 0.2$ and $\beta_3 = 0.1$. Finally, the *Global optimality* factor is then calculated as the average of the optimalities of all programs in the system:

$$OptGlobal(P_t) = \frac{1}{Card(P_t)} \sum_{p \in P_t} Opt(p) \tag{2}$$

  2. *Average Size of Rules* (ϕ_2): measures the average complexity of all rules in R, using the RP measure mentioned above.
  3. *Average Size of programs* (ϕ_3): measures the average cardinality of all the programs in P_t in terms of the number of rules.
- **Actions.** Each rule is described by a tuple of features $\rho = \langle \varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5, \varphi_6, \varphi_7, \varphi_8 \rangle$ from which we extract relevant information:
  1. *Size* (φ_1): expressiveness of the rule using RP.
  2. *Positive Coverage Rate* (φ_2).
  3. *Negative Coverage Rate* (φ_3).
  4. *NumVars* (φ_4): number of variables of ρ.
  5. *NumCons* (φ_5): number of constants (functors with arity 0) of ρ.
  6. *NumFuncs* (φ_6): number of functors with arity greater than 0 of ρ.
  7. *NumStructs* (φ_7): number of structures (lists, graphs, ...) of ρ.
  8. *isRec* (φ_8): indicates if the rule ρ is recursive or not.

[1] The probable infinite number of states and rules makes the abstraction of states and rules necessary.

We use a natural index as the only feature for operators. As an action consists of a choice of operator and rule, an action is finally a tuple of *nine* features.
- **Transitions.** Transitions are deterministic. A transition τ evolves the current sets of rules and programs by applying the operators selected (together with the rule) and the combiners.
- **Rewards.** The optimality criteria seen above is used to feed the rewards. In particular, we use the result returned by equation (1) as reward.

At each point in time, the reinforcement learning policy π can be in one of the states $s_t \in \mathcal{S}$ and may select an action $a_t = \pi(s_t) \in \mathcal{A}$ to execute. Executing such action a_t in s_t will change the state into $s_{t+1} = \tau(s_t, a_t)$, and the policy receives a reward $w_t = \omega(s_t, a_t)$. The policy does not know the effects of the actions, i.e. τ and ω are not known by the policy and need to be learned. This is the typical formulation of reinforcement learning [10] but using features to represent the states and the actions. With all these elements, the aim of our decision process is to find a policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$ that maximises:

$$V^{\pi}(s_t) = \sum_{i=0}^{\infty} \gamma^i w_{t+i} \tag{3}$$

for all s_t, where $\gamma \in [0, 1]$ is the *discount parameter* which determines the importance of the future rewards ($\gamma = 0$ only considers current rewards, while $\gamma = 1$ strives for a high long-term reward).

4.2 Modelling the state-value function: using a regression model

In our setting, for the RL module, we use a hybrid between model-free value-function methods (which update a state-value matrix) and model-based methods (which learn models for τ and ω) [10]. In particular, our approach uses the *state-value* function ($Q(s, a)$, which returns quality (q) *values*, $q \in \mathbb{R}$, as in Q-learning [13]) generalising it with a regression model. A model $M : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ calculates the optimality or q value for each state and action and tries to generalise Q. By using $a_t = \arg\max_{a \in \mathcal{A}} \{M(s_t, a_i)\}$ we get the best action for state s_t.

In order to train the model we use a 'matrix' Q (which is actually a table), whose rows are in $\mathcal{S} \times \mathcal{A} \times \mathbb{R}$ where $\mathcal{S}$ is a tuple of state features, $\mathcal{A}$ is the tuple of rule features and operator, and $\mathbb{R}$ is a real value for q. Table 1 shows an example of Q. Abusing notation, to work with Q as a function (like the original Q-matrix in many RL methods), we will denote by $Q[s, a]$ the value of q in the row of Q for that state s and action a. Each row in Q records the state and action for each time step in the system. So, Q grows in terms of the number of rows.

state (s)			**action** (a)								**q**
ϕ_1	ϕ_2	ϕ_3	o	φ_1	φ_2	φ_3	φ_4	φ_5	φ_6	φ_7	φ_8	
1.223	1.473	6.431	2	17	3	0	4	1	2	0	0	**0.78**
1.301	1.511	6.253	2	16	3	1	3	2	1	0	0	**0.65**
...												

Table 1: Example of a Q matrix (represented as a table).

A supervised model M (using q as the output) is retrained periodically from table Q (gErl uses linear regression by default). Once the system has started, at each step, Q is updated using the following formula, as in Q-learning:

$$Q[s_t, a_t] \leftarrow \alpha \left[w_{t+1} + \gamma \max_{a_{t+1}} M(s_{t+1}, a_{t+1}) \right] + (1 - \alpha) Q[s_t, a_t] \tag{4}$$

where the maximum future value is obtained by the model instead of a Q-matrix. The previous formula has two parameters: the discount parameter $\gamma \in [0, 1]$, and the *learning rate* α ($\alpha \in [0, 1]$), which determines to what extent the newly acquired information will override the old information ($\alpha = 0$ makes the agent not to propagate anything, while $\alpha = 1$ makes the agent consider only the most recent information). By default, $\alpha = 0.5$ and $\gamma = 0.5$.

5 Reusing past policies

In this section, we describe how to reuse and apply policies.The abstract representation of states and rules in actions (the ϕ and φ features) facilitates the transfer of learning information between related but also different tasks (rules and actions may be completely different but their *features* can still be similar).

As we have seen in Section 2, in other TL methods the knowledge is transferred in several ways (via modifying the learning algorithm, biasing the initial action-value function, etc.) and, if the source and target task are very different, a mapping between actions and/or states is needed. Instead of that, in gErl the reuse of previous acquired knowledge is done in a totally transparent way.

The main reason why we can use the past policies (the learned "matrix" Q) in order to accelerate the learning of different new tasks is due to the abstract representation of states and actions which allows the system does not start from the scratch and reuse the optimal information, namely, actions successfully applied from certain states (from the previous task) when it reaches a similar (with similar features) new state. Due this abstract representation, how different are the source and target task does not matter, so the reuse of knowledge in gErl as a transfer learning approach would overlap with any of the TL families showed in Section 2 and goes beyond, allowing to deal with totally different tasks.

The scope of this knowledge transfer as a model transferring is summarised as follows: when gErl reaches the solution of a given problem (or it executes a maximum number of steps), Q_{source} and the model M_{source} can be viewed as knowledge acquired that can be transferred to the new situation. Concretely, when gErl learns the new task, Q_{source} is used to train a new model which is used from the beginning (M_{target}) and it is updated with the new information acquired in each time step of the system (in order to retrain the model M_{target}). Figure 2 briefly shows the process.

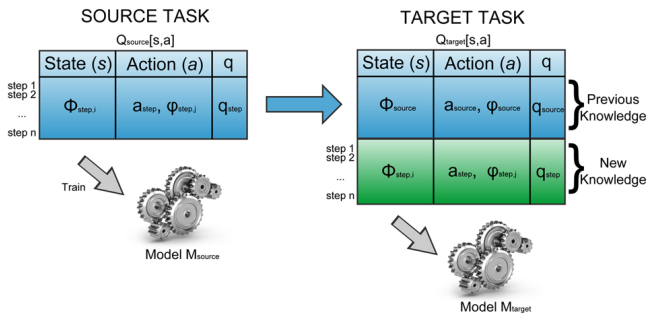


Fig. 2: The knowledge transfer process in the gErl system.

6 Empirical study

In this section, we describe an experiment which illustrates the policy reuse strategy used in gErl[2].

We use list processing problems as a structured prediction [1] domain where not only the input is structured but also the output. For the first part of our study we have selected five different problems that use the Latin alphabet $\Sigma = \{a, b, \ldots, z\}$ as a finite set of symbols and perform the following transformations:

1. $d \rightarrow c$: replaces *"d"* with *"c"*. Instances would look like this: $trans([t, r, a, d, e]) \rightarrow [t, r, a, c, e]$
2. $e \rightarrow ing$: replaces *"e"* with *"ing"* located at the last position of a list. Instances would look like this: $trans([t, r, a, d, e]) \rightarrow [t, r, a, d, i, n, g]$
3. $d \rightarrow pez$: replaces *"d"* with *"pez"* located at any position of a list. Instances would look like this: $trans([t, r, a, d, e]) \rightarrow [t, r, a, p, e, z, e]$
4. $Prefix_{over}$: adds the prefix *"over"*. Instances would look like this: $trans([t, r, a, d, e]) \rightarrow [o, v, e, r, t, r, a, d, e]$
5. $Suffix_{mark}$: adds the suffix *"mark"*. Instances would look like this: $trans([t, r, a, d, e]) \rightarrow [t, r, a, d, e, m, a, r, k]$

According to the data structure, we need a way to navigate the structure and apply local or global changes. In order to do this we need to define appropriate operators. The first operator, *oneSust*, is a two-step mechanism that takes the input and output lists in order to go through them comparing the lists until it finds any difference. As a result, it returns the input rule where its *rhs* has been substituted by a high-order function *map* (which applies a parametrised function to the whole list). The definition of this operator is written in Erlang, but it can be informally defined using an example: $oneSust(trans([a, b, c]) \rightarrow [d, b, c]) \Rightarrow trans([a, b, c]) \rightarrow map(a \rightarrow d, [a, b, c])$.

The second operator, *nSust*, which fits with the second and the third problem, also is a two-step mechanism similar to the previous one, but, instead of returning only one rule with a simple substitution, it returns as many rules as possible "complex" changes can be done from the difference between the input and output lists. This operator can be informally defined using an example: $nSust(trans([a, b, c]) \rightarrow [a, d, e, c]) \Rightarrow \{ \begin{array}{l} trans([a, b, c]) \rightarrow map(b \rightarrow de, [a, b, c]) \\ trans([a, b, c]) \rightarrow map(b \rightarrow dec, [a, b, c]) \end{array}$.

The third and fourth operators, $addPrefix$ and $addSuffix$, just check whether the input list is a suffix or prefix, respectively, of the output list and take the difference. If so, it is returned the input rule where the *rhs* has been substituted by the concatenation of the difference and the input list (prefix) or by the concatenation of the input list and the difference. For instance: $addPrefix(trans([a, b, c]) \rightarrow [z, a, b, c]) \Rightarrow trans([a, b, c]) \rightarrow [z] + +[a, b, c])$, $addSuffix(trans([a, b, c]) \rightarrow [a, b, c, z]) \Rightarrow trans([a, b, c]) \rightarrow [a, b, c] + +[z])$.

Finally, we need a way of generalising the rules. That is performed by the fifth and the sixth operators, $genLHS$ and $genRHS$, which generalise the input and output strings, respectively: $genLHS(trans(X) \rightarrow Y) \Rightarrow trans(V_S) \rightarrow Y$ and $genRHS(trans(X) \rightarrow Y) \Rightarrow trans(X) \rightarrow V_S$ where V_S is a string variable.

[2] gErl is available at `http://users.dsic.upv.es/~fmartinez/gerl.html`

With these six operators gErl is able to solve any of the previous problems simply applying a sequence of the correct operators for each problem. For instance, given the instance $trans([a,b,c]) \rightarrow [a,d,c]$, we have the sequence of operator applications:

$$\begin{array}{rcl} genPat(trans([a,b,c]) \rightarrow [a,d,c]) & \Rightarrow & trans([a,b,c]) \rightarrow map(b \rightarrow d,[a,b,c]) \\ genLHS(trans([a,b,c]) \rightarrow map(b \rightarrow d,[a,b,c])) & \Rightarrow & trans(V_S) \rightarrow map(b \rightarrow d,[a,b,c]) \\ genRHS(trans(V_S) \rightarrow map(b \rightarrow b,[a,b,c])) & \Rightarrow & trans(V_S) \rightarrow map(b \rightarrow d,V_S) \end{array}$$

where the latter rule $trans(V_S) \rightarrow map(t \rightarrow u, V_S)$ is the solution.

Since we want to analyse the ability of the system to improve the learning process when reusing past policies, we will solve each of the previous problems and we will reuse the policy (each model) obtained on a previous problem to solve the rest of the problems (including itself). We will also add a few number of non-relevant operators to increase the difficulty to solve each problem. The set of operators O used has ten operators. To make the experiments independent of the operator index, we will set up 5 random orders for them. Each problem has 20 positive instances e^+ and no negative ones. From each problem we will extract 5 random samples of ten positive instances in order to learn a policy from them with each of the five order of operators (5 problems $\times$ 5 samples $\times$ 5 operator orders =125 different experiments).

First, we show the results without the use of previous policies. The mean number of steps needed to solve each problem is shown in Table 2. Next, we see the results with policy reuse in Table 3, showing the aggregated means (in number of steps) of each sample and operator order.

To analyse whether the differences are significant (between the results that do not reuse and those that do), we used the Wilcoxon signed-ranks test with a confidence level of $\alpha = 0.05$ and $N = 25$ (5 samples $\times$ 5 operators orders). The results which significantly improve the performance obtained without using previous policies are shown in bold. Interestingly, the results obtained reusing works for most combinations (in those combinations where the difference is not significant, it is still better in magnitude), including those cases where the problems have nothing to do and do not reuse any operator, suggests that the abstract description of states and rules is beneficial even when the problems are not related and gives support to the idea of a general system that can perform better as it sees more and more problems, one of the reasons why the reinforcement model and the abstract representations were conceived in gErl.

	$l \rightarrow c$	$e \rightarrow ing$	$d \rightarrow pez$	$Prefix_{over}$	$Suffix_{mark}$
Steps	108.68	76.76	74.24	61.28	62.28

Table 2: Results not reusing previous policies.

	Problem				
PCY from	$l \rightarrow c$	$e \rightarrow ing$	$d \rightarrow pez$	$Prefix_{over}$	$Suffix_{mark}$
$l \rightarrow c$	**65.68**	**58**	70, 64	**48.84**	**49.12**
$e \rightarrow ing$	**66.48**	**50.04**	**56.4**	**45.2**	**45.36**
$d \rightarrow pez$	**56.36**	**49.6**	**57.32**	52.24	**45.84**
$Prefix_{over}$	**58.8**	**48.96**	**60.6**	**43.8**	**46.88**
$Suffix_{mark}$	102, 72	**64.4**	67.32	56.16	**57.48**
Average	70.01	54.2	62.46	49.25	48.94

Table 3: Results reusing policies

7 Conclusions and future work

One of the problems of reusing knowledge from previous learning problems to new ones is the representation and abstraction of this knowledge. In this paper we

have investigated how policy reuse can be useful even in cases where the problems have no operators in common, simply because some abstract characteristics of two learning problems are similar at a more general level. In order to make more conclusive claims, more experiments should be done on other domains.

There are many other things to explore in the context of gErl. We would like to include features for the operators as well. In more general terms, we think that a way (or measure) of similarity between problems would help us to better understand when the system is able to detect these similarities, from the point of view of better assessing the achievements of the system. Finally, while we have focussed on the system gErl, we think that many of the ideas in this paper could also be applied to other kinds of systems, most especially learning classifier systems, reinforcement learning and other evolutionary techniques.

Acknowledgements. This work was supported by the MEC projects CONSOLIDER-INGENIO 26706 and TIN 2010-21062-C02-02, GVA project PROMETEO/2008/051, and the REFRAME project granted by the European Coordinated Research on Long-term Challenges in Information and Communication Sciences & Technologies ERA-Net (CHIST-ERA), and funded by the Ministerio de Economía y Competitividad in Spain. Also, F. Martínez-Plumed is supported by FPI-ME grant BES-2011-045099.

References

1. T. Dietterich, P. Domingos, L. Getoor, S. Muggleton, and P. Tadepalli. Structured Machine Learning: the next ten years. *Machine Learning*, 73:3–23, 2008.
2. K Driessens. *Relational Reinforcement Learning. Phd Thesis*. Department of Computer Science, K.U.Leuven, Leuven, Belgium, 2004.
3. F. Fernandez and M. Veloso. Probabilistic policy reuse in a Reinforcement Learning agent. In *AAMAS 06*, pages 720–727. ACM Press, 2006.
4. C. Ferri, J. Hernández-Orallo, and M.J. Ramírez-Quintana. Incremental learning of functional logic programs. In *FLOPS*, pages 233–247, 2001.
5. J. Holland and Booker. What is a learning classifier system? In *Learning Classifier Systems*, volume 1813 of *LNCS*, pages 3–32. 2000.
6. J.Carroll. Fixed vs Dynamic Sub-transfer in Reinforcement Learning. In *ICMLA'02*. CSREA Press, 2002.
7. Y. Liu and P. Stone. Value-function-based transfer for reinforcement learning using structure mapping. AAAI, pages 415–20, July 2006.
8. F. Martínez, C. Ferri, J. Hernández, and M. J. Ramírez. Learning with configurable operators and rl-based heuristics. In *NFMCP*, volume 7765 of *LNCS*, pages 1–16. Springer-Verlag, 2013.
9. N. Mehta. Transfer in variable-reward hierarchical reinforcement learning. In *In Proc. of the Inductive Transfer workshop at NIPS*, 2005.
10. R. Sutton. *Reinforcement Learning: An Introduction*. MIT Press, 1998.
11. R. Sutton. Between MDPs and semi-MDPs: A framework for temporal abstraction in Reinforcement Learning. *Artificial Intelligence*, 112:181–211, 1999.
12. M. Taylor and P. Stone. Transfer learning for Reinforcement Learning domains: A survey. *Journal of Machine Learning Research*, 10(1):1633–1685, 2009.
13. C. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8:279–292, 1992.

Improving the Scalability of a Distributed Learning Algorithm Based on Frontier Vector Quantization and Information Theory

Diego Peteiro-Barral and Bertha Guijarro-Berdiñas

Dept. of Computer Science, University of A Coruña,
Campus de Elviña s/n, 15071 A Coruñn, Spain
{dpeteiro,cibertha}@udc.es
http://www.lidiagroup.com

Abstract. In machine learning, scalability is a trending issue. The challenge is to find a tradeoff among accuracy, training time, and allocated memory. The DFVQIT is a distributed algorithm based on frontier vector quantization and information theory that implements a genetic algorithm (GA) to combine the models inferred from distributed sites. The DFVQIT has proven effective in several data sets but the scalability of the GA is improvable. In order to do this, we propose two novel strategies: a cluster-based method, and a two-level approach in which the first level shrinks the search space for the second level. In average, cluster-based methods perform more than 3.22 times faster than GA-based methods at the expense of a 3.77% loss in accuracy. Additionally, the two-level approach reduces the training time by 25.56% whilst maintains accuracy. These results demonstrate the efficacy of the proposed methods in order to improve the scalability of the DFVQIT.

Keywords: Machine learning; distributed learning; FVQIT; distributed FVQIT; neural networks; genetic algorithms; clustering

1 Introduction

In recent years, the unconstrained growth of the volume of data has raised new challenges in machine learning regarding scalability. Scalability comprises not simply accuracy but several other measures regarding computational resources. In machine learning, scalability is defined by the effect that an increase in the size of the training set has on the computational performance of an algorithm (accuracy, training time and allocated memory). So the challenge is to find a tradeoff among them or, in other words, getting "good enough" solutions as "fast" as possible and as "efficiently" as possible. A sample of the interest generated by scalability was revealed with the organization of the workshop PASCAL Large Scale Learning Challenge [1] at the 25th International Conference on Machine learning (ICML'08). A more recent sample of the relevance of scalability in learning, also known as "big learning", was disclosed with the conference of the Neural Information Processing Systems Foundation (NIPS'2011), aiming to provide a forum for exchanging solutions that address big learning problems. The relevance of these conferences for researchers and practitioners is meaningful.

In a previous research [2], the authors developed a novel distributed learning algorithm built upon the Frontier Vector Quantization based on Information Theory (FVQIT) method [3,4]. The FVQIT showed good classification capacity but poor training time performance so that distributed learning seemed appropriate. One can take advantage of the distributed paradigm to speed up the training process of the FVQIT. The current trend of reducing speed of processors in favor of multi-core processors and computer clusters leads to a suitable context for implementing distributed learning algorithms. One of the most promising lines of research towards learning from distributed data sets is the MapReduce paradigm [5]. In the first stage (map), learners are trained on independent subsets of data –ideally, in parallel. Then, in the second stage (reduce), learners are integrated in some manner. MapReduce is appropriate in the case of the FVQIT and, for this purpose, we used a genetic algorithm (GA) to optimize the integration method. The results obtained from twelve classification data sets demonstrate the efficacy of the distributed method. In average, the distributed FVQIT (DFVQIT) performed 4.26 times faster than the FVQIT and improves classification accuracy by 5.25%.

As can be seen from these results, distributed learning may be significantly faster than batch learning. However, the use of a GA as integration method adds a quite expensive temporal and computational overhead on the DFVQIT. In order to improve the scalability of the DFVQIT, two novel implementations are presented in this research. On the one hand, we propose the use of a clustering method rather than a GA during the integration stage because the former are faster and computationally less expensive than the latter. On the other hand, we propose a divide-and-conquer approach to speedup the integration process. Thus, we plan to prune the distributed instances of the FVQIT during the map, parallel stage in order to shrink the search space for the integration method during the reduce phase. Note that this approach does not depend on the integration algorithm so it will be evaluated in both cluster and GA-based methods.

The rest of the paper is structured as follows: Section 2 describes the DFVQIT, Section 3 expounds the two novel approaches proposed in this research, Section 4 shows the experimental study, and Section 5 presents the conclusions and future work.

2 Background: Distributed FVQIT

The idea behind the FVQIT is to place a set of process elements (PEs) on the frontier between classes, in such a way that each PE will represent a local model. In other words, the FVQIT builds a piecewise representation of the borderline between classes in a classification problem. Figure 1 shows a simple two-class bi-dimensional example for illustrative purposes. Each PE defines a region by proximity in the feature space, in which the local model associated to it is composed of the nearest samples according the the Euclidean distance in the feature space.

In order to do this, the FVQIT minimizes the energy function J that calculates the divergence between the Parzen estimator of the distribution of data samples and the estimator of the distribution of the PEs,

$$J(\mathbf{c}) = 2\log\int f^{+}(\mathbf{x})\,g(\mathbf{x})\,dx - 2\log\int f^{-}(\mathbf{x})\,g(\mathbf{x})\,dx + \log\int g^{2}(\mathbf{x})\,dx \qquad (1)$$

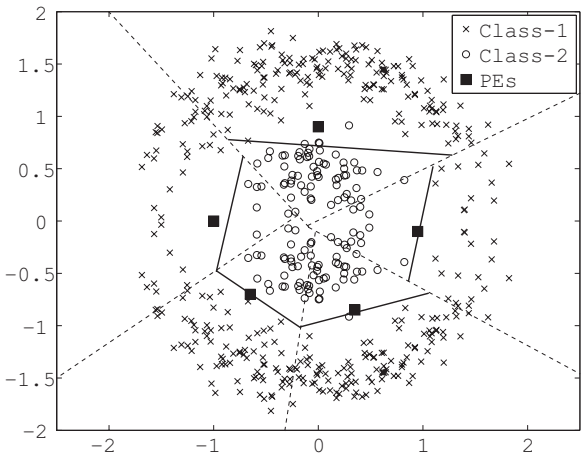


Fig. 1. Example of operation of FVQIT. Local models and frontier between classes. The division in local models is shown with dotted lines and the solid lines depict the decision regions defined by each local model.

where $f(\mathbf{x})$ and $g(\mathbf{x})$ are the Parzen density estimators of the distribution of data samples and PEs, respectively; $f^+(\mathbf{x})$ and $f^-(\mathbf{x})$ are the estimators of the distributions of data for the two nearest classes to the PE; $\mathbf{x} \in \Re^n$ are data samples, and $\mathbf{c} \in \Re^n$ are the positions associated to the PEs. Assuming this formulation, when the PEs are placed on the minimum of the energy function $J(\mathbf{c})$, they will find themselves well distributed on the frontiers between classes. Therefore, we utilize the gradient descent method to move the PEs toward such situation. The algorithm is summarized as follows,

1. Initialize the positions $\mathbf{c}_i, i = 1,\ldots,M$ of the PEs randomly.
2. Compute which class repels each PE and which class attracts it using the *k*-nearest neighbor rule [6]. The mode of the classes of the *k* closest samples will be the repelling class for that PE. The mode of the classes of the *k* closest samples that do not belong the the repelling class will be the attracting class.
3. Compute the cross information potential $C_+ = \int f^+(\mathbf{x})\, g(\mathbf{x})\, dx$ between each PE and the samples from the repelling class, and the cross information potential $C_- = \int f^-(\mathbf{x})\, g(\mathbf{x})\, dx$ between each PE and the data from the attracting class. Additionally, the entropy $V = \int g^2(\mathbf{x})\, dx$ computes the force of repulsion between PEs in order to avoid accumulation of PEs on a point and favor distribution.
4. Update the positions of each PE through gradient descent,

$$\mathbf{c}_i(n+1) = \mathbf{c}_i(n) - \eta \left(\frac{\nabla V}{V} + \frac{\nabla C_+}{C_+} - \frac{\nabla C_-}{C_-} \right), i = 1,\ldots,M \tag{2}$$

   where *n* is the iteration and η is the step size.
5. Repeat from 2 until the maximum number of iterations is reached.

Once the PEs are situated on the frontiers between classes, the goal is to construct a classifier for each local model. This classifier will be in charge of classifying samples

in the region assigned to its local model and will be trained only with the points of the training set in this region (see Figure 1). The FVQIT utilizes one-layer neural networks, a lightweight classifier trained with the efficient algorithm proposed in [7]. This algorithm allows rapid supervised training and requires less computational resources than classic methods. The key idea of the algorithm is to measure the error prior to the nonlinear activation functions. In this manner, the weights $\mathbf{w}$ of the ANN can be easily computed by solving a system of linear equations,

$$\mathbf{A} \cdot \mathbf{w} = \mathbf{b} \tag{3}$$

where

$$\begin{aligned} \mathbf{A} &= \mathbf{x} \cdot \mathbf{x}^T \\ \mathbf{b} &= f^{-1}(\mathbf{d}) \cdot \mathbf{x} \end{aligned} \tag{4}$$

with x being the input samples, d the desired outputs, and f the nonlinear activation function. Additionally, the advantage of this method is its ability to learn in an incremental fashion. If we assume that data set $1 \cup 2$ contains the union of data sets 1 and 2, the following relation holds

$$\mathbf{A}_{1\cup 2} \cdot \mathbf{w} = \mathbf{b}_{1\cup 2} \Leftrightarrow (\mathbf{A}_1 + \mathbf{A}_2) \cdot \mathbf{w} = \mathbf{b}_1 + \mathbf{b}_2 \tag{5}$$

The former distributed algorithm DFVQIT [2] is based on this relation. Note that a

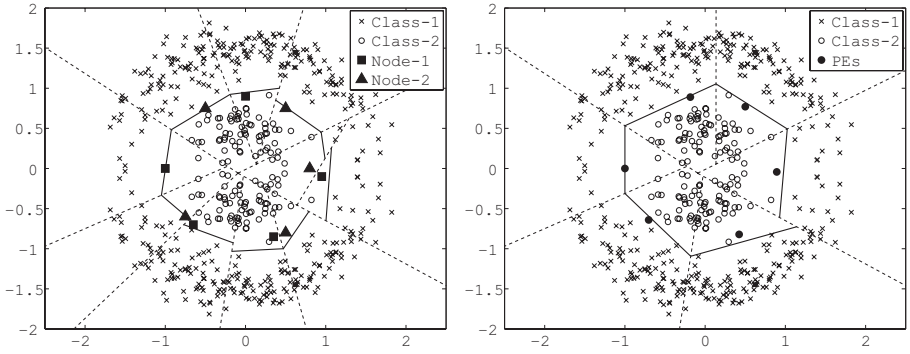


Fig. 2. Example of integration of two nodes in the DFVQIT. The upper figure shows the union of the PEs of nodes 1 and 2. The lower figure shows the combination of the PEs of the two nodes using the integration method.

straightforward integration may lead to redundant, crowded area in the input space that will affect the predictive ability of the algorithm, as can be seen in Figure 2. It is important to remark that a smaller number of PEs is related with better generalization performance and fast execution. Thus, in order to combine in a more effective manner the different instances of the FVQIT trained in the distributed sites, we implemented a genetic algorithm (GA) as integration method following the Michigan approach [8]. The initial population consists of the union of every PE from the distributed sites. The GA was defined by the following items,

- The fitness function is the class accuracy of the entire population because of the piecewise representation of the FVQIT. This change in approach will cause that the optimization process will be driven by fitness-maximization of the population rather than best individual.
- The selection function is based on the locality of the FVQIT and implemented using binary tournament selection [9] by pairwise Euclidean distance between PEs; the closer the PEs the larger the probability of being selected for breeding.
- The crossover operator is defined as the two parents (p_1 and p_2) procreate a child which its location is the midpoint of their locations $\mathbf{c}' = (\mathbf{c}_{p_1} + \mathbf{c}_{p_2})/2$ and its decision boundary is determined by the sum of their matrices of coefficients $\mathbf{A}' = \mathbf{A}_{p_1} + \mathbf{A}_{p_2}$, and $\mathbf{b}' = \mathbf{b}_{p_1} + \mathbf{b}_{p_2}$ which exploits the incremental features of the algorithm described above.
- Finally, the mutation operator changes the location $\mathbf{c}$ and the matrices of coefficients $\mathbf{A}$ and $\mathbf{b}$ of the individuals by small random disturbances.

The DFVQIT outperformed the batch FVQIT algorithm in terms of time, complexity of the resultant model and test accuracy [2]. However, the use of a GA as integration method in the DFVQIT adds a quite expensive temporal and computational overhead that may result in poor scalability particularly on very large data sets. In order to improve the scalability of the DFVQIT, two novel proposal are presented in the next section.

3 Novel Proposals

In the first place, we propose the use of a clustering method rather than a GA during the integration stage. In the second place, we propose to prune the distributed instances of the FVQIT during the map, parallel stage in order to shrink the search space for the integration method during the reduce phase. As demonstrated in [2], the integration method of the DFVQIT can be also seen as a pruning method of the FVQIT inasmuch as it diminishes the number of PEs and maintains, or even improves, performance.

3.1 Cluster-Based Pruning

Clustering is expected to be much faster than GA at the expense of a less flexible result. The cluster-based pruning method consists of the following steps,

1. Compute the pairwise Euclidean distance between PEs.
2. Build a tree of hierarchical clusters, dendrogram of the PEs using the shortest distance method. A dendrogram (see example in Figure 3) consists of many U-shaped lines that connect PEs in a hierarchical tree. The height of each U represents the distance between the two PEs being connected.
3. Compute the smallest height at which a horizontal cut through the dendrogram leaves n clusters. Find the optimal set of clusters by cross-validation.

It is expected that the output model will be a more general representation of the training data with fewer PEs.

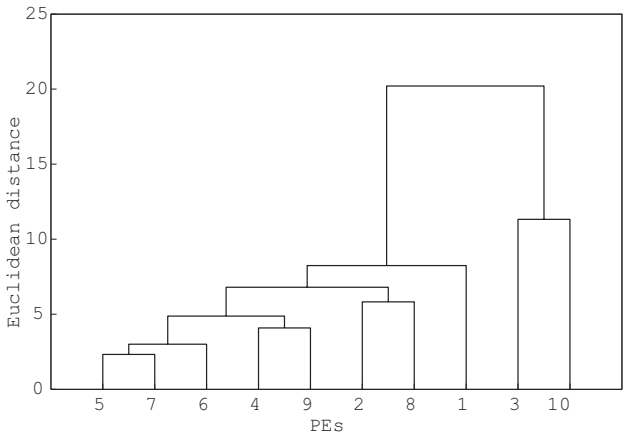


Fig. 3. Example of dendrogram.

3.2 Two-Level Pruning

The algorithm proposed in [2] and that proposed in this research, are implemented as a one-level integration method, i.e. global pruning. As an improvement in performance, we can think about a two-level algorithm for the integration method, with a first, local pruning (during the map stage of MapReduce) and a second, global pruning (during the reduce phase). The hypothesis is that the first, local pruning will quickly shrink the search space for the second, global pruning. Note that this approach does not depend on the integration algorithm so it will be applicable in both GA and cluster-based methods. The two-level method is summarized as follows using MapReduce,

Map, In every partition...

- Divide the local data into training and validation sets.
- Run the FVQIT (see Section 2) on the local training data.
- Run the local pruning method (GA or cluster-based methods) on the local validation data.

Reduce

- Broadcast the FVQIT-models to a designated site.
- Run the global pruning method (GA or cluster-based methods).

Note that the local pruning method is run in parallel in every node. A smaller search space is expected to speedup training, particularly when the size of the population or the data set are very large.

4 Experimental Study

The objective of this section is to experimentally evaluate the performance of the methods proposed in this research. We compare the following four implementations of the

DFVQIT: one-level GA-based approach (former DFVQIT [2], “gold standard” for comparison), one-level cluster-based approach, two-level GA-based approach, and two-level cluster-based approach.

4.1 Materials and Methods

The algorithms are evaluated to classify twelve data sets of diverse kinds of tasks. Table 1 summarizes the number of input features, samples, and output classes of the data sets. A more detailed description of the twelve data sets can be found in [10].

Table 1. Brief description of the data sets.

Name	Features	Samples	Classes
Abalone	8	4,177	28
Adult	33	30,162	2
Chess	6	28,056	18
Connect4	42	67,557	3
Forest	54	101,241	7
Letter	16	20,000	26
Magic	10	19,020	2
Mushroom	22	8,124	2
Nursery	16	12,960	5
Poker	10	25,010	10
Shuttle	9	43,500	7
Waveform	21	5,000	3

4.2 Experimental Procedure

As far as possible, identical conditions have been chosen to ensure a fair comparison of the different results. In distributed scenarios, training data have been scattered across 5 different nodes in which each node contains 10 PEs. Thus, the number of PEs after applying the reduce method may be up to 50.

The evaluation of the methods has been done using holdout validation: 90% of data for training and 10% for testing. Additionally, 10% of the training data have been used for validation. Experiments were run 10 times with random partitions of the data set to ensure reliable results. We use the methodology proposed by Demsar [11] to perform a statistical comparison of the algorithms over the multiple data sets. First a Friedman test [12] is done and then, if significant differences are found, the Bonferroni-Dunn test [13] is considered.

4.3 Results and Discussion

The training time of the four implementations of the algorithm is shown in Table 2. In average, cluster-based methods perform 3.51 and 3.23 times faster than GA-based

methods in the one and two-level pruning approaches, respectively. As expected, these differences are significant with respect to the former implementation of the FVQIT. Moreover, the two-level approach is worth it in terms of time. This approach reduces by 28.67% the training time of GA-based method and by 22.45% of cluster-based methods. These results support the initial assumption that states that the first, local pruning would quickly reduce the search space for the second, global pruning. Note also that this response is expected to be boosted for larger numbers of PEs.

Table 3 presents the number of PEs at the end of training. Both GA-based and cluster-based pruning methods are able to significantly reduce the number of PEs (around 81% the former and 68% the latter). As conjectured, the stochastic behavior of GAs plays in favor of flexible configurations of PEs against the deterministic behavior of clustering. Moreover, the two-level methods reduce the number of PEs by and extra 16.86% and 1.59% in GA and cluster-based algorithms, respectively, in comparison with one-level pruning methods. However, these differences are not statistically significant.

Finally, Table 4 shows the test error of the four implementations. In average, the differences in terms of error between the one and two-level approaches are negligible ($< 0.23\%$). In the question concerning the performance of GA-based against cluster-based methods, the former outperform the latter by 3.77%. Significant differences are found between GA-based and cluster-based methods, but not between one and two-level approaches.

In summary, these results suggest that cluster-based methods are faster than those based on GA at the expense of slight higher number of PEs and performance error. If we compare one-level against two-level approaches, the latter are faster and simpler –less number of PEs– whilst maintains performance error. In light of the above, we can say that if the learning process is strongly constrained by the training time then the two-level cluster-based DFVQIT is recommendable. Contrarily, if the learning process is constrained by maximum-accuracy then the two-level GA-based DFVQIT is the most appropriate election.

5 Conclusions and Future Work

Distributed learning is a promising line of research to improve the scalability of batch learning algorithms. In this research, we exploit the advantages of MapReduce paradigm in order to improve the performance of the distributed algorithm DFVQIT. On the one hand, the clustering method implemented has been demonstrated efficient in terms of training time (more than 3.22 times faster) at the expense of a slight degradation in performance (3.77% loss in accuracy) with respect to the GA-based approach. On the other hand, the divide-and-conquer, two-level approach has been proven effective in both GA and cluster-based algorithms: the performance error is maintained whilst the complexity of the algorithm –number of PEs– is diminished up to 16.86% and the training time is reduced by 25.56%. These results demonstrate the performance of the methods proposed in this research in comparison with previous implementations.

For future work, we plan to extend this research to distributions of data with skewness. Moreover, the use of a more complex classifier will be considered in order to

Table 2. Training time (s).

Data set	One-level pruning		Two-level pruning	
	GA-based	*Cluster-based*	*GA-based*	*Cluster-based*
Abalone	6.41 ± 0.51	1.68 ± 0.02	4.55 ± 0.58	1.37 ± 0.09
Adult	11.58 ± 2.00	8.17 ± 0.68	7.57 ± 2.66	5.25 ± 0.30
Chess	17.23 ± 1.61	3.08 ± 0.02	13.96 ± 1.06	3.03 ± 0.13
Connect4	33.80 ± 8.66	17.97 ± 0.22	19.40 ± 8.99	10.97 ± 0.25
Forest	142.67 ± 8.97	29.56 ± 0.30	104.69 ± 9.06	24.61 ± 0.88
Letter	32.41 ± 1.49	3.37 ± 0.04	24.46 ± 1.81	3.53 ± 0.03
Magic	7.51 ± 0.75	2.84 ± 0.01	4.69 ± 0.42	2.46 ± 0.06
Mushroom	7.63 ± 0.37	4.38 ± 0.40	5.81 ± 0.52	3.05 ± 0.26
Nursery	7.69 ± 0.76	2.46 ± 0.02	5.36 ± 0.78	2.26 ± 0.11
Poker	5.43 ± 2.11	3.29 ± 0.02	2.52 ± 1.52	2.49 ± 0.07
Shuttle	15.89 ± 0.69	3.86 ± 0.06	12.60 ± 1.35	3.72 ± 0.13
Waveform	5.27 ± 0.58	3.05 ± 0.33	3.76 ± 0.64	2.18 ± 0.24

Table 3. Number of PEs.

Data set	One-level pruning		Two-level pruning	
	GA-based	*Cluster-based*	*GA-based*	*Cluster-based*
Abalone	7.00 ± 1.70	7.80 ± 2.30	4.70 ± 1.42	9.50 ± 5.68
Adult	2.40 ± 1.35	1.40 ± 0.84	1.30 ± 0.67	2.10 ± 1.14
Chess	13.30 ± 2.54	19.60 ± 6.95	11.90 ± 1.73	14.30 ± 5.02
Connect4	2.40 ± 1.65	7.00 ± 3.43	1.50 ± 0.85	8.10 ± 3.88
Forest	17.89 ± 1.90	23.90 ± 6.30	15.22 ± 3.63	21.70 ± 7.11
Letter	27.40 ± 2.99	32.60 ± 5.85	29.00 ± 3.20	33.50 ± 3.67
Magic	7.10 ± 1.66	8.30 ± 4.57	5.00 ± 1.05	12.40 ± 6.12
Mushroom	11.30 ± 1.25	34.50 ± 11.19	9.30 ± 1.49	31.40 ± 8.10
Nursery	9.80 ± 2.44	18.90 ± 14.68	4.60 ± 1.58	19.30 ± 5.32
Poker	1.50 ± 0.71	1.60 ± 0.84	1.10 ± 0.32	1.00 ± 0.00
Shuttle	15.10 ± 2.56	32.00 ± 8.74	12.20 ± 2.62	27.40 ± 8.19
Waveform	6.20 ± 1.32	7.20 ± 4.57	5.10 ± 1.66	11.00 ± 8.76

Table 4. Performance error (%).

Data set	One-level pruning		Two-level pruning	
	GA-based	*Cluster-based*	*GA-based*	*Cluster-based*
Abalone	77.06 ± 2.58	80.12 ± 1.92	77.32 ± 1.46	79.69 ± 1.03
Adult	17.25 ± 0.83	17.20 ± 1.08	16.71 ± 0.61	17.35 ± 1.02
Chess	68.39 ± 1.17	69.57 ± 1.00	69.42 ± 1.38	69.98 ± 1.06
Connect4	25.96 ± 0.82	27.89 ± 0.81	25.99 ± 0.70	27.28 ± 1.01
Forest	34.87 ± 0.75	35.77 ± 0.68	34.71 ± 0.64	36.68 ± 1.07
Letter	22.84 ± 1.16	22.50 ± 1.73	23.45 ± 1.12	22.84 ± 1.28
Magic	15.69 ± 0.83	18.10 ± 0.76	15.87 ± 0.49	17.31 ± 1.23
Mushroom	11.34 ± 0.78	13.79 ± 0.84	12.44 ± 1.34	13.35 ± 1.90
Nursery	8.31 ± 0.87	8.87 ± 0.74	8.52 ± 1.04	8.73 ± 1.25
Poker	51.60 ± 1.13	52.56 ± 1.99	51.46 ± 1.44	51.68 ± 1.95
Shuttle	0.54 ± 0.12	1.28 ± 0.56	0.57 ± 0.24	1.10 ± 0.35
Waveform	17.16 ± 1.73	17.12 ± 2.53	17.26 ± 2.49	16.72 ± 1.39

evaluate whether accuracy is maintained, or improved, with a lower number of PEs. It is an open question if this will be worth it in terms of training time.

Acknowledgments. This research has been economically supported in part by the Secretaría de Estado de Investigación of the Spanish Government through the research project TIN 2012-37954; and by the Consellería de Industria of the Xunta de Galicia through the research projects CN2011/007 and CN2012/211; all of them partially funded by FEDER funds of the European Union. Diego Peteiro-Barral acknowledges the support of Xunta de Galicia under *Plan I2C* Grant Program.

References

1. S. Sonnenburg, V. Franc, E. Yom-Tov, and M. Sebag. PASCAL Large Scale Learning Challenge. `http://largescale.first.fraunhofer.de/workshop/`, 2008.
2. D. Peteiro-Barral and B. Guijarro-Berdiñas. A distributed learning algorithm based on frontier vector quantization and information theory. *International Conference on Artificial Neural Networks (ICANN)*, 2013. Submitted.
3. David Martínez-Rego, Oscar Fontenla-Romero, Iago Porto-Díaz, and Amparo Alonso-Betanzos. A new supervised local modelling classifier based on information theory. In *International Joint Conference on Neural Networks (IJCNN)*, pages 2014–2020, 2009.
4. Iago Porto-Díaz, David Martínez-Rego, Amparo Alonso-Betanzos, and Oscar Fontenla-Romero. Information theoretic learning and local modeling for binary and multiclass classification. *Progress in Artificial Intelligence*, pages 1–14, 2012.
5. Cheng Chu, Sang Kyun Kim, Yi-An Lin, YuanYuan Yu, Gary Bradski, Andrew Y Ng, and Kunle Olukotun. Map-reduce for machine learning on multicore. *Advances in neural information processing systems*, 19:281, 2007.
6. T. Cover and P. Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27, 1967.
7. E. Castillo, O. Fontenla-Romero, B. Guijarro-Berdiñas, and A. Alonso-Betanzos. A Global Optimum Approach for One-Layer Neural Networks. *Neural Computation*, 14(6):1429–1449, 2002.
8. David E Goldberg. Genetic algorithms in search, optimization, and machine learning. 1989.
9. David E Goldberg and Kalyanmoy Deb. A comparative analysis of selection schemes used in genetic algorithms. *Urbana*, 51:61801–2996, 1991.
10. A. Frank and A. Asuncion. UCI machine learning repository, 2010.
11. Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006.
12. Milton Friedman. A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, 11(1):86–92, 1940.
13. Olive Jean Dunn. Multiple comparisons among means. *Journal of the American Statistical Association*, 56(293):52–64, 1961.

PQ-learning: Aprendizaje por refuerzo multiobjetivo

Manuela Ruiz-Montiel, Lawrence Mandow, and José-Luis Pérez-de-la-Cruz

ETSI Informática, Universidad de Málaga
Bulevar Louis Pasteur nº35, 29071, Málaga, España
{mruiz,lawrence,perez}@lcc.uma.es

Resumen En este artículo describimos y analizamos PQ-learning, un algoritmo para problemas de aprendizaje por refuerzo multiobjetivo. El algoritmo es una extensión de Q-learning, un algoritmo para problemas de aprendizaje por refuerzo escalares. Al contrario que otros algoritmos, PQ-learning no requiere información de preferencias sobre los objetivos, es aplicable a problemas con fronteras de Pareto no convexas y permite recuperar a partir de los Q-valores las secuencias de acción correspondientes a diferentes políticas Pareto-óptimas. PQ-learning ha sido aplicado a dos problemas pertenecientes a un banco de pruebas propuesto en la literatura de aprendizaje por refuerzo multiobjetivo.

Key words: Aprendizaje por Refuerzo, Optimización Multiobjetivo, Q-learning

1. Introducción

En este trabajo presentamos PQ-learning, un nuevo algoritmo para determinar cómo un agente debe comportarse en un entorno de modo que se maximice la recompensa acumulada a lo largo del tiempo, donde tal recompensa es un vector $\overrightarrow{r} \in \mathbb{R}^n$, es decir, consta de un componente por cada objetivo a maximizar.

En este contexto, el entorno se especifica típicamente como un *proceso de decisión markoviano* (PDM), un formalismo usado para modelar la toma de decisiones en situaciones que presentan incertidumbre. La mayoría de las aproximaciones existentes trabajan con procesos de decisión markovianos donde la recompensa es escalar. Formalmente, un PDM escalar se define por un conjunto de estados S, un conjunto de acciones A y por la *dinámica del entorno* (el *modelo*). Esta dinámica viene dada por dos funciones: (1) $P_{sa}(s')$, que es la probabilidad de alcanzar el estado s' tomando la acción a en el estado s, y (2) $R_{sa}(s') : S \times A \times S \rightarrow \mathbb{R}$, que es la recompensa obtenida al alcanzar el estado s' escogiendo la acción a en el estado s. La solución a un PDM escalar es una política de actuación $\pi : S \rightarrow A$ que indica qué acción tomar en cada estado, de manera que se maximice el valor de la recompensa acumulada a partir de un instante t, $R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \Sigma_{k=0}^{\infty} \gamma^k r_{t+k+1}, 0 \leq \gamma \leq 1$. El factor de descuento γ determina la importancia que el agente otorga a recompensas obtenidas en el futuro. Si $\gamma = 1$, tienen el mismo valor que las obtenidas inmediatamente; si $\gamma < 1$, las obtenidas inmediatamente tienen más peso.

Los métodos para obtener la política óptima de un PDM escalar se clasifican a grandes rasgos en: (1) programación dinámica, donde se asume un conocimiento perfecto de la dinámica del entorno y la política óptima se calcula de manera exacta, y (2) aprendizaje por refuerzo (AR), donde la política óptima se aproxima mediante un proceso de interacción con el entorno y por tanto no se asume un conocimiento perfecto del modelo. Un ejemplo de método de AR es el algoritmo Q-learning [1], ampliamente usado en la literatura, donde el agente va optimizando los valores $Q(s,a) : S \times A \to \mathbb{R}$. Dada una política, estos Q-valores representan la calidad de una combinación s, a, es decir, el valor esperado al seguir la política tras tomar la acción a en el estado s. Formalmente, $Q(s,a) = E\{\Sigma_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a\}$.

Muchos procesos de decisión markovianos del mundo real se formulan mejor en términos de optimización multiobjetivo, donde las recompensas son vectores de n dimensiones, una por cada objetivo a maximizar. La solución a un PDM multiobjetivo es una política de actuación $\pi : S \to A$ que maximiza el valor de la recompensa acumulada a partir de un instante t, $\overrightarrow{R}_t = \Sigma_{k=0}^{\infty} \gamma^k \overrightarrow{r}_{t+k+1}$. Al no poder establecerse un orden total entre vectores, en este caso el resultado de optimizar la recompensa no es un único valor, sino un conjunto de valores *no dominados* denominado *frontera de Pareto*. Decimos que un vector $\overrightarrow{v}$ domina a un vector $\overrightarrow{w}$, $\overrightarrow{v} \succ \overrightarrow{w}$, cuando $\exists i : v_i > w_i \wedge \nexists j : v_j < w_j$, donde v_i es la componente i-ésima de un vector $\overrightarrow{v}$. Dado un conjunto de vectores Q, su frontera de Pareto contiene aquellos vectores que no son dominados por ningún otro vector en Q, es decir, $\{\overrightarrow{v} \in Q : \{\overrightarrow{w} \in Q : \overrightarrow{w} \succ \overrightarrow{v}\} = \varnothing\}$.

La investigación dedicada a resolver procesos de decisión markovianos multiobjetivo es relativamente reciente. Las aproximaciones se clasifican en *single-policy* y *multi-policy* [2]. Los algoritmos single-policy averiguan una única política de actuación según cierta información de preferencias que permite establecer un orden total entre los valores. Por ejemplo, Gábor et al. [3] amplían un algoritmo de programación dinámica con un orden lexicográfico sobre los objetivos. Natarajan y Tadepalli [4] amplían técnicas de programación dinámica y de AR con escalarización lineal: se establece un peso por cada objetivo, y así puede obtenerse un único valor escalar (la suma ponderada de los objetivos) con el cual trabajar. Por su parte, los algoritmos multi-policy averiguan un conjunto de políticas cuyos valores acumulados pretenden aproximar la frontera de Pareto del problema de optimización multiobjetivo. Uno de los primeros representantes de este tipo de algoritmos es el propuesto por White [5], basado en programación dinámica. En este algoritmo se van manteniendo conjuntos de vectores no dominados en lugar de valores escalares. La propuesta de Barrett y Narayanan [6] es similar a la de White, pero mantiene envolventes convexas en lugar de conjuntos no dominados, lo cual proporciona ventajas en cuanto a eficiencia. Sin embargo, cuando la frontera de Pareto del problema es no convexa, esta propuesta no averigua todas las políticas. Más recientemente, Issabekov y Vamplew [7] han estudiado el comportamiento de Q-learning ampliado con información de preferencias lexicográficas o lineales. En general, la incorporación de preferencias a cualquier tipo de algoritmo de AR multiobjetivo permite aliviar el proceso de

aprendizaje, pero tiene algunos inconvenientes: (1) para especificarlas suele ser necesario cierto conocimiento del modelo, y (2) existen limitaciones en el uso de la escalarización lineal a la hora de averiguar fronteras no convexas [8].

En este trabajo proponemos un nuevo algoritmo de AR multiobjetivo, PQ-learning, basado en el algoritmo de AR escalar Q-learning [1], que pretende solventar los inconvenientes de las aproximaciones existentes. Aprende un conjunto de políticas que aproximan la frontera de Pareto (es multi-policy), sin asumir el conocimiento del modelo que suponen las técnicas de programación dinámica, ni ningún tipo de información de preferencias. Además, PQ-learning es aplicable a problemas con fronteras no convexas y, a diferencia de las aproximaciones anteriores, proporciona información suficiente para recuperar las secuencias de acción de diferentes políticas. Hasta donde sabemos, PQ-learning es el único algoritmo que reúne estas características.

PQ-learning ha sido aplicado a dos problemas pertenecientes a un banco de pruebas propuesto en la literatura de AR multiobjetivo [2]. Los resultados obtenidos se han comparado con los arrojados por una extensión de Q-learning que utiliza técnicas habituales en programación dinámica multiobjetivo.

La Sección 2 describe y analiza el algoritmo PQ-learning. La Sección 3 presenta algunos resultados empíricos del algoritmo. Finalmente, la Sección 4 recopila las conclusiones y apunta las líneas posibles de continuación de este trabajo.

2. Pareto Q-learning

En esta sección describimos el algoritmo propuesto, basado en el algoritmo Q-learning [1,9], donde la política aprendida π viene dada por la expresión $\pi(s) = argmax_a Q(s,a)$. Q-learning se clasifica como un algoritmo *off-policy*, es decir, la política que utiliza para interactuar con el entorno es distinta de la política que aprende. Por esta razon partimos de Q-learning: como queremos aprender más de una política a la vez, tenemos que usar un algoritmo que utilice una política distinta de la aprendida para interactuar con el entorno [6]. En el Cuadro 1 se muestra el esquema propuesto para el algoritmo Q-learning multiobjetivo. A continuación detallamos algunas consideraciones:

1. El operador ND devuelve la frontera de Pareto de un conjunto de vectores
2. Cada $Q(s,a)$ es un conjunto y no un valor escalar. Inicializamos cada $Q(s,a)$ al conjunto vacío (línea 1)
3. Cuando se decide explotar la tabla Q (línea 5), con probabilidad $1-\epsilon$ escogeremos una acción de entre todas las correspondientes a los valores del conjunto $NDU(s) = ND(\bigcup_a (Q(s,a)))$. A cada una de estas acciones se le asocia una probabilidad de ser escogida, proporcional a la cantidad de valores asociados a ella en el conjunto $NDU(s)$. Con probabilidad ϵ, escogeremos una acción aleatoriamente, para explorar el entorno (esta política de exploración se denomina ϵ-greedy). El estado s' es el estado al que conduce la acción escogida en este paso.
4. En Q-learning escalar, $Q(s,a)$ se actualiza con el valor $r + \gamma max_{a'} Q(s',a')$. En Q-learning multiobjetivo, el conjunto $Q(s,a)$ se actualiza con $\overrightarrow{r} + \gamma NDU(s')$. Si s' es final, $Q(s,a)$ se actualiza con $\{\overrightarrow{r}\}$ (línea 7)

5. La traslación y escala de conjuntos no dominados (línea 7) se define como sigue: $\overrightarrow{u} + bQ = \{\overrightarrow{u} + bq : \overrightarrow{q} \in Q, Q \neq \varnothing\}$, $\overrightarrow{u} + \varnothing = \varnothing$, $bQ = \{b\overrightarrow{q} : \overrightarrow{q} \in Q\}$

En este esquema queda por definir el método utilizado para actualizar el valor estimado hasta el momento (línea 7), $Q(s, a)$, con la nueva estimación $\overrightarrow{r} + \gamma NDU(s')$. A continuación veremos, en primer lugar, una manera directa pero inviable de hacerlo, y en segundo, el método que da lugar al algoritmo propuesto PQ-learning.

2.1. Aproximación Ingenua: Actualización Exhaustiva

En Q-learning escalar, la actualización del valor $Q(s, a)$ con el valor $r + \gamma max_{a'} Q(s', a')$ se realiza mediante la expresión $Q(s, a) = (1 - \alpha)Q(s, a) + \alpha(r + \gamma max_{a'} Q(s', a'))$, es decir, mediante una suma de valores escalares. En Q-learning multiobjetivo nos limitaremos de momento a sumar los conjuntos de valores no dominados $Q(s, a)$ y $\overrightarrow{r} + \gamma NDU(s')$. Cuando ambos conjuntos son no vacíos, la expresión de actualización es $Q(s, a) = (1 - \alpha)Q(s, a) + \alpha(\overrightarrow{r} + \gamma NDU(s'))$. Sean $Q = (1 - \alpha)Q(s, a)$ y $U = \alpha(\overrightarrow{r} + \gamma NDU(s'))$ ($U = \alpha\{\overrightarrow{r}\}$ si s' es final). En trabajos previos de programación dinámica multiobjetivo [5,6] la suma de conjuntos de valores no dominados, que denominaremos como *exhaustiva*, se define como sigue: $Q + U = ND\{\overrightarrow{q} + \overrightarrow{u} : \overrightarrow{q} \in Q, \overrightarrow{u} \in U\}$. Si $Q(s, a) = \varnothing$, la actualización se realiza mediante la expresión $Q(s, a) = \overrightarrow{r} + \gamma NDU(s')$ ($Q(s, a) = \{\overrightarrow{r}\}$ si s' es final). Si $\overrightarrow{r} + \gamma NDU(s') = \varnothing$, $Q(s, a)$ no cambia.

Sin embargo, a continuación veremos un ejemplo del uso del esquema propuesto donde esta suma exhaustiva no da buenos resultados. En la Figura 1 podemos ver un ejemplo de PDM multiobjetivo. El agente comienza en el estado s_0 y ha de llegar al estado final s_4. La tasa de descuento γ es igual a 1. En este modelo sólo hay dos políticas posibles que el agente puede seguir (ambas son óptimas): (1) la que en s_1 toma la acción a_{11} (recompensa acumulada esperada: $(-3, 2)$), y (2) la que en s_1 toma la acción a_{12} (recompensa acumulada esperada: $(-2{,}5, 1{,}5)$). En el resto de estados sólo hay una acción disponible, luego no cabe duda de cuál escoger. Esto quiere decir que al finalizar el proceso de aprendizaje, en la posición $Q(s_0, a_0)$ de la tabla debe haber dos vectores, correspondientes a las dos políticas anteriores.

En la primera columna del Cuadro 2 se muestra una sucesión de posibles episodios de aprendizaje de Q-learning multiobjetivo con actualización exhaus-

Algoritmo: Q-learning multiobjetivo

```
1. Inicializar todas las posiciones de la tabla Q a ∅
2. Repetir (para cada episodio):
3.     Inicializar s
4.     Repetir (para cada paso del episodio):
5.         Elegir a desde s, usando política ε-greedy derivada de Q
6.         Tomar acción a, observar r⃗ y s'
7.         Q(s, a) ← actualizar(Q(s, a), r⃗ + γNDU(s'))
8.         s ← s'
9.     Hasta que s sea terminal
```

Cuadro 1. Esquema propuesto para Q-learning con múltiples objetivos

tiva ($\alpha = 0{,}5$). La segunda columna muestra las posiciones de la tabla Q que se se ven modificadas en cada secuencia. En la séptima fila (segunda columna), en el conjunto $Q(s_0, a_0)$, el vector $(-2{,}75, 1{,}75)$ ha surgido de actualizar $(-3, 2)$, que surgió al realizar la secuencia $s_0, a_0, s_1, a_{11}, s_2, a_2, s_4$ (tercera fila), con el vector $(-2{,}5, 1{,}5)$ proveniente de sumarle la recompensa al vector $(-1{,}5, 1{,}5)$ de $Q(s_1, a_{12})$, en la sexta fila. Es decir, se está *mezclando* la secuencia $s_0, a_0, s_1, a_{11}, s_2, a_2, s_4$ con la secuencia $s_0, a_0, s_1, a_{12}, s_3, a_3, s_4$.

En aprendizaje con un único objetivo, las posiciones de la tabla Q no son conjuntos sino valores escalares. En tal situación, mezclar secuencias no da lugar a ningún problema: el hecho de actualizar un valor asociado a una secuencia determinada con el valor correspondiente a otra distinta, significa que la nueva secuencia es mejor que la anterior. Ese único valor irá convergiendo hasta su valor óptimo conforme avanzan los episodios de aprendizaje. Sin embargo, en aprendizaje multiobjetivo, donde las posiciones de la tabla Q son conjuntos, éstos pueden crecer de manera descontrolada si usamos la suma exhaustiva, haciendo el algoritmo inviable. Podemos apreciar este fenómeno en la octava fila del Cuadro 2. Vemos que aparecen dos nuevos vectores en $Q(s_0, a_0)$. Sabemos que en $Q(s_0, a_0)$ sólo tiene sentido que haya dos vectores, luego dos de los cuatro vectores sobran. Intuitivamente, parece lógico pensar que deberíamos quedarnos con el primero y

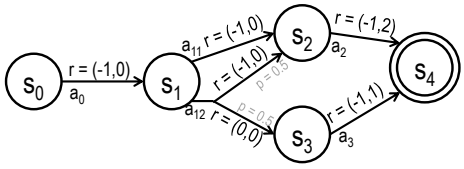


Figura 1. Ejemplo de PDM multiobjetivo

Entrenamiento	**Act. exhaustiva**	**Act. sellos**
$s_0, a_0, s_1, a_{11}, s_2, a_2, s_4$	$Q(s_2, a_2) = \{(-1, 2)\}$	$Q(s_2, a_2) = \{(-1, 2)_{(-1,2)f3}\}$
$s_0, a_0, s_1, a_{11}, s_2, a_2, s_4$	$Q(s_1, a_{11}) = \{(-2, 2)\}$	$Q(s_1, a_{11}) = \{(-2, 2)_{(-2,2)f3}\}$
$s_0, a_0, s_1, a_{11}, s_2, a_2, s_4$	$Q(s_0, a_0) = \{(-3, 2)\}$	$Q(s_0, a_0) = \{(-2, 2)_{(-3,2)f3}\}$
$s_0, a_0, s_1, a_{12}, s_2, a_2, s_4$	$Q(s_1, a_{12}) = \{(-2, 2)\}$	$Q(s_1, a_{12}) = \{(-2, 2)_{(-2,2)f3}\}$
$s_0, a_0, s_1, a_{12}, s_3, a_3, s_4$	$Q(s_3, a_3) = \{(-1, 1)\}$	$Q(s_3, a_3) = \{(-1, 1)_{(-1,1)f3}\}$
$s_0, a_0, s_1, a_{12}, s_3, a_3, s_4$	$Q(s_1, a_{12}) = \{(-1, 5, 1, 5)\}$	$Q(s_1, a_{12}) =$ $\{(-1{,}5, 1{,}5)_{(-2,2)(-1,1)f2},$ $(-1, 1)_{(-1,1)f3}\}$
$s_0, a_0, s_1, a_{12}, s_3, a_3, s_4$	$Q(s_0, a_0) =$ $\{(-3, 2), (-2{,}75, 1{,}75)\}$	$Q(s_0, a_0) = \{(-3, 2)_{(-3,2)f1},$ $(-2{,}5, 1{,}5)_{(-2,5,1,5)f3}, (-2, 1)_{(-2,1)f3}\}$
$s_0, a_0, s_1, a_{11}, s_2, a_2, s_4$	$Q(s_0, a_0) = \{(-3, 2), (-2{,}75, 1{,}75),$ $(-2{,}875, 1{,}875), (-2{,}625, 1{,}625)\}$	$Q(s_0, a_0) = \{(-3, 2)_{(-3,2)f1},$ $(-2{,}5, 1{,}5)_{(-2,5,1,5)f1}, (-2, 1)_{(-2,1)f1}\}$

Cuadro 2. Posibles trazas de Q-learning multiobjetivo con actualización exhaustiva (columna 2) y de PQ-learning (columna 3)

con el cuarto, ya que el primer vector se corresponde con la política 1, y el cuarto es el que se parece más a la política 2, ya que ha sido actualizado dos veces con el vector correspondiente a tomar la acción a_{12}. Es más, podríamos ahorrarnos las actualizaciones intermedias que llevarían desde el vector $(-3, 2)$ al vector $(-2{,}5, 1{,}5)$, puesto que $(-3, 2)$ corresponde a una secuencia distinta. Es decir, en la septima fila, donde actualizamos el conjunto $Q(s_0, a_0) = \{(-3, 2)\}$ con el conjunto $\overrightarrow{r} + \gamma NDU(s_1) = \{(-3, 2), (-2{,}5, 1{,}5)\}$, podríamos copiar directamente el vector $(-2{,}5, 1{,}5)$ en $Q(s_0, a_0)$, luego $Q(s_0, a_0) = \{(-3, 2), (-2{,}5, 1{,}5)\}$. El vector $(-3, 2)$ de $Q(s_0, a_0)$ se seguiría actualizando con el vector $(-3, 2)$ de $\overrightarrow{r} + \gamma NDU(s_1)$, luego no cambiaría.

2.2. PQ-learning: Actualización con Sellos

Proponemos una alternativa para definir la suma de dos conjuntos, $Q + U$, de manera que se mantengan únicamente los vectores correspondientes a las secuencias que tienen sentido. Para esto utilizaremos *sellos*, que determinan qué vector de Q ha de actualizar cada vector de U. Un *vector sello* V es una terna $\langle \overrightarrow{v_V}, sello_{V\leftarrow}, sello_{V\rightarrow} \rangle$ donde $\overrightarrow{v_V} \in \mathbb{R}^n$, $sello_{V\leftarrow} : S \rightarrow \mathbb{N} \cup \{-1\}$, $sello_{V\rightarrow} : S \times A \rightarrow \mathbb{N} \cup \{-1\}$. La función $sello_{Q\leftarrow}$ de un vector sello $V_Q \in Q(s, a)$ indica, para cada estado s', qué vector del conjunto $\overrightarrow{r} + \gamma NDU(s')$ ha actualizado previamente a V_Q (independientemente de $\overrightarrow{r}$, ya que las operaciones de traslación y escala no afectan a las funciones sello). La función $sello_{U\rightarrow}$ de un vector sello $V_U \in U$ indica, para cada estado s y cada acción a, qué vector del conjunto $Q(s, a)$ ha sido actualizado previamente por V_U. Es decir, si el vector V_U ha actualizado previamente a V_Q, entonces $sello_{Q\leftarrow}(s') = sello_{U\rightarrow}(s, a)$. Si $sello_{Q\leftarrow}(s') = -1$, significa que V_Q nunca ha sido actualizado por ningún vector de $\overrightarrow{r} + \gamma NDU(s')$. Si $sello_{U\rightarrow}(s, a) = -1$, significa que V_U nunca ha actualizado a ningún vector de $Q(s, a)$. El algoritmo de actualización con sellos consta de tres fases a la hora de actualizar el conjunto $Q(s, a)$ con el conjunto $\overrightarrow{r} + \gamma NDU(s')$:

1. Actualizar los vectores de $Q(s, a)$ que ya han sido actualizados previamente por algún vector de $\overrightarrow{r} + \gamma(NDU(s'))$, con el vector correspondiente
2. Es posible que haya valores en $Q(s, a)$ que nunca hayan sido actualizados por ningun vector de $\overrightarrow{r} + \gamma NDU(s')$, en situaciones en las que una misma acción pueda llevar a estados distintos. En este caso, se realiza una actualización exhaustiva para estos vectores huérfanos de $Q(s, a)$ con todos los vectores de $\overrightarrow{r} + \gamma NDU(s')$, creando nuevos vectores en $Q(s, a)$
3. Inspeccionar $\overrightarrow{r} + \gamma NDU(s')$ en busca de vectores que no hayan copiado su valor en $Q(s, a)$, o cuyo valor asociado en $Q(s, a)$ se haya eliminado en algún episodio previo del algoritmo. En cualquiera de los dos casos, esta fase del algoritmo intenta añadir el vector de $\overrightarrow{r} + \gamma NDU(s')$ a $Q(s, a)$. En caso de que el motivo anterior fuera el primero, se genera un nuevo sello y se modifican las funciones sello correspondientes

El Cuadro 3 muestra detalladamente este proceso dividido en tres fases. Es necesario realizar algunas aclaraciones:

1. Cada sello generado tiene asociado un *factor de confianza* (θ_{sello}). Esto permite calcular el *factor de confianza global*, Θ_V, de un vector sello: $\Theta_V = min\{\theta_{sello} :$

$sello \in I_{sello_{V\leftarrow}} \setminus \{-1\}\}$. A la hora de comparar dos vectores, si alguno de ellos ha sido actualizado por más de un estado distinto, el operador $ND_\Theta(Q)$ no hará efectiva la comparación si el factor de confianza global de alguno de ellos es bajo ($\Theta_V < \Theta_{umbral}, 0 \leq \Theta_{umbral} \leq 1$). Así, aunque uno de los dos vectores sea dominado, se mantendrá en la frontera, ya que es posible que más adelante, cuando ese vector sea más fiable por estar más avanzado el proceso de aprendizaje, pase a estar en la frontera por derecho propio. Si por el contrario tiene un factor de confianza global alto ($\Theta_V > \Theta_{umbral}$), entonces puede ser eliminado.

2. El método *generarSello*() (líneas 18 y 29) devuelve el sello más pequeño que aún no ha sido usado
3. La modificación que la expresión $sello_{V_U\rightarrow}(s,a) = sello$ (líneas 20 y 31) realiza sobre V_U, se refleja también en el vector sello del conjunto $Q(s',a')$ correspondiente
4. El código completo para el algoritmo PQ-learning es el mismo que el del Cuadro 1, reemplazando la línea 7 por:
   $Q(s,a) \leftarrow actualizarSellos(Q(s,a), \overrightarrow{r} + \gamma NDU_\Theta(s'), s, a, s')$

actualizarSellos(Q, U, s, a, s')
1. Si $U = \varnothing$
2. Devolver Q
3. $Q_{s'} = \{V \in Q : sello_{V\leftarrow}(s') \neq -1\}$
4. $\overline{Q_{s'}} = \{V \in Q : sello_{V\leftarrow}(s') = -1\}$
5. $Q_{nuevo} = \varnothing$
6. $U_{ligados} = \varnothing$
7. Repetir (para cada $V_Q \in Q_{s'}$) // **Fase 1**
8. Si $\exists V_U \in U : sello_{V_U\rightarrow}(s,a) = sello_{V_Q\leftarrow}(s')$
9. $v_{nuevo} = (1-\alpha)\overrightarrow{v}_{V_Q} + \alpha\overrightarrow{v}_{V_U}$
10. $Q_{nuevo} = ND_\Theta(Q_{nuevo} \cup \{\langle v_{nuevo}, sello_{V_Q\leftarrow}, sello_{V_Q\rightarrow}\rangle\})$
11. $U_{ligados} = U_{ligados} \cup \{V_U\}$
12. Incrementar $\theta_{sello_{V_Q\leftarrow}(s')}$
13. Repetir (para cada $V_Q \in \overline{Q_{s'}}$) // **Fase 2**
14. Repetir (para cada $V_U \in U$)
15. $\overrightarrow{v}_{nuevo} = (1-\alpha)\overrightarrow{v}_{V_Q} + \alpha\overrightarrow{v}_{V_U}$
16. $V_{nuevo} = \langle \overrightarrow{v}_{nuevo}, sello_{V_Q\leftarrow}, sello_{V_Q\rightarrow}\rangle$
17. Si $sello_{V_U\rightarrow}(s,a) = -1$
18. $sello = generarSello()$
19. $\theta_{sello} = 0$
20. $sello_{V_U\rightarrow}(s,a) = sello$
21. $sello_{V_{nuevo}\leftarrow}(s') = sello$
22. En otro caso
23. $sello_{V_{nuevo}\leftarrow}(s') = sello_{V_U\rightarrow}(s,a)$
24. Incrementar $\theta_{sello_{V_U\rightarrow}(s,a)}$
25. $Q_{nuevo} = ND_\Theta(Q_{nuevo} \cup \{V_{nuevo}\})$
26. Repetir (para cada $V_U \in U$) // **Fase 3**
27. $V_{nuevo} = \langle \overrightarrow{v}_{V_U}, -1, -1\rangle$
28. Si $sello_{V_U\rightarrow}(s,a) = -1$
29. $sello = generarSello()$
30. $\theta_{sello} = 0$
31. $sello_{V_U\rightarrow}(s,a) = sello$
32. $sello_{V_{nuevo}\leftarrow}(s') = sello$
33. $Q_{nuevo} = ND_\Theta(Q_{nuevo} \cup \{V_{nuevo}\})$
34. En otro caso, si $V_U \notin U_{ligados}$ y s' no es final
35. $sello_{V_{nuevo}\leftarrow}(s') = sello_{V_U\rightarrow}(s,a)$
36. $Q_{nuevo} = ND_\Theta(Q_{nuevo} \cup \{V_{nuevo}\})$
37. Devolver Q_{nuevo}

Cuadro 3. Algoritmo de actualización con sellos

Los vectores sello correspondientes a las políticas aprendidas partiendo del estado s son los correspondientes al conjunto $NDU(s)$. Para explotar la información aprendida una vez que se ha realizado el aprendizaje, el agente ha de decidirse por un vector sello $V \in NDU(s)$, y tomar la acción asociada (es decir, la acción a tal que $Q(s,a)$ dio lugar al valor V), que le llevará a un estado s'. Una vez en s', ha de elegir la acción asociada al vector sello W de $NDU(s')$ tal que $sello_{W\rightarrow}(s,a) = sello_{V\leftarrow}(s')$.

Ilustraremos el funcionamiento de PQ-learning mediante el PDM multiobjetivo de la Figura 1. Para cada vector $\overrightarrow{v}$ de los conjuntos Q, utilizaremos la notación $\overrightarrow{v}_{\overrightarrow{v}_a, \overrightarrow{v}_{b,\ldots,f_n}}$, donde $\overrightarrow{v}_a, \overrightarrow{v}_b, \ldots$ son los valores que han actualizado alguna vez a $\overrightarrow{v}$, y $f_n \in \{f_1, f_2, f_3\}$ es la fase de PQ-learning que ha dado lugar a ese valor (véase la tercera columna del Cuadro 2).

Las cinco primeras secuencias dan lugar a los mismos valores que la suma exhaustiva. En la sexta secuencia se perciben cambios significativos entre la suma exhaustiva y la que utiliza sellos: en $Q(s_1, a_{12})$ podemos ver que hay un valor extra que surge de copiar mediante la fase 3 el valor $(-1, 1)$. Esta operación dará lugar a un valor redundante, pero es necesaria para asegurar, en casos más complejos, que se tienen en cuenta todas las combinaciones de secuencias. Estos dos valores de $Q(s_1, a_{12})$ pasan a $Q(s_0, a_0)$ mediante la fase 3, en la siguiente ejecución de la misma secuencia (séptima fila). Por último, en la octava fila, vemos como el conjunto $Q(s_0, a_0)$ no crece de manera descontrolada, como sucede en la suma exhaustiva, gracias a los sellos. El valor $(-2, 1)$ dará lugar a un valor redundante que será eliminado por el operador ND, pues se irá aproximando al vector $(-2, 5, 1, 5)$ cuando el vector que lo actualiza, $(-1, 1) \in Q(s_1, a_{12})$, se actualice con el vector $(-2, 2) \in \overrightarrow{r} + NDU(s_2)$.

3. Resultados

En esta sección analizaremos los resultados arrojados por PQ-learning al aplicarlo a dos problemas de un banco de pruebas propuesto en la literatura de AR multiobjetivo [2]. Tal y como sugieren Vamplew et al [2], el rendimiento se ha medido en base al *hipervolumen* de la frontera aproximada, que es el volumen encerrado entre los puntos de la frontera y un punto de referencia que es dominado por todos los puntos de cualquier frontera aproximada. También mostramos el comportamiento del algoritmo que utiliza la suma exhaustiva de conjuntos sobre estos mismos problemas.

El primer problema considerado es *Deep Sea Treasure* (DST), de dos objetivos[1]. La frontera real para este problema es no convexa y tiene diez puntos y un hipervolumen de 535. La configuración paramétrica ha sido: $\alpha = 0{,}1, \gamma = 1$(ya que el problema es episódico), $\epsilon = 0{,}4, \Theta_{umbral} = 0{,}3$. El segundo problema considerado es *Resource Gathering* (RG), de tres objetivos[2]. Este problema presenta indeterminismo y su frontera real tiene ocho puntos y un hipervolumen de

[1] `http://uob-community.ballarat.edu.au/~pvamplew/MORL.html`

[2] Véase `http://uob-community.ballarat.edu.au/~pvamplew/MORL.html`. Aquí hemos reformulado RG en episodios

1,8299. La configuración paramétrica ha sido: $\alpha = 0{,}1, \gamma = 0{,}9$ (hemos mantenido la tasa de descuento inferior a 1 para dar más valor a las recompensas que son obtenidas antes, y así primar los caminos más cortos), $\epsilon = 0{,}4, \Theta_{umbral} = 0{,}3$. Para los parámetros α, ϵ y Θ_{umbral} se han elegido unos valores prudentes que han funcionado satisfactoriamente. En la Figura 2 se muestra la evolución de la media del hipervolumen (tomada de diez ejecuciones del experimento) de la frontera aproximada durante 70000 episodios de aprendizaje ($\sim$ 25 min.) para DST y 3000 para RG ($\sim$ 2 min.).

El algoritmo Q-learning multiobjetivo con actualización exhaustiva también ha sido aplicado tanto a DST como a RG, con la misma parametrización que PQ-learning (salvo por el parámetro Θ_{umbral}, que en la actualización exhaustiva no entra en juego). En DST, tras una hora únicamente había completado 9000 episodios de aprendizaje, momento en el cual el conjunto $NDU(s_i)$ tiene unos 50 vectores que no representan secuencias legítimas, tal y como veíamos en el ejemplo de la Sección 2.1. En RG, tras una hora únicamente había completado 500 episodios, momento en el cual el conjunto $NDU(s_i)$ tiene unos 70 vectores que no representan secuencias legítimas.

4. Conclusiones

Este artículo describe y analiza PQ-learning, una extensión de Q-learning para problemas de AR multiobjetivo. Como Q-learning, el algoritmo no asume un conocimiento perfecto de la dinámica del entorno. PQ-learning descubre un conjunto de políticas que pretende aproximar a la frontera real del problema. A diferencia de otros trabajos, el algoritmo no requiere información de preferencias, es aplicable a problemas con fronteras de Pareto no convexas y permite recuperar las secuencias de acciones correspondientes a distintas políticas Pareto-óptimas.

Hasta donde sabemos, PQ-learning es el primer algoritmo destinado a resolver procesos de decisión markovianos multiobjetivo sin ningún tipo de información del modelo ni de preferencias. Por esta razón no ha sido comparado con otros algoritmos de la literatura. El algoritmo ha sido aplicado a dos problemas

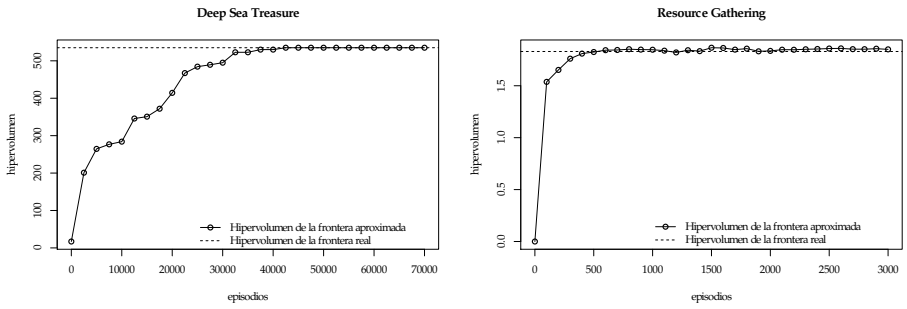


Figura 2. Resultados al aplicar PQ-learning sobre DST y RG

pertenecientes a un banco de pruebas propuesto en la literatura de AR multiobjetivo. PQ-learning ha sido comparado con una extensión de Q-learning que emplea técnicas habituales en programación dinámica multiobjetivo, resultando este último inviable para los problemas analizados.

La continuación natural del trabajo aquí expuesto es la realización de un estudio teórico completo del algoritmo, de la influencia de diferentes configuraciones de parámetros sobre el rendimiento y de escalabilidad, así como la aplicación del algoritmo a problemas reales. En concreto, los autores han aplicado el algoritmo Q-learning a problemas de diseño computacional en trabajos previos [10,11,12] que potencialmente podrán beneficiarse de PQ-learning.

Agradecimientos. Este trabajo está parcialmente financiado por el Plan Nacional de I+D+I, proyecto TIN2009-14179 (Gobierno de España, Ministerio de Ciencia e Innovación) y por la Universidad de Málaga, Campus de Excelencia Internacional Andalucía Tech. Manuela Ruiz-Montiel disfruta de una beca FPU (Gobierno de España, Ministerio de Educación).

Referencias

1. Watkins, C.J.: Learning from delayed rewards. PhD thesis, University of Cambridge (1989)
2. Vamplew, P., Dazeley, R., Berry, A., Issabekov, R., Dekker, E.: Empirical evaluation methods for multiobjective reinforcement learning algorithms. Mach. Learn. **84**(1-2) (2011) 51–80
3. Gábor, Z., Kalmár, Z., Szepesvári, C.: Multi-criteria reinforcement learning. ICML '98, Morgan Kaufmann Publishers Inc. (1998) 197–205
4. Natarajan, S., Tadepalli, P.: Dynamic preferences in multi-criteria reinforcement learning. ICML '05, ACM (2005) 601–608
5. White, D.J.: Multi-objective infinite-horizon discounted markov decision processes. Journal of Mathematical Analysis and Applications **89** (1982)
6. Barrett, L., Narayanan, S.: Learning all optimal policies with multiple criteria. ICML '08, ACM (2008) 41–47
7. Issabekov, R., Vamplew, P.: An empirical comparison of two common multiobjective reinforcement learning algorithms. In: AI 2012: Advances in Artificial Intelligence. Springer (2012) 626–636
8. Vamplew, P., Yearwood, J., Dazeley, R., Berry, A.: On the limitations of scalarisation for multi-objective reinforcement learning of pareto fronts. In: AI 2008: Advances in Artificial Intelligence. Springer (2008) 372–378
9. Sutton, R.S., Barto, A.G.: Reinforcement learning: an introduction. MIT Press, Cambridge, Ma. (1998)
10. Ruiz-Montiel, M., Boned, J., Gavilanes, J., Jiménez, E., Mandow, L., Pérez-de-la Cruz, J.L.: Proyecto arquitectónico mediante gramáticas de formas sencillas y aprendizaje. In: Avances en Inteligencia Artificial: Actas de CAEPIA 11, 1. (2011)
11. Ruiz-Montiel, M., Mandow, L., Pérez-de-la Cruz, J.L., Gavilanes, J.: Shapes, grammars, constraints and policies. In: CEUR: SHAPES 1.0, 812. (2011)
12. Ruiz-Montiel, M., Boned, J., Gavilanes, J., Jiménez, E., Mandow, L., de-la Cruz, J.L.P.: Design with shape grammars and reinforcement learning. Advanced Engineering Informatics **27**(2) (2013) 230 – 245

Tri-Training con sobremuestreo para aprendizaje semi-supervisado.

Isaac Triguero[1], Salvador García[2], and Francisco Herrera[1]

[1] Dept. de Ciencias de la Computación e Inteligencia Artificial, CITIC-UGR. Universidad de Granada. 18071, Granada, España
triguero@decsai.ugr.es, herrera@decsai.ugr.es
[2] Dept. de Informática. Universidad de Jaén. 23071, Jaén, España
sglopez@ujaen.es

Resumen Las técnicas de aprendizaje semi-supervisado nacen para abordar aquellos problemas en los que se dispone de pocos datos etiquetados y un volumen relativamente alto de no etiquetados. En este trabajo se propone un nuevo algoritmo, basado en la filosofía del co-entrenamiento, que usa una técnica de sobremuestreo en la fase de inicialización. Concretamente, se extiende el modelo de *Tri-Training* presentado por Zhou y Li (2005), generando nuevos ejemplos sintéticos de clase conocida para ofrecer mayor diversidad a cada clasificador. Se realiza un estudio completo del rendimiento de este método contrastando los resultados mediante técnicas estadísticas que avalan la utilidad de esta propuesta.

Palabras clave: Tri-training, sobremuestreo, co-entrenamiento, aprendizaje semi-supervisado

1. Introducción

En muchos ámbitos, tales como la bioinformática o la minería web, es mucho más fácil obtener datos no etiquetados debido a que requieren un menor esfuerzo, experiencia y tiempo. En este contexto, los métodos de aprendizaje supervisado están limitados por los datos de los que sí se conoce la clase. Sin embargo, el aprendizaje semi-supervisado (*Semi-supervised Learning*, SSL) nace como el paradigma que se encarga de diseñar modelos en presencia de ambos tipos de datos. En esencia, estos métodos usan los datos no etiquetados para modificar las hipótesis aprendidas de los datos etiquetados.

El paradigma del SSL puede considerarse una extensión de los aprendizajes supervisado y no supervisado mediante el uso de información adicional típica del otro paradigma de aprendizaje. En la literatura, el SSL se divide en dos enfoques: clasificación semi-supervisada (*Semi-supervised Classification*, SSC) [6] y agrupamiento semi-supervisado [14]. El primero se centra en mejorar la clasificación supervisada, minimizando los errores en los datos etiquetados, siendo al mismo tiempo compatible con la distribución de los datos no etiquetados. El segundo trata de obtener mejores agrupaciones (*clusters*) que las obtenidas solo con datos no etiquetados. Este trabajo se centra en SSC.

Es importante diferenciar el paradigma del SSL de un tema estrechamente ligado: el aprendizaje activo (*active learning*) [9]. Esta metodología trata de discernir los ejemplos más relevantes dentro del conjunto de los no etiquetados. Esos ejemplos son etiquetados posteriormente por un experto [13].

Los métodos de SSC tienen dos modos de funcionamiento: conocidos como transductivo e inductivo. El aprendizaje transductivo se refiere al problema de predecir la clase de los datos no etiquetados de los que se dispone durante el proceso de entrenamiento. El aprendizaje inductivo considera que los datos etiquetados y no etiquedos forman el conjunto de entrenamiento, su objetivo es predecir casos aún no vistos. En esta contribución abordamos ambos enfoques.

En la última década se han propuesto numerosos métodos de SSC, como por ejemplo máquinas de vectores de soporte (S3VMs) [12] o métodos basados en grafos [3]. Estos métodos asumen diferentes relaciones entre los datos etiquetados y no etiquetados. Sin estas hipótesis la SSC carece de utilidad. Un enfoque destacado es el algoritmo de *Tri-Training* propuesto en [18], basado en un modelo de co-entrenamiento (*Co-Training*) [4], que fue desarrollado para solucionar las principales debilidades del enfoque *Co-Training*: las características pueden ser divididas en dos bloques y cada uno de ellos es suficiente para aprender, además son condicionalmente independiente el uno del otro respecto de la clase.

El algoritmo de *Tri-Training* no requiere esta división a nivel de características. En este sentido, este modelo no hace ninguna hipótesis sobre los datos de entrada, aceptando que sus propias predicciones tienden a ser correctas. Este algoritmo extiende al modelo de *Co-Training* estándar haciendo uso de tres clasificadores que realizan un aprendizaje mutuo. La hipótesis inicial de cada clasificador se aprende sobre un conjunto muestreado [5] del conjunto etiquetado. Este procedimiento de inicialización es la mayor debilidad de este método ya que con el número de datos etiquetados tan reducido del que se dispone, la diversidad obtenida por el muestreo utilizado es limitada.

En este trabajo se propone un mecanismo de inicialización que permita añadir diversidad al algoritmo de *Tri-Training*. Para ello, se diseña un algoritmo de sobremuestreo basado en SMOTE [7] que genera nuevos ejemplos artificiales a partir del conjunto de etiquetados. Cada clasificador es inicializado con el conjunto etiquetado y algunos de esos ejemplos artificiales. La propuesta se denomina "*Tri-Training* con sobremuestreo"(TriSM).

En el estudio experimental, se analizan las capacidades transductivas e inductivas de la propuesta sobre un total de 20 conjuntos de datos con un número reducido de ejemplos etiquetados, incluyendo un análisis basado en tests estadísticos no paramétricos [11].

El resto del trabajo se organiza como sigue: La Sección 2 define el problema de la SSC y resume el trabajo previo realizado en este ámbito. La Sección 3 describe en profundidad el modelo propuesto. La Sección 4 define el estudio experimental y analiza los resultados. Finalmente, la Sección 5 concluye el trabajo.

2. Preliminares

En esta sección se define el problema del aprendizaje semi-supervisado (Sección 2.1) y se describe brevemente en qué consisten los enfoques más relevantes de la literatura (Sección 2.2).

2.1. Clasificación semi-supervisada

El aprendizaje semi-supervisado puede ser definido como sigue: Sea $\mathbf{x}_p$ una instancia donde $\mathbf{x}_p = (\mathbf{x}_{p1}, \mathbf{x}_{p2}, ..., \mathbf{x}_{pD}, \omega)$, con $\mathbf{x}_p$ perteneciente a la clase ω y a un espacio D-dimensional donde $\mathbf{x}_{pi}$ es el valor de la i-ésima característica del ejemplo p-ésimo. Se asume que existe un conjunto de ejemplos etiquetados L que consisten en $\mathbf{n}$ ejemplos $\mathbf{x}_p$ con clase ω conocida. Además, existe un conjunto de datos no etiquetado U que está formado por $\mathbf{m}$ instancias $\mathbf{x}_q$ con ω desconocida, siendo $\mathbf{m} >> \mathbf{n}$. La unión $L \cup U$ forma el conjunto de entrenamiento TR. El propósito del aprendizaje semi-supervisado es el de obtener una hipótesis robusta usando TR en lugar de sólo L, que puede ser aplicado en dos formas distintas: aprendizaje transductivo e inductivo.

El aprendizaje transductivo se describe como la aplicación de una técnica de SSC para etiquetar las $\mathbf{m}$ instancias $\mathbf{x}_q$ de U con su clase correspondiente. La clase asignada debería representar la distribución de clases eficientemente, basándose en la distribución de los datos etiquetados y no etiquetados.

Siendo TS un conjunto de test compuesto por $\mathbf{t}$ instancias $\mathbf{x}_r$ no usadas en la etapa de entrenamiento del método y de clase ω desconocida, la fase inductiva consiste en clasificar los ejemplos de TS en base a lo aprendido con TR.

2.2. Trabajo previo

En la literatura se han propuesto diferentes enfoques para abordar el aprendizaje semi-supervisado [19]. Éstos normalmente realizan diferentes hipótesis sobre la relación existente entre las distribuciones de datos etiquetados y no etiquetados. Los modelos generativos [10] aprenden un modelo probabilístico que asume que los datos siguen un determinado modelo paramétrico. El proceso de auto-entrenamiento (*Self-Training*)[17] consiste en un clasificador que es entrenado inicialmente con L para clasificar ejemplos no etiquetados. A continuación dicho clasificador es re-entrenado con sus predicciones más confiables, aumentando el tamaño de L. Este modelo asume que sus propias predicciones son correctas. La metodología de *Co-Training* estándar [4] asume que el espacio de características es divisible en dos grupos de atributos condicionalmente independientes, y que cada grupo de atributos permite predecir las clases perfectamente. Cada clasificador es entrenado con cada grupo de características y después cada clasificador enseña al otro los ejemplos para los cuales tiene más confianza en el etiquetado realizado. El aprendizaje multi-vista (*multiview*) [18] se considera una extensión del modelo de *Co-Training*, sin necesidad de dividir el espacio de características o de aplicar una enseñanza mutua entre clasificadores. En lugar de ello, se centran en aplicar varios clasificadores combinando sus predicciones.

Existen otros algoritmos como las máquinas de vectores de soporte para inferencia transductiva [12] que asumen que las clases están bien separadas. Además, el aprendizaje semi-supervisado también puede ser interpretado como un grafo [3], donde dos instancias están conectadas si sus etiquetas son la misma. En este modelo, la construcción del grafo determina el comportamiento de este tipo de algoritmos.

3. TriSM: *Tri-training* con sobremuestreo

En esta sección se describe el modelo TriSM. Primero se introduce el modelo de *Tri-Training* (Sección 3.1) en el que se basa TriSM y después se explica la motivación para usar sobremuestreo y cómo funciona (Sección 3.2).

3.1. Tri-training

El algoritmo de *Tri-Training* fue propuesto en [18] para mejorar el modelo de *Co-Training* estándar, de modo que no se requiera división natural de las características en dos subconjuntos. Este requerimiento no es aplicable en muchos problemas reales, como se analizó en [8].

Por esta razón, el enfoque de *Tri-Training* aborda este problema mediante el uso de tres clasificadores h_1, h_2 y h_3, que usan el mismo algoritmo de aprendizaje supervisado, como por ejemplo un árbol de decisión [15]. De este modo, la diversidad entre los clasificadores solo se puede conseguir manipulando el conjunto original de datos etiquetados L. Concretamente, el proceso de inicialización consiste de un muestreo con reemplazo de L [5].

A continuación, el modelo entra en un bucle que determina los ejemplos no etiquetados para los que se obtiene una mayor confianza en la clasificación y así aumentar el tamaño del conjunto L. En cada iteración, cada ejemplo de U tiene la oportunidad de ser etiquetado para cada clasificador h_i. Un ejemplo no etiquetado x_q puede ser etiquetado para h_i siempre que los otros dos clasificadores estén de acuerdo en la clase de este ejemplo. Esto implica que x_q será usado para modificar el modelo aprendido por h_i.

Bajo este esquema es posible que dos clasificadores (por ejemplo, h_2, h_3) predigan una clase incorrecta para el ejemplo no etiquetado, añadiendo ruido en la hipótesis aprendida por el otro clasificador (h_1). Para aliviar este problema, el modelo incorpora un mecanismo para compensar la influencia negativa de los ejemplos mal etiquetados, añadiendo una cantidad suficiente de datos etiquetados.

El bucle se repite hasta que el modelo aprendido por los tres clasificadores no varía. La clasificación final se realiza mediante una regla de mayoría de los clasificadores entrenados.

3.2. TriSM

El modelo de *Tri-Training* es, en resumen, un multiclasificador en el que todos los clasificadores usan el mismo tipo de algoritmo de aprendizaje sin dividir

el espacio de características. Por lo tanto, la diversidad entre los clasificadores tiene que ser buscada por otras vías, limitadas al uso de los datos de entrada. De hecho, el algoritmo de *Tri-Training* usa un esquema de inicialización basado en muestreo con reemplazo como mecanismo para introducir diversidad en los modelos aprendidos por cada clasificador.

Sin embargo, en muchos problemas reales que tratan de abordar el paradigma de SSL, el número de ejemplos etiquetados tiende a ser muy pequeño. Este modelo de muestreo crea nuevos conjuntos etiquetados para cada clasificador h_i modificando el conjunto L. En general, esta operación produce distintos conjuntos de datos etiquetados respecto del original, pero las diferencias no son significativas cuando el conjunto etiquetado es muy pequeño y/o existen algunos datos erróneos (*outliers*) en él. Este hecho puede provocar que no se represente de forma precisa el dominio del problema, condicionando a los clasificadores.

Figura 1: Método de sobremuestreo

Entrada: Conjunto $Original$, Ratio de sobremuestreo $r\,\%$ Número de vecinos k.
salida: Conjunto $Sobremuestreado$
$Sobremuestreado = \emptyset$
for $i = 1$ a $NumeroDeClases$ **do**
  $PC[i]$ = deLaClase ($Original$, i)
  $paraAnadir[i] = (r/100) \cdot \#PC$
end for
if $r < 100$ **then**
  Barajar el conjunto $Original$.
  $PorPrototipo = 1$
else
  $PorPrototipo = (int)r/100$
end if
for $i = 1$ a $NumeroDeClases$ **do**
  for $j = 1$ a $\#paraAnadir[i]$ **do**
    $Generados = 0$
    repeat
      $vecinos[1..k]$ = Calcular los k vecinos más cercanos de $PC[j]$
      nn = Número aleatorio entre 1 y k
      $Ejemplo = PC[j]$
      $Cercano = Original[vecinos[nn]]$
      for $m = 1$ a $NumeroDeAtributos$ **do**
        $dif = Cercano[m] - Ejemplo[m]$
        gap = Número aleatorio entre 0 y 1.
        $Sintetico[m] = Ejemplo[m] + gap * dif$
      end for
      $Sobremuestreado = Sobremuestreado \cup Sintetico$
      $Generado++$
    until $Generado < PorPrototipo$
  end for
end for

Para resolver este problema, diseñamos un nuevo modo de introducir diversidad en la inicialización de *Tri-Training* mediante sobremuestreo. El objetivo de este método es reforzar las regiones de decisión de todas las clases. Este proceso introduce nuevos ejemplos sintéticos etiquetados, generados mediante ciertas operaciones sobre los datos originales, que esperan mejorar las capacidades de generalización.

Al comienzo del proceso de *Tri-Training*, el conjunto de datos etiquetado es usado al completo para la etapa de sobremuestreo. La Figura 1 describe la técnica de sobremuestreo usada. Este método se basa en el algoritmo SMOTE propuesto en [7] que fue originalmente diseñado para problema no balanceados y se limita a sobremuestrear la clase minoritaria. En nuestra propuesta extendemos la idea de SMOTE para generar ejemplos de todas las clases. Cada clase estará proporcionalmente sobremuestreada usando sus ejemplos e introduciendo ejemplos sintéticos a lo largo de los segmentos que los unen a cualquiera de sus k vecinos más cercanos de la misma clase. Se generan datos artificiales como la diferencia entre un ejemplo existente y uno de sus vecinos más cercano, elegido aleatoriamente. Después, la diferencia se escala mediante un número aleatorio en el rango [0,1], y se añade al ejemplo considerado. De esta forma, se está eligiendo un punto entre dos ejemplos. Nótese que la clase del ejemplo generado es la misma de la muestra considerada para sobremuestrear.

Al final de esta etapa se tiene un conjunto de ejemplos artificiales que serán usados en el proceso de *Tri-Training*. En el algoritmo TriSM, la fase de inicialización consiste en que los tres clasificadores usen los datos etiquetados más algunos de los datos generados aleatoriamente. En particular, los datos generados son seleccionados aleatoriamente sin reemplazo, usando el mismo número de ejemplos para cada clasificador. Después de esto, el resto del proceso del método procede del mismo modo.

4. Estudio experimental y análisis de resultados

Esta sección presenta el estudio experimental llevado a cabo. Se muestran las principales características de los problemas utilizados (Sección 4.1), los algoritmos de comparación y sus respectivos parámetros (Sección 4.2). Finalmente se realiza un análisis de los resultados obtenidos (Sección 4.3).

4.1. Conjuntos de datos

Para caracterizar el rendimiento del modelo propuesto se han empleado un total de 20 conjuntos de datos representando problemas de clasificación. Han sido tomados del repositorio *KEEL-dataset*[3][2]. El Cuadro 1 resume las características de los datos seleccionados, mostrando su número de instancias (#In.), número de atributos (#Ats.) y número de clases (#Cl.).

Todos los conjuntos han sido particionados empleando la técnica de validación cruzada de 10 campos. Cada partición de entrenamiento se divide

[3] `http://www.keel.es/datasets.php`

en datos etiquetados y no etiquetados. Usando las recomendaciones de [16], el proceso de división no mantiene la proporción de clases en ambos conjuntos. Por lo tanto, se usa una selección aleatoria de ejemplos que serán anotados como instancias etiquetadas, y para el resto de ejemplos la clase será eliminada. Aseguramos que cada clase tiene al menos un representante.

Esta experimentación toma un ratio del 10 % de etiquetados que produce un número muy reducido de datos etiquetados.

Cuadro 1: Conjuntos de datos utilizados

Conjunto	**#In.**	**#Ats.**	**#Cl.**	**Conjunto**	**#In.**	**#Ats.**	**#Cl.**
autos	205	25	6	pima	768	8	2
bupa	345	6	2	ring	7.400	20	2
cleveland	297	13	5	segment	2.310	19	7
ecoli	336	7	8	spambase	4.597	55	2
flare-solar	1.066	9	2	spectheart	267	44	2
glass	214	9	7	tic-tac-toe	958	9	2
iris	150	4	3	twonorm	7.400	20	2
lymphography	148	18	4	wisconsin	683	9	2
magic	19.020	10	2	yeast	1.484	8	10
phoneme	5.404	5	2	zoo	101	17	7

4.2. Algoritmos y parámetros utilizados

Además del modelo *Tri-Training* original, se han empleado dos métodos clásicos de aprendizaje semi-supervisado: *Self-Training* [17] y *Co-Training* [4] para comparar.

La configuración de parámetros es común para todos los problemas y fue elegida acorde a la recomendación de los autores. El Cuadro 2 presenta los valores utilizados. En esta tabla, el ratio de sobremuestreo y número de vecinos del algoritmo TriSM se corresponden con los mejores valores encontrados después de realizar numerosos experimentos previos. Además, dado que el proceso de inicialización de *Tri-Training* y TriSM es estocástico, estos métodos han sido ejecutados tres veces por partición.

Todos estos métodos permiten usar distintos algoritmos de aprendizaje como clasificador base. Se usarán dos clasificadores bien conocidos: KNN [1] y C4.5 [15]. A continuación se describe brevemente cómo se calcula la confianza en la predicción para los esquemas de *Self-Training* y *Co-Training*:

- KNN: La confianza en la clasificación se aproxima como la distancia al conjunto etiquetado.
- C4.5: La confianza se obtiene de la precisión de la hoja que realiza la predicción. Esta precisión es el porcentaje de ejemplos de entrenamiento bien etiquetados del total cubiertos.

4.3. Análisis de resultados

En esta subsección se analizan los resultados obtenidos. El estudio se divide acorde al clasificador base utilizado. En la Figura 2 se muestran los resultados

Cuadro 2: Parámetros empleados

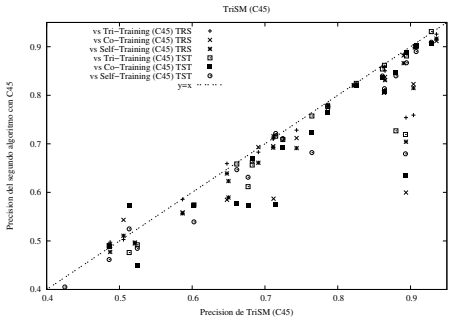
Algoritmo	*Parámetros*
KNN	Número de vecinos = 3, Distancia Euclidea
C4.5	Confianza: $c = 0{,}25$, Poda activada, Min. items por hoja =2
Self-Training	$MAX_ITER = 40$
Co-Training	$MAX_ITER = 40$, Conjunto inicial de no etiquetados=75
Tri-Training	Sin parámetros especificados
TriSM	Ratio de sobremuestreo = 400 %, Número de vecinos = 5

medios obtenidos en cada conjunto de datos usando C4.5 (Figura 2a) y KNN (Figura 2b). En estas figuras, cada punto compara al método TriSM con un segundo algoritmo en un solo conjunto de datos. La posición X del punto indica la precisión de TriSM en las fase transductiva e inductiva (indicado como TRS y TST, respectivamente, en las figuras), y en el eje Y se muestra la precisión del algoritmo de comparación. Así, los puntos que están bajo la recta $y = x$ corresponden a los conjuntos de datos donde TriSM mejora al segundo algoritmo.

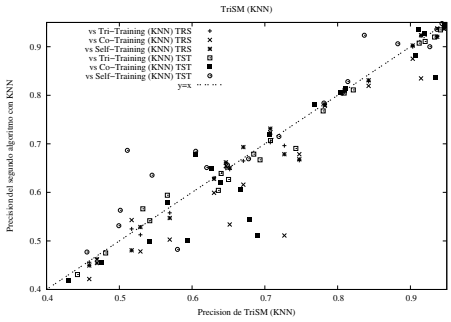
El Cuadro 3 resume los resultados medios de precisión (Acc.) y su correspondiente desviación típica (SD.) en los 20 conjuntos de datos, tanto en la fase transductiva como en la inductiva. El clasificador base se especifica entre paréntesis. Los mejores resultados se han destacado en negrita.

Figura 2: Precisión transductiva e inductiva

(a) C4.5 como clasificador base



(b) KNN como clasificador base



Considerar solo resultados medios puede llevarnos a conclusiones erróneas. Por ello, se van a emplear test estadísticos no paramétricos para constrastar los resultados. Se usa el test de Friedman como método para detectar diferencias significativas entre los resultados transductivos e inductivos y el método de Holm como test *post-hoc* para caracterizar las diferencias encontradas [11].

Los Cuadros 4 y 5 resumen el estudio estadístico con C4.5 y KNN, respectivamente. Los rangos de Friedman representan la efectividad de cada modelo. El p-valor obtenido con este test se presenta la última fila de las tablas, mostrando que existen diferencias significativas entre los métodos (*p*-valor <0,001). Además, el mejor (menor) rango obtenido se marca como '*' que muestra el método que ha sido tomado como método de control por el test *post-*

hoc. Junto al rango, se muestra el p-valor obtenido por el método de Holm. Los p-valores destacados en negrita hacen referencia a aquellos métodos que son mejorados significativamente por el método de control con un nivel de confianza $\alpha = 0,05$.

Cuadro 3: Resultados medios en las fases transductiva e inductiva

	Transductivo Acc.		**Inductivo Acc.**	
Algoritmo	**Acc.**	**SD.**	**Acc.**	**SD.**
TriSM (C45)	**0,7269**	0,0310	**0,7309**	0,0600
Tri-Training (C45)	0,7039	0,0337	0,7003	0,0540
Co-Training (C45)	0,6845	0,0408	0,6840	0,0623
Self-Training (C45)	0,6940	0,0347	0,6955	0,0606
TriSM (KNN)	0,7208	0,0262	0,7193	0,0532
Tri-Training (KNN)	0,7113	0,0278	0,7111	0,0535
Co-Training (KNN)	0,6832	0,0340	0,6854	0,0559
Self-Training (KNN)	0,7117	0,0298	0,7090	0,0566

Cuadro 4: Rangos de Friedman y p-valores ajustados para C4.5

	Transductiva		**Inductiva**	
Algoritmo	Rango	p-valor de Holm	Rango	p-valor de Holm
TriSM (C45)	1,35*	–	1,35*	–
Tri-Training (C45)	2,45	**0,0070**	2,48	**0,0058**
Co-Training (C45)	3,10	**0,0000**	3,25	**0,0000**
Self-Training (C45)	3,10	**0,0000**	2,93	**0,0002**

$p-$valor de Friedman en la fase transductiva = 0,000019
$p-$valor de Friedman en la fase inductiva = 0,000017

Cuadro 5: Rangos de Friedman y p-valores ajustados para KNN

	Transductiva		**Inductiva**	
Algoritmo	Rango	p-valor de Holm	Rango	p-valor de Holm
TriSM (NN)	1,65*	–	1,70*	–
Tri-Training (NN)	2,70	**0,0202**	2,85	**0,0097**
Co-Training (KNN)	3,20	**0,0004**	2,95	**0,0066**
Self-Training (KNN)	2,45	**0,0500**	2,50	**0,0500**

$p-$valor de Friedman en la fase transductiva = 0,001766
$p-$valor de Friedman en la fase inductiva = 0,008970

A partir de estos resultados, se pueden extraer las siguientes conclusiones:

- El método propuesto en conjunto con C4.5 obtiene los mejores resultados de precisión media transductiva e inductiva. La Figura 2a muestra claramente este hecho ya que la mayor parte de los puntos están bajo la recta $y = x$. El test estadístico confirma que TriSM (C45) mejora al resto de algoritmos.
- Atendiendo al clasificador usado, el test de Holm indica que las diferencias de TriSM sobre el resto de métodos es siempre significativa ($\alpha = 0,05$). No obstante, cabe mencionar que en la Figura 2, se observa como TriSM (C45) consigue un mejor rendimiento que TriSM (KNN).
- En la comparación de TriSM con el método original de *Tri-Training*, se aprecia la utilidad del esquema de sobremuestreo propuesto. Encontramos que por ejemplo, el esquema TriSM (C45) mejora al Tri-Training (C45) en 19 de 20 en aprendizaje transductivo y en 17 de 20 en la fase inductiva.

5. Conclusiones

En este trabajo se ha presentando un nuevo método de SSC llamado TriSM que es una extensión del algoritmo de *Tri-Training*. Se ha incorporado un proceso de sobremuestreo que genera prototipos sintéticos con el fin de añadir diversidad al método subyacente. El estudio experimental muestra que TriSM es un método adecuado para abordar problemas con pocos datos etiquetados. El test estadístico muestra que las conclusiones obtenidas son significativas. Como trabajo futuro podría ser interesante estudiar el uso de datos sintéticos en otros enfoques semi-supervisados.

Referencias

1. Aha, David W., Kibler, Dennis, and Albert, Marc K. Instance-based learning algorithms. *Machine Learning*, 6(1):37–66, 1991.
2. Alcalá-Fdez, Jesús, Fernandez, Alberto, Luengo, Julián, Derrac, Joaquín., García, Salvador, Sánchez, L., and Herrera, F. KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing*, 17(2-3):255–277, 2011.
3. Blum, Avrim and Chawla, Shuchi. Learning from labeled and unlabeled data using graph mincuts. In *Proceedings of the 18th International Conference on Machine Learning*, pp. 19–26, 2001.
4. Blum, Avrim and Mitchell, Tom. Combining labeled and unlabeled data with Co-Training. In *Proceedings of the Annual ACM Conference on Computational Learning Theory*, pp. 92–100, 1998.
5. Breiman, Leo. Bagging predictors. *Machine Learning*, 24:123–140, August 1996. ISSN 0885-6125.
6. Chapelle, Olivier, Schölkopf, Bernhard, and Zien, Alexander. *Semi-Supervised Learning*. The MIT Press, first edition, 2006.
7. Chawla, Nitesh V., Bowyer, Kevin W., Hall, Lawrence O., and Kegelmeyer, W. Philip. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16: 321–357, 2002.
8. Du, Jun, Ling, Charles X., and Zhou, Zhi-Hua. When does co-training work in real data? *IEEE Transactions on Knowledge and Data Engineering*, 23(5):788–799, 2010.
9. Freund, Yoav, Seung, H. Sebastian, Shamir, Eli, and Tishby, Naftali. Selective sampling using the query by committee algorithm. *Machine Learning*, 28:133–168, 1997.
10. Fujino, Akinori, Ueda, Naonori., and Saito, Kazumi. Semisupervised learning for a hybrid generative/discriminative classifier based on the maximum entropy principle. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3):424–437, 2008.
11. García, Salvador and Herrera, Francisco. An extension on "Statistical comparisons of classifiers over multiple data sets"for all pairwise comparisons. *Journal of Machine Learning Research*, 9:2677–2694, 2008.
12. Joachims, Thorsten. Transductive inference for text classification using support vector machines. In *Proceedings of the 16th International Conference on Machine Learning*, pp. 200–209. Morgan Kaufmann, 1999.
13. Muslea, Ion, Minton, Steve, and Knoblock, Craig. Selective sampling with redundant views. In *Proceedings of the 17th National Conference on Artificial Intelligence and 20th Conference on Innovative Applications of Artificial Intelligence*, pp. 621–626. AAAI Press, 2000.
14. Pedrycz, Witold. Algorithms of fuzzy clustering with partial supervision. *Pattern Recognition Letters*, 3:13–20, 1985.
15. Quinlan, J. Ross. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers, San Francisco, CA, USA, 1993. ISBN 1-55860-238-0.
16. Wang, Yu, Xu, Xiaoyan, Zhao, Haifeng, and Hua, Zhongsheng. Semi-supervised learning based on nearest neighbor rule and cut edges. *Knowledge-Based Systems*, 23(6):547–554, 2010.
17. Yarowsky, David. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pp. 189–196, 1995.
18. Zhou, Zhi-Hua and Li, Ming. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge and Data Engineering*, 17:1529–1541, 2005. ISSN 1041-4347.
19. Zhu, Xiaojin. *Semi-supervised learning literature survey. Technical report, Computer Sciences* University of Wisconsin-Madison, 2005
20. Zhu, Xiaojin and Goldberg, Andrew B. *Introduction to Semi-Supervised Learning*. Morgan and Claypool, first edition, 2009.

17-20 sept 2013
MULTICONFERENCIA
CAEPIA'13
Madrid

IX Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados [MAEB 2013]

Editores asociados:

José Manuel Colmenar
Abraham Duarte
Eduardo G. Pardo
José Ignacio Hidalgo
Juan Lanchares
Juan José Pantrigo
Antonio S. Montemayor

Simposio organizado por

Prefacio

Las Metaheurísticas, los Algoritmos Evolutivos y los Algoritmos Bioinspirados se encuentran entre los métodos más relevantes y exitosos para resolver un conjunto enorme de problemas complejos y computacionalmente muy costosos de interés teórico y práctico que aparecen en diversas áreas de la ciencia y la ingeniería. Estos métodos se pueden describir como entornos algorítmicos que requieren modificaciones relativamente pequeñas para ser adaptados a problemas concretos. Algunos ejemplos de este tipo de métodos son (manteniendo su nombre original en inglés): Simulated Annealing, Tabu Search, Multi-start Methods, Iterated Local Search, Variable Neighbourhood Search, GRASP, Memetic Algorithms, Scatter Search, Evolutionary Algorithms, o Ant Colony Optimization.

La investigación en Metaheurísticas, Algoritmos Evolutivos y Algoritmos Bioinspirados se ha convertido en un campo de investigación de marcado carácter interdisciplinar, en la que confluyen áreas de conocimiento como la Inteligencia Artificial, la Ciencia de los Computadores, o la Investigación Operativa, entre otras. Estas técnicas han recibido una enorme atención por parte de la comunidad científica, hecho que se aprecia si se considera la gran cantidad de trabajos publicados en revistas de alto impacto, como por ejemplo, IEEE Trans. on Evolutionary Computation, Evolutionary Computation o Journal of Heuristic; cientos de libros editados por las editoriales más prestigiosas (Springer, Kluwer o Taylor&Fracis) con una temática centrada en este tipo de técnicas; o las conferencias temáticas como MIC (Metaheuristic International Conference). IEEE CEC (IEEE Conference on Evolutionary Computation), GECCO (Genetic and Evolutionary Computation Conference) o Evostar.

El congreso MAEB se ha convertido, por derecho propio, en el foro de encuentro, discusión y transferencia de conocimiento entre investigadores de habla hispana del campo de las Metaheurísticas, los Algoritmos Evolutivos y Bioinspirados. Este volumen contiene los trabajos aceptados para su presentación oral en el IX Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB 2013), celebrado en Madrid del 17 al 20 de Septiembre de 2013. No cabe duda de que MAEB es un congreso especial, que ha sabido crecer en estos años, a pesar de las diferentes vicisitudes en que se ha visto envuelto, como la variabilidad en la periodicidad de celebración, su inclusión o no en el CEDI, su celebración como congreso independiente o confederado con otras conferencias (como ocurre en esta edición con la Multiconferencia CAEPIA) o el hecho de no contar con el apoyo de ninguna asociación científica. El éxito de MAEB refleja la importancia de este campo de investigación en la comunidad científica española, pero también el hecho de que MAEB no es sólo una reunión de expertos, sino también una reunión de amigos, que se reúne cada 18 meses (aproximadamente) para intercambiar avances, experiencias, etc.

Este buen funcionamiento se ha puesto de nuevo de manifiesto si se observa el nivel de compromiso demostrado ante la llamada a la participación de MAEB 2013. Incluso en una situación tan difícil para la Ciencia en España (en particular) y en el mundo (en general) como la actual, hemos recibido la excelente cifra de 90 trabajos, de los cuales, 80 han sido seleccionados para ser presentados oralmente, tras un proceso de revisión anónimo. Además, en esta edición hemos contado como es habitual, con un excelente conferenciante invitado, al que queremos agradecer su predisposición a colaborar con MAEB 2013 desde el primer momento: Nenad Mladenovic. El resumen de su conferencia puede encontrarse en este volumen. Otra de las actividades destacadas de este MAEB 2013 es la exposición de pintura sobre Algoritmos Evolutivos Desconectados, presentada por la Universidad de Extremadura. Se trata de una exposición de 50 obras de pintura enmarcados dentro de un ambicioso proyecto de investigación dirigido por el Profesor Francisco Fernández de Vega, que ya ha recibido el reconocimiento internacional en conferencias como GECCO (best paper award) o CEC.

En esta edición se convocarán de nuevo tres premios a trabajos presentados en MAEB: premio al mejor artículo metodológico, premio para el mejor artículo aplicado (o de aplicación a problemas reales) y, por último, premio al mejor trabajo pre-doctoral. Los dos primeros premios están patrocinados por la Asociación Española para la Inteligencia Artificial (AEPIA), mientras que el último está patrocinado por Granada Excellence Network of Innovation Laboratories GENIL (genil.ugr.es). Como en la pasada edición, será la audiencia de MAEB la que elegirá los trabajos merecedores de este premio de entre un grupo de finalistas.

Nos gustaría expresar también nuestro agradecimiento al Comité Director de MAEB, por haber depositado su confianza en nosotros para la organización de esta edición y por su ayuda siempre que la hemos necesitado; a los miembros del Comité de Programa, por su diligencia a la hora de difundir la convocatoria y revisar los artículos asignados; y a los compañeros de nuestros respectivos grupos de investigación, sin cuyo esfuerzo como miembros del Comité Organizador de MAEB 2013, esta aventura no habría sido posible.

No podía faltar en estas actas un sentido y sincero homenaje al profesor Juan Manuel Sánchez Pérez, Catedrático de Universidad, que lo fue de la Universidad de Extremadura y uno de los promotores del primer MAEB que se celebró en Mérida en 2002. Animamos a todos los lectores a leer el obituario que incluimos en estas actas con todo nuestro aprecio y respeto.

Madrid, Septiembre 2013

Abraham Duarte
J. Ignacio Hidalgo

Organización

Comité de Dirección

Enrique Alba	Universidad de Málaga
Francisco Fernández	Universidad de Extremadura
José Antonio Gámez	Universidad de Castilla la Mancha
Francisco Herrera	Universidad de Granada
César Hervás	Universidad de Córdoba
Rafael Martí	Universidad de Valencia
Juan Julián Merelo	Universidad de Granada
José A. Moreno	Universidad de La Laguna
Luciano Sánchez	Universidad de Oviedo

Comité Organizador

Abraham Duarte (Co-Presidente)	Universidad Rey Juan Carlos
J. Ignacio Hidalgo (Co-Presidente)	Universidad Complutense de Madrid
José Manuel Colmenar	Universidad Complutense de Madrid
Juan Lanchares	Universidad Complutense de Madrid
Antonio S. Montemayor	Universidad Rey Juan Carlos
Eduardo G. Pardo	Universidad Rey Juan Carlos
Juan J. Pantrigo	Universidad Rey Juan Carlos

Comité de Programa

Jesús Aguilar, U. Pablo Olavide	Enrique Alba, U. Málaga
Francisco Almeida, U. La Laguna	Ada Álvarez, U. A. Nuevo León
Ramón Álvarez-Valdés, U. Valencia	Lourdes Araujo, U. Nacional a Distancia
Maribel G. Arenas, U. Granada	Jaume Bacardit, U. of Nottingham
Julio R. Banga, CSIC	Joaquín Bautista, U. P. Cataluña
José Manuel Benítez, U. Granada	Christian Blum, U. P. Cataluña
Julio Brito, U. la Laguna	Rafael Caballero, U. Málaga
Vicente Campos, U Valencia	Erick Cantú Paz, Yahoo! Inc.
Jorge Casillas, U. Granada	Pedro A. Castillo, U. Granada
Francisco Chávez, U. Extremadura	Francisco Chicano, U. Málaga
Carlos A. Coello, CINVESTAV - IPN	José Manuel Colmenar, U. C. Madrid
Ángel Corberán, U. Valencia	Antonio Córdoba, U. Sevilla
Oscar Cordón, ECSC	Carlos Cotta, U. Málaga
Alfredo Cuesta,U. C. Madrid	Pedro Cuesta, U. Las Palmas de GC

Sergio Damas, ECSC
María J. del Jesús, U. Jaén
Julián Dorado, U. Coruña
Abraham Duarte, U. Rey J. Carlos
José Egea, U. P. Cartagena
Anna Esparcia, ITI - UPV
Elena Fernández, U. P. Cataluña
Julia Flores, U. Castilla-La Mancha
Blas Galván, U. las Palmas de GC
Carlos García, U. Córdoba
Salvador García, U. de Jaén
Raúl Giraldez, U. Pablo Olavide
Antonio González, U. Granada
Pedro González, U. Jaén
Francisco Herrera, U. Granada
José Ignacio Hidalgo, U. C. Madrid
Angel A. Juan, U. Oberta de Catalunya
Juan Lanchares, U. C. Madrid
Pedro Larrañaga, U. P. Madrid
José Antonio Lozano, U. País Vasco
Francisco Luna, U. Málaga
Luís Magdalena, ECSC
Francisco Martínez, U. Córdoba
Alexander Mendiburu, U. País Vasco
Daniel Molina, U. Cádiz
Antonio S. Montemayor, U. Rey J. Carlos
J. Marcos Moreno, U. La Laguna
Pablo Moscato, U. of Newcastle
Julio Ortega, U. Granada
José Otero, U. Oviedo
Felipe Padilla, ÉTS
Eduardo G. Pardo, U. Rey J. Carlos
Mª del Carmen Pegalajar, U. Granada
David Pelta, U. Granada
M. Elena Pérez, U. Valladolid
Juan R. Rabuñal, U. Coruña
Ignacio Requena, U. Granada
José Riquelme, U. Sevilla
Víctor Rivas, U. Jaén
Rubén Ruiz, U. P. Valencia
Sancho Salcedo, U. Alcalá
Ángel Sánchez, U. Rey J. Carlos
Luciano Sánchez, U. Oviedo
Thomas Stützle , U. Libre de Bruxelles
Ángel Udías, U. Rey J. Carlos

Luis de la Ossa, U. Castilla-La Macha
Adenso Díaz, U. Oviedo
Bernabé Dorronsoro, U. Luxemburgo
Richard Duro, U. Coruña
Francisco Javier Elorza, U. P. Madrid
Antonio J. Fernández, U. Málaga
Francisco Fernández, U. Extremadura
Miguel Frade, I. Politécnico Leiria
José A. Gámez, U. Castilla-La Mancha
Nicolás García, U. Córdoba
Josep M. Garrell, U. Ramon Llull
Juan A. Gómez Pulido, U. Extremadura
José L. González-Velarde, I. T. Monterrey
José Antonio Gutiérrez, U. Córdoba
Cesar Hervás, U. Córdoba
Fernando Jiménez, U. Murcia
Manuel Laguna, U. de Colorado
Dario Landa Silva, U. of Nottingham
Enrique López, U. León
Manuel Lozano, U. Granada
Gabriel J. Luque, U. Málaga
Rafael Martí, U. València
Belén Melián, U. La Laguna
Juan Julián Merelo, U. Granada
Julián Molina, U. Málaga
Claudio Moraga, ECSC
José A. Moreno, U. La Laguna
Antonio J. Nebro, U. Málaga
Domingo Ortiz, U. Córdoba
Joaquín Pacheco, U. Burgos
Juan J. Pantrigo, U. Rey J. Carlos
Francisco Parreño, U. Castilla La Mancha
José I. Peláez, U. Malaga
Antonio Peregrín, U. Huelva
J. Miguel Puerta, U. Castilla La Mancha
Helena Ramalhinho, U. Pompeu Fabra
Roger Ríos, U. A. Nuevo León
Jose L. Risco Martín, U. C. Madrid
Carlos Andrés Romano, U. P. Valencia
Yago Sáez, U. Carlos III
Ana Mª Sánchez, U. Granada
Juan Manuel Sánchez, U. Extremadura
Roberto Santana, U. P. Madrid
Marco Tomassini, Swiss Federal I. of Tech.
Miguel A. Vega Rodríguez, U. Extremadura

Oswaldo Vélez, U. Rey J. Carlos
José Luís Verdegay, U. Granada
Pedro Villar, U. Granada
Amelia Zafra, U. Córdoba
Sebastián Ventura, U. Córdoba
Gabriel Villa, U. Sevilla
Gabriel Winter, U. las Palmas GC

In Memoriam Juan Manuel Sánchez Pérez

Juan Manuel Sánchez Pérez, Catedrático de Universidad en la Universidad de Extremadura (UEX) en el área de Arquitectura y Tecnología de Computadores, falleció el sábado 13 de abril de 2013 tras una larga enfermedad. Sus compañeros queremos rendirle un humilde homenaje mediante este obituario.

Quienes le conocimos damos fe de su enorme calidad humana, cercanía y profundo interés por sus compañeros y estudiantes, tesón en el trabajo, dedicación a su familia, y tantas otras muestras que atestiguaron su bonhomía. Esta gran humanidad que le caracterizaba se puso de manifiesto a lo largo de su enfermedad, afrontándola con paciencia, esperanza y ganas de superación y, en último término, una gran paz en la despedida.

Juan Manuel obtuvo el título de Doctor en CC. Físicas por la Universidad Complutense de Madrid en el año 1976 bajo la dirección del profesor Antonio Vaquero, con la tesis titulada "Sistema electrónico de enseñanza programada". Tras ocupar diferentes puestos docentes en la UCM, gana su plaza de Profesor Titular de Universidad en el año 1983, permaneciendo en la Complutense hasta 1996, año en el que es nombrado Profesor Titular en la Universidad de Extremadura. En 1999 culminó su carrera académica al ser nombrado Catedrático de la Universidad de Extremadura por el área de Arquitectura y Tecnología de Computadores. Fue además director del Departamento de Informática de la UEX durante varios años y coordinador de diversos planes de estudios de másteres universitarios.

Sus primeros pasos en el mundo de la investigación estuvieron orientados al estudio de la enseñanza programada, campo en el que tiene publicaciones con los profesores José Garcia Santesmases y Antonio Vaquero como el artículo "Addressing methods for external sequential photodigital memories for teaching machines" publicado en el X International Symposium on Educational Technology en el año 1976.

Con posterioridad fue testigo del crecimiento en la capacidad de integración en los circuitos digitales y de lo inviable de las técnicas de diseño existentes. Consciente de esta dificultad, Juan Manuel centró su investigación en el desarrollo de herramientas CAD para el diseño lógico de circuitos integrados combinacionales y secuenciales. En este campo, su grupo de investigación fue de los primeros a nivel nacional en aplicar técnicas basadas en los procesos naturales tales como los algoritmos evolutivos, o el recocido simulado (simulated annealing) en la optimización de circuitos lógicos multinivel, así como en la utilización de diagramas binarios de decisión (BDDs) para la verificación de circuitos integrados. En el año 1998 funda el grupo de investigación ARCO (Arquitectura de Computadores y Diseño Lógico) en la Universidad de Extremadura, en el

que orientó su investigación, por un lado, al ámbito de los dispositivos lógicos reconfigurables basados en FPGAs (prototipado y procesadores empotrados) y, por otro, a la resolución de problemas de optimización combinatoria mediante metaheurísticas y computación evolutiva (utilizando técnicas de computación paralela y distribuida).

Durante sus años de investigación dirigió más de una decena de tesis doctorales, generó una abundantísima producción científica y participó en 20 proyectos de I+D+i. También cabe destacar su contribución a la transferencia de resultados de investigación a la sociedad, al haber sido miembro fundador de la empresa de base tecnológica SENSOLOG S.L.

Su actividad docente se extiende no sólo a la Universidad Complutense de Madrid y la Universidad de Extremadura. Ha impartido clases, además, en la Universidad de León y en el Instituto Politécnico de Leiría (Portugal). Ahora que está tan de moda el uso de las nuevas tecnologías aplicadas a la enseñanza podemos decir sin temor a equivocarnos que fué pionero en este campo teniendo más de 8 artículos publicados entre los años 1975 y 1979 relacionados con la enseñanza programada y el uso de los microprocesadores como herramientas docentes. Como rasgo importante a destacar también figura su profunda entrega a las tareas docentes, en las que siempre se preocupó por sus alumnos, tanto a nivel académico como personal.

En definitiva, este legado supone para todos los que tuvimos la suerte de conocerle, además de un recuerdo imborrable de su persona, un aliciente más en nuestra dedicación a la Universidad, desde las tres perspectivas en las que tanto contribuyó: docencia, investigación y gestión.

D. E. P.

GRASP con Path Relinking para el Problema del Orienteering

Vicente Campos

Dept. Estadística i Investigació Operativa, Universitat de València, Spain
vicente.campos@uv.es

Rafael Martí

Dept. Estadística i Investigació Operativa, Universitat de València, Spain
rafael.marti@uv.es

Abstract. En este trabajo, proponemos la resolución del problema de optimización conocido por orienteering, que combina elementos del problema del agente viajero y el de la mochila. El problema surgió en el contexto de los deportes pues existe uno consistente en que los jugadores deben conseguir la máxima puntación posible al visitar ciertos puntos a su elección para lo cual disponen de un tiempo máximo. En el caso particular en el que el punto inicial y final coinciden, el camino recorrido es un circuito de ahí que el problema reciba el nombre de problema del agente viajero selectivo, incluso en esta versión el problema es NP-difícil. Desde 1980 se han propuesto varios métodos de solución y aplicándose a diversos ámbitos como el turismo y problemas de rutas. El enfoque empleado en este artículo es usar un procedimiento metaheurístico basado en las metodologías GRASP y Path Relinking. Se han analizado diferentes métodos constructivos y se han usado dos entornos de búsqueda local en GRASP. Se ha utilizado Path Relinking con una elite de soluciones proporcionadas por GRASP. Se ofrecen resultados computacionales sobre una colección de 73 ejemplos reportados previamente y se comparan los resultados obtenidos con otras metodologías.

Keywords: Metaheurísticas, GRASP, Path Relinking, Orienteering.

1 Introducción

El problema del Orienteering (OP), consiste en determinar un camino desde un origen a un destino prefijados en un grafo (dirigido o no), cada nodo tiene una puntuación asociada y el camino obtiene una puntuación que es la suma de las puntuaciones de los nodos visitados. Se trata de maximizar la puntuación del camino sin exceder un tiempo límite. Las aplicaciones de este problema y variantes son numerosas (ver por ejemplo Pacheco et al., 2009), incluyendo enfoques multi-objetivo (Jozefowiez et al., 2008; Schilde et al., 2009). En turismo, el problema surge cuando se quieren diseñar itinerarios en una gran ciudad para maximizar el atractivo total (cultura, ocio,…), disponiendo de valoraciones de los puntos de interés, con diferentes tiempos disponibles (Tsiligirides, 1984).

El OP ha sido formulado como un problema lineal entero (Vansteenwegen et al., 2011), en el caso de un grafo dirigido, formulación inspirada en la propuesta por Miller et al. (1960) para el problema del agente viajero. Por otro lado Fischetti et al. (1998) propusieron un algoritmo branch-and-cut para resolver el caso del grafo no dirigido y la solución debe ser un circuito. En Vansteenwegen et al. (2011) se determina que los mejores métodos actuales para resolver el OP son el Ant Colony Optimization (ACO) y Variable Neighborhood Search (VNS) propuestos por Schilde et al. (2009), superando al clásico heurístico de Chao et al. (1996). La metodología de Path Relinking se usó para resolver el Team Orienteering Problem en Soffriau et al. (2010), se trata de una variante del OP en donde hay que encontrar un conjunto de rutas con origen común, que no excedan de un tiempo límite y que maximicen la puntuación total de los nodos visitados. La aportación más importante del presente trabajo es el desarrollo de un método de solución para el OP basado en las metodologías Greedy Randomized Adaptive Search Procedure (GRASP) conjuntamente con Path Relinking (PR) (Resende and Ribeiro, 2003). GRASP consiste esencialmente en la aleatorización de un algoritmo constructivo voraz (greedy), que se adapta a la solución parcial, así se genera un conjunto de soluciones dispersas en el espacio de soluciones susceptibles de ser mejoradas con un algoritmo de búsqueda local. PR permite conectar soluciones entre sí y en el camino de unión encontrar mejores soluciones.

En este trabajo se analizan varios diseños sobre una vasta colección de ejemplos comparando con el mejor valor conocido. La experimentación demuestra que nuestros métodos son competitivos con los mejores heurísticos reportados en la literatura.

2 GRASP

Presentamos un nuevo GRASP para resolver el OP, describimos en esta sección cuatro constructivos C1, C2, C3 y C4, un método de mejora de búsqueda local y la comparativa entre ellos.

Sea un grafo dirigido G = (V, A) en donde V es el conjunto de nodos y A el de arcos, donde debemos construir el camino con origen en el nodo 1 y final en el *n*, cada arco (i, j) tiene un tiempo asociado t_{ij} y la puntuación asociada a cada nodo *i* es s_i.

2.1 Métodos constructivos

En todos los constructivos descritos a continuación, $0<\alpha\leq 1$ es un parámetro que debe ajustarse y determina la fracción de candidatos que debe ser considerada.

C1 comienza con un camino $P = \{1, n\}$ desde 1 hasta *n* a través del arco (1, *n*). *P* representa la solución parcial y T_P su tiempo asociado (inicialmente $T_P = t_{1n}$). La lista de candidatos *CL* contiene a los nodos que no están en *P* pero que pueden ser insertados en *P* dentro del tiempo límite *T*. En el primer paso pues

$$CL = \{i \epsilon V \backslash P : t_{1i} + t_{in} \leq T\}.$$

En cada paso, C1 elige un elemento candidato $i \in CL$ para insertar en la solución parcial. Cada candidato se evalúa de acuerdo con su puntuación para construir la lista restringida de candidatos *RCL*1 de la que se elige aleatoriamente un elemento. Concretamente, primero se calcula la máxima puntuación $S = max_{i \in CL} s_i$ de los elementos en *CL* y luego se construye *RCL*1 con todos los candidatos cuya puntuación supera una fracción α de la puntuación máxima.

$$RCL1 = \{i \in CL : s_i \geq \alpha S\}.$$

Finalmente, C1 selecciona aleatoriamente un elemento de *RCL*1 el cual se inserta en la mejor posición de *P* (la que produce el menor incremento en tiempo del camino). C1 realiza iteraciones reconstruyendo *CL* mientras sea posible insertar algún nodo al camino parcial. Cuando *CL*=∅, C1 devuelve el camino *P* como resultado y para.

C2 es un constructivo alternativo introducido en Resende y Werneck (2004), exitosamente utilizado en Resende et al., (2010). C2 construye *CL*, como en C1. La lista restringida de candidatos *RCL*2 se construye de forma diferente, primero se determina su tamaño como una fracción α de los candidatos en *CL* ($|RCL2| = \lceil \alpha |CL| \rceil$), a continuación, se extrae una muestra aleatoria sin elementos repetidos de tamaño |*RCL*2|, insertando en la solución parcial el candidato con mayor puntuación en *RCL*2. Notar que en C1 cuanto mayor es α menor es el grado de aleatorización, mientras que en C2 ocurre lo contrario. Por otra parte, en C1 los elementos con mejor puntuación tienen más probabilidad de ser elegidos que en C2 pues en *RCL*2 los candidatos se eligen independientemente de su puntuación.

C3 obtiene la lista de candidatos *CL* de forma análoga a C1 y C2, pero la evaluación greedy e_i del elemento *i*, es la puntuación s_i por incremento en tiempo, Δt_i, al insertar el elemento *i* en la mejor posición de la solución parcial, es decir $e_i = \frac{s_i}{\Delta t_i}$.

La lista restringida de candidatos *RCL*3 se construye con los elementos en *CL* cuya evaluación al menos iguale una fracción α de la máxima evaluación

$$RCL3 = \{i \in CL: e_i \geq \alpha E\} \ \ donde\ E = max_{i \in CL} e_i$$

C4, al igual que C2, construye la lista restringida de candidatos *RCL*4 con una muestra aleatoria de *CL* de tamaño $\lceil \alpha |CL| \rceil$. Se calcula e_i para cada $i \in RCL4$ y se selecciona el elemento con mayor e_i insertándolo en la mejor posición del camino parcial.

Cada iteración GRASP genera una solución y debido a la aleatorización, se obtiene un muestreo estratégico del espacio de soluciones. Las soluciones GRASP son independientes y no guardan memoria, a diferencia de, por ejemplo, tabu search. La mejor solución GRASP es la mejor de entre las obtenidas por algoritmos independientes.

2.2 Comparación de métodos constructivos

Es importante que un método constructivo produzca buenas soluciones para el método principal. Esto es importante también en el caso de GRASP con PR. En ambos contextos, se prefieren soluciones con buena calidad con respecto al objetivo pero también que sean diversas para explorar diferentes regiones del espacio de soluciones.
Para comparar los constructivos de la sección anterior, generamos un conjunto de soluciones con cada uno y medimos su calidad y su diversidad. La calidad se mide por el objetivo. Para medir la diversidad usamos el cálculo que sigue: dadas dos soluciones, $A = \{1, a_1, a_2, \dots, a_k, n\}$ y $B = \{1, b_1, b_2, \dots, b_p, n\}$ su diversidad es

$$div(A, B) = \frac{|A-B|+|B-A|}{|A|+|B|}$$

En la expresión anterior no se contabiliza el origen y destino del camino. Para ilustrarlo, supongamos las dos soluciones siguientes en un grafo con n = 21, $A = \{1,7,10,4,8,3,12,20,21\}$ y $B = \{1,12,3,9,11,7,21\}$. El número de elementos en A no presentes en B es 4, y el número de elementos en B que no están en A es 2. Por otra parte, el número total de elementos en las dos soluciones exceptuando origen y destino es 12, por lo tanto la diversidad entre A y B es $div(A, B) = \frac{4+2}{7+5} = 0.5$. Notar que esta medida es independiente del tamaño del grafo. Generamos 100 soluciones con cada constructivo y obtenemos la diversidad media entre todos los pares de soluciones. Por otro lado, calculamos el valor objetivo de cada solución y su desviación relativa del valor de la mejor solución conocida. Las medias de estas desviaciones y diversidades producen una evaluación total del método.

En las pruebas, utilizamos 33 ejemplos con óptimo conocido en Fischetti et al. (1998). La Tabla 1 muestra para cada uno de los cuatro métodos, la desviación objetivo media (Dev) y la diversidad media (Div). El mejor método respecto a la calidad es C2 (con α = 0.2) puesto que obtiene la menor desviación 7.2%. El método que mayor diversidad produce es C1 (con α = 0.2) con un valor de 0.57. Es difícil comparar directamente ambas medidas (Div y Dev) pues no van unidas.

	α	0.2	0.4	0.6	0.8
C1	Dev/Div	19.3% - 0.57	13.5% - 0.49	12.1% - 0.43	13.0% - 0.32
C2	Dev/Div	7.2% - 0.28	7.9% - 0.20	8.4% - 0.15	9.9% - 0.11
C3	Dev/Div	11.8% - 0.41	13.1% - 0.33	13.8% - 0.27	14.2% - 0.21
C4	Dev/Div	7.4% - 0.25	7.3% - 0.18	7.9% - 0.14	9.2% - 0.11

Tabla 1. Desviación y diversidad de los métodos constructivos

En el diagrama de dispersión de la Figura 1, aparece un total de 16 puntos al considerar cuatro selecciones de α por cada constructivo. Representamos 1-Dev sobre Div. Sólo los métodos C1(0.2), C1(0.4), C1(0.6), C2(0.2) y C3(0.2) no están dominados, es decir que no son inferiores en ambas medidas a ningún otro, entre paréntesis el valor de α. Seguiremos utilizando los 5 métodos no dominados en la búsqueda local.

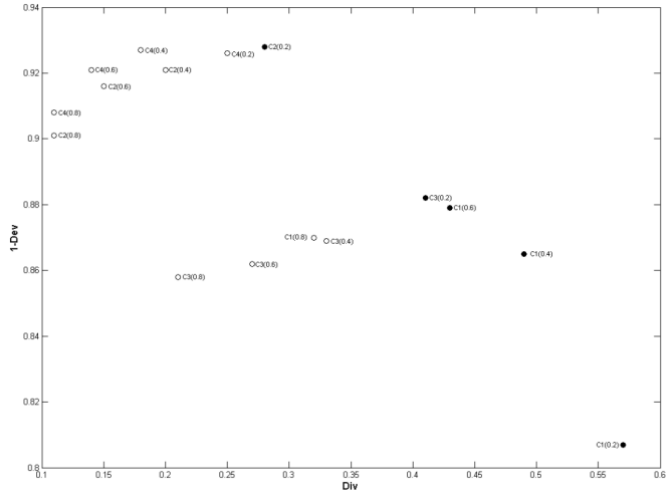


Figura 1. Desviación y diversidad de los métodos constructivos

2.3 Método de mejora basado en búsqueda local

Hemos diseñado una búsqueda local de dos fases. La primera está basada en intercambios mientras que la segunda está basada en inserciones.

El entorno de la primera fase se basa en el intercambio uno-a-uno entre un nodo *v* del camino *P* y otro nodo *w* no en *P* (sin exceder el tiempo máximo límite *T*). La diferencia entre las puntuaciones de ambos nodos (s_w - s_v) nos da el valor del movimiento asociado. Se examinan los nodos del camino y, para cada uno de ellos, se considera su intercambio con cada nodo fuera del camino. Calculamos el valor de cada combinación y se realiza el mejor movimiento encontrado. En el caso de que no exista movimiento de mejora para un determinado vértice *v* en *P*, se intenta reducir la longitud del camino sin reducir la puntuación total del mismo. Concretamente, se exploran de

nuevo los nodos fuera de *P* pero solo consideramos los que tienen la misma puntuación que *v* y chequeamos si el intercambio con *v* reduce el tiempo del camino. En este último caso se realiza el movimiento aunque sin incremento de la puntuación. En caso contrario se continúa la búsqueda con el siguiente vértice *v* del camino *P*.
Si se consigue realizar un intercambio en la fase anterior, se intenta, en una segunda fase, un movimiento de inserción que consiste en insertar en *P* un nodo fuera del camino. Hay que hacer notar que ni la inserción ni el intercambio conllevan necesariamente un aumento del tiempo del camino, es factible incluso una reducción del mismo en la inserción si no se satisface la desigualdad triangular. La inserción, si es posible, se hace en la mejor posición (i.e. aquélla que supone un menor aumento del tiempo del camino), en este caso se volvería a la primera fase intentado nuevos intercambios. En caso contrario se continúa hasta la exploración de todos los nodos y se reinicia la primera fase si se ha realizado algún movimiento.

El procedimiento de búsqueda local finaliza cuando se examinan todos los nodos en *P* y ya no son posibles más intercambios.

2.4 Comparación de métodos con búsqueda local

En la sección anterior hemos probado 16 métodos constructivos sobre 33 ejemplos. Hemos identificado cinco de ellos, C1(0.2), C1(0.4), C1(0.6), C2(0.2) y C3(0.2) como los mejores en términos de calidad y diversidad. Vamos a ver ahora su comportamiento cuando la búsqueda local, que representamos abreviadamente por LS, se aplica a 10 soluciones construidas con cada uno. Para cada método y cada ejemplo, obtenemos la mejor solución encontrada de entre las 100 soluciones generadas más la búsqueda local descrita anteriormente. La Tabla 2 muestra la desviación media con respecto al valor óptimo (Dev) y el número de mejores soluciones que cada método es capaz de alcanzar (#best).

La tabla 2 muestra que el mayor número de mejores soluciones 17 se obtienen con C1(0.2)+LS, capaz de alcanzar la menor desviación 3.42% usando 33 ejemplos. Por otro lado, los métodos son muy rápidos, por debajo de 0.1 seg. de CPU en un Intel Core I5 a 3.20 GHz en cada ejemplo.

	C1(0.2)+LS	C1(0.4)+LS	C1(0.6)+LS	C2(0.2)+LS	C3(0.2)+LS
Dev	3.42%	3.97%	5.65%	3.96%	7.76%
#best	17	13	7	16	7

Tabla 2. Desviación y mejores soluciones de constructivos con búsqueda local.

Al comprobar las mejores soluciones que cada método es capaz de alcanzar, descubrimos que algunos métodos son capaces de obtener distintas mejores soluciones. Esto es especialmente cierto si comparamos dos métodos basados en diferentes constructivos, por ejemplo entre las 16 mejores soluciones obtenidas por C2(0.2)+LS y las 17 conseguidas por C1(0.2)+LS, solamente 10 coinciden y entre ambos métodos son capaces de generar 23 mejores soluciones.

Hemos realizado un experimento preliminar adicional para determinar si existía una correlación fuerte entre el valor del método constructivo con el valor del constructivo reforzado con la búsqueda local. En particular, hemos generado 100 soluciones del ejemplo gil262 con C1(0.2) obteniendo su valor y su valor mejorado resultando coeficiente de correlación $r = 0.284$ indicando una correlación débil. Por lo tanto no podemos concluir que las mejores soluciones con la búsqueda local procedan con alta confianza de las mejores soluciones constructivas.

El análisis anterior nos ha llevado a alternar los procedimientos C1(0.2)+LS, C2(0.2)+LS en el conjunto de iteraciones GRASP.

3 Path Relinking

Método sugerido por Glover y Laguna (1997) para reforzar la búsqueda local. Se generan soluciones nuevas explorando las trayectorias que conectan soluciones de alta calidad. Laguna y Martí (1999) adaptaron PR en el contexto de GRASP como forma de intensificación.

PR tiene dos fases, en la primera se genera un conjunto de soluciones diferentes obtenida por GRASP, en la segunda aplicamos el reencadenamiento a cada par de soluciones en el conjunto. Con el fin de explorar mejor el camino entre las soluciones del par hemos añadido el procedimiento de búsqueda local como se hizo en Laguna y Martí (1999) para el problema del cruce de arcos.

En esta sección comparamos GRASP y GRASP con PR sobre 33 ejemplos usados en las Secciones 3 y 4 para evaluar la contribución de PR. La Tabla 3 muestra la media del objetivo (Valor), la desviación media respecto al óptimo (Dev), el número de óptimos (#opt) y número de mejores soluciones (#best) que cada método es capaz de alcanzar y tiempo CPU en segundos (CPU). Se corrieron 500 construcciones con GRASP y con GRASP con PR se crearon todos los pares de soluciones diferentes.

	GRASP	GRASP con PR
Valor	2078.30	2099.82
Dev	1.75%	0.84%
#best	14	33
#opt	14	18
CPU	1.72	26.87

Tabla 3. Comparación sobre 33 ejemplos

En la Tabla 3 vemos que los resultados con PR superan a GRASP, excepto en el tiempo de computación. Sin embargo, pruebas complementarias muestran que aumentar el número de construcciones GRASP no conduce a una mejora progresiva de la mejor solución, más bien se produce un estancamiento que no alcanza la calidad de la

mejor solución reportada por PR. Es decir a largo plazo GRASP con PR es mejor que GRASP aun dejando a este último el mismo tiempo de ejecución.

4 Comparación con métodos previos

En esta sección comparamos los métodos propuestos con los mejores heurísticos previos: CGW de Chao et al. (1996), GLS de Vansteenwegen et al. (2009), ACO de Schilde et al. (2009) y VNS de Schilde et al. (2009).
Chao et. al. (1996) propuso un conjunto de 40 ejemplos, 14 de ellos con 64 nodos llamado p64 y 26 ejemplos de 66 nodos. Cada conjunto está basado en el mismo grafo difiriendo en el tiempo máximo T para completar el camino.

CGW	GLS	ACO		VNS	
valor	valor	valor	CPU	valor	CPU
790.7	786.0	795.0/792.0	1.970	795.0/790.7	2.570

Tabla 4. Comparación de los métodos mejores previos sobre p64

La Tabla 4 muestra las medias de los valores obtenidos con los cuatro métodos mencionados (CGW, GLS, ACO y VNS). Los valores se tomaron de Schilde et al. (2009), corresponden a un Pentium 4D a 3.2 Ghz. Los tiempos no aparecen en los métodos CGW y GLS. Por otra parte los métodos ACO y VNS fueron replicados 10 veces en cada ejemplo y se reportan el mejor y el peor resultados obtenidos, respectivamente y solo un valor cuando coinciden. Los tiempos de computación (CPU) corresponden al de obtener la mejor solución en la mejor de las 10 ejecuciones. El método ACO es el mejor y por lo tanto vamos a compararlo con nuestros GRASP y GRASP + PR.

ACO		GRASP		GRASP+PR	
valor	CPU	valor	CPU	valor	CPU
795.0/792.0	1.970	794.6/791.1	0.089	795.0/792.9	0.208

Tabla 5. Comparación con el mejor método previo sobre p64

La Tabla 5 muestra que nuestros métodos son competitivos con el mejor método heurístico publicado para este problema. La comparación directa de CPU no es posible debido a que los dos GRASP se ejecutaron con un Intel Core i5 650 at 3.20 GHz. Pero la ratio entre la media de CPU de GRASP con PR y ACO es de 9.47, lo que parece indicar que nuestro método es más rápido que ACO.

ACO		VNS		GRASP		GRASP+PR	
valor	CPU	valor	CPU	valor	CPU	valor	CPU
952.3	0.26	952.3/947.5	1.38	952.3/948.3	0.07	952.3	0.12

Tabla 6. Comparación de los mejores métodos sobre p66

La Tabla 6 compara los dos mejores heurísticos previos, ACO y VNS con nuestros dos métodos GRASP y GRASP con PR, sobre el conjunto de ejemplos p66. La co-

lumna valor muestra el promedio de los valores obtenidos con cada método (ACO, VNS, GRASP y GRASP+PR). Las pruebas son idénticas a las del experimento previo, replicando 10 veces la ejecución de cada método sobre cada ejemplo. Las medias mejor/peor de VNS y GRASP son, respectivamente, 952.3/947.5 y 952.3/948.3. Por otra parte, ACO y GRASP+PR son capaces de obtener siempre en todas las ejecuciones el mejor valor conocido en los 26 ejemplos de p66. GRASP requiere solo 0.07 seg. de media por 0.12 seg. que requiere GRASP+PR. ACO y VNS requieren 0.26 y 1.38 seg., respectivamente, pero considerando que nuestro ordenador es más moderno, podemos concluir que GRASP tiene un comportamiento similar a VNS mientras que GRASP+PR lo tiene respecto a ACO en este grupo de ejemplos.

Fischetti et al. (1998) propusieron un procedimiento exacto (branch-and-cut) para el caso en que el camino es un circuito, y obtienen el valor óptimo en muchos casos, sin embargo, su método consume hasta 5 horas sin obtener el óptimo en algunos de los ejemplos más grandes. Como comparación inicial, hemos corrido el algoritmo de branch-and-cut, cedido amablemente por los autores, sobre una de las colecciones de 33 ejemplos citados en su artículo, con nuestro ordenador Intel Core i5 650 at 83.2 Ghz con la versión actualizada CPLEX 12.0. La tabla pormenorizada es demasiado grande para la extensión de esta artículo pero los resultados se resumen, sobre estos 33 ejemplos, de la forma que sigue: El algoritmo de branch-and-cut requiere en media 1060.46 segundos, frente a 3.43 para GRASP y 90.57 para GRASP con PR, respectivamente, si bien el método exacto obtiene el óptimo en los 33 ejemplos, mientras que los heurísticos lo aproximan. GRASP con PR es capaz de obtener 19 óptimos y su alejamiento medio de la solución óptima es tan solo del 1.42%. GRASP presenta un alejamiento del óptimo del 2.37%, consiguiendo un total de 14 óptimos.

5 Conclusiones

El Orienteering es un problema combinatorio difícil y constituye una plataforma excelente para estudiar la eficacia de mecanismos de búsqueda. Nuestro interés principal era investigar el efecto de combinar distintos entornos dentro de GRASP así como el estudio del post-proceso basado en Path Relinking. Después de una vasta experimentación, hemos determinado los beneficios de añadir estrategias mejoradas de búsqueda a los procedimientos básicos. Hemos comparado nuestros métodos con los mejores identificados en estudios previos. Claramente, la comparación con nuestros métodos muestran que son competitivos con los que determinan el estado del arte actual.

Agradecimientos

Esta investigación fue parcialmente financiada por el Ministerio de Educación y Ciencia de España (TIN2009-07516 y TIN2012-35632-C02). Damos las gracias a los profs. Schilde, Doerner, Hartl, y Kiechle por haber compartido sus resultados y datos con nosotros. Queremos también agradecer a los profs. Fischetti, Salazar y Toth por

cedernos su código de branch-and-cut y a Abraham Duarte y Jesús Sánchez-Oro por su adaptación a CPLEX 12.0 y ejecución sobre un conjunto de ejemplos.

Referencias

Chao, I.M., Golden, B.L. and Wasil, E.A. (1996) A fast and effective heuristic for the orienteering problem, European Journal of Operational Research 88, 475-489.

Fischetti, M., Salazar, J.J. and Toth, P. (1998) Solving the orienteering problem through branch and cut, INFORMS Journal on Computing 10, 133- 148.

Glover, F. , Laguna, M. (1997) Tabu Search, Kluwer Academic Publishers, Boston.

Jozefowiez, N., F. Glover, and M. Laguna (2008) Multi-objective Meta-heuristics for the Traveling Salesman Problem with Profits. Journal of Mathematical Modelling and Algorithms 7(2), 177-195.

Laguna, M. and Martí, R. (1999) GRASP and path relinking for 2-layer straight line crossing minimization. INFORMS Journal on Computing, 11:44–52.

Miller, C. E., A. W. Tucker, and Zemlin, R. A. (1960) Integer programming formulations and traveling salesman problems, Journal ACM, 7, 326–329.

Pacheco, J., A. Alvarez, S. Casado and J. L. González-Velarde (2009) A Tabu Search Approach to an Urban Transport Problem in Northern Spain, Computers and Operations Research 36, pp. 967-979.

Resende, M.G.C. and Werneck, R.F. (2004) A hybrid heuristic for the p-median problem, Journal of Heuristics, 10:59–88.

Resende, M.G.C. and Ribeiro, C.C. (2003) Greedy randomized adaptive search procedures. F. Glover, G. Kochenberger, eds. State-of-the-art Handbook in Metaheuristics, Kluwer Academic Publishers, Boston, MA. 219–250.

Resende, M.G.C., R. Martí, M. Gallego and A. Duarte (2010), GRASP and Path Relinking for the Max-Min Diversity Problem, Computers and Operations Research 37(3), 498-508

Schilde, M., Doerner, K.F., Hartl, R.F.and Kiechle, G. (2009), Metaheuristics for the bi-objective orienteering problem, Swarm Intell., 3, 179 -201.

Souffriau, W., Vansteenwegen, P., Vanden Berghe, G. and Oudheusden, D.V. (2010) A path relinking approach for the team orienteering problem, Computers and Operations Research, 37, 1853-1859.

Tsiligirides, T. (1984) Heuristic methods applied to orienteering, Journal of the Operational Research Society, 35(9), 797—809.

Vansteenwegen, P., Souffriau, W. and Oudheusden, D.V. (2011) The orienteering problem: A survey, European Journal of Operational Research 209, 1-10.

Vansteenwegen, P., Souffriau, W., Vanden Berghe, G. and Oudheusden, D.V. (2009) A guided local search metaheuristic for the team orienteering problem. European Journal of the Operational Research, 196(1), 118-127.

A Tabu Search Approach for a Territory Design Problem with Stochastic Demands

Mónica Elizondo-Amaya[1], Roger Z. Ríos-Mercado[1], David P. Morton[2], and Erhan Kutanoglu[2]

[1] Graduate Program in Systems Engineering
Universidad Autónoma de Nuevo León
San Nicolás de los Garza, NL 66450, México
{moni,roger}@yalma.fime.uanl.mx
[2] Graduate Program in Operations Research & Industrial Engineering
The University of Texas at Austin
Austin, TX 78712, USA
{morton,erhan}@mail.utexas.edu

Abstract. Territory design can be defined as the problem of grouping small geographic units into larger geographic clusters called territories in such a way that the latter satisfy certain planning criteria. A stochastic version of a commercial territory design problem in presence of uncertain demands is addressed in this work. The objective is to minimize the territory imbalances with respect to the product demand subject to planning criteria such as territory connectivity, compactness, and balance with respect to the number of customers. A tabu search metaheuristic for large instances is developed for this problem. The proposed method incorporates advanced techniques such as strategic oscillation and specific neighborhood exploration strategies. Empirical evidence shows the value of using such advanced techniques. To the best of our knowledge, this is the first study addressing the stochastic nature of the demand in a commercial territory design problem.

Keywords: Commercial territory design, metaheuristics, tabu search, stochastic programming.

1 Introduction

In the general context, the territory design problem (TDP) may be viewed as the problem of grouping small geographic basic units (BUs) into larger geographic clusters, called territories, in such a way that the territories are acceptable (or optimal) according to certain planning criteria. This problem belongs to the family of districting problems that have a broad range of applications such as political districting (Hojati [1], Ricca and Simeone [2], Mehrotra et al. [3], Bozkaya et al. [4]) and sales territory design (Zoltners and Sinha [5], Fleischmann and Paraschis [6], Drexl and Haase [7]) to name the most relevant. The reader can find an extensive survey on approaches to districting problems in the works of

Kalcsics et al. [8] and Duque et al. [9]. The problem addressed in this paper is a commercial territory design problem motivated by a real-world application from the bottled beverage distribution industry. Given a set of city blocks, where two different activities are present in each block (number of customers and product demand)the firm wants to partition the area of the city into disjoint territories according to several criteria such as:(i) Balanced territories: Territories must be similar in size, with respect to each of the two block activity measures; (ii) connectivity: For each formed territory, BUs can reach each other by traveling within the territory; (iii) compactness: BUs assigned to a territory are relatively close to each other and (iv) a fixed number of territories.

Several variations of the deterministic version of this problem (that is, when the demands are known) have been studied in the past; however, the stochastic version of the problem has not been addressed before to the best of our knowledge. This work introduces the stochastic version of the commercial territory design problem (STDP) that considers the uncertainty of the customers demand. A solution approach based on tabu search (Glover [10]) is proposed to obtain solutions for this problem. The proposed algorithm incorporates advanced mechanisms to improve the local search performance such as strategic oscillation, a dynamic neighborhood generation technique and a candidate list strategy. Our empirical work includes an evaluation of the algorithmic performance of the proposed approach. Computational experiments report that the quality of feasible solutions is improved when these strategies are employed in the local search phase.

2 Problem Statement

Let $G = (V, E)$ be an undirected graph, where V is the set of nodes (city blocks) and E is the set of edges that represents adjacency between blocks. That is, a block or basic unit (BU) is associated with a node $j \in V$, and an edge connecting nodes i and j exists if BUs i and j are located in adjacent blocks. Multiple attributes are associated to each block $j \in V$ such as geographical coordinates (c_x^j, c_y^j), number of customers denoted by w_j and sales volume (product demand) which is a discrete random variable ξ_j with realizations indexed by $\omega \in \Omega$. The number of territories p is fixed and is given as a parameter. It is required that each node is assigned to only one territory. A p-partition of the set V is denoted by $X = (X_1, \ldots, X_p)$ where $X_k \subset V$ is called a territory of V. Let $w(X_k) = \sum_{i \in X_k} w_i$ denote the number of customers in the territory X_k. The balancing requirement with respect to the number customers is modeled by introducing a tolerance parameter τ. This tolerance parameter is user specified and it represents a limit on the maximum deviation from an ideal target allowed. This target value is given by the average size $\mu = \sum_{i \in V} w_i / p$. Another important feature is that all of the nodes assigned to each territory are connected by a path contained totally within the territory (i.e., each of the territories X_k must induce a connected subgraph of G). Figure 1 illustrates the balancing and connectivity constraints. In addition, it is required that in each of the territories, blocks

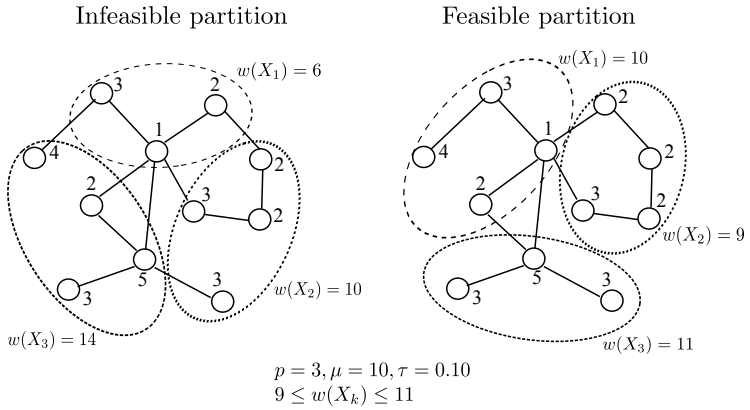


Fig. 1. Illustration of balancing and connectivity.

must be relatively close to each other. To account for this, we use a dispersion measure based on the p-center problem (pCP) and a maximal distance T for the dispersion of the territories is set. The dispersion of a territory X_k is given by $\max_{k=1,\ldots,p} \max_{j \in X_k} \left\{ d_{c(k),j} \right\}$, where $c(k)$ denotes the index of the center of the territory X_k, and is determined by:

$$c(k) = \arg \min_{j \in X_k} \min_{i \in X_k} \{d_{ij}\}.$$

Location of territory centers is not a requirement of the problem but a feature of the proposed formulation that was adopted for convenience when measuring dispersion of territories. Finally, the objective is to minimize the territory imbalances with respect to the stochastic demand. Let Π be the collection of all p-partitions of V and let $W(X_k) = \sum_{j \in X_k} \xi_j$ be the size of demand in the territory X_k. The combinatorial optimization model is given as follows:

Model (STDP)

$$\min_{X \in \Pi} \quad f(X) = \mathbb{E}\left[\max_{k \in K} \{W(X_k)\} \right] \tag{1}$$

$$\text{subject to:} \quad \frac{w(X_k)}{\mu} \in [(1-\tau), (1+\tau)] \qquad k \in K, \tag{2}$$

$$\max_{k=1,\ldots,p} \max_{j \in X_k} \left\{ d_{c(k),j} \right\} \leq T, \tag{3}$$

$$G_k = G\left(V_k, E(V_k)\right) \text{ is connected} \qquad k \in K. \tag{4}$$

Objective (1) minimizes the territory imbalances with respect to the product demand by minimizing the expectation of the maximum territory size with respect to the product demand. Constraints (2) represent the territory balance with respect to the number of customers as it establishes that the size of each territory with respect to this attribute must lie within a range (measured by tolerance parameter τ) around its average size. Constraint (3) establishes a compactness

measure as it limits the maximum distance between a territory center and its assigned BUs. This parameter can be a policy established by the firm. Therefore, the problem can be formally described as finding a p-partition of the graph G that meets balance, connectivity and compactness constraints and minimizes the territory imbalances with respect to stochastic customer demands. This problem has been modeled in terms of integer programming and empirically validated using the Branch and Bound (B&B) procedure implemented by the commercial solver CPLEX 11.2. Small instances of just 20 nodes, 2 territories and 3 demand scenarios were solved to optimality. Due to the fact that real-world instances are even greater than 500 nodes, the use of a heuristic procedure is proposed. A tabu search procedure was developed and tested on a collection of data instances obtaining promising results.

3 Tabu search approach to STDP

Tabu search (Glover and Laguna [11]) is a well-known metaheuristic that has been widely used for successfully solving many combinatorial optimization problems. Particularly, in political districting, TS has shown an efficient performance [4]. Tabu search guides a local search procedure to explore the solution space beyond local optimality. Starting from an initial solution, TS iteratively moves from the current solution to its best neighbor, even if this new solution is worse than the one available, until a pre-specified stopping criterion is met. In order to avoid cycling and becoming trapped in local optima, certain moves (or solution attributes) that lead to previously explored regions are forbidden (i.e.,classified as tabu), forming the short-term memory of TS. The tabu status of a move may be overriden making it an allowable move if an aspiration criterion is satisfied (if, for instance, the tabu move leads to a new best solution). The tabu tenure is the length of time during which a certain move is classified as tabu. It can be kept constant or varied dynamically throughout the search. These key aspects of the TS algorithm tailored to STDP in this paper are described below.

3.1 Initial solution generation

At a given iteration, the construction phase consists of building p territories simultaneously in such a way that connectivity is always satisfied while infeasibility in terms of balance and dispersion is allowed to some extent. This procedure starts by selecting p territory seed centers $c(1), \ldots, c(p)$ which are the first BUs assigned to each territory (i.e., $c(k) \in X_k, k \in K = \{1, \ldots, p\}$). Selecting seed centers by using a purely random approach could lead to obtain inappropriate initial territory centers (e.g., centers close to each other). To avoid this, the problem of choosing an appropriate set of p initials seeds is viewed as a p-Dispersion Problem (Erkut et al. [12]). Territories are then built iteratively in three stages. In the first stage, a fraction δ of the total of BUs are iteratively assigned to territories by considering the threshold distance T and connectivity. At each iteration, a BU $i \in V$ is assigned to an adjacent territory X_{k^*} (i.e., the basic unit

i is connected by an edge to a BU already assigned to X_{k^*}) such that:

$$k^* = \arg \min_{k \in K_T(i)} E\left[\max\left\{W(X_1), \ldots, W(X_k \cup \{i\}), \ldots, W(X_p)\right\}\right], \quad (5)$$

where $K_T(i) = \{k \in K | d_{c(k),i} \leq T\}$. The latter process is iterated until a fraction δ of the total of BUs have been assigned to p territories, where the centers $c(1), \ldots, c(p)$ are updated every L iterations. From this stage, the p territories have been built through a greedy function that completely ignores the customer balance constraints. The rationale behind this is that at the start of the construction phase all territories are likely to violate these constraints thus it is not suitable to penalize them until they have a considerable size with respect to this activity measure. The second stage of the construction phase tries to assign the remaining nodes that were not assigned in stage one. For this stage a BU is assigned to an adjacent territory by considering both the objective function and balance constraints. For each basic unit $i \in V$, the cost of assign it to a territory X_k, $k \in K_T(i)$ is evaluated according to the following merit function:

$$\begin{aligned}\phi_k(i) = \lambda \left(\frac{1}{\gamma}\right) E\left[\max\left\{W(X_1), \ldots, W(X_k \cup \{i\}), \ldots, W(X_p)\right\}\right] \\ + (1-\lambda)\left(\frac{1}{\mu}\right) \max\left\{w(X_k \cup \{i\}) - (1+\tau)\mu, 0\right\} \quad (6)\end{aligned}$$

where $\gamma = \sum_{i \in V} E(\xi_i)/p$ (i.e., the expected value for the average size of demand in the territories), is used for normalizing the objective function. The second term of the Equation (6) represents the represents the relative infeasibility with respect to the upper bound of the balance requirement. Both factors objective function and balancing infeasibility are weighted by a parameter λ. A candidate list (RCL) restricted by a quality parameter α is constructed, from which a territory is randomly selected to assign the current basic unit. The process is repeated for every basic unit i. If a territory exceeds the average weight for the number of customers it is considered closed and no further node can be assigned to it. The latter process iterates until no basic unit can be assigned to a territory within the threshold distance T, or every territory is considered closed or if all BUs have been assigned. Since previous stages do not guarantee that all nodes will be assigned to a territory, a final stage is applied in which unassigned nodes are assigned by using Equation (5) or to their nearest adjacent territory in case that no territory center is within the maximal dipersion distance. We now describe the mechanisms for improving the constructed initial solution.

3.2 Neighborhood generation

The aim of the local search is to improve the objective function and at the same time to reduce the infeasibilities of the balance and the compactness constraints as much as possible. Insert moves and pairwise exchanges (swaps) are frequently used move types for neighborhood generation in combinatorial problems. Let $X_{t(i)}$ denotes the territory to which node i belongs, $t(i) \in 1, \ldots, p$ we next define two different moves for the neighborhoods construction:

- $move_A(i,k)$: An insertion move reassign a basic unit i from its current territory $X_{t(i)}$ to another territory X_k; $t(i) \neq X_k$; $t(i), k \in 1,\ldots,p$.
- $move_B(i,j)$: A swap move exchanges basic units i and j between its respective territories $X_{t(i)}$ $X_{t(i)}$ and $X_{t(j)}$; $t(i) \neq t(j)$; $t(i), t(j) \in 1,\ldots,p$.

Connectivity must be kept, then only moves $move_A(i,k)$ where $X_{t(i)} \setminus \{i\}$ remains connected and for any $l \in X_k$, $\exists(i,l) \in E$ are allowed. The same criteria apply for moves $move_B(i,j)$. We denote as $N_A(X)$ and $N_B(X)$ the neighborhoods of a solution X generated from $move_A(i,k)$ and $move_B(i,j)$ respectively.

3.3 Merit function and strategic oscillation

A variant of the objective function is used to guide the local search phase. This function weights the objective function value and the infeasibility with respect to the compactness and balancing constraints is used. Specifically, the merit function for a given territory design $X = (X_1,\ldots,X_p)$ is given by:

$$F(X) = \left(\frac{1}{\gamma}\right) E\left[g(X)\right] + \beta_1 f_1(X) + \beta_2 f_2(X), \tag{7}$$

where,

$$g(X) = \left(\frac{1}{\gamma}\right) \max_{k=1,\ldots,p} \{W(X_k)\} \tag{8}$$

$$f_1(X) = \left(\frac{1}{d_{\max}}\right) \max_{k=1,\ldots,p} \left\{ \max_{j \in X_k} \left\{d_{c(k),j}\right\} - T, 0 \right\} \tag{9}$$

$$f_2(X) = \sum_{k=1}^{p} \left(\frac{1}{\mu}\right) \max\left\{w(X_k) - (1+\tau)\mu, (1-\tau)\mu - w(X_k), 0\right\} \tag{10}$$

Then, the quality of a move is determined by (7). Equation (8) is the normalized objective function. Expresion (9) represents the relative infeasibility with respect to the compactness requirement. Equation (10) is the sum of all relative infeasibilities of the balancing constraints. Finally, β_1 and β_2 are penalty parameters to be dynamically updated according to strategic oscillation (Glover [13]). The penalty coefficients are then self-adjusted each r iterations. If compactness constraint was violated in all the r^* previous solutions, then $\beta_1 = \psi\beta_1$, $\psi > 1$ and the same occurs with β_2 when the balance constraints are violated in all r^* previous solutions. If all r^* previous solutions were feasible, the values of both parameters are reduced as follows: $\beta_q = \frac{1}{\psi}\beta_q$, $q = 1,2$, $\psi > 1$. Otherwise β_q remains unchanged. The factor ψ as well as the values for r and $r*$ are user-defined parameters. With this strategy we can guide the search to a larger space by allowing infeasible moves.

3.4 Neighborhood exploration strategy

The proposed TS uses a dynamic neighborhood exploration strategy. The neighborhood structure is not a static set as it can change according to an infeasibility

threshold $\epsilon \in [0, 1]$. Since insertion moves are expected to perform a better job than swap moves in terms of recovering feasibility with respect to balancing constraints, then if the sum of relative infeasibilities in a solution X is greater than a threshold ϵ (i.e., $f_1(X) + f_2(X) > \epsilon$), the neighborhood $N_A(X)$ is constructed, otherwise the neighborhood structure $N_A(X) \cup N_B(X)$ is then considered.

3.5 Recency-based memory and tabu tenure

Whenever a $move_A(i, k)$ or $move_B(i, j)$ is performed, any move that puts the BU i back into territory $t(i)$ or BU j back into territory $t(j)$ is declared tabu and forbidden for the next θ iterations. In this study θ is randomly selected in some interval $[\theta_{\min}; \theta_{\max}]$ following a uniform distribution, where $\theta_{\min}$ and $\theta_{\max}$ are user-defined parameters. Then, a new tabu tenure is generated for the attributes that becomes tabu at a given iteration. This is a well known strategy in TS literature to induce a good balance between intensification and diversification (see [14]). The tabu status of a move is overridden making it allowable if such move leads to a new best solution.

3.6 Candidate list strategy

Considering every possible move from the current solution to select the best neighbor may be extremely time consuming and computationally expensive. A list of candidates is proposed were only those moves involving the k_1 worst territories with respect to quality (i.e., those with larger demand) and the worst k_2 with respect to infeasibility (i.e., those with the highest values $f_1 + f_2$) are considered. The aim of this selection is to avoid performing moves that do not reflect a significant improvement in the evaluation of the merit function.

4 Computational evaluation

In this section we present the preliminary experimental results on the strategic oscillation and the neigborhood exploration mechanism in our local search algorithm. The proposed TS was coded in C++ and all experiments were carried out on a Intel Core i5, 2.30 GHz computer. For the experiments, we generated randomly problem instances based on real-world data provided by the industrial partner. This data set is generated according to the characteristics described in [15]. In that work, full details on how the instances are generated can be found. We experimented with 10 instances of each size $(n, p) \in \{(100, 6), (500, 10)\}$. For all instances, the allowable deviation for the balancing constraints was set at 5% ($\tau = 0.05$). For this preliminar study, the random demand is described by 10 discrete scenarios, each with a fixed probability. The scenario probabilities are: 0.01, 0.04, 0.15, 0.02, 0.34, 0.14, 0.09, 0.1, 0.06, 0.05. The process terminates when $maxIter_1$ iterations have been performed or when $maxIter_2$ consecutive iterations have been performed without improving the best solution found during the search. The parameters of the algorithm were set as described in Table 1.

(n,p)	T	δ	α	λ	ψ	$\theta_{\min}$	$\theta_{\max}$	ϵ	r	r^*	k_1	k_2	$maxIter_1$	$maxIter_2$
(100,6)	200	0.5	0.4	0.7	2	5	10	0.003	10	3	2	2	1000	250
(500,10)	150	0.5	0.4	0.7	2	5	10	0.003	10	3	4	4	1000	250

Table 1. Algorithm parameters.

4.1 Strategic oscillation efficiency

The improvement produced when the strategic oscillation is incorporated into the local search is first addressed. To this aim, both penalty coefficients β_1 and β_2 in the merit function (Equation (7)) are fixed to a value φ (1,10,100) and solutions obtained (NSO solutions) are compared with those generated when the penalty coefficients are self-autoadjusted (SO solutions).

Instance	SO	NSO			RD(%)		
		$\varphi = 1$	$\varphi = 10$	$\varphi = 100$	$\varphi = 1$	$\varphi = 10$	$\varphi = 100$
1	24.3234	-	25.2348	24.9342	-	3.7470	2.5112
2	22.0237	-	23.149	25.2034	-	5.1095	14.4376
3	24.9014	-	26.7921	28.2347	-	7.5927	13.3860
4	25.6236	26.8923	29.1356	26.1073	4.9512	13.7061	1.8877
5	23.1421	-	24.2997	24.7231	-	5.0021	6.8317
6	24.3513	-	24.9762	27.1892	-	2.5662	11.6540
7	27.6923	-	27.6923	27.9217	-	0.0000	0.8284
8	20.2892	-	22.2349	22.8592	-	9.5898	12.6668
9	24.168	27.02334	23.9152	23.7462	11.8145	-1.0460	-1.7453
10	24.2107	-	25.8551	24.5266	-	6.7920	1.3048

Table 2. Performance of strategic oscillation for 100-node instances.

Instance	SO	NSO		RD(%)	
		$\varphi = 10$	$\varphi = 100$	$\varphi = 10$	$\varphi = 100$
1	17.4282	19.3927	20.2254	11.2720	16.0499
2	15.3521	17.2589	17.8924	12.4205	16.5469
3	18.9346	23.0282	24.4627	21.6197	29.1958
4	17.5503	20.4339	19.2389	16.4305	9.6215
5	19.8923	21.9934	23.4602	10.5624	17.9361
6	18.0034	21.3467	21.5469	18.5704	19.6824
7	17.3943	19.3992	20.0429	11.5262	15.2268
8	17.7723	20.389	19.4533	14.7235	9.4585
9	16.5814	19.5496	19.2369	17.9008	16.0149
10	20.4502	24.2421	25.9085	18.5421	26.6907

Table 3. Performance of strategic oscillation for 500-node instances.

Results of the empirical comparison for 100 nodes are summarized in Table 2. Columns 2 to 5 show the objective function value of solutions obtained from SO and NSO strategies. A relative deviation (RD) between SO and NSO solutions is computed as $100 \times [(sol(\text{NSO}) - sol(\text{SO}))/sol(\text{SO})]$. As it can be noticed, in 90% of the instances the quality of the SO solutions is better than NSO solutions for $\varphi = 10$ and 100 with average relative improvement of 5.30% and 6.37% respectively. For the smallest penalty coefficient value ($\varphi = 1$) no feasible solutions were identified in 80% of the instances. This superiority in the quality of the solutions generated by using strategic oscillation is better apreciated in

Table 3 for 500-node instances where the relative average RD are 15.35% and 17.64% for $\varphi = 10$ and $\varphi = 100$ respectively.

4.2 Evaluation of the neighborhood exploration strategy

The efficiency of using a dynamic neighborhood structure that variates depending on the level of relative infeasibility at each iteration is tested. As in the previous experiment, solutions obtained by using a dynamic neighborhood exploration strategy (DN) are compared with those obtained when the proposed TS procedure uses an static neighborhood (SN). Then, neighborhood $N_A(X)$ is considered for the first 500 iterations, after that the neighborhood $N_A(X) \cup N_A(X)$ is used. Computational results presented in Table 4 confirm the solution quality improvement when the neighborhood structure is chosen by using a threshold ϵ for the relative infeasibility. The relative deviations (computed as in the previous test) between both DN and SN solutions are on average 1.76% and 15.32% for 100 and 500 node instances, respectively, which is a remarkably high improvement particulary in the case of 500-node instances.

Instance	(n,p)=(100,6)			(n,p)=(500,10)		
	SN	DN	RD(%)	SN	DN	RD(%)
1	24.3234	24.7821	-1.8509	19.4356	17.4282	11.5181
2	22.0237	20.9237	5.2572	17.1267	15.3521	11.5593
3	24.9014	25.6221	-2.8128	20.1514	18.9346	6.4263
4	25.6236	25.6082	0.0601	20.6567	17.5503	17.7
5	23.1421	22.0732	4.8425	23.0561	19.8923	15.9046
6	24.3513	23.1892	5.0114	22.3783	18.0034	24.3004
7	27.6923	26.6351	3.9692	22.7923	17.3943	31.0332
8	21.6371	22.3146	-3.0361	19.0342	17.7723	7.1004
9	24.168	23.8375	1.3865	18.5168	16.5814	11.6721
10	24.2107	23.1003	4.8069	23.7197	20.4502	15.9876

Table 4. Comparison between static and dynamic neighborhood exploration strategies.

5 Conclusions and work in progress

In this paper, we introduced the stochastic version of a commercial territory design problem. A tabu search algorithm is presented for this problem. To the best of our knowledge the demand uncertainty has not been treated before in commercial territory design. The proposed TS procedure incoporates sophisticated mechanisms such as strategic oscillation, which constitute the core of many adaptive memory programming algorithms, and strategies for neighborhood exploration. Preliminary experimental results show the positive impact of these mechanisms as the quality of the solutions is significantly improved when these mechanisms are incorporated into the local search phase.

Several issues are still work in progress. Currently we are working on the calibration of parameters, we are studying the self-tuning of the algorithm parameters during the search according to evolution status. We plan to execute a more extensive experimentation considering instances of 1,000 BUs and up as well as increasing the number of demand scenarios. The study of the behavior of

the other components of the local search such as the tabu tenure which can be adjusted during the search according to the search history is to be done. The incorporation of long-term memory strategies to guide the search into unexplored regions of the solution space (diversification) and performing a more thorough examination in some good or promising regions (intensification) may be needed as well.

Acknowledgments: This research has been supported by the Mexican National Council for Science and Technology (grant CB-2011-01-166397). In addition, the first author is supported by a fellowship for graduate studies from CONACYT.

References

1. Hojati, M.: Optimal political districting. Computers & Operations Research **23**(12) (1996) 1147–1161
2. Ricca, F., Simeone, B.: Local search algorithms for political districting. European Journal of Operational Research **189**(3) (2008) 1409–1426
3. Mehrotra, A., Johnson, E.L., Nemhauser, G.L.: An optimization based heuristic for political districting. Management Science **44**(8) (1998) 1100–1113
4. Bozkaya, B., Erkut, E., Laporte, G.: A tabu search heuristic and adaptive memory procedure for political districting. European Journal of Operational Research **144**(1) (2003) 12–26
5. Zoltners, A.A., Sinha, P.: Sales territory design: Thirty years of modeling and implementation. Marketing Science **24**(3) (2005) 313–332
6. Fleischmann, B., Paraschis, J.N.: Solving a large scale districting problem: A case report. Computers & Operations Research **15**(6) (1988) 521–533
7. Drexl, A., Haase, K.: Fast approximation methods for sales force deployment. Management Science **45**(10) (1999) 1307–1323
8. Kalcsics, J., Nickel, S., Schröder, M.: Toward a unified territorial design approach: Applications, algorithms, and GIS integration. TOP **13**(1) (2005) 1–56
9. Duque, J.C., Ramos, R., Suriach, J.: Supervised regionalization methods: A survey. International Regional Science Review **30**(3) (2007) 195–220
10. Glover, F.: Tabu search – Part I. ORSA Journal on Computing **1**(3) (1989) 190–206
11. Glover, F., Laguna, M.: Tabu Search. Kluwer, Boston, EUA (1997)
12. Erkut, E., Ülküsal, Y., Yeniçerioğlu, O.: A comparison of p-dispersion heuristics. Computers and Operations Research **21**(10) (1994) 1103–1113
13. Glover, F., Hao, J.K.: The case for strategic oscillation. Annals of Operations Research **183**(1) (2011) 163–173
14. Glover, F., Laguna, M., Martí, R.: Principles of tabu search. In T.González, ed.: Approximation Algorithms and Metaheuristics. Chapman & Hall/CRC (2007) 1–12
15. Ríos-Mercado, R.Z., Fernández, E.: A reactive GRASP for a commercial territory design problem with multiple balancing requirements. Computers & Operations Research **36**(3) (2009) 755–776

GRASP + *Path Relinking* aplicado al problema del *Cyclic Bandwidth*

Alfonso Fernández, David Candelas

Dpto. Ciencias de la Computación
Universidad Rey Juan Carlos
28933 Móstoles

alfonso.fernandez@urjc.es
dcandelas1989@gmail.com

Abstract. El problema del *Bandwidth Reduction* de una matriz cuadrada consiste en encontrar una permutación de sus filas y sus columnas, de tal forma que sus elementos no nulos se encuentren lo más cerca posible de la diagonal principal. La resolución del problema se plantea como un etiquetado de los vértices de un grafo (que representa a la matriz), donde el objetivo perseguido es minimizar la máxima diferencia entre las etiquetas de los vértices adyacentes. En este trabajo vamos a proponer un algoritmo que, basado en un constructivo *GRASP*, aplica una búsqueda local basada en un esquema de *Path Relinking* para resolver una variante de esta problemática, denominada *Cyclic Bandwidth*. Los resultados obtenidos confirman la idoneidad del método desarrollado para tratar este problema.

1 Introducción

El problema de la reducción del ancho de banda (más conocido como *Bandwidth Reduction*) tiene su origen en la década de los 60, basado en la necesidad de resolución de sistemas complejos de ecuaciones, siendo planteado a mediados de esta década por Harper [1] y por Harari [2]. Dichos sistemas de ecuaciones podían ser representados como una matriz cuadrada, donde cada columna representaba una de las variables del problema y cada fila una de las ecuaciones. Dado que las herramientas computaciones empleaban gran cantidad de tiempo en encontrar una solución, se hacía necesario un pre-tratamiento del sistema de ecuaciones con objeto de simplificar su grado de complejidad. Este nivel de dificultad podía ser reducido si se conseguía manipular la matriz de tal forma que los elementos no nulos de la misma estuvieran lo más cerca posible de la diagonal principal. No solo el problema representa la resolución de un sistema de ecuaciones sino que tiene aplicaciones más allá

de las matemáticas como son el diseño de sistemas de transmisión de energía a gran escala o la planificación de circuitos electrónicos VLSI o incluso en labores de recuperación de información en hipertextos en procesamiento del lenguaje [2].

Descripción del problema.
Para definir el ancho de banda de una matriz $M = \{m_{ij}\}_{n\,x\,n}$ es necesario conocer el ancho de banda de cada una de sus filas. Entendemos por dicho ancho de banda como el máximo de las distancias de los elementos no nulos de esta fila a la diagonal principal. En base a esto, el ancho de banda de la matriz será el mayor de los anchos de banda de todas y cada una de las filas que componen la matriz:

$B_i(M) = max\ \{|i-j|: m_{ij} \neq 0\}$, sería el ancho de banda de la fila i-esima
$B(M) = max\ \{B_i(M): i=1..n\}$, sería el ancho de banda de la matriz

El problema será encontrar una permutación de las filas y las columnas de M, de tal manera que minimicemos el ancho de banda de la matriz XMX^{-1}, donde $X \in \Re^{nxn}$ es una matriz de permutación.
Una manera sencilla y que facilita en medida la modelización de este problema es la equiparar el sistema de ecuaciones a un grafo no dirigido y no ponderado. En dicho grafo, la matriz de adyacencias representaría el sistema de ecuaciones y el problema se transformaría en etiquetar los vértices del grafo de tal forma que minimicemos las diferencias entre las etiquetas de los vértices del grafo. De esta manera podemos expresar el problema de la reducción del ancho de banda del grafo $G(V,A)$, donde V sería el conjunto de vértices del grafo y A el conjunto de aristas, como una biyección entre V y $\{1,..,n\}$. El *bandwidth* para un etiquetado f sería:

$B(f,G) = max\ \{|f(j)-f(i)|: (i,j) \in A\}$, sería el ancho para un etiquetado f

$B(G) = min\ \{B(f,G): \forall f\}$, sería el ancho de banda del grafo G

siendo $f(u)$ la etiqueta del vértice u para el etiquetado f. Esta es la notación que introdujo Harper en 1985 [3].
El problema del *Bandwidth Reduction* ha sido ampliamente estudiado, siendo Papadimitrou [4] el que demostró su dificultad NP-Completa,

justificando la necesidad de desarrollar un algoritmo heurístico para su resolución.
En nuestro caso nos centraremos una variante del problema anteriormente presentado denominada *Cyclic Bandwidth Reduction* (*CBR*). Esta versión del problema fue introducida por Leung y colaboradores en 1984 [5] y en este caso el objetivo es minimizar la máxima diferencia entre dos vértices adyacentes en un etiquetado circular del grafo. El *bandwidth* del grafo *G(V,A)* se calcularía de igual forma que en el caso del *bandwidth* normal

$B_c(f,G)= max \{|f(j)-f(i)|_c: (i,j) \in A\}$, sería el ancho para un etiquetado *f*

pero, en este caso $|f(j)-f(i)|_c$ sería

$|f(j)-f(i)|_c=min\{|f(j)-f(i)|,n-|f(j)-f(i)|\}$

definiéndose ahora el *bandwidth* de la matriz como

$$B_c(G)= min \{B_c(f,G): \forall f\}$$

La aplicaciones a este subproblema del *bandwidth* las encontramos en el diseño de redes circulares y de estructuras de datos. Además se ha encontrado [6] que

$$B(G)/2 \leq B_c(G) \leq B(G)$$

demostrando Lam y colaboradores [7] que su dificultad es similar a la del *bandwith* general.

Estado del arte.
A pesar de su relevancia teórica y práctica, la importancia dada a este subproblema respecto a la prestada al *bandwidth* general ha sido mucho menor. De lo encontrado hasta ahora podemos destacar que en 2002 Zhou [8] propuso un método para obtener cotas inferiores para el problema del *Bandwidth* y del *Cyclic Bandwidth*. Más tarde, de Klerk et al. [9] propusieron dos nuevos métodos de relajación para el *Bandwidth* y

el *Cyclic Bandwidth* basándose en una reformulación de un antiguo problema de asignación cuadrática (*QAP*). Anteriormente a todos ellos, en 1995, Yuan y Zhou [10] demostraron que para los grafos de intervalos unitarios, existe un etiquetado que es, al mismo tiempo, óptimo para los siguientes seis problemas de etiquetado: *Bandwidth, Cyclic Bandwidth, Profile, Cutwidth, Modified Cutwidth y Sum Bandwidth*. En las últimas fechas, otros trabajos como el de Romero-Monsivais y colaboradores [11], han propuesto un *Branch and Bound* con el objetivo de resolver esta problemática.

2 Metaheurísticas GRASP y *Path Relinking*

Metaheuristica GRASP.
GRASP es una técnica de diseño de algoritmos englobada dentro de las metaheurísticas multi-arranque y orientada para problemas de optimización combinatoria [12]. Cualquier algoritmo GRASP tiene dos fases bien definidas:

• Construcción: esta fase se diseña en base a un algoritmo iterativo en el que, partiendo de una solución vacía y, tras un número de pasos, se va conformando la solución. Dicho proceso es voraz, porque en cada paso intenta incluir en la solución aquella opción que mejor resultado dé. Sin embargo, y para que el proceso no sea determinista, se incluye una componente estocástica. Esta variante aleatoria implica la construcción, en cada iteración, de un conjunto de elementos susceptibles de ser incorporados a la solución que se está construyendo. Dicho conjunto se denomina Lista de Candidatos Restringidos (*RCL*). En cada iteración, se construye una nueva lista de la cual se elige aleatoriamente uno de los elementos, que pasa a formar parte de la solución que se está construyendo. Cada una de estas incorporaciones influirá tanto en los elementos incluidos en la solución como en los candidatos a formar parte de ella, de ahí que el proceso sea adaptativo.
• Mejora: en esta fase se intenta optimizar la solución obtenida en la fase anterior. Para ello se suele recurrir a procedimientos de búsqueda local o a hibridaciones con metodologías más elaboradas. En nuestro caso implementaremos dos sencillos procedimientos de mejora basados en estrategias de inserción e intercambio.

Metaheuristica *Path Relinking*.

El algoritmo de *Path Relinking* es una metaheurística evolutiva basada en los principios del *Scatter Search*, formulados originalmente por Glover [13] en la década de los 70. Dicha metaheurística remplaza el espacio euclídeo de la búsqueda dispersa y lo sustituye por vecindandes como base de una técnica combinatoria.
El método trabaja con un conjunto de soluciones denominado *RefSet* y utiliza técnicas combinatorias y de mejora para generar nuevas soluciones más eficientes a partir de las existentes. Dichas soluciones pueden provenir de una fase constructiva anterior, aplicando *Path Relinking* mecanismos de diversificación e intensificación para actualizar ese conjunto referencia de partida.
La base del *Path Relinking* es la explorar trayectorias en el espacio de soluciones. Por ello se parte de una de las soluciones del conjunto de referencia (solución inicial) y, teniendo como objetivo alcanzar otra de las soluciones de dicho conjunto (solución guía), se genera un camino de soluciones entre ambas. Dichas soluciones intermedias son obtenidas gracias a sucesivos cambios en los atributos de la solución inicial por atributos de la solución guía. De esta manera es posible encontrar entre estas soluciones intermedias, mejores soluciones que las ya obtenidas hasta entonces. Dichas soluciones mejoradas pueden pasar a formar parte del conjunto de referencia.
Variaciones a dicho método podemos encontrar desde intercambiar el papel de la solución inicial y la guía a la de generar caminos entre múltiples padres.

3 Algoritmo Desarrollado

Una vez presentadas las bases de nuestro trabajo, vamos a detallar como ha sido implementado nuestro algoritmo. Dicho método consta de dos partes principales: parte constructiva, inspirada en una estrategia GRASP y una parte de mejora, basada en una estrategia *Path Relinking*. Vamos a detallar brevemente cada una de las secciones.

Parte Constructiva.
La base de esta parte constructiva se fundamenta en la elección aleatoria de un vértice a etiquetar, escogido de la lista de candidatos confeccionada para cada iteración, excepto en la primera iteración, donde el nodo a marcar con la etiqueta 1 será seleccionado aleatoriamente. Dicha lista de candidatos (*CL*) va a estar formada, en cada iteración, por

aquellos nodos adyacentes al último nodo etiquetado que no han sido etiquetados, así como por aquellos nodos adyacentes a los nodos ya etiquetados y que todavía no tienen su correspondiente etiqueta. A continuación, seleccionamos el nodo a etiquetar siguiendo el siguiente criterio. De todos los candidatos que conforman la *CL*, escogemos un porcentaje arbitrario de los mejores, siendo estos los vértices cuya mayor diferencia entre sus nodos adyacentes etiquetados sea mayor que un cierto valor dado. De entre estos mejores candidatos, etiquetamos uno al azar, actualizando la lista de candidatos para la siguiente iteración. Este proceso se repite hasta que no quede ningún nodo por etiquetar. A continuación se aplica un proceso de búsqueda local, implementado con dos sencillos métodos basados en estrategias de sustitución o intercambio de etiquetas entre nodos.

Fase de Mejora.
Una vez finalizada la fase constructiva y, disponiendo ya de una solución, dos técnicas basadas en una estrategia *Path Relinking* fueron implementadas.

Path Relinking 1.
Este primer método basado en *Path Relinking* trata de encontrar una mejor solución en la trayectoria que une dos soluciones óptimas generando el camino que lleva a la una a la otra mediante permutaciones. Iterativamente y por orden, vamos colocando los nodos de la solución inicial en igual disposición que los de la solución guía mediante intercambios, hasta llegar a dicha solución final. A cada una de estas soluciones intermedias, se les aplica una búsqueda local. Si el resultado de esa búsqueda local es una solución mejor, dicha solución pasaría ser etiqueta como tal. Se muestra un ejemplo gráfico en la Figura 1:

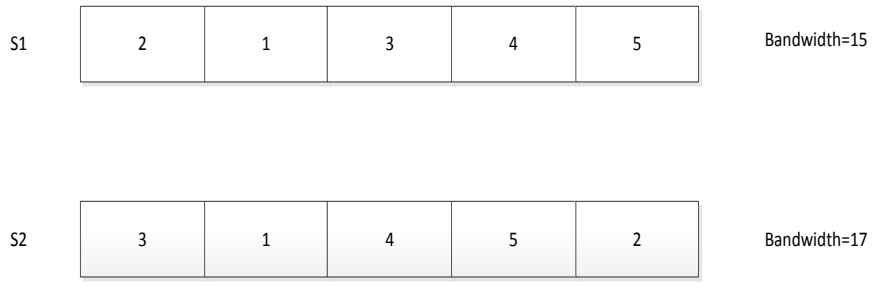


Fig. 1. Dos soluciones al problema del cyclic *bandwidth* de un grafo con 5 nodos.

La secuencia de pasos que daría nuestro método *Path Relinking*1 aparecen expuestos en la Figura 2:

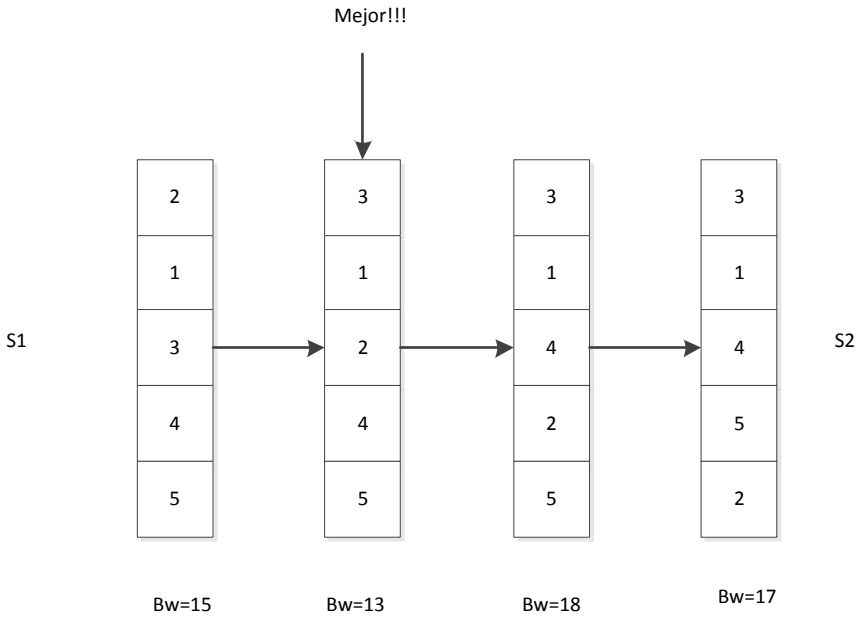


Fig. 2. Ejemplo *Path Relinking*1

En nuestro caso, no solo vamos a buscar la trayectoria entre dos soluciones óptimas si no que nos vamos a quedar con las 10 mejores soluciones al problema obtenidas mediante el procedimiento GRASP y vamos a aplicar el *Path Relinking* para la combinación de todas ellas.

Path Relinking 2.
Este método se diferencia del anterior en que exploramos un mayor número de soluciones intermedias. En cada iteración, colocamos cada una de las etiquetas de la solución guía en la solución de partida obteniendo *n* soluciones intermedias. De esas soluciones nos quedamos con la que tenga mejor *bandwidth*. De nuevo, volvemos a colocar todas las etiquetas obteniendo *n-1* soluciones intermedias (no podemos a evaluar otra vez la misma ya que no avanzaríamos) volviéndonos a quedar con la que tenga un mejor *bandwidth* y así sucesivamente hasta que llegamos a la solución guía. La mejor solución será aquella solución inter-

media que tenga el menor *bandwidth* de todas. Al igual que en el método anterior calcularemos las trayectorias para las 10 mejores soluciones. La Figura 3 muestra esta secuencia de pasos del método *Path Relinking* 2:

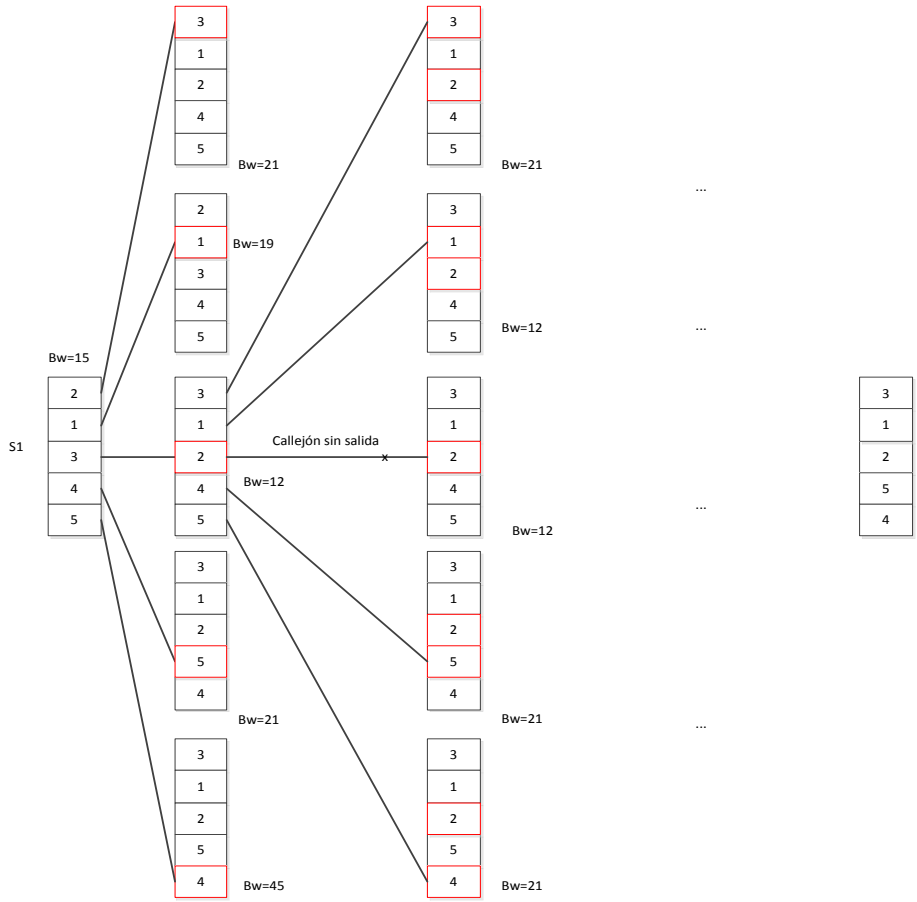


Fig. 3. Ejemplo *Path Relinking*2

4 Resultados

Para la realización de este trabajo, se emplearon una serie de datos tomados del famoso repositorio de Harper y Boeing del NIST, disponible en:*http://math.nist.gov/MatrixMarket/data/Harwell-Boeing/*. Hemos elegido conjuntos conformados por un número de instancias inferior a

300 elementos. Hemos desestimado el trabajar con bases de datos grandes, siendo esta una de las tareas a realizar en futuros trabajos.
El procedimiento anteriormente presentado fue desarrollado en lenguaje Java, llevándose a cabo las ejecuciones en un ordenador Pentium IV Core 2 Duo 2.27 GHz y 2 GB de memoria RAM.
La Tabla 1 muestra las características de cada matriz junto con los resultados obtenidos al aplicar nuestros algoritmos.

Instancia		Bandwidth	GRASP + PR
Nombre	**Nodos**		
can___24	24	11	7
can___61	61	42	14
can___62	62	12	12
can___73	73	22	24
can___96	96	17	27
can__187	187	33	16
can__229	229	120	73
can__256	256	148	97
can__268	268	165	98

Tabla 1. Resultados obtenidos en este trabajo, comparados con los del trabajo de Czibula y colaboradores [14] para el problema de *Bandwidth* general.

Como podemos observar, los resultados obtenidos en este trabajo se encuentran en su mayoría y teniendo en cuenta las cotas marcadas por el trabajo de Zhou [8], dentro de los márgenes obtenidos a partir de los resultados de Czibula y colaboradores [14]. No ha sido posible comparar nuestros resultados con los obtenidos en otros trabajos que trabajasen con la misma problemática al ser este un problema con reciente interés adquirido.

5 Conclusiones y futuros trabajos

En este trabajo hemos desarrollado un algoritmo GRASP con una fase de mejora basada en un esquema *Path Relinking* y lo hemos aplicado sobre el problema del *Cyclic Bandwidth*. Los resultados, pertenecientes a un estudio aun en una fase preliminar, muestran que el algoritmo diseñado funciona adecuadamente para este tipo de problemas. Queda

pendiente para un futuro la consolidación del procedimiento desarrollado y su aplicación a otros problemas más complejos.

6 Referencias

1. Harper, L.H.(1966). Optimal numberings and isoperimeter problems on graphs, J. Combin. Theory 1, 385-393.
2. Harary, F. (1967) Problem 16, in: M Fiedler (Ed.), Theory of Graphs and its Applications, Czechoslovak Academy of Science, Prague, p. 161.
3. Harper, L.H. (1985). Optimal numberings and isoperimetric problems on graphs. Journal of Combinatorial Theory, 1:385-393.
4. Papadimitriou, C.H. (1976). The NP-completeness of the bandwidth minimization problem. Computing, 16:263-270
5. Leung, J., Vornberger, O. and Witthoff, J.D. (1984). On Some Variants of the Bandwidth Minimization Problem. SIAM Journal of Computing, 13:650-667.
6. Lai, Y.L. and Williams, K. (1999). A survey of solved problems and applications on bandwidth, edge-sum, and profile of graphs. Journal of Graph Theory, 31:75-94.
7. Lam, P., Shiu, W. and Chan, W. (2002). Characterization of graphs with equal bandwidth and cyclic bandwidth. Discrete Mathematics, 242:283-289.
8. Zhou, S.: Bounding the bandwidths for graphs. Theoretical computer science 249(2) (2002) 357-368.
9. de Klerk, E., Eisenberg-Nagy, M., Sotirov, R.: On semidenite programming bounds for graph bandwidth. Technical report, Centrum Wiskunde & Informatica, (2011)
10. Yuan, J., Zhou, S.: Optimal labelling of unit interval graphs. Applied Mathematics, A Journal of Chinese Universities 10B(3) (1995) p. 337-344.
11. Romero, H., Rodriguez, E. and Ramirez, G. A New Branch and Bound Algorithm for the Cyclic Bandwidth Problem. MICAI (2) (2012), p. 139-150.
12. Feo, T and Resende, M. Greedy adaptive search procedures. Journal of Global Optimization 6 (1995) 109-133.
13. Laguna, M., Glover, F. and Marti, R. Fundamentals of Scatter Search and Path Relinking. Control and Cybernetics 29(3) (2000), p. 653-684.
14. G.Czibula, G-C.Crişan, C-M.Pintea, I-G.Czibula, Soft Computing approaches on the Bandwidth Problem, INFORMATICA 24(1) (2013), p.1-12

Un Algoritmo Voraz Iterativo con Memoria de Destrucciones Tabú para el Problema de las Múltiples Mochilas Cuadráticas

Carlos García-Martínez[1], Francisco J. Rodriguez[2], and Manuel Lozano[2]

[1] Dpt. de Informática y Análisis Numérico, Universidad de Córdoba, Córdoba 14071, España, cgarcia@uco.es
[2] Dept. Ciencias de la Computación e Inteligencia Artificial, Universidad de Granada, Granada 18071, España

Resumen Los algoritmos voraces iterativos son metaheurísticas simples pero eficaces para atacar problemas combinatorios. Su flexibilidad permite incorporar componentes de otras metaheurísticas para generar modelos híbridos más eficaces. Proponemos el uso de un mecanismo de destrucción con memoria tabú para estos algoritmos, que almacena las componentes de la solución que se han eliminado recientemente y evita extraerlas nuevamente en las próximas iteraciones. El objetivo es ofrecer un nivel de diversificación y de éxito mayor que el del mecanismo de destrucción estándar, el cual selecciona componentes aleatoriamente.
Hemos llevado a cabo experimentos sobre el problema de las múltiples mochilas cuadráticas, para el que incorporamos un procedimiento de optimización local novedoso, para medir el beneficio de la propuesta. Los resultados muestran que el algoritmo voraz iterativo con memoria de destrucciones tabú, con el procedimiento de refinamiento local, es capaz de producir soluciones de calidad superior a las de los algoritmos estado-del-arte correspondientes.

Keywords: Algoritmos voraces iterativos, memoria tabú, problema de las múltiples mochilas cuadráticas

1. Introducción

Los algoritmos voraces iterativos (*iterated greedy* en inglés; IG) [7] son metaheurísticas simples y efectivas que han recibido especial atención por su alto rendimiento a la hora de resolver problemas combinatorios complejos. Éstos siguen un principio simple basado en la *destrucción parcial* y *reconstrucción heurística* de una sola solución. Durante la destrucción, algunas componentes de la solución son eliminadas de ésta, generando una solución incompleta. Seguidamente, la reconstrucción utiliza una función heurística para completar esta solución parcial. Adicionalmente, se suele introducir un operador de optimización local que refina las soluciones que visita el algoritmo.

En este trabajo extendemos el modelo IG estándar con una memoria de destrucciones tabú [3] que previene la extracción de componentes que fueron

eliminadas de la solución actual en iteraciones previas y recientes. El objetivo es dotar al modelo de un aporte extra de diversificación en la generación de soluciones parciales y así explorar el espacio de búsqueda más eficazmente.

Para su evaluación hemos considerado el problema de las múltiples mochilas cuadráticas (*quadratic multiple knapsack problem* en inglés; QMKP) [4]. Éste consiste en la asignación de un conjunto de objetos a un conjunto de mochilas con restricciones de capacidad, de forma exclusiva y con el objetivo de maximizar la suma de los beneficios de los objetos incluidos. Sin embargo, los beneficios se encuentran asociados no sólo a los objetos individuales, sino también a los pares de objetos que se insertan en la misma mochila. Este hecho hace que las técnicas tradicionales que no consideran las interacciones entre objetos no sean eficaces a la hora de tratar este problema [8]. Los experimentos desarrollados no sólo muestran que nuestra modelo, instancia de un algoritmo IG con memoria de destrucciones tabú para el problema QMKP, es capaz de atacar el problema exitosamente explotando el conocimiento inherente del problema, sino que además es muy competitiva frente a los modelos de la literatura.

El resto del artículo se estructura como sigue. En la Sección 2, describimos el problema QMKP y enumeramos las metaheurísticas propuestas para su solución. En la Sección 3, describimos la propuesta de un algoritmo IG con memoria de destrucciones tabú y su instanciación para resolver el problema QMKP. En la Sección 4, describimos los experimentos llevados a cabo para evaluar nuestra propuesta y los resultados obtenidos. La Sección 5 cierra el trabajo con la exposición de conclusiones y trabajos futuros.

2. El Problema de las Múltiples Mochilas Cuadráticas

En el QMKP [4], se dispone de un conjunto de n objetos y K mochilas. Cada objeto $i \in \{1, 2, \ldots, n\}$ tiene un beneficio p_i y un peso w_i, cada par de objetos (i y j) tiene un beneficio p_{ij}, y cada mochila $k \in \{1, 2, \ldots, K\}$ tiene una capacidad C_k. El beneficio p_{ij} asociado al par de objetos i y j se suma al beneficio total sólo si ambos objetos se introducen en la misma mochila. El objetivo introducir cada objeto en una mochila de forma que la suma de los pesos de los objetos en cada mochila k no exceda su capacidad C_k y el beneficio total de los objetos introducidos sea máximo. Formalmente, sean las variables binarias x_{ij} que indican si el objeto i se incluye o no en la mochila k, entonces el QMKP se puede formular como:

$$\text{Maximizar} \quad \sum_{i=1}^{n}\sum_{k=1}^{K} x_{ik}p_i + \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\sum_{k=1}^{K} x_{ik}x_{jk}p_{ij}, \tag{1}$$

$$\text{cumpliendo} \quad \sum_{i=1}^{n} x_{ik}w_i \leq C_k, \quad \forall k \in \{1, 2, \ldots, K\}, \tag{2}$$

$$\text{y} \quad \sum_{i=1}^{K} x_{ik} \leq 1, \quad \forall i \in \{1, 2, \ldots, n\}. \tag{3}$$

El QMKP aparece en muchas situaciones reales donde un conjunto de recursos con diferentes niveles de interacción entre ellos tienen que asignarse a diferentes tareas, por ejemplo, en la asignación de operarios (sus potenciales se consideran individualmente y en pares) a proyectos. EL QMKP es una extensión de otros dos problemas combinatorios, el problema de las múltiples mochilas y el de la mochila cuadrática. Al igual que sus predecesores, el QMKP es NP-duro [4].

Hiley y Julstrom [4] fueron los primeros en estudiar este problema y propusieron tres métodos para su solución: un procedimiento constructivo voraz (comentado en la Sección 3.1), un método de optimización local que extrae un conjunto de objetos de sus mochilas y después aplica el procedimiento constructivo anterior, y un algoritmo genético generacional. Singh y Baghel [13] y Sara y Sipahioglu [12] describen otros dos algoritmos genéticos que obtienen mejores resultados, respectivamente.

La propuesta más reciente para el QMKP, y estado del arte, es un algoritmo de colonia de abejas artificiales con un procedimiento de optimización local propuesto por Sundar y Singh [14]. Las abejas representan procedimientos de explotación de soluciones previas, representadas como fuentes de comida. Las abejas se clasifican en obreras, supervisoras o exploradoras. Inicialmente, cada abeja obrera se asocia a una fuente de comida generada mediante un procedimiento voraz aleatorio. En cada iteración, las abejas obreras exploran una nueva fuente de comida en la vecindad de su fuente asociada y la evalúan. Si la cantidad de néctar de la nueva fuente es superior a la de la fuente actual, la abeja obrera se traslada a la nueva fuente de comida abandonando la anterior, en otro caso permanece en la fuente anterior. Seguidamente, las abejas supervisoras exploran una nueva fuente en la vecindad de una fuente asociada a una obrera, y la reemplazan si ésta resulta ser mejor. Cuando una fuente de comida no se mejora durante un número de iteraciones, la abeja obrera correspondiente la abandona y se convierte en una exploradora. Esta exploradora genera una nueva solución y se asocia a ésta como obrera. En el caso del QMKP, a la hora de explorar una nueva fuente de comida en la vecindad de otra anterior, las abejas aplican un procedimiento de reemplazo de mochilas entre dos soluciones completas o un método de alteración, según un parámetro del algoritmo. Además, se aplica un proceso de optimización local cuando se generan soluciones suficientemente buenas.

3. Algoritmo Voraz Iterativo con Memoria de Destrucciones Tabú

El algoritmo IG comienza a partir de una solución completa generada mediante un procedimiento voraz (Sección 3.1). Después, produce iterativamente una solución parcial S_d en la fase de destrucción (Sección 3.2), que reconstruye aplicando un procedimiento voraz devolviendo una nueva solución completa S_c. Además, se suele aplicar un método de optimización local para mejorar las soluciones completas (Sección 3.3). Antes de continuar con la siguiente iteración, se decide si la solución generada tras el procedimiento de optimización

local se convierte en la nueva solución actual mediante un criterio de aceptación (Sección 3.4). El proceso se repite hasta que se alcance alguna condición de parada, por ejemplo, número máximo de iteraciones o un tiempo máximo de computación. Finalmente, se devuelve la mejor solución generada.

3.1. Procedimiento de Construcción Voraz

Hiley y Julstrom [4] describen un método constructivo para el QMKP basado en la *contribución* ($\Delta(i,k)$) y *densidad* ($D(i,k)$) que tiene un objeto i con respecto a una mochila k. Éstos se definen como la suma de los valores de beneficio asociados al objeto i y los que ya se encuentran en la mochila k, y su división por el peso del objeto i, respectivamente:

$$\Delta(i,k) = p_i + \sum_{j \in k} p_{ij}, \quad D(i,k) = \Delta(i,k)/w_i$$

El procedimiento de construcción voraz consiste en insertar iterativamente el objeto i, de entre los que no están en ninguna mochila, en la mochila k tal que maximiza el valor $D(i,k)$ y cumple que $w_i + \sum_{j \in k} w_j \leq C_k$. El proceso se repite hasta que no hay ningún objeto con una contribución $\Delta(i,k) \geq 0$ que entre en la mochila k. En cada iteración, el beneficio de la nueva solución se calcula eficientemente sumando la contribución $\Delta(i,k)$.

3.2. Memoria de Destrucciones Tabú

El objetivo de la fase de destrucción del modelo IG es el de ofrecer soluciones de partida a la fase de construcción de forma que se exploren diferentes regiones del espacio. El mecanismo de destrucción del algoritmo IG estándar elimina componentes aleatorios de la solución actual [10,11,16], pero en la literatura podemos encontrar mecanismos más elaborados. Por ejemplo, Fanjul-Peyro y Ruiz [1] proponen funciones heurísticas para la selección estocástica de elementos a eliminar para el problema de la planificación de trabajos en máquinas paralelas.

Con la intención de mejorar la diversidad de las soluciones parciales que genera la fase de destrucción, proponemos el uso de una memoria a corto plazo que evite que los elementos que han sido extraídos de la solución recientemente (y que con cierta probabilidad se insertan en la fase de construcción), vuelvan a ser eliminados en las siguientes iteraciones. Para ello, cada vez que una componente de la solución se extrae de ésta, la memoria de destrucciones tabú almacena esta componente. Entonces, el mecanismo de destrucción evita extraer aquellas componentes que se encuentra actualmente en la memoria de destrucciones tabú. El número de iteraciones que la componente permanece en la memoria viene especificado por un parámetro de usuario llamado *tenencia tabú*.

El mecanismo de destrucción con memoria tabú recibe la solución de donde se deben extraer algunas componentes (S), el número de componentes que se deben eliminar (n_d), el estado actual de la memoria tabú (L_{tabu}), y el valor de la tenencia tabú (tt); y devuelve una solución parcial (S_d) y el nuevo estado

de L_{tabu}. A la hora de elegir una componente a eliminar, se busca alguna que no se encuentre en L_{tabu}. Seguidamente, se marca la componente elegida en la memoria de destrucciones tabú y se le asigna el valor tt a su tenencia tabú. Finalmente, en cada iteración se decrementan los valores de tenencia tabú de las componentes almacenadas en L_{tabu}.

3.3. Métodos de Optimización Local

Sundar y Singh [14] proponen un método de optimización local que examina posibles intercambios entre objetos introducidos en alguna mochila y objetos que no, de forma que se seleccionen aquellos intercambios que mejoren la calidad de la solución sin violar las restricciones de capacidad de las mochilas. Tan pronto como se encuentre un intercambio favorable, el procedimiento de refinamiento local aplica dicho cambio, es decir, implementa la estrategia *primera-mejora*. En este estudio, referenciamos a este método como PrimerOL. Además, consideramos la aplicación de la estrategia *máxima-mejora*, y el procedimiento se referenciará como MejorOL.

Por otro lado, hemos diseñado otros dos métodos de optimización local que consideran todos los posibles intercambios entre cualquiera dos objetos del problema, bien estén asignados a una mochila o no[3]. Nuestra hipótesis es que estos métodos pueden ser beneficiosos para el QMKP porque consideran movimientos adicionales que los optimizadores anteriores no, en particular, el caso del intercambio de dos objetos que ya se encuentran incluidos en mochilas (diferentes). Hemos implementado otros dos métodos de refinamiento local con estos intercambios utilizando las estrategias primera-mejora y máxima-mejora. Los métodos se referenciarán como $PrimerOL*$ y $MejorOL*$, respectivamente.

Hiley y Julstrom [4] describen otro método de refinamiento local que extrae un número de objetos aleatorios de sus mochilas y después completa la solución siguiendo un orden aleatorio. Dado que el procedimiento descrito es similar en cierto grado a la operación básica de los algoritmos IG, no se ha estudiado entre los diferentes modelos de optimización local.

3.4. Criterio de Aceptación

Hemos explorado los dos siguientes criterios de aceptación para nuestro algoritmo IG con destrucciones tabú:

- *Reemplazar si mejor* (RM): La nueva solución se acepta sólo si es mejor que la solución actual [17].
- *Reemplazar siempre* (RS): Los modelos IG que usan el criterio RM pueden estancarse en óptimos locales por una falta de diversificación [9,10]. En el otro extremo encontramos al criterio reemplazar siempre, que siempre aplica el procedimiento de destrucción sobre la última solución generada, independientemente de su valor objetivo para el problema. Este criterio favorece

[3] Se puede encontrar la definición de la nueva métrica de contribución de intercambios en http://www.uco.es/grupos/kdis/kdiswiki/index.php/TIG-QMKP

claramente la diversificación sobre la intensificación, porque promueve una búsqueda estocástica sobre el espacio de óptimos locales.

4. Experimentos

En esta sección analizamos el rendimiento de nuestra propuesta de un algoritmo IG con destrucciones tabú para el problema QMKP, el cual referenciaremos a partir de ahora como TIG-QMKP. La Sección 4.1 describe el marco general de los experimentos llevados a cabo. El resto de las secciones muestran los resultados. Nuestra intención es: (1) comparar sus resultados con los de otras metaheurísticas para el QMKP (Sección 4.2) y (2) analizar las componentes algorítmicas de la propuesta que efectivamente explotan el conocimiento asociado al problema (Sección 4.3)

4.1. Marco Experimental

Los algoritmos se implementaron en C++ y compilaron con gcc 4.6.3. Los experimentos se ejecutaron en una computadora con procesador i7-930 2.8GHz Intel(R) Core(TM) (8MB de caché, 4 núcleos y 8 hilos de ejecución) con 12GB de RAM y sistema operativo Linux Fedora(TM) V15. Cada ejecución se configuró para usar un único hilo de ejecución. Se realizaron experimentos sobre las 60 instancias del problema usadas en [4,12,13,14], que se caracterizan por su densidad d (proporción de valores de beneficio por pares p_{ij}; $\{0{,}25, 0{,}75\}$), número de objetos n ($\{100, 200\}$), y número de mochilas K ($\{3, 5, 10\}$) (5 instancias por combinación de parámetros). Se asignaron las capacidades de las mochilas al 80 % de la suma de los pesos de los objetos dividido por el número de mochilas.

Todos los algoritmos se ejecutaron 40 veces sobre cada instancia del problema, consumiendo un máximo de 5 segundos para problemas con 100 objetos y 30 para problemas con 200 objetos.

4.2. Comparación con Otras Metaheurísticas

En esta sección, comparamos la propuesta TIG-QMKP con otras metaheurísticas para el QMKP: el modelo de programación entera [4,12] (fórmulas 1-3) analizado con CPLEX 12 y los modelos referenciados en la Sección 2, HJ-SHC [4], HJ-GCA [4] (el mejor de éstos dos), SS-GA [12], SB-SSGGA [13], y SS-ABC [14].

Previamente, se llevó a cabo un ajuste de parámetros para TIG-QMKP[4], obteniendo que la mejor combinación de estrategias y parámetros era: PrimerOL* como método de optimización local, RS como criterio de aceptación, n_d igual a $0{,}1n$ y tt igual a 7. Adicionalmente, los análisis estadísticos basados en test no paramétricos [2] mostraron que el uso de la memoria de destrucciones tabú ofrecía mejoras significativas frente al mecanismo de destrucción estándar del algoritmo IG.

[4] Se puede encontrar información acerca del ajuste de parámetros de TIG-QMKP en http://www.uco.es/grupos/kdis/kdiswiki/index.php/TIG-QMKP

La Tabla 1 muestra los resultados medios de los algoritmos en las instancias del problema (la media sobre 5 instancias y 40 ejecuciones por instancia). Los resultados de los algoritmos, excepto para CPLEX y TIG-QMKP, fueron recogidos de [14] (para algunas instancias del problema no se disponía de resultados de SS-GA). Ante la ausencia de indicaciones de un número máximo de evaluaciones en [14], hemos añadido los resultados de nuestra propuesta y de CPLEX habiendo finalizado sus simulaciones según el tiempo indicado en [14] para cada instancia del problema (la media se indica en la cuarta columna). CPLEX se ejecutó una vez por instancia del problema, permitiendo la compresión de la información generada y guardado en disco cuando se requería más de 400MB de memoria RAM. El carácter '-' indica que CPLEX no pudo ofrecer ninguna solución para la instancia correspondiente en el tiempo indicado. Los mejores resultados y desviación estándar de 40 ejecuciones se incluyen para la mejor propuesta de la literatura y TIG-QMKP. Los mejores resultados (media de todos los algoritmos y mejor valor entre SS-ABC y TIG-QMKP) se resaltan en negrita.

Cuadro 1. Resultados de los algoritmos para el QMKP (las ejecuciones de TIG-QMKP y CPLEX se truncaron según el tiempo indicado en [14])

Instance					HJ-	SS-	SB-S	SS-ABC			TIG		
n	K	C	Time	CPLEX	SHC[1]/GGA[2]	GA	SGGA	Best	Avg	SD	Best	Avg	SD
$d = 0{,}25$													
100	3	723,2	3.534	25737.4	26977.4[2]	26678.4	27369.4	28180	27705	261.09	**28221.8**	**28063.9**	**88.864**
100	5	433.8	2,894	17738.6	20544.2[2]	20219.8	20927	21709.2	21351	187.894	**21758.6**	**21602.4**	**86.916**
100	10	216.8	1.752	7341.67	13627[1]	14217.4	14579.8	15261	14854.3	195.352	**15549.4**	**15385.8**	**89.69**
200	3	1346.6	28.12	90347	100666[1]	79179.9	101396	**102796**	102281	289.634	102671	**102429**	**94.066**
200	5	807.8	21.66	62350.2	72912.4[1]	58069.6	74143.4	**76247.6**	75667.8	332.602	76095.2	**75760.7**	**153.542**
200	10	403.6	10.764	-	48490.4[1]	38933.6	49519	51318.6	50372.8	401.39	**52227.2**	**51779.1**	**174.93**
$d = 0{,}75$													
100	3	680.6	1.958	63757.8	68741.4[1]	65404.1	69270.8	69496.6	69039.6	267.8	**69602.4**	**69596.6**	**20.724**
100	5	408	1.916	41008	47955[1]	46395.9	48410	40323.2	48829.5	221.93	**49154.8**	**49022.5**	**52.596**
100	10	203.8	1.358	16794	29353.2[1]	28371	29500.2	30432.2	30013.2	195.798	**30439.6**	**30209.9**	**94.468**
200	3	1393.6	16.67	236357	261717[1]	254301	262755	264103	261867	1111.51	**264836**	**264836**	**104.39**
200	5	835.8	15.574	148983	177174[1]	170447	177860	180233	178872	747.896	**181095**	**180757**	**157.04**
200	10	417.6	9.732	-	105687[1]	100940	106452	107807	106641	576.592	**109197**	**108747**	**185.694**

La Tabla 1 muestra que TIG-QMKP obtiene mejores resultados medios que el resto de algoritmos en todos los problemas. Además, la media de sus mejores resultados son superiores a los de SS-ABC en 10 parametrizaciones del problema y su desviación estándar siempre es menor. La Tabla 2 analiza los resultados de TIG-QMKP frente a los de otros algoritmos de la literatura mediante los test estadísticos de Iman-Davenport, Holm y Wilcoxon. Para los dos primeros se consideraron dos conjuntos de problemas y algoritmos: uno, sólo con los problemas para los cuales todos los algoritmos ofrecen resultados (se muestra el ranking de los algoritmos en la columna Holm Todos Algs) y otro, sólo con los algoritmos que ofrecen resultados para todos los problemas (se muestra el ranking de los algoritmos en la columna Holm Todos Probs). El test de Iman-Davenport detectó diferencias significativas en los conjuntos de resultados (estadísticos 172,73 y 539,31, y valores críticos 2,14 y 2,41 con 0.05 como factor de significancia, respectivamente), y el test de Holm detectó diferencias significativas (destacadas en negrita) entre el mejor algoritmo en ambos casos (TIG-QMKP) y el resto

de modelos excepto SS-ABC para el conjunto con menos problemas. Para el test de Wilcoxon se calcularon los valores críticos según el número de instancias resueltas por cada algoritmo con 0.05 como factor de significancia (los valores R+ se asocian a TIG-QMKP). Observamos que el test de Wilcoxon encuentra diferencias significativas a favor de TIG-QMKP en todos los casos. Por tanto, concluimos que nuestra propuesta es muy competitiva con respecto al estado del arte para el QMKP.

Cuadro 2. Comparativa entre el estado del arte y TIG-QMKP

	Holm		**Wilcoxon**			
	Todos Algs	**Todos Probs**	**R+**	**R−**	**V. Crítico**	**Diferencias?**
TIG-QMKP vs ...	1.15	1.07				
... SS-ABC	2.06	**2.13**	1809	21	648	+
... SB-SSGGA	**3.24**	**2.85**	1830	0	648	+
... SS-GA	**4.62**	-	902	1	294	+
... HJ-SHC	**4.79**	**4.2**	1830	0	648	+
... HJ-GGA	**5.26**	**4.75**	1830	0	648	+
... CPLEX	**6.91**	-	1081	0	361	+

4.3. Explotación del Conocimiento del Problema

En esta sección, comparamos el rendimiento a lo largo de la ejecución de TIG-QMKP con el de algoritmos específicos con la intención de determinar las características de TIG-QMKP que lo hacen especialmente interesante para el problema QMKP. En particular, se analizarán las gráficas de convergencia de TIG-QMKP, un algoritmo IG estándar ajustado (IG), un IG estándar sin optimización local de soluciones (IG-NoOL), una búsqueda Tabú (Tabu), un enfriamiento simulado (SA), y una búsqueda Tabú y enfriamiento simulado comenzando a partir de una solución heurística (Tabu* y SA*, respectivamente). Los parámetros de los algoritmos fueron ajustados en experimentos preliminares.

La Figura 1 muestra el resultado medio y normalizado de los algoritmos a lo largo de 40 ejecuciones sobre las 60 instancias del problema QMKP.

Según estos resultados podemos destacar que:

1. Comenzar la búsqueda a partir de una solución generada mediante un procedimiento heurístico beneficia claramente a la metaheurística. Este hecho se observa al ver que SA y Tabu, las únicas metaheurísticas que no comienzan a partir de una solución heurística, no consiguen buenos resultados.
2. El uso del método de optimización local PrimerOL* eleva la calidad de las soluciones sin deteriorar la eficiencia del algoritmo. Esto se aprecia al notar que la diferencia entre los resultados de IG-NoOL e IG es prácticamente constante, y favorable a IG.
3. El avance progresivo de TIG, IG e IG-NoOL frente al estancamiento de Tabu* y SA* muestra que la reconstrucción iterativa y heurística de soluciones parciales es más efectiva que la exploración de vecindarios de propósito general llevada a cabo por Tabu* y SA* (y Tabu y SA).

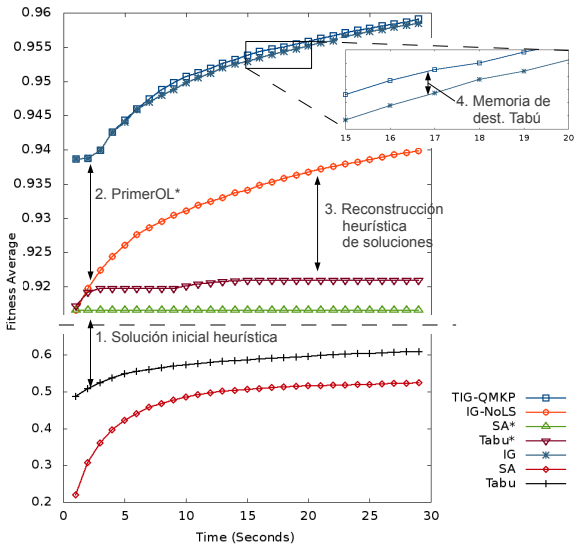


Figura 1. Gráficas de convergencia

4. Finalmente, el uso de la memoria de destrucciones tabú permite a TIG-QMKP mejorar aún más los resultados de un algoritmo IG estándar con ajuste de parámetros. Esta mejora revela sus beneficios después de unas iteraciones iniciales, cuando la memoria a corto plazo se ha inicializado apropiadamente y el mecanismo de destrucción explota dicha información para producir soluciones diversas en el espacio de búsqueda.

5. Conclusiones

Hemos presentado un algoritmo IG con memoria de destrucciones tabú que mejora el rendimiento del modelo IG estándar con la producción de soluciones parciales diversas. Esta memoria almacena las componentes que han sido recientemente eliminadas de la solución actual, y previene su eliminación en iteraciones próximas. Hemos analizado el rendimiento de la propuesta sobre el problema QMKP, para el cual además se han diseñado nuevos operadores de optimización local. A partir de los resultados, hemos concluido que el mecanismo de destrucción con memoria tabú ofrece mejoras sobre el mecanismo de destrucción estándar del IG. Además, el modelo final ha mostrado ser muy competitivo frente al estado del arte para este problema. Finalmente hemos analizado las componentes de la propuesta que explotan el conocimiento asociado al problema y consiguen ofrecer unos resultados superiores a otras propuestas.

Consideramos que la propuesta es merecedora de futuros estudios. En particular, nos planteamos el diseño de algoritmos IG con destrucciones tabú para otros problemas combinatorios, así como hibridar nuestro modelo con otras metaheurísticas para obtener resultados aún mejores.

Agradecimientos

Este trabajo ha sido financiado por los proyectos TIN2012-37930-C02-01/02 y P08-TIC-4173 del Ministerio de Economía y Competitividad y de la Junta de Andalucía, respectivamente.

Referencias

1. Fanjul-Peyro, L., Ruiz, R.: Iterated greedy local search methods for unrelated parallel machine scheduling, Eur. J. Oper. Res. 207 (1), 55–69 (2010)
2. Garcia, S., Molina, D., Lozano, M., Herrera, F. A Study on the Use of Non-Parametric Tests for Analyzing the Evolutionary Algorithms' Behaviour: A Case Study on the CEC'2005 Special Session on Real Parameter Optimization, J. Heuristics 15 (6) 617–644 (2009)
3. Glover, F., Laguna, M.: Tabu Search, Kluwer Academic Publishers, 1997.
4. Hiley, A., Julstrom, B.: The Quadratic Multiple Knapsack Problem and three heuristic approaches to it. In: Proc. of the Genetic and Evolutionary Computation Conference (GECCO), 2006, pp. 547–552.
5. Holm, S.: A simple sequentially rejective multiple test procedure, Scand. J. Stat. 6, 65–70 (1979)
6. Iman, R., Davenport, J.: Approximations of the critical region of the Friedman statistic. In: Communications in Statistics, 1980, pp. 571–595.
7. Jacobs, L., Brusco, M.: A local-search heuristic for large set-covering problems, Nav. Res. Log. 42, 1129–1140 (1995)
8. Julstrom, B.A.: Greedy, genetic, and greedy genetic algorithms for the quadratic knapsack problem. In: Proceedings of the 2005 conference on Genetic and evolutionary computation, GECCO '05, ACM, New York, NY, USA, 2005, pp. 607–614.
9. Lozano, M., Molina, D., García-Martínez, C. Iterated Greedy for the Maximum Diversity Problem, Eur. J. Oper. Res. 214 31–38 (2011)
10. Ruiz, R., Stützle, T.: A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. Eur. J. Oper. Res. 177, 2033–2049 (2007)
11. Ruiz, R., Stützle, T.: An Iterated Greedy heuristic for the sequence dependent setup times flowshop problem with makespan and weighted tardiness objectives, Eur. J. Oper. Res. 187, 1143–1159 (2008)
12. Saraç, T., Sipahioglu, A.: A Genetic Algorithm for the Quadratic Multiple Knapsack Problem. In: BVAI (LNCS 4729), 2007, pp. 490–498.
13. Singh, A., Baghel, A.: A New Grouping Genetic Algorithm for the Quadratic Multiple Knapsack Problem. In: EvoCOP (LNCS 4446), 2007, pp. 210–218.
14. S. Sundar, A. Singh, A Swarm Intelligence Approach to the Quadratic Multiple Knapsack Problem, in: ICONIP (LNCS 6443), 2010, pp. 626–633.
15. Wilcoxon, F.: Individual comparisons by ranking methods, Biometrics 1, 80–83 (1945)
16. Ying, K.-C.: Solving non-permutation flowshop scheduling problems by an effective iterated greedy heuristic, Int. J. Adv. Manuf. Technol. 38 (3–4), 348–354 (2008)
17. Ying, K.-C., Cheng, H.-M.: Dynamic parallel machine scheduling with sequence-dependent setup times using an iterated greedy heuristic, Exp. Syst. App. 37 (4) 2848–2852 (2010)

Foto-renderización con Inteligencia de Enjambre

Carlos M. Fernandes[1,2], Antonio M. Mora[1], Juan Julián Merelo[1], Agostinho C. Rosa[2]

[1] Departmento de Arquitectura y Tecnología de Computadores,
Universidad de Granada, Granada, España
[2] LaSEEB-ISR-IST, Technical University of Lisbon, Lisbon, Portugal
{cfernandes,acrosa}@laseeb.org
{amorag, jmerelo}@geneura.ugr.es

Resumen. Este artículo describe la aplicación de un algoritmo de inteligencia de enjambre en la generación de imágenes utilizando datos de fotografías. El algoritmo, llamado KANTS, se basa en los modelos artificiales de colonias de hormigas, y fue desarrollado originalmente como un método de agrupamiento (*clustering*) y clasificación de datos. KANTS consta de una población de vectores de datos de entrada que se comunican a través de una rejilla de vectores, mediante una actualización de dichos vectores. En los últimos trabajos, el algoritmo ha sido movido de su entorno científico inicial y utilizado como generador estocástico de imágenes. En dichos trabajos, se han utilizado diferentes tipos de datos como vectores de entrada. En este artículo se consideran valores RGB de fotografías para inicializar los vectores de entrada y los vectores de la rejilla, con el fin de generar diferentes versiones o transformaciones de dichas fotos. Llamamos a las imágenes *dibujos ferogénicos*, ya que los datos de salida utilizados para generarlos son los mapas de feromonas del algoritmo. De esta forma, el método se ha contextualizado como una herramienta creativa dentro del arte generativo.

1. Introducción

Los algoritmos de inteligencia de enjambre (*swarm intelligence*) [1] son sistemas auto-organizados en los que individuos simples interactúan a nivel local, llevando al sistema a la emergencia de patrones globales. La interacción puede estar restringida a la comunicación entre los individuos, o puede utilizar el entorno del sistema como un medio para la comunicación. Cuando los individuos interactúan con o a través del medio ambiente, el sistema se dice estigmérgico, un término introducido en [8] para describir la capacidad de los insectos sociales de utilizar el ambiente como medio de comunicación. Una de las clases principales de algoritmos de inteligencia de enjambre son los algoritmos inspirados en colonias de hormigas, en los que se encuadra KANTS, el algoritmo descrito en este trabajo.

KANTS es un algoritmo de hormigas propuesto en [12] para la agrupación (*clustering*) y clasificación de datos. El método se inspira en el Ant System [2], un modelo de colonia de hormigas descrito a través de conjunto de reglas locales. Estas reglas, tras ajustarse, conducen al enjambre hacia un estado de auto-organización en el que emergen patrones de comportamiento complejos.

En el Ant System, esos patrones se caracterizan por líneas de hormigas auto-organizadas y estables. En KANTS, las reglas del sistema fueron adaptadas con el objetivo de que las hormigas se auto-organicen en grupos de características similares, guiando al sistema hacia un estado de agrupamiento de datos.

Con este fin, en lugar de la rejilla homogénea de 2 dimensiones utilizada en el Ant System como hábitat para el enjambre, KANTS trabaja con una rejilla de 2 dimensiones en la que hay un vector de variables con valores reales asignados a cada celda; es decir, el entorno es una rejilla de vectores. Los agentes también se diferencian del Ant System, dado que KANTS utiliza los vectores de datos de entrada (con el mismo tamaño que los vectores del entorno) como hormigas artificiales. Éstas se mueven a través de la rejilla, ajustando los vectores que visitan de modo que se acerquen (sean más parecidos) a su propio vector. A su vez, las hormigas son atraídas hacia regiones donde hay vectores similares al suyo.

KANTS ha sido utilizado en los últimos años por los autores de este artículo como herramienta de arte de enjambre (*swarm art*). Más exactamente, el algoritmo fue utilizado en la generación de imágenes no figurativas de 2 dimensiones, usando diferentes tipos de conjuntos de datos correlacionados, tales como electroencefalogramas (EEG) [6], o valores de RGB de pinturas abstractas [7]. En este artículo, se estudia el potencial de KANTS como generador de imágenes a partir de los datos de fotografías en color. Además, en lugar de la estrategia de inicialización (aleatoria) utilizada en los trabajos anteriores, el entorno es ahora inicializado con los valores RGB de la misma fotografía que genera los datos de entrada. Por lo tanto, las hormigas se mueven en un entorno que está sesgado hacia la forma original y hacía la estructura cromática de la foto. La imagen resultante es una versión, en cierto modo, filtrada de la fotografía. Los efectos de los diferentes valores de los parámetros del algoritmo son estudiados y presentados en este artículo.

2. Conceptos principales y trabajos relacionados

El *arte de enjambre*, entendido como la generación de trabajos artísticos u objetos ornamentales mediante sistemas de inteligencia de enjambre, es un campo incluido dentro del llamado *arte generativo*. Este término se utiliza para clasificar creaciones artísticas que se han desarrollado principalmente por medio de sistemas basados en métodos de inteligencia artificial u otros modelos computacionales. Existen numerosos trabajos en este área, de modo que el arte generativo está dividido en otros campos, como la música artificial o el arte evolutivo. A continuación describiremos algunos de los trabajos más representativos en relación a los conceptos que tratamos en este artículo.

Del mismo modo que KANTS, los *dibujos de enjambre* de Leonel Moura [13] también están basadas en el Ant System. El citado autor comenzó experimentando con dibujos en la pantalla de su ordenador. Sin embargo, los resultados fueron decepcionantes hasta que decidió utilizar una máquina de diseño

gráfico y un pincel, y crear objetos físicos (reales). Desde entonces, Moura ha estado experimentando con enjambres, auto-organización y robótica [14].

Al igual que Moura, Monmarché et al. [11] han usado hormigas en su investigación del potencial de los enjambres como 'artistas no humanos'. Estos autores discuten el paradigma de las hormigas artificiales para generar dibujos y música. Usando terminología de Historia del Arte, los dibujos de Moura y Monmarché podrían catalogarse como abstractos, mientras que la propuesta de Collomosse [3], que utilizó Computación Evolutiva para evolucionar técnicas para el renderizado de fotografías es un trabajo figurativo.

Semet et al. [17] investigaron también la renderización automática, proponiendo un método no fotorrealista basado en hormigas artificiales. Dichas hormigas se mueven y perciben el entorno (la imagen), y depositan 'tinta' en una imagen de salida, considerando su posición y el estado de una pequeña memoria. El usuario interacciona con la colonia y determina los parámetros, define los 'mapas de importancia' y decide cuándo termina el renderizado.

En 2001, Ramos y Almeida [15] propusieron una modificación del Ant System en la que las hormigas se mueven sobre una imagen en escala de grises, detectando los bordes o fronteras a la vez que generan mapas de feromonas similares a un bosquejo de la imagen. Más tarde Fernandes et al. [4] describieron una extensión evolutiva del modelo. En 2010, Fernandes [5] introdujo el término *pherographia* (que significa dibujar con feromonas) como denominación para los mapas de feromonas de los algoritmos de hormigas y planteó una línea de proyectos creativos basados en la pherographia que derivaron en numerosos trabajos artísticos (ver [14], por ejemplo).

Estos son sólo algunos ejemplos de arte de enjambre, relacionados con nuestro trabajo. Sin embargo, hay diversas variantes de arte generativo, como se puede ver en la recopilación de Lewis [10], que analiza los métodos más importantes. Otra referencia relevante es el libro de Romero y Machado [16].

En el presente artículo, pretendemos contribuir a este campo utilizando el algoritmo KANTS, inicialmente creado como método de agrupación de datos, como herramienta de creación de arte de enjambre. Las creaciones que ofrece esta técnica se denominan *dibujos ferogénicos*, ya que se generan basándose en la feromona. Podrían considerarse una extensión de las citadas pherographias [5], dado que KANTS deriva del mismo sistema (Ant System), y las imágenes son en realidad los mapas de feromonas del algoritmo, una vez concluido su proceso (movimiento de las hormigas en el medio). Sin embargo, usamos el término ferogénicos para diferenciarlos del esquema propuesto en [15]. Además, la pherographia, entendida según Fernandes [5], da como resultado trabajos figurativos siempre, mientras que los dibujos de enjambre pueden producir creaciones abstractas si el medio (rejilla de vectores) se inicializa de forma aleatoria. KANTS utiliza datos correlacionados que interaccionan de forma pseudo-estocástica en un entorno heterogéneo. Los resultados son, por tanto, impredecibles, como se comentó anteriormente. Aun así, KANTS tam-

bién puede generar imágenes figurativas, si dicho entorno (rejilla) se inicializa de forma controlada. En la Sección 4 se describe un proyecto de arte de enjambre basado en fotografías de las que los dibujos ferogénicos resultantes son fotorrealistas. Pero antes, el algoritmo es descrito en la sección siguiente.

3. KANTS

El algoritmo KANTS [12] es un método de agrupamiento (clustering) y clasificación. Este término deriva de Kohonen y Ants, dado que el método está inspirado en parte en los Mapas Auto-organizativos de Kohonen (*Self-Organizing Maps o SOM*) [9] y, a su vez, se trata de un algoritmo basado en el Ant System [2], del que hereda algunos mecanismos. La principal diferencia respecto a este algoritmo es la forma en que se implementa la feromona, así como el entorno en el que se mueven las hormigas, siendo una rejilla cuadrada en ambos casos, pero contando en KANTS con un vector de pesos en cada celda.

Las ecuaciones en [2] fueron adaptadas en KANTS, para conseguir que las hormigas, que son vectores de datos de entrada, tiendan a moverse hacia otros patrones (hormigas) similares. Cuando una hormiga recorre una zona determinada del hábitat (*grid*), actualiza los vectores en ese punto y en un vecindario cercano para hacerlos más similares a su propio vector (que permanece constante durante toda la ejecución), reduciendo la distancia Euclídea entre ellos. Este enfoque se inspira en SOM. Sin embargo, existen varias diferencias con ese método, como las limitaciones de movilidad topológica que tienen las hormigas (sólo pueden moverse a celdas cercanas o directamente conectadas con la actual), o la pseudo-estocasticidad que tiene el método a la hora de seleccionar la siguiente celda a la que se moverá una hormiga.

Esta sección describe una versión simplificada de KANTS, en la que se han eliminado parámetros que no influyen en el desarrollo del arte de enjambre. El lector puede dirigirse a [12] para ver una descripción del KANTS original.

KANTS se basa en las propiedades emergentes de un conjunto de unidades simples (hormigas) que se mueven en una rejilla. El hábitat se mapea sobre una matriz de dimensiones $NxNxd$, donde d es la dimensión de los vectores de datos del problema, y NxN es la dimensión de la rejilla (número de celdas). Por tanto, cada celda tendrá asociado un vector con d variables. Las hormigas también 'cargan' con un vector que corresponde a un patrón de entrada (*sample*), habiendo tantas hormigas como patrones haya en el problema.

La idea principal del algoritmo es disponer de un conjunto de hormigas moviéndose y actualizando los vectores del hábitat (que tienen el mismo tamaño). Las dimensiones del grid sobre el que se mueven los patrones afectan al rendimiento del algoritmo respecto a las posibilidades de agrupamiento, siendo, en general, una buena proporción $:3$ o $:2$ respecto al número de hormigas. Los vectores del grid se inicializan con valores aleatorios con distribución uniforme en $[0,1]$. Posteriormente, los patrones son normalizados en

el mismo rango y se sitúan de forma aleatoria en el hábitat. En cada iteración, cada hormiga elige la siguiente celda a la que moverse (con cierta probabilidad) considerando las ecuaciones 1 y 2, inspiradas por el Ant System.

$$w(j) = \left(1 + \frac{\sigma}{1 + \delta\sigma}\right)^{\beta} \tag{1}$$

$$P_{i \to j} = \frac{w(\mathrm{j}).\,\mathrm{r(j)}}{\sum_{l \in Moore\ neig.} w(\mathrm{j})} \tag{2}$$

La ecuación 1 mide la probabilidad relativa de moverse a la celda *j* con una densidad de feromona s σ. El parámetro β ($\beta \geq 0$) está asociado con la llamada sensitividad osmotropotáxica. La osmotropotaxis fue reconocida por Wilson [18] como uno de los tipos de detección y procesado de feromona por parte de las hormigas y postula el seguimiento instantáneo de los gradientes de feromona. En otras palabras, β controla el grado de aleatoriedad con el que las hormigas seguirán un nuevo rastro de feromona. El parámetro δ ($\delta \geq 0$) define la capacidad sensorial ($1/\delta$) y modela la pérdida de sensibilidad que padecen las hormigas ante grandes concentraciones de feromona.

Esto significa que una hormiga tenderá a moverse, eventualmente, lejos de un rastro cuando el nivel de feromona alcanza una gran concentración. Esto se traduce en que la función que modela el comportamiento de las hormigas en relación al tiempo medio que están en un determinado rastro muestra grandes oscilaciones, debido a las variaciones en la concentración de feromona.

La ecuación 2 muestra la probabilidad de que una hormiga se mueva a una celda concreta *j* dentro del hábitat, dentro del actual vecindario de Moore. Ésta se define tras hacer una discretización del tiempo y el espacio: $P_{i \to j}$ es la probabilidad de moverse de la celda *i* a la *j*, *w(j)* se obtiene de la ecuación 1 y *r(j)* se fija a 1 si la celda j está dentro de un radio definido por el usuario alrededor de la celda *i* (o cualquier otro tipo de forma alrededor de la misma), y 0 en caso contrario. La densidad de feromona σ (ecuación 3) se define como la inversa de la distancia Euclídea entre el vector que lleva la hormiga *n*, v_{an} y el vector en la celda *(i,j)* en el momento *t*, $v_{cij}(t)$.

$$\sigma = \frac{1}{d(\vec{v}_{an}, \vec{v}_{cij}(t))} \tag{3}$$

De forma que una hormiga tiende a moverse a celdas cuyos vectores, que son modificados con el paso de las hormigas, sean lo más similares posible al suyo propio, que es fijo. Las hormigas modifican los vectores de la celda en la que están (y un vecindario), según la ecuación 4, en la que $\alpha \in [0,1]$ es el factor de aprendizaje que controla la 'intensidad' con la que los vectores del hábitat adquieren la información de las hormigas.

$$\overrightarrow{v_c}(t) = \overrightarrow{v_c}(t-1) + \alpha\left[1 - d\left(\overrightarrow{v_a}, \overrightarrow{v_{cij}}(t)\right)\right].(\overrightarrow{v_a} - \overrightarrow{v_{cij}}(t-1)) \tag{4}$$

Dicha ecuación muestra cómo se cambia el hábitat. El 'refuerzo' es proporcional a la distancia Euclídea entre el vector de entrada y el vector del grid. Finalmente, los vectores del grid se evaporan en cada iteración. Dicha evaporación se hace siguiendo la expresión de la ecuación 5, en la que $k \in [0, \ ]$ (normalmente un valor en $[0.001, 0.1]$) es el factor de evaporación y v_{lc} es el estado inicial de un vector (en el instante *t=0*). En esencia, la evaporación ajusta los vectores para volver a sus valores iniciales.

$$\overrightarrow{v_c}(t) = \overrightarrow{v_c}(t) - k.(\overrightarrow{v_c}(t) - \overrightarrow{v_{lc}}) \quad (5)$$

Con estas ecuaciones, las hormigas transforman el entorno, comunicándose mediante él, y agrupándose en clusters que representarán en mayor o menor medida las clases de los patrones de entrada. Dado que este método no es supervisado, KANTS utiliza el mapa de feromonas sólo para la comunicación entre hormigas, siendo más importantes, desde el punto de vista de algoritmo, los clusters que se consiguen. Aun así, la sección 4 mostrará cómo este grid puede ser visualizado como un tipo de 'huella dactilar' de los datos.

4. Renderización Fotográfica con KANTS

Hasta ahora, KANTS ha sido utilizado como una herramienta de arte enjambre para la generación de representaciones abstractas de vectores de datos o, más bien, de la interacción e interdependencia de esos mismos datos (ver [6]). La naturaleza abstracta de los mapas de feromonas resultantes emerge únicamente a partir de la estructura del algoritmo original, de modo que, a la rejilla, inicializada aleatoriamente, no se le impone ningún tipo de mecanismo de tendencia. Un ejemplo se muestra en el trabajo *Abstracting the Abstract*, que ganó en 2012 la competición de Arte, Diseño y Creatividad Evolutiva del congreso GECCO [7]. Los datos de entrada son los valores RGB de un fichero conteniendo una imagen de un cuadro abstracto y los dibujos ferogénicos surgen como resultado de la acción del enjambre en una rejilla inicializada aleatoriamente. Sin embargo, se podría considerar un enfoque alternativo, en el que el resultado podría estar sesgado por una inicialización no aleatoria. En este trabajo, proponemos la inicialización de la rejilla con los valores RGB de fotos en color. De hecho, la rejilla será la propia fotografía, representada por una matriz de vectores de valores RGB. A continuación, una versión más pequeña de tamaño de la misma foto se utiliza para la generación de los vectores de datos de entrada. El resultado es la consecuencia de la interacción de los vectores de entrada con la imagen del hábitat. Se espera que el dibujo final sea un esbozo de la foto original.

Para probar esta hipótesis, la rejilla del medio ambiente se inicializa con los valores RGB de una foto de tamaño 200×200 píxeles. Es decir, en $t = 0$, la rejilla es una copia exacta de la foto. Los vectores de entrada se extraen de una versión de la misma foto de 120×120 píxeles. Por lo tanto, hay 40000 vectores en la rejilla y 14400 de entrada. La relación entre el número de vectores de entrada y de rejilla está dentro del rango indicado en [12], que es [1:3,1:2].

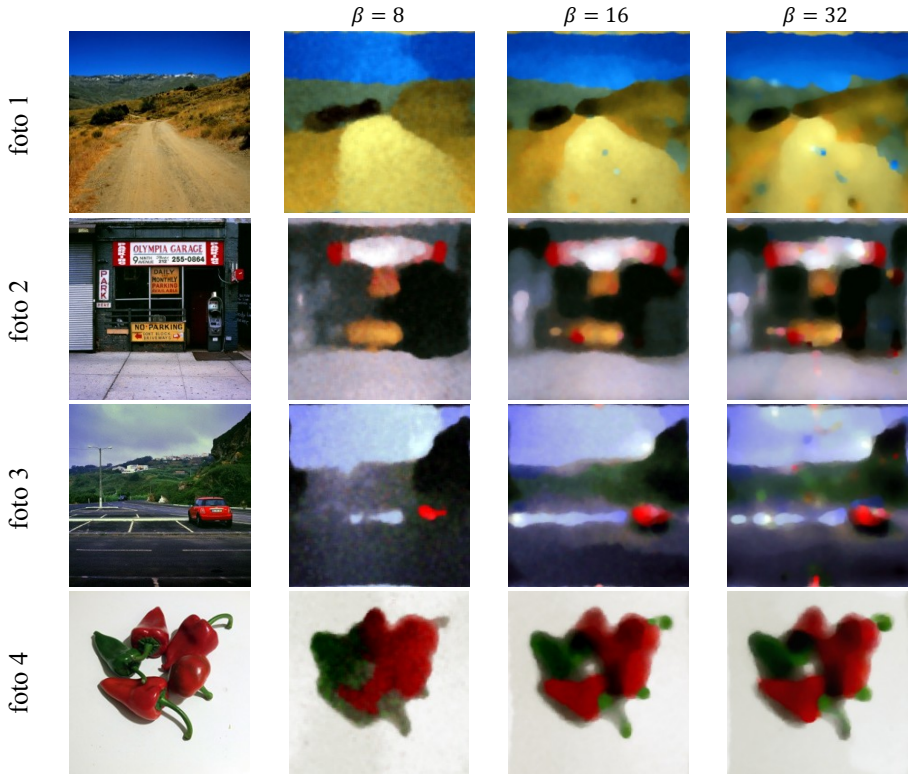


Figura 1. Dibujos ferogénicos utilizando una foto como rejilla inicial y los valores RGB de la misma foto como vectores de entrada. Resultados generados con diferentes valores de β.

Dado que en $t = 0$ la rejilla es una copia de la foto, las hormigas van a encontrar fácilmente regiones con vectores similares a los suyos, en dirección a las cuales se moverán, convergiendo así hacia la forma y la estructura de color de la foto. Como se pude ver en la Figura 1, los mapas de feromonas, después de 50 iteraciones, son esbozos de la foto original. No son copias exactas debido a que los vectores, aunque tienden a trasladarse a regiones similares, se rigen por un esquema pseudo-estocástico. El enjambre tiene su *hoja de ruta* en la configuración inicial de la red, pero, a continuación, las interacciones complejas de las hormigas dan su propia interpretación de la imagen.

La Figura 1 muestra varios ejemplos de fotos con diferentes características (paisaje, paisaje urbano, mixto y naturaleza muerta). Los resultados se refieren a diferentes β. Este es el rango en el que los patrones auto-organizados emergen, una región de comportamiento que se encuentra entre las regiones ordenadas y caóticas, donde no hay estructuras de complejidad y de cooperación.

El aumento de β (que significa que se reduce el factor aleatorio en la regla de seguimiento de feromona) genera pinturas ferogénicas con más detalle. Sin

embargo, en algunas imágenes los detalles importantes no pueden ser reproducidos por el enjambre. Por ejemplo, en el paisaje urbano de la foto 2, los detalles se pierden. Por otro lado, la estructura simple de la foto 1, con regiones cromáticas bien definidas y formas sencillas, genera un dibujo interesante, al estilo de una acuarela impresionista. Según la experiencia de los autores con el sistema, las fotos de paisajes generan los mejores resultados. Sin embargo, esta es una evaluación subjetiva dichos resultados.

La Figura 2 describe el progreso de la red durante la ejecución de KANTS. En $t = 1$ el enjambre ha añadido *ruido* a la imagen original. Con el avance del tiempo, los detalles se desvanecen. En $t = 50$, el mapa de feromonas es un boceto de la foto original y solo reproduce las formas y colores principales.

La versión de KANTS utilizada en este trabajo tiene seis parámetros que pueden ser ajustados a fin de controlar los resultados. β, δ y $r1$ controlan el movimiento de las hormigas, mientras que α, k y $r2$ ajustan el esquema de actualización de los vectores de la rejilla. Es de esperar que una combinación de baja intensidad de evaporación y pequeña tasa de aprendizaje permita al sistema mantener las características de la foto, al menos por un tiempo más largo. Del mismo modo se supone que un α elevado y una intensa evaporación conducirán al cambio rápido de la rejilla inicial, que a su vez conduce a una pintura ferogénica más diferenciada de la referencia. La Figura 3 confirma esta última hipótesis. Con $k = 0.0001$ y $\alpha = 0.01$, el dibujo es similar a la foto original porque la etapa de actualización es menos intensa (menos refuerzo y evaporación). La acción del enjambre en el hábitat es lenta. Cuando los valores de ambos parámetros aumentan, el sistema se mueve hacia un estado que genera imágenes abstractas. Con valores elevados de α y k, el sistema es destructivo: el entorno es rápidamente alterado por las hormigas, las cuales dependen más de su interacción con la red que de la configuración inicial.

Estos resultados describen la capacidad de KANTS para el renderizado fotorrealista. Los dibujos se pueden ajustar utilizando el conjunto de parámetros del algoritmo, de forma que mediante un ajuste de dichos parámetros, las imágenes variarán desde una interpretación ultra-realista de la foto hasta una representación abstracta de los valores cromáticos. En el hábitat, KANTS tiene la capacidad de generar bocetos de las fotos al estilo de acuarelas. Tras haber sido utilizado para el arte abstracto de enjambre, este trabajo demuestra las posibilidades del algoritmo como herramienta de arte figurativo y fotorrealista.

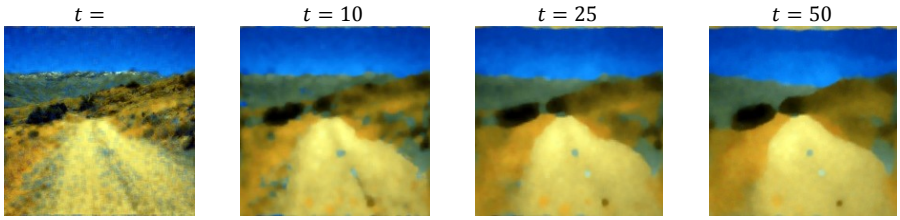


Figura 2. Progreso del dibujo ferogénico. $\boldsymbol{\beta = 16}$; $\boldsymbol{\delta = 0.2}$; $\boldsymbol{\alpha = 0.1}$; $\boldsymbol{r1 = 50}$; $\boldsymbol{r2 = 2}$

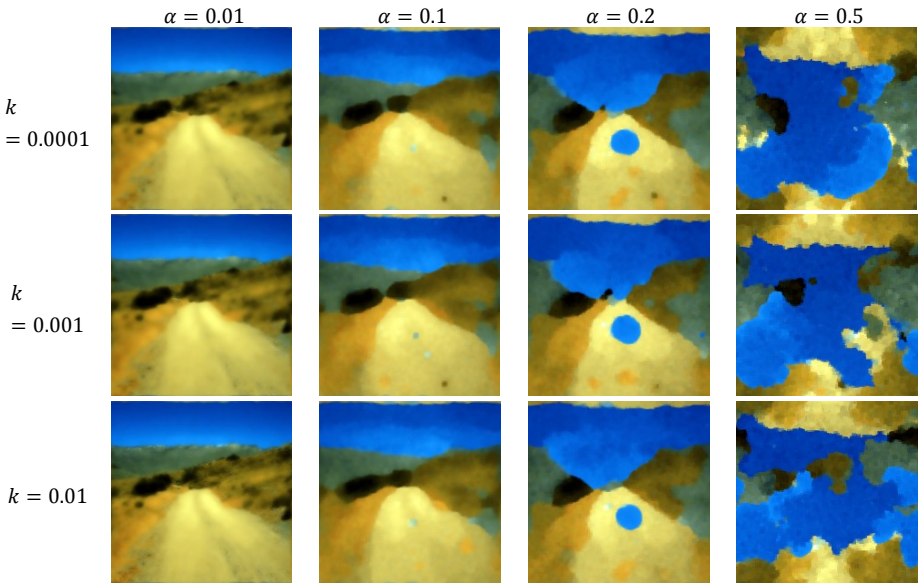


Figura 3. Dibujos ferogénicos de la misma foto, cambiando la tasa de refuerzo α y la evaporación k. 50 iteraciones. Parámetros: $\boldsymbol{\beta = 16}$; $\boldsymbol{\delta = 0.2}$; $\boldsymbol{r1 = 50}$: $\boldsymbol{r2 = 2}$.

5. Conclusiones

En este trabajo se describe una serie de experimentos de arte de enjambre realizados con un algoritmo basado en el comportamiento de las hormigas. El algoritmo es capaz de crear agrupaciones de datos al permitir que esos datos (que serán las propias hormigas) se muevan a través de un entorno heterogéneo. Las hormigas se comunican a través del hábitat al tiempo que lo modifican. Dicho hábitat es una rejilla de vectores, que es utilizado para generar los llamados *dibujos ferogénicos* (se basan en la feromona que depositan las hormigas en el entorno), siendo por tanto, representaciones de la interacción de los datos de entrada con el hábitat.

El proyecto presentado en este artículo usa los valores RGB de una fotografía como vectores de entrada, mientras que los vectores de la rejilla se inicializan con los valores de la misma fotografía. Las imágenes resultantes son pinturas figurativas de la foto. Es posible ajustar el estilo de la imagen generada, pasando de ser ultra-realista a abstracta. Con este fin, el artículo presenta también los efectos de los parámetros en los resultados.

Como primera línea de trabajo futuro, se estudiará la llamada memoria del sistema, dado que la rejilla puede ser considerada como un mapa de memoria, en la que los niveles de feromona son reforzados (aprendizaje) y evaporados (olvido). De esta forma podremos explorar estas características usando fotos

diferentes para reinicializar la rejilla y así estudiar la aptitud del enjambre en el trato y síntesis de múltiples imágenes.

Agradecimientos Fernandes agradece FCT, Ministerio Portugués de la Ciencia, la beca SFRH/BPD/66876/2009. Este trabajo se está desarrollando gracias a la financiación de los proyectos EvOrq (TIC-3903), CANUBE (CEI2013-P-14) y ANYSELF (TIN2011-28627-C04-02).

Referencias

1. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. New York, NY: Oxford University Press, 1999.
2. Chialvo, D., Millonas, M.: How Swarms Build Cognitive Maps. *The Biology and Technology of Intelligent Autonomous Agents*, No. 144, NATO ASI Series, pp. 439-450, 1995.
3. Collomosse, J.P.: Evolutionary Search for the Artistic Rendering of Photographs. *The Art of Artificial Evolution - A Handbook of Artificial Art and Music*, J. Romero, and P. Machado, Eds, Springer, pp. 39-62, 2007.
4. Fernandes, C.M., Ramos, V., Rosa, A.C.: Self-Regulated Artificial Ant Colonies on Digital Image Habitats. *International Journal of Lateral Computing*, 2(1), pp. 1-8, 2005.
5. Fernandes, C.M.: Pherographia: Drawing by Ants. *Leonardo* 43(2), pp. 107-112, 2010
6. Fernandes, C.M., A.M. Mora, J.J. Merelo, A.C. Rosa, Pherogenic Drawings: Generating 2-dimensional Abstract Representations of Sleep EEG with the KANTS Algorithm. Proceedings of the 4th Int. Conf. on Evolutionary Computation Theory and Applications, 2012.
7. Gagné, C., Hoover, A.K, McCormack, J.: 2012 Evolutionary Art, Design and Creativity Competition, 2012 [online: http://eadcc.sigevolution.org/]
8. Grassé, P.-P.: La reconstrucion du nid et les coordinations interindividuelles chez bellicositermes et cubitermes sp. La théorie de la stigmergie: Essai d'interpretation du comportement des termites constructeurs', *Insectes Societies*, No. 6, pp. 41-80, 1959.
9. Kohonen, T.: *The Self-Organizing Maps*. Springer-Verlag, Berlin, 2001
10. Lewis, M.: Evolutionary Visual Art and Design. *The Art of Artificial Evolution - A Handbook of Artificial Art and Music*. J. Romero, P. Machado, Eds, Springer, pp. 3-38, 2007.
11. Monmarché, N., Mahnich, I., Slimane, M.: Artificial Art Made by Artificial Ants. In *The Art of Artificial Evolution - A Handbook of Artificial Art and Music*, J. Romero, and P. Machado, Eds, Springer, pp. 227-24, 2007.
12. Mora, A.M., Fernandes, C.M., Merelo, J.J., Ramos, V., Laredo, J.L.J., Rosa, A.C.: KohonAnts: A Self-Organizing Ant Algorithm for Clustering and Pattern Classification. In Proc. of the 11th Int. Conf. on Artificial Life, MIT Press, pp. 428-435, 2008.
13. Moura, L.: Swarm Paintings-Non-human Art. Architopia. *Utopia Biennial*, 11-24, 2001.
14. Moura, L., (Ed.): *Inside: Art and Science*. Catalogue, Ed. LXXL, Lisbon, 2009.
15. Ramos, V., Almeida, F.: Artificial Ant Colonies in Digital Image Habitats: a Mass Behaviour Effect Study on Pattern Recognition. Proceedings of the 2nd International Workshop on Ant Algorithms, pp. 113-116, 2000.
16. Romero, J., Machado, P. (Eds.): *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music*. Springer, Berlin-Heidelberg, 2007.
17. Semet, Y, O'Reilly, Durand, F.: An Interactive Artificial Ant Approach to Non-photorealistic Rendering. In Proceedings of Genetic and Evolutionary Computation Conference (GECCO), K. Deb et al., Eds., LNCS 3102, Springer-Verlag, pp. 188-200, 2004.
18. Wilson, E.O.: Insect Societies. Cambridge: Belknam Press, 1971.

EvoSpace-Interactivo: Una herramienta para el Arte y Diseño Interactivo y Colaborativo

Francisco Fernández de Vega[2], Mario García-Valdez[1], Leonardo Trujillo[1], Juan J. Merelo Guervós[3], and Gustavo Olague[4]

[1] Instituto Tecnológico de Tijuana, Tijuana BC, Mexico
[2] Grupo de Evolución Artificial, Universidad de Extremadura, Mérida, Spain
[3] Universidad de Granada, Granada, Spain
[4] Centro de Investigación Científica y de Educación Superior de Ensenada, Ensenada BC, Mexico
mario@tectijuana.edu.mx, leonardo.trujillo@tectijuana.edu.mx
fcofdez@unex.es, jmerelo@geneura.ugr.es, olague@cicese.mx

Abstract. Las tendencias más nuevas en las áreas de investigación relacionadas con la computación evolutiva no suelen incorporar algunas tecnologías del panorama actual: la computación en nube (muy presente en las nuevas ofertas tecnológicas) y las redes sociales (fundamentales para los usuarios) así como la la conexión entre estos dos tendencias son prácticamente inexistentes. Este trabajo pretende tender un puente que integre los modelos de computación evolutiva, y más concretamente aquellos basados en interacción, con los sistemas distribuidos basados en modelos *cloud*, haciendo uso de las redes sociales para una colaboración efectiva en los procesos de generación de arte/diseño. El uso de redes sociales en este contexto podría permitir un uso real del concepto de *meme*, idea que se transmite entre personas en un proceso ayudado por la tecnología. Este trabajo presenta EvoSpace-Interactivo, una herramienta de software libre para el desarrollo de Algoritmos Evolutivos Colaborativos e Interactivos (AECI). La herramienta facilita el uso de los Algoritmos Evolutivos Interactivos (AEI) para el diseño artístico. Además de describir la herramienta, presentamos una prueba de concepto de la herramienta, que integrada en Facebook muestra evolución artística y colaboraciń mediante uso de redes sociales.

Keywords: Evolución Interactiva, Diseño Colaborativo, Arte Evolutivo, Computación Cloud, Redes Sociales

1 Introducción

Aunque los Algoritmos Evolutivos se diseñaron inicialmente como métodos para resolver problemas de optimización [6], su naturaleza estocástica pronto reveló su fuente de creatividad y su capacidad para ser aplicados en el mundo del arte, la música y el diseño [15]. Los AEs son así empleados hoy día de forma frecuente en procesos de arte y diseño generativo [1].

Sin embargo todavía persisten algunos problemas cuando pedimos *creatividad* a un AE [9]. En primer lugar resulta difícil dedicir como codificar conceptos relacionados con el mundo del arte, la música y el diseño en una estructura de datos que será un individuo de una población que evolucione; tampoco resulta trivial el diseño de operadores genéticos que realicen una búsqueda efectiva. Por último, uno de los principales problemas en este dominio de aplicación es encontrar un buen método para evaluar la calidad de los individuos, método que debe evaluar aspectos estéticos de las soluciones consideradas. El problema de fondo es que los principios estéticos y creativos no son comprendidos en su totalidad, lo que impide definir con precisión una función de evaluación que mida su valor correctamente.

En vista de las dificultades inherentes al proceso creativo, los investigadores, en términos coloquiales, optaron por la tangente: decidieron recurrir al cerebro humano para realizar tareas de evaluación estética, definiendo así los Algoritmos Evolutivos Interactivos (AIEs): un Algoritmo Evolutivo por lo demás estandar, cuyo proceso de evaluación es realizado por un ser humano en una etapa interactiva.

Con el nuevo modelo, el bucle principal del AE incluye la intervención humana para asignar la calidad a cada una de las soluciones evolucionadas. Pero la solución no está exenta de problemas: la fatiga que surge en la repetición sistemática del proceso de evaluación. Ya hay trabajos en la literatura que tratan de mitigar este efecto indeseado [5]. A pesar de todo los AEIs han demostrado su utilidad [1, 14, 16].

Debemos mencionar por último que normalmente son artistas con formación en ciencias de la computación, o científicos con interés en arte/diseño los investigadores que normalmente trabajan en esta área. Pero es difícil para un artista clásico, sin conocimientos de programación, entrar en este dominio. Además, las herramientas que facilitan el uso de los AEs a los investigadores no están especialmente concebidas para artistas. Así, algunas propiedades particularmente útiles para una herramienta destinada a artistas, y que hoy día no se encuentra disponible, son las siguientes:

1. Una forma directa y sencilla de codificación de individuos para arte/diseño generativo. Los usuarios que consideramos aquí son artistas sin formación específica en ciencias de la computación.
2. Una implementación efectiva que permita la influencia directa entre diferentes artistas participando en un proceso creativo colaborativo.

El mundo artístico se mueve en redes de influencia que permiten a las ideas viajar entre las mentes de los artistas. Una auténtica evolución memética de ideas que surgen de las manos y mentes de los artistas debería ser facilitada por una herramienta del tipo considerado. El término memético en su original concepción (según Richard Dawkins [3]), sería finalmente un componente principal del proceso evolutivo.

El presente trabajo describe una herramienta que explícitamente se enfrenta al modelo colaborativo e interactivo desde el punto de vista del artista y su proceso creativo.

El resto del artículo se estructura del siguiente modo: La sección 2 presenta una revisión del área de Evolución Interactiva. La sección 3 describe la plataforma desarrollada y sus objetivos principales. El modelo colaborativo-interactivo se presenta en la sección 4, y el trabajo experimental junto con los resultados en la sección 5. Finalmente incluimos nuestras conclusiones en la sección 6.

2 Evolución Interactiva

El término Evolución Interactiva (EI) incorpora decisiones y preferencias humanas en la presión selectiva que guía la búsqueda [15].

Específicamente, la EI realiza una búsqueda en la que la función de evaluación tradicional es reemplazada por una evaluación subjetiva llevada a cabo por el usuario del sistema. La naturaleza no definida del objetivo de la búsqueda amplía el conjunto de soluciones factibles que pueden ser alcanzadas, promoviendo así la diversidad y la exploración en el *espacio de diseño*.

Entre los primeros ejemplos que sigue este modelo se encuentra Biomorphs [4]. Nosotros estamos particularmente interesados en la posible colaboración de múltiples usuarios que participen en un proceso creativo.

Uno de los primeros ejemplos de evolución interactiva basada en web fue presentado por Langdon para evolucionar formas fractales [8]. Un servidor web almacenaba la población y era distribuida en porciones a clientes remotos que utilizaban Javscript. De forma similar, Secretan et al. [12], y Clune y Lipson [2] usaron AEIs basados en web para evolucionar imágenes de dos y tres dimensiones con una componente artística. En ambos casos se ofrece una interfaz web en la que los usuarios interactuan a partir de sus preferencias personales, y pueden colaborar entre ellos reutilizando diseños de usuarios previos. Los usuarios además califican los resultados e influyen mutuamente en los procesos evolutivos. El sistema además permite estudiar la red de influencias mediante el linaje genético que se mantiene almacenado para cada imagen.

Otro ejemplo interesante es el propuesto por Kowaliw et al. [7] basado en un sistema multiagentes.

En este trabajo se extiende el modelo colaborativo utilizando una implementación paralela y dinámica basada en el modelo cloud que explícitamente alienta la colaboración mediante el uso de redes sociales.

3 EvoSpace

EvoSpace es un hábitat, un almacén para el desarrollo de algoritmos evolutivos que se pretenden ejecutar en un modelo de computación cloud. Está diseñado para que sea versátil, dado que la población es independiente del modelo evolutivo particular que quiera utilizarse. EvoSpace es asíncrono y los procesos clientes, que denominamos *EvoWorkers*, interaccionan de modo asíncrono y dinámico con EvoSpace realizando las rutinas básicas de una búsqueda evolutiva. Los EvoWorkers pueden despleagarse tanto en clientes remotos como en la

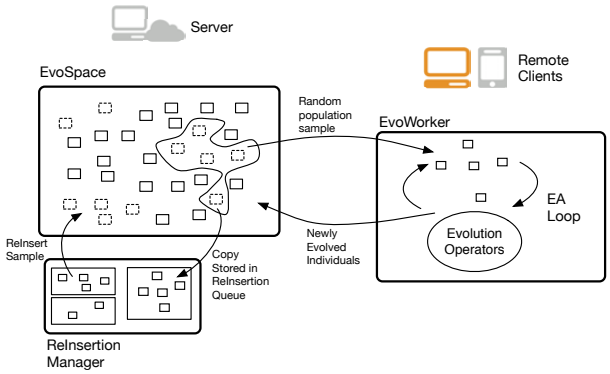


Fig. 1. Flujo de datos y componentes principales de EvoSpace.

propia plataforma que alberga al servidor. La figura 1 muestra los componentes principales del flujo de datos dentro de EvoSpace.

Evospace se basa en el modelo de *espacio de tuplas*, un modelo de direccionamiento asociativo de espacio de memoria para varios procesos.

Proceso Servidor El servidor EvoSpace crea y activa el contenedor para la población y ejecuta los métodos de la interfaz con los clientes. Dos procesos adicionales son necesarios: (i) *InitPopulation* inicializa una población de *popsize* individuos aleatorios; (ii) *ReInsertionMgr* se utiliza para control de errores, y chequea periódicamente si el tamaño de la población ha caído por debajo de un umbral, reinsertando en ese caso los individuos necesarios.

Procesos clientes El proceso `EvoWorker` solicita una muestra de individuos al servidor para aplicarle operaciones genéticas y realizar el ciclo evolutivo sobre el subconjunto recibido. Puede así considerarse que cada EvoWorker trabaja con una población local a la que se aplica el ciclo evolutivo durante un número de generaciones, al cabo del cual, los individuos evolucionados son devueltos al servidor. Pero también es posible construir EvoWorkers especializados en un sólo operador genético, o varios, si fuera necesario.

Implementación La población de individuos es manejada por el servidor sobre memoria RAM, utilizando la base de datos *Redis*. Se eligió Redis por manejar conjuntos mediante claves hash, lo que reduce la complejidad computacional de los procesos de búsqueda. EvoSpace está implementado como un módulo Python ofrecido como un servicio web utilizando los frameworks HTTP *cherrypy* y *Django*. El servicio web EvoSpace puede interactuar con cualquier lenguaje que soporte *json-rpc* o peticiones *ajax*. Todo el sofware está disponible con *Simplified BSD License* [5].

[5] `http://evospace.org`

4 Una herramienta para Evolución Colaborativa e Interactiva

Como se describe anteriormente, el *framework* que se ha desarrollado se denomina EvoSpace-Interactivo, y sus componentes principales se muestran en la figura 2.

Los desarrolladores de aplicaciones colaborativas basadas en Evolución Interactiva han sido liberados, gracias al nuevo framework, de la carga de trabajo asociada a diseño y programación requerido en este tipo de proyectos. Sólamente tres componentes deben definirse adaptados al problema concreto que desee abordarse -marcado con doble línea en la figura-: la *representación del individuo*; el *script* que renderice dicho individuo; y por último el proceso *worker* que incluya los operadores genéticos que deberán ser definidos en el contexto del problema y representación elegidos. No obstante, esperamos en futuras versions ofrecer parte de este trabajo ya predefinido de modo genérico, pero con la flexibilidad suficiente para que los usuarios avanzados puedan modificar lo que necesiten.

La herramienta, implementada con componentes de código abierto, permite el manejo de sesiones con la interfaz Facebook, utilizando para ello el protocolo OAuth 2.0. Además se permite el manejo de colecciones por parte de los usuarios, gestionadas mediante PostgreSQL DBMS. Se utilizan librerías Javascript para gestionar la parte interactiva en el cliente, y json-rpc para la comunicación entre componentes. Toda la funcionalidad está repartida en diferentes servicios, lo que permite intercambiar cualquiera que se considere sin afectar al resto.

Se utiliza un modelo *Responsive Web Design (RWD)* para la implementación GUI de la interfaz interactiva del usuario, lo que permite una adaptación automática a diferentes tipos de dispositivos móviles. Es suficiente el uso de definiciones CSS para conseguir la apariencia deseada.

Interfaz de Usuario El usuario interactúa mediante una interfaz web como la mostrada en la figura 3(a), compuesta por cinco elementos. En la parte superior izquierda se incluye la gestión del usuario y autenticación. Los usuarios pueden conectarse utilizando sus cuentas Facebook, o, si lo desean, pueden permanecer como usuarios anónimos. Si el usuario utiliza su cuenta Facebook, los amigos que también tienen enlazada la aplicación serán incluidos en la parte izquierda, para animar la colaboración entre usuarios. El tercer elemento es el muro central, donde se incluye una muestra de n individuos de la población. Estos individuos son tomados al azar del servidor EvoSpace.

El usuario puede interactuar de dos modos: seleccionando los individuos que prefiera -lo que significa *like* en términos Facebook-. El usuario también puede añadir individuos a su propia colección y almacenarlo. Una colección no es más que un almacén de individuos. Una vez que el usuario termina de interactuar con los individuos mostrados, puede pedir otro conjunto al servidor, mediante el botón *GetMore* situado en la parte superior de la ventana. La acción asociada devuelve el grupo actual al servidor, y trae un nuevo conjunto al cliente. Cada vez que el usuario activa esta acción, en la parte del servidor

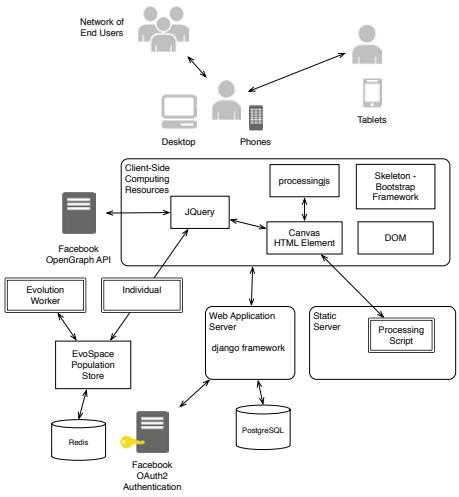
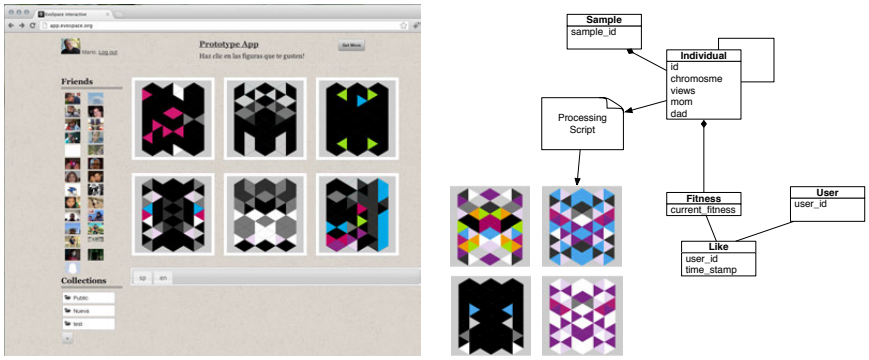


Fig. 2. Diseño Conceptual del modelo EA-CI

se lanza un evento *Breeding*, que sirve para lanzar un proceso específico asociado a la aplicación concreta en que se trabaje. El último elemento de la interfaz es el situado en la parte inferior izquierda, que corresponde a las colecciones. El usuario puede manejar varias para organizar sus imágenes. Por otro lado, cada individuo puede mostrar cuántos usuarios lo han seleccionado, permite enlazar a sus padres, operadores genéticos que lo crearon y su genealogía.

Codificación de Individuos Los individuos son representados internamente como diccionarios, de acuerdo con las especificaciones Redis. Así cada individuo posee: un identificador único `id`; un cromosoma definido por el usuario; el número de veces que el individuo ha sido seleccionado y devuelto a la población (propiedad llamada `vista`), información sobre los operadores genéticos utilizado para generarlo junto con los identificadores de sus padres; el valor de `fitness actual` que contiene el valor de fitness más reciente; y un diccionario fitness en el que cada clave es una concatenación de identificador de usuario, valor de instante temporal, y valor numérico que representa la aptitud del individuo. La figura 3(b) muestra una representación UML del individuo.

Espacio de trabajo y lenguaje de programación *Processing* es un lenguaje de programación desarrollado inicialmente como herramienta de esbozo artístico en un contexto visual. Es utilizada por artistas, diseñadores, arquitectos e investigadores para visualizar aplicaciones, juegos, proyectos de animación interactiva, etc. [11]. Processing es muy utilizado en la actualidad por artistas generativos [10], y ésta es la razón principal por la que elegimos Processing como herramienta de trabajo para los artistas en EvoSpace-Interactivo, dado el tipo de



(a) Interfaz de usuario.

(b) Representación interna del individuo

Fig. 3. (a) Interfaz de Usuario y (b) Representación

usuario destinatario de la herramienta. Existe además un complemento de librería de scripts java para Processing, que permite ejecutar scripts en este lenguaje dentro de cualquier navegador HTML5.

En nuestro caso, los scripts escritos en Processing son responsables de la renderización del individuo -cualquiera que se ésta: animación, vídeo, audio o elemento interactivo. Antes de ejecutarse una llamada al método encargado de la renderización, se introduce vía parámetro el cromosoma del individuo en cuestión. Cada script tiene asociado su espacio físico de trabajo -lienzo para el artista-, definido en el estandar HTML5.

5 Prueba de concepto: Shapes

Para probar la herramienta, se ha implementado una aplicación básica de AEIC llamada *Shapes*. En *Shapes* se evolucionan imágenes inspiradas en el estilo Op-Art. Las imágenes consisten en configuraciones bidimensionales de triángulos equilateros. Los triángulos se disponen en un arreglo de 11 por 6 y su color se toma de una paleta de doce colores.

Representación de Individuos y Processo de Aparamiento. La matriz de triángulos se codifica con un vector cromosoma de 66 elementos $\mathbf{v} = (v_1, ..v_{66})$, con $v_i \in \{1, 2, ..11\}$. El fondo es de color gris claro, como este color también se incluye en la paleta de once colores se puede tener un efecto de “triangulo faltante" cuando un triángulo tiene el mismo color que el fondo. Se utiliza un script en Processing para dibujar una versión estática de las imágenes. El proceso de apareamiento utiliza una selección tipo torneo de tamaño 6 para seleccionar a dos individuos de EvoSpace y generar una descendencia de dos. La descendencia reemplaza a los dos peores individuos. Si los participantes en el torneo no tienen por lo menos cinco vistas el proceso termina. Los operadores de cruce se utilizan con

una proporción de cruce de 1. Se cuenta con cruces de un punto, tanto verticales como horizontales. Utilizando una proporción de mutación de 0.3, se pueden aplicar los siguientes operadores de mutación: (1) mutación de punto simple; (2) *espejos* verticales y horizontales sobre un punto aleatorio. (3) *barajar* lo que da una nueva permutación del cromosoma.

Evaluación de Aptitud. Para evaluar la aptitud de cada individuo se toman en consideración las evaluaciones realizadas por varios usuarios. En *Shapes* cuando un usuario ve una pintura y ésta se le hace atractiva, lo indica seleccionándola con lo cual se le asigna un *Like* (Me gusta), en el caso contrario el usuario simplemente no selecciona la pintura. Este tipo de evaluación considera entonces solo evaluaciones positivas. En cualquier caso cuando una pintura es considerada por un usuario, se incrementa su contador de *vistas*. Se considera que un individuo tiene una baja aptitud si tiene un número bajo de *Likes* y un alto número de *vistas*. En cambio un individuo con un alto número vistas y *Likes* se considera con mayor aptitud. La proporción *Likes*/*Vistas* puede brindar esta información, pero no hace distinción entre un individuo con muchas o pocas *vistas*, si el número de *Likes* es cero; también se debe considerar que *vistas* siempre debe ser mayor a cero. Para evitar los casos anteriores se propone que la aptitud sea dada por $(Likes + 1)/(Views + 1)$.

Configuración del Sistema. La población se inició con 500 individuos generados de manera aleatoria. Para evaluar a los individuos los usuarios toman muestras de seis individuos los cuales son regresados a EvoSpace una vez evaluados, para tomar inmediatamente después una nueva muestra. Este ciclo se repite para todos los usuarios que participan del experimento. Cada dos muestras regresadas, se dispara un proceso de apareamiento, lo que agrega nuevos individuos a la población. Se configuró el método `ES.ReInsert()` para que se ejecutara en caso de que la cola de muestras llegase al umbral de 20 muestras. Para este experimento inicial se esperaban un máximo número de 40 usuarios concurrentes. El sistema se instaló en un servidor virtual con 500MB de RAM y ha estado en operación por varios meses en `http:\\shapes.evospace.org`.

Resultados. En operación desde octubre de 2012, actualmente se cuenta con un total de 90 usuarios registrados. Se han generado más de 12 mil individuos y entregado más de 26 mil muestras. Una muestra de los individuos en cierta colección es mostrada en la Fig. 4. El Histograma mostrado en la Fig. 5(a) muestra el número de Likes que han recibido los individuos: se observa que una gran proporción no recibe ningun like y son pocos los individuos que logran una cantidad considerable. El número de vistas que logran los individuos para esta cantidad de usuarios, se muestra en el histograma de la Fig. 5(b).

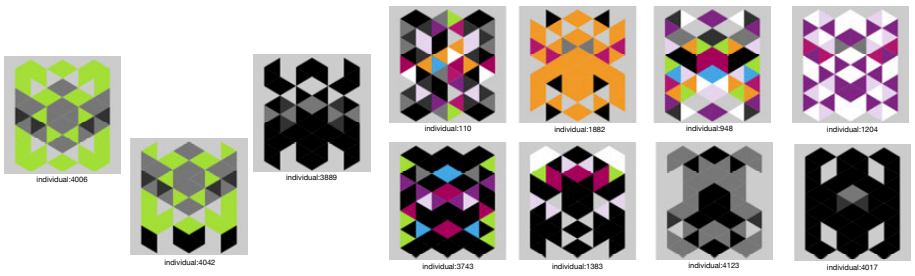
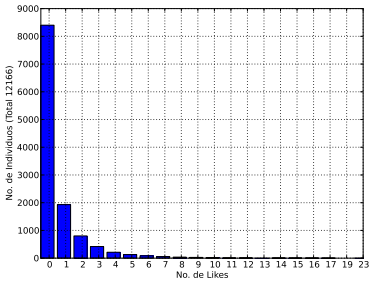
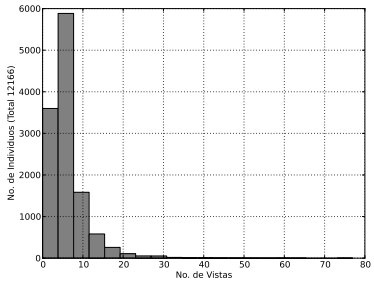


Fig. 4. Colección de Individuos.



(a) Histograma del número de *Likes* por individuo.



(b) Histograma del número de vistas por individuo.

Fig. 5. Histogramas de vistas y likes.

6 Conclusiones

EvoSpace-Interactivo ha sido desarrollada como marco para la utilización de Algoritmos Evolutivos Colaborativos e Interactivos, y ha sido probado con usuarios en un proceso de diseño artístico con resultados alentadores.

El sistema se integra con redes sociales populares, y hace uso de sus medios de promoción en el proceso específico de evolución artificial colaborativa: permite asignación de fitness mediante acción de un grupo de usuarios y suministra a los investigadores información contextual relevante sobre el proceso, lo que permite una mejor comprensión del comportamiento y preferencias de los usuarios.

El uso del lenguaje *Processing* permitirá una fácil adaptación por parte de artistas y rápido desarrollo de representaciones artísticas, más allá de lo que la prueba de concepto Shapes ha mostrado. Esperamos que la herramienta EvoSpace facilite el desarrollo de proyectos de arte evolutivo basados en el modelo interactivo y colaborativo.

Agradecimientos Este trabajo ha sido financiado por los proyectos siguientes: DEGEST-ProIFOPEP (Mexico) Research Project 4616.12-P; CONACYT (Mexico) Basic Science Research Project No. 178323; Gobierno de Extremadura, Consejería de Empleo, Empresa e Innovación y fondos FEDER, proyecto GRU10029; Ministerio de Ciencia e Innovación, proyecto TIN2011-28627-C04-02 and -03 (ANYSELF), y proyecto P08-TIC-03903 Junta de Andalucía.

References

1. P. Bentley. An introduction to evolutionary design by computers. In P. J. Bentley, editor, *Evolutionary Design by Computers*, chapter 1, pages 1–73. Morgan Kaufman, San Francisco, USA, 1999.
2. J. Clune and H. Lipson. Evolving three-dimensional objects with a generative encoding inspired by developmental biology. In *Proceedings of the European Conference on Artificial Life*, pages 144–148, 2011.
3. R. Dawkins. *The Selfish Gene*. Oxford University Press, Oxford, UK, 1976.
4. R. Dawkins. *Climbing Mount Improbable*. W.W. Norton & Company, 1996.
5. M. Frade, F. Fernandez de Vega, and C. Cotta. Evolution of artificial terrains for video games based on accessibility. In C. Di Chio, S. Cagnoni, C. Cotta, M. Ebner, A. Ekart, A. I. Esparcia-Alcazar, C.-K. Goh, J. J. Merelo, F. Neri, M. Preuss, J. Togelius, and G. N. Yannakakis, editors, *EvoGAMES*, volume 6024 of *LNCS*, pages 90–99, Istanbul, 7-9 Apr. 2010. Springer.
6. K. A. D. Jong. *Evolutionary computation - a unified approach*. MIT Press, 2006.
7. T. Kowaliw, A. Dorin, and J. McCormack. Promoting creative design in interactive evolutionary computation. *Evolutionary Computation, IEEE Transactions on*, 16(4):523 –536, 2012.
8. W. B. Langdon. Global distributed evolution of l-systems fractals. In M. Keijzer, U.-M. O'Reilly, S. M. Lucas, E. Costa, and T. Soule, editors, *Genetic Programming, Proceedings of EuroGP'2004*, volume 3003 of *LNCS*, pages 349–358. Springer-Verlag, 5-7 April 2004.
9. J. McCormack. Open problems in evolutionary music and art. In *Proceedings of the 3rd European conference on Applications of Evolutionary Computing*, EC'05, pages 428–436, Berlin, Heidelberg, 2005. Springer-Verlag.
10. M. Pearson. *Generative Art*. Manning Publications, pap/psc edition, July 2011.
11. C. Reas and B. Fry. *A programming handbook for visual designers and artists*. The MIT Press, 2007.
12. J. Secretan, N. Beato, D. B. D'Ambrosio, A. Rodriguez, A. Campbell, J. T. Folsom-Kovarik, and K. O. Stanley. Picbreeder: A case study in collaborative evolutionary exploration of design space. *Evol. Comput.*, 19(3):373–403, Sept. 2011.
13. Y. Semet. Evolutionary computation: a survey of existing theory. Technical report, University of Illinois, 2002.
14. K. Sims. Artificial evolution for computer graphics. In *Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '91, pages 319–328, New York, NY, USA, 1991. ACM.
15. H. Takagi. Interactive evolutionary computation: fusion of the capabilities of ec optimization and human evaluation. *Proceedings of IEEE*, 89(9):1275–1296, 2001.
16. S. Todd and W. Latham. *Evolutionary art and computers*. Academic Press, 1992.

Evolutionary Elaboration of Daily News as a Poetic Stanza

Pablo Gervás

Universidad Complutense de Madrid,Spain,
pgervas@sip.ucm.es,
WWW home page: http://nil.fdi.ucm.es

Abstract. Among the existing efforts at automatic poetry generation, either from scratch or from a given semantic input, little effort has been paid to the issue of sustained innovation. Sustained innovation requires that repeated attempts at generation produce results significantly different from prior system output. This is a characteristic of human creativity that is difficult to achieve in automatic systems built on deterministic procedures. The present paper describes a system that aims at sustained innovation by interleaving with the generation process a stage of knowledge collection or renewal. It extends an existing evolutionary solution for poetry generation to cover the generation of poems about what has happened in the world on a given day, taking as a knowledge source the set of news articles from a specific online newspaper for the given day. The collection of the newspaper articles for the day enacts the knowledge renewal stage. The system described creates a population of text drafts about the chosen day, and evolves them over a number of generations applying a fitness function that measures its metric and linguistic qualities, relying on operators designed to change the drafts with respect to the desired features.

1 Introduction

The issue of sustained innovation in creativity relates to the ability of an agent to produce significantly different results on a given generation attempt from those obtained earlier. In human creators this arises naturally from the fact that humans lead a life of exposure to continuous stimuli, and the knowledge base from which they feed their creative processes is continuosly updated with new material, whether from personal experience or from consumption of content rich products such as books, films, or conversations. For automatic systems, modelling of the creative processes in themselves is complex enough without attempting a parallel effort of modelling the processes of learning and interpretation that underlie such activities. Existing models of computer poetry generation tend to operate from a set knowledge base, usually a carefully crafted resource requiring significant development effort.

The present paper describes a poetry generator that is capable of producing a different set of outputs everyday, by using the set of newspaper articles for

the day as a training knowledge base for its generation processes. The text for the newspaper articles is downloaded from the online edition of the newspaper. A language model is trained based on these texts and used to provide an initial population of drafts which are then submitted to an evolutionary process where they are repeatedly revised, shuffled and judged until an acceptable poetic elaboration of the initial material is reached.

Section 2 reviews briefly prior efforts at poetry generation. Section 3 outlines the evolutionary procedure employed and presents some results. Section 4 includes a discussion of the limitations of the resulting system as a model of sustained innovation in creativy, and section 5 presents conclusions and further work.

2 Brief Review of Existing Poetry Generation Work

Existing poetry generators are reviewed in terms of the specific AI technologies that are employed in the generation process.

2.1 Generate and Test

The generate & test paradigm of problem solving has also been widely applied in poetry generators. Because metric restrictions are reasonably easy to model computationally, very simple generation solutions coupled with an evaluation function for metric constraints are likely to produce acceptable results (given an assumption of poetic licence as regards to the content). An example of this approach is the early version of the WASP system [1]. Initial work by Manurung [2] also applied a generate & test approach based on chart generation, but added an important restriction: that poems to be generated must aim for some specific semantic content, however vaguely defined at the start of the composition process. This constitutes a significant restriction on the extent of poetic licence allowed.

2.2 Case-Based Reasoning

Another important tactic that human authors are known to use is that of reusing ideas, structures, or phrasings from previous work in new results. This is very similar to the AI technique of Case-Based Reasoning (CBR) [3]. Some poetry generators have indeed explored the use of this technique as a basic generation mechanism. An evolution of the WASP system [4] used CBR to build verses for an input sentence by relying on a case base of matched pairs of prose and verse versions of the same sentence. Each case was a set of verses associated with a prose paraphrase of their content. An input sentence was used to query the case base and the structure of the verses of the best-matching result was adapted into a verse rendition of the input. This constituted a different approach to hardening the degree of poetic licence required to deem the outputs acceptable (the resulting verses should have a certain relation to the input sentence).

2.3 Grammar-Based Generation

Another important mechanism that has been employed by automatic poets is grammar-based generation. By using a grammar to produce grammatically correct combinations of words, the results obtained start to resemble understandable sentences. As Chomsky mentioned in 1957 [5], the fact that a sentence is grammatically correct does not imply that it will be interpretable. However, in the context of automatically generated poetry, sentences like Chomsky's classic counterexample ("Colorless green ideas sleep furiously") acquire a special interest, as they provide both a sense of validity (due to their syntactic correctness) and a sense of adventure (due to the impossibility of pinpointing a specific meaning for them). On reading such sentences, the human mind comes up with a number of conflicting interpretations, none fully compatible with its literal meaning. This multiplicity of shifting meanings is very atractive in the light of modern theories about the role of reader interpretation in the reading process.

In 1984 William Chamberlain published a book of poems called "The Policeman's Beard is Half Constructed" [6]. In the preface, Chamberlain claimed that all the book (but the preface) had been written by a computer program. The program, called RACTER, managed verb conjugation and noun declension, and it could assign certain elements to variables in order to reuse them periodically (which gave an impression of thematic continuity). Although few details are provided regarding the implementation, it is generally assumed that RACTER employed grammar-based generation. The poems in Chamberlain's book showed a degree of sophistication that many claim would be impossible to obtain using only grammars, and it has been suggested that a knowledgeable combination of grammars, carefully-crafted templates and heavy filtering of a very large number of results may have been employed.

2.4 Stochastic Language Modelling

The use of n-grams to model the probability of certain words following on from others has proven to be another useful tecnique. An example of poetry generation based on this is the cybernetic poet developed by Ray Kurtzweil [7, 8]. RKCP (Ray Kurtzweil Cybernetic Poet)[9] is trained on a selection of poems by an author or authors and it creates from them a language model of the work of those authors. From this model, RKCP can produce original poems which will have a style similar to the author on which they were trained. The generation process is controlled by a series of additional parameters, for instance, the type of stanza employed. RKCP includes an algorithm to avoid generating poems too close to the originals used during its training, and certain algorithms to maintain thematic coherence over a given poem. Over specific examples, it could be seen that the internal coherence of given verses was good, but coherence within sentences that spanned more than one verse was not so impressive.

2.5 Evolutionary Solutions

Manurung went on to develop in his PhD thesis [10] an evolutionary solution for this problem (now described in [11]). Evolutionary solutions seem particularly apt to model this process as they bear certain similarities with the way human authors may explore several possible drafts in parallel, progressively editing them while they are equally valuable, focusing on one of them when it becomes better valued than others, but returning to others if later modifications prove them more interesting. Manurung's MCGONAGALL used a linguistic representation based on Lexicalized Tree Adjoining Grammar (LTAG) over which operated several genetic operators – from baseline operators based on LTAG syntactic operations to heuristic semantic goal-directed operators – and two evaluation functions – one that measured how close the solutions stress pattern was a a target metre, and one that measured how close the solutions propositional semantics was to the target semantics.

A redesigned version of WASP [12] was used to produce a selection of 10 poems which has been published in a book about the possibilities of computers writing love poems [13]. In that version, the overall style of the resulting poems is strongly determined by the accumulated sources used to train the content generators, which are mostly n-gram based. The poems presented in the book were produced with content generators trained on collections of texts by Federico García Lorca [14], Miguel Hernández [15, 16] and a selection of Sixteenth Century Spanish poets [17]. Readers familiar with the sources can detect similarities in vocabulary, syntax and theme.

3 An Evolutionary Approach to Poetry Generation

The computational model proposed in this paper for the generation of poetry brings together two basic insights obtained from the study of the existing poetry generators: the ability to iterate over a draft applying successive modifications in search of a best fit, and the ability to measure metric forms. The concept of a *draft* that holds the current best solution for a given poem and which gets progressively modified towards an optimal solution, is fundamental to the proposed model. The concept of *reviser*, a module that operates on a draft to progressively improve it, completes the picture to cover the first insight. Such drafts need to be evaluated for conformance with the desired poetic form, and the results of this evaluation need to be taken into account in any subsequent operations on the draft. The concept of a *judge*, a module capable of evaluating partial results according to desired criteria, covers the second insight. In the model, judges can evaluate aspects concerning form, but also content, linguistic validity, fluency, or innovation (in the form of similarity with previous known poems). As a third insight, the model builds on the idea that poets do work at the same time on several possibilities for completing a line, keeping options open to see which may match better with the rest of the poem. When computers are considered to take on an equivalent task, this approach can be taken a step further, so a poetry generator can not just work on one poem but write several

at the same time. The model will therefore operate not on a single draft but over a *population* of candidate drafts.

The existence of a population of candidate solutions, that evolves over time as a result of operations carried out upon it, and that is evaluated based on specific criteria, conforms with the structure of an evolutionary solution, which is one of the candidate technologies to apply. However, our aim is to provide the means for bringing together a number of these technologies. We do this in two different ways. First, we allow a set of alternatives for the creation of the drafts in the initial population. To this end we introduce the concept of *babbler*, a module in charge of producing an initial draft. By allowing a population of babblers to produce the initial population, we introduce the possibility of relying on more than one technology to produce them. Grammar, ngram, or case based solutions can be included among the set of babblers. Second, we introduce a set of alternatives for operating upon the initial drafts, by allowing a population of revisers, possibly employing different technologies. Finally, to allow for revision operations specific to poetic form, we introduce the concept of a *poet*, a module in charge of transforming a draft with a view to matching a specific poetic form. In the spirit of the model, we allow a population of poets, to contemplate more than one possible target form.

The resulting set of elements constitutes a set of families of automatic experts: one family of content generators or babblers (which generate a flow of text that is taken as a starting point by the poets), one family of poets (which try to convert flows of text into poems in given strophic forms), one family of judges (which evaluate different aspects that are considered important), and one family of revisers (which apply modifications to the drafts they receive, each one oriented to correct a type of problem, or to modify the draft in a specific way). These families work in a coordinated manner like a cooperative society of readers/critics/editors/writers. All together they generate a population of drafts over which they all operate, modifying it and pruning it in an evolutionary manner over a number of generations of drafts, until a final version, the best valued effort of the lot, is chosen. Judges evaluate what babblers produce, revisers modify it taking into account what the judges have said. Bad sequences are eliminated during pruning, not so bad ones are modified to make them better.

3.1 WASP Redesigned: Evolutionary Approach to Poetry Generation

A redesigned version of the WASP poetry generator has been built following the model described above. In this version, the overall style of the resulting poems is strongly determined by the accumulated sources used to train the content generators, which are n-gram models obtained from the set of news articles from a specific online newspaper for the given day.

The various judges assign scores on specific parameters (on poem length, on verse length, on rhyme, on stress patterns of each line, on similarity to the sources, fitness against particular strophic forms...) and an overall score for each draft is obtained by combining all individual scores received by the draft. A

specific judge is in charge of penalising instances of excessive similarity with the sources, which then get pushed down in the ranking and tend not to emerge as final solutions.

Additional judges used assign scores for:

- overall poem length (with respect to a desired value used as input)
- length of the verses in the poem (with respect to a desired value used as input)
- average length of sentences in the poem (set empirically at 30 words, to achieve acceptable final poems)
- acceptability of the placement of stressed syllables with respect to each verse (as established by metric tradition for the desired verse length)
- appearance in the poem of words that are metrically impossible to parse (foreign words that do not follow the syllable patterns of the Spanish language)

Poets operate mainly by deciding on the introduction of line breaks over the text they receive as input.

Revisers rely on scores assigned by judges to introduce changes to drafts. Modifications can be of several types: deletion of spans of text, substitution of spans for newly generated ones, word substitution, sentence elimination, and simple cross-over of fragments of poems to obtain new ones. Revisers act as mutator operators, taking an existing draft and changing it in some way. The set of revisers includes:

- dropping a sentence at random
- inserting an additional sentence (produced from the same trained language model) at the end of the draft
- shifting the position of the line breaks in the draft to produce a new draft with different distribution of words over lines
- replacing some words of the poem selected at random with valid replacements according to the ngram model

Because an initial draft produced by an n-gram based content generator is then processed many times over by poets and revisers, final results oscillate between surprising faithfulness to the sources and very radical surreal compositions.

The implementation described here is a very simple one that combines two of the technologies previously used by story generators (evolutionary solutions and stochastic language models).

3.2 Empirical Setting of Evolutionary Parameters

The evolutionary WASP system has a number of parameters that govern its evolutionary process. The main ones are the size of the population to consider and the number of generations to be built. To establish acceptable values for

these parameters, a number of experiments was run with the set of newspaper articles from the EL País newspaper for 21/05/2013. The system was tested with values of 100, 50, 20 and 10 for the population size and values of 100, 50, 20 and 10 for the number of generations. For each possible configuration, the system was run and the average score of the final population was recorded. Results are reported in Table 1. The time taken by each run is also included.

Pop Size	# gen	Av Score	Time in ms.
100	50	81	3159934
100	20	77	1369584
100	10	73	1025167
50	100	83	3308220
50	50	80	1624226
50	20	78	806429
50	10	74	490583
20	100	82	1505449
20	50	80	1028309
20	20	78	401746
20	10	75	279184
10	100	83	1337811
10	50	77	697729
10	20	79	264351
10	10	77	209875

Table 1. Average scores for different configurations of evolutionary parameters

The scores obtained increase steadily with the size of the populations and the number of generations. However, the time taken increases even more rapidly. As runs with 100 generations for a population of 50 or even 20 drafts result in comparable or higher scores than runs with a population of 100 drafts over 50 generations, and in less time (almost half in some cases), it was decided to carry out further experiments with smaller populations and larger number of generations.

3.3 Searching for Aesthetic Value

It is important to note that none of the judges included scores the poem for aesthetic value, or for grammaticality. The resulting population produced for any given run can be explored in search of acceptable poems. The construction procedure based on a language model trained on human-produced texts guarantees grammaticality to a certain extent, at least to ensure that the poems come up with sentences that sound acceptable in most cases. Those drafts that have suffered a high number of interventions (mutations) by the reviser that replaces words show less grammatical sentences. The interaction between revisers and judges guarantees that the poems in the final population satisfy metric constraints on verse length and stress placement according to traditional metrics.

Users of the system can navigate over the resulting population to select a poem that they find pleasing.

The system includes a number of parameters that can be employed to guide the generation process towards specific types of results, in terms of poem length, or verse length. Poem lengths considered are 14, 8 and 4, which are traditional values for classic stanzas of Spanish poetry (*sonetos*, *octavas reales*, and *cuartetos*). Verse lengths considered are 11 and 8 syllables, which are the traditional meters for Spanish poetry (*endecasílabos* y *octosílabos*).

Some examples of these poems, resulting from training with news articles from the EL País newspaper for 21/05/2013, are given in Table 2 and Table 3.

Valdano. Nosotros. Mourinho le había	*Valdano. Us. Mourinho had given him*
unos alumnos había	*some students had*
hecho música pero ambos chiítas	*made music but both Shiites*
los procedimientos sancionadores	*the sanctioning procedures*
y de cómo se apuntó	*and of how he claimed*
una mancha de justicia.	*a stain of justice.*

Table 2. Example of a poem produced by WASP after training on a collection of news articles from the EL País newspaper for 21/05/2013, with an approximate English translation.

Tengo nada que figuran	*I have nothing that figures*
con nuestra cultura es un	*with our culture is a*
laboratorio financiado	*laboratory financed*
con preferentes está	*with prefered (shares) is*
convirtiendo cada año.	*converting each year.*

Table 3. Example of a poem produced by WASP after training on a collection of news articles from the EL País newspaper for 21/05/2013, with an approximate English translation.

The poem in Table 2 resulted from a run of 10 generations with a population of 50 drafts, aiming for poems of 8 verses 8 syllables long. It obtained a score of 74 and it was rated 23rd of its generation.

The poem in Table 3 resulted from a run of 10 generations with a population of 50 drafts, aiming for poems of 8 verses 8 syllables long. It obtained a score of 75 and it was rated 18th of its generation.

4 Discussion

The poetry generator described here combines an evolutionary solution to model the creative process and a very simple model of the process of renewal of the source of inspiration. In humans, inspiration may come from an innumerable set of possible sources, from direct personal experience to material that has been read, or seen, or heard. The material employed may have been experienced since the last attempt at generation, or simply remembered from an earlier experience. To deal with this vast set of possible sources, humans apply filtering criteria, selecting from the whole set of available material just a small portion which is used to trigger the generation process. Additionally, humans can select different subsets of their remembered material and combine them in surprising ways to produce new results. The model of poetry generation presented here is significantly simpler in that it only considers a single set of resources: the set of texts that it employs as an inspiring set, in the sense described by Ritchie [18]. No selection is carried out over it, and renewal is radical: when the new texts for a day are loaded in the system all texts considered previously are abandoned. This is a limitation of the model.

Improvements can be considered along the lines of more elaborate processes of modification of the inspiring set, possibly by combining new material with the old, possibly by applying a process of selection to decide which parts of the old material are to be combined with the new.

An additional alternative might be to consider an inspiring set of texts that changes dynamically over time, influenced by system outputs, as described in [19].

5 Conclusions and Further Work

A computational model for poetry generation has been proposed that relies on evolutionary approaches to produce poems after training on a set of texts. This set of texts can be varied every day to emulate the way in which humans produce poetry inspired by different material on each successive ocasion that they set out to write. The system has been tested on the set of news articles from the digital edition of a Spanish daily newspaper for a given day, producing a set of poetic elaborations on the topics that are newsworthy on that day.

In many cases, pleasing poems ocur at the lower end of the scores in the population. This suggests that pleasing poems may be culled early during the evolutionary process. As future work, a judge capable of scoring poems on aesthetic potential may be explored. As aesthetic is a subjective quality, this task has significant inherent difficulty.

Acknowledgements

I would like to thank the referees for their comments which helped improve this paper. The research reported in this paper was partially supported by the Ministerio de Educación y Ciencia (TIN2009-14659-C03-01).

References

1. Gervás, P.: WASP: Evaluation of different strategies for the automatic generation of spanish verse. In: Proceedings of the AISB-00 Symposium on Creative & Cultural Aspects of AI. (2000) 93–100
2. Manurung, H.M.: Chart generation of rhythm-patterned text. In: Proc. of the First International Workshop on Literature in Cognition and Computers. (1999)
3. Aamodt, A., Plaza, E.: Case-based reasoning; foundational issues, methodological variations, and system approaches. AI COMMUNICATIONS **7**(1) (1994) 39–59
4. Gervás, P.: An expert system for the composition of formal Spanish poetry. Journal of Knowledge-Based Systems **14**(3–4) (2001) 181–188
5. Chomsky, N.: Syntactic Structures. Mouton, The Hague (1957)
6. Chamberlain, W.: The Policeman's Beard is Half Constructed. Warner Books, New york (1981)
7. Zacks, R.: The story of the 21st century - TR Associate Editor Rebecca Zacks interviews Ray Kurzweil. MIT Technology Review (2000)
8. Riordan, T.: The muse is in the software. The New York Times, 24 November (2003)
9. Kurzweil, R.: Ray Kurzweil's Cybernetic Poet. http://www.kurzweilcyberart.com/poetry/rkcp_overview.php Last visit: 7/03/2013 (2001)
10. Manurung, H.M.: An evolutionary algorithm approach to poetry generation. PhD thesis, University of Edimburgh, Edimburgh, UK (2003)
11. Manurung, R., Ritchie, G., Thompson, H.: Using genetic algorithms to create meaningful poetic text. J. Exp. Theor. Artif. Intell. **24**(1) (2012) 43–64
12. Gervás, P.: Computational modelling of poetry generation. In: Proceedings of the AISB 13 Symposium on AI and Poetry. (2013)
13. Gervás, P.: Diez poemas emocionales generados por un computador. In Cañas, D., González Tardón, C., eds.: ¿Puede un computador escribir un poema de amor?, Editorial Devenir (2010) 189–196
14. Lorca, F.: Poema del cante jondo ; Romancero gitano. Mil letras/ Thousand Letters. Ediciones Cátedra (2009)
15. Hernández, M.: Poemas de adolescencia: Perito en lunas ; Otros poemas. Biblioteca contemporánea. Editorial Losada (1963)
16. Hernández, M., Ballesta, J.: Viento Del Pueblo: Poesía en la Guerra. Letras Hispánicas. Cátedra (1989)
17. Rivers, E.: El Soneto español en el siglo de oro. Nuestros Clasicos Series. Akal (1993)
18. Ritchie, G.: Some empirical criteria for attributing creativity to a computer program. Minds & Machines **17** (2007) 67–99
19. Gervás, P.: Dynamic inspiring sets for sustained novelty in poetry generation. In: Proceedings of the International Conference on Computational Creativity 2011. (2011)

Software *Auto-tuning* con Metaheurísticas Multi-Objetivo

Juan J. Durillo and Thomas Fahringer

Institute for Computer Sciences
University of Innsbruck, Austria
juan@dps.uibk.ac.at , tf@dps.uibk.ac.at

Abstract. La optimización de programas de ordenador es una tarea tradicionalmente aplicada por los programadores. Sin cambiar la semántica de un programa, diferentes implementaciones del mismo pueden dar lugar un rendimiento totalmente distinto. En la era multi-core o many-core, muchos programas son paralelos, situación que, junto a los cada vez más complejos diseños hardware, hacen que el espacio de posibles implementaciones para un mismo programa explote. En estas circunstancias, es necesario diseñar herramientas que busquen de manera automática la mejor variante (*software auto-tuning*). Este trabajo describe cómo el problema de la opmización de código puede ser abordado con metaheuristicas multi-objetivo y presenta un *framework* para software *auto-tuning* consistente en un compilador (the Insieme Compiler) y un entorno de ejecución.

Keywords: Optimización multi-objetivo, metaheurísticas, software auto-tuning

1 Introducción

La puesta a punto o *tuning* de aplicaciones paralelas es una tarea costosa que tiende a introducir errores. Esta tarea consiste en hacer modificaciones en el código del programa y medir el rendimiento o tiempo de ejecución del mismo con respecto a la versión original. La idea de dichas modificaciones es usar de manera eficiente el hardware sobre el que se ejecuta dicha aplicación. El principal problema de esta tarea es que transformaciones en el código de una aplicación para una determinada plataforma pueden ser no beneficiosas para otra plataforma. Lo mismo ocurre con distintas aplicaciones: modificaciones sobre un determinado trozo de código no tienen por qué ser beneficiosas para otro programa.

En el contexto anterior, el *tuning* automático de aplicaciones o *auto-tuning* es un atractivo y necesario paradigma para mejorar el rendimiento de aplicaciones. La idea de *auto-tuning* es automatizar el proceso de transformar y modificar el código fuente de un programa para obtener el máximo rendimiento. En este sentido, en la teoría de compiladores existen una serie de transformaciones que permiten modificar sintácticamente un trozo de código sin variar su semántica. Ejemplo de estas transformaciones son: el desenrrollo de bucles (*loop unrolling*),

bloqueo de datos (*blocking or tiling*), o el intercambio de bucles (*loop interchanging*). En muchos casos, dichas transformaciones tienen asociadas una serie de parámetros que permiten afinar aún más el comportamiento del programa. Como consecuencia del alto número de posibless transformaciones y parámetros, estos *auto-tuners* se enfrentan a un enorme espacio de búsqueda.

Un enfoque para *auto-tuning* consiste en aplicar alguna estrategia de búsqueda sobre el conjunto de posibles variantes de código durante el tiempo de compilación del programa. En este sentido, *Simplex* ha sido una de las alternativas preferidas junto a algún *algoritmo genético* (GA) [10, 16]. El principal problema es, sin embargo, que dichos mecanismos de *auto-tuning* han sido tradicionalmente aplicados para la optimización de un sólo objetivo [18, 17, 12, 7, 15, 6, 2, 1, 3, 8]: el tiempo de ejecución de un programa, consumo de memoria o uso de recursos, etc. Pocos, o ningún trabajo, han tratado varios objetivos contradictorios al mismo tiempo.

Un ejemplo con varios objetivos contradictorios aparece en aplicaciones paralelas. Es un hecho bien conocido que en muchos casos el *speed-up* de una aplicación paralela con respecto a la version secuencial de la misma no escala linealmente con el número de hilos de ejecución. De hecho, la ley de Amdahl establece que el máximo *speed-up* que se puede obtener está limitado por la porción secuencial de dicho programa. En dichos casos, el tiempo de ejecución de un programa paralelo y su eficiencia pueden verse como objetivos contradictorios: el tiempo de ejecución disminuirá a medida que lo hace también la eficiencia de nuestro programa. Por ejemplo, un programa que use un sólo hilo de ejecución tiene una eficiencia igual al 100%. El mismo programa usando dos hilos de ejecución tendrá una eficiencia menor que el 100%. A medida que el número de hilos aumenta, la eficiencia de nuestro programa decrece. Obviamente, todos queremos minimizar el tiempo de ejecución y tener el programa más eficiente posible. En este caso, el frente de Pareto estaría formado por versiones de dicho programa usando diferentes hilos de ejecución.

Este trabajo presenta un *framework* para hacer *auto-tuning* de aplicaciones paralelas. Dicho *framework* se ha implementado dentro del proyecto *Insieme Compiler*[1] y consiste en: (1) un compilador que genera un conjunto de programas que son Pareto óptimos con respecto al tiempo de ejecución y su eficiencia; y (2) un entorno de ejecución que en tiempo real selecciona el mejor programa, de acuerdo a las preferencias del usuario que ejecuta la aplicación. En particular, el presente trabajo describe la búsqueda implementada en la parte del compilador de nuestro *framework*. El mecanismo de búsqueda usado por nuestro compilador consiste en un híbrido entre el algoritmo de búsqueda diferencial (GDE3) (*differential evolution*) [9] y la teoría de conjuntos difusos (*rough sets*) [11].

El resto de este trabajo se estructura de la siguiente manera. La siguiente sección describe el problema. La sección 3 describe el método empleado para optimizar código. Tras ello, la sección 4 presenta los resultados obtenidos por nuestro método en una serie de problemas. Finalmente se presentan las conclusiones y trabajo futuro.

[1] http://www.insieme-compiler.org/

2 Descripción del Problema

Consideremos que t_s es el tiempo de ejecución la versión secuencial más rápida s y $t_p(x)$ el tiempo de p usando x threads. El *speed-up* de p usando x threads se define como $s(x) = \frac{t_s}{t_p(x)}$ y su eficiencia se define como $e(x) = \frac{s(x)}{x}$. En esta ecuación podemos a su vez sustituir el denominador de forma que tenemos $e(x) = \frac{t_s}{x \times t_p(x)}$. Ambas ecuaciones dependen de la versión secuencial más rápida, lo cual no siempre es conocido. Sin embargo, t_s es una constante en nuestras ecuaciones que puede ser omitida cuando comparamos la calidad de diferentes versiones paralelas de un programa. En particular, si queremos maximizar la eficiencia de nuestro programa, lo único que tendremos que hacer es minimizar los términos $t_p(x)$ y $x \times t_p(x)$. De esta manera, estaremos buscando las soluciones más rápidas y eficientes [2].

3 El algoritmo de optimización RS-GDE3

Esta sección describe el algoritmo de optimización RS-GDE3. Dado un programa de ordenador al que se le han aplicado una serie de transformaciones, las cuales tienen varios parámetros por determinar, nuestro objetivo es encontrar los valores óptimos para dichos parámetros de forma que se minimice el tiempo de ejecución y se aumente la eficiencia de dicho programa. En este trabajo proponemos un algoritmo que combina el uso de GDE3 y la teoría de conjuntos difusos. La idea es que GDE3 guíe la búsqueda iterando sobre el posible espacio de configuraciones y usar la teoría de conjuntos difusos para reducir el espacio de búsqueda en la medida de lo posible. Esta segunda condición viene impuesta por el hecho de que queremos minimizar el tiempo de compilación. A diferencia de otros mecanismos de reducción del espacio de búsqueda en el contexto de *auto-tuning* [14], nuestro método no requiere de ningún conocimiento específico del dominio de la aplicación, ni tampoco derivar complejos método analíticos, siendo por tanto un mecanismo estándar e independiente de la aplicación a ejecutar.

La figura 3 (a) muestra una representación gráfica de nuestro técnica. Ésta consiste en la búsqueda iterativa de nuevas y mejores configuraciones de nuestro programa. En cada iteración, un conjunto de nuevas configuraciones es generado por medio de GDE3. Tras ello, los límites del espacio de búsqueda se actualizan de acuerdo a la información proporcionada por las soluciones dominadas y no-dominadas conocidas hasta este momento. Dicha actualización se realiza utilizando la mencionada teoría de conjuntos difusos. En la fase inicial, dichos límites se determinan por los valores máximos y mínimos que pueden adquirir los parámetros a modificar en nuestro código. Como criterio de parada, nuestro

[2] El lector debe tener en cuenta que el cálculo de la eficiencia tiene en cuenta el número de hilos de ejecución. De esta manera, es un razonamiento equivocado afirmar que la versión más rápida es la más eficiente. Esta afirmación únicamente es cierta cuando dos versiones usan el mismo número de hilos de ejeución. Sin embargo, como se explica más adelante, el número de hilos de ejecución es una de las variables de nuestro problema.

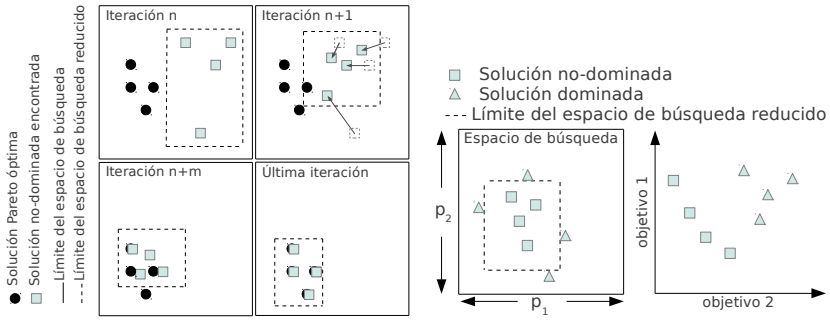


Fig. 1. a) Proceso iterativo en RS-GDE3. b) Reducción del espacio de búsqueda usando conjuntos difusos.

método considera que ha encontrado un conjunto de soluciones Pareto óptimas aceptables cuando durante un periodo de 5 iteraciones no es capaz de generar soluciones que dominen al conjunto de Pareto actual.

3.1 Código a Optimizar

En este trabajo RS-GDE se aplica para optimizar código escrito en OpenMP. Dicho paradigma permite paralelizar un programa mediante el uso de *pragmas* o anotacioines que define que partes de un programa pueden ser ejecutadas en paralelo. Es decir, usando dichas anotaciones se puede definir cuantos hilos deben ejecutar de forma paralela un trozo de código dentro de un programa. Un ejemplo de anotación se muestra a continuación.

```
#pragma omp parallel for
for (int i=0; i < 10; i++) {
do_something(i)
}
```

La anotación anterior indica que el bucle *for* definido justo después es paralelo y, por tanto, todas sus iteraciones se pueden ejecutar de forma simultanea.

Además, en nuestro caso, para cada programa OpenMP se aplican una serie de transformaciones. Dichas transformaciones tienen como objetivo reutilizar de la manera más eficiente posible los datos del programa que se encuentran ya en la memoria *cache* del procesador. En este trabajo hemos usado la transformación conocida como *tiling*. Supongamos que tenemos un bucle como el siguiente para multiplicar dos matrices a y b y almacena el resultado en c.

```
for (i=0; i < n; i++) {
c[i] = 0;
for (j=0; j < n; j++) {
c[i] = c[i] + a[i][j] * b[j];
}
}
```

El bucle anterior puede ser transformado en un bucle equivalente pero que reutiliza los datos en la *cache* de manera más eficiente. Dicho nuevo bucle se puede escribir como sigue a continuación. El nuevo programa multiplica las dos matrices multiplicando pequeños bloques, que pueden ser lo suficientemente pequeños para caber en la *cache* (en este ejemplo se usan bloques 2×2). [3].

[3] El lector puede verificar que el resultado de ambos programas es el mismo.

```
for (i=0; i < n; i+=2) {
c[i] = 0;
c[i+1] = 0;
for (j = 0; j < n; j+=2) {
for (ii = i; ii < min(i+2, n); ii++) {
for (jj = j; jj < min(j+2, n); jj++) {
c[ii] = c[ii] + a[ii][jj] * b[jj];
}
}
}
}
```

El tamaño de los bloques es un parámetro a determinar, cuya única restricción es ser mayor que 0 y menor que la dimensión de la matriz a multiplicar. En este caso, nuestro método debe encontrar el tamaño ideal de bloque para multiplicar dichas matrices y el número de hilos, dando lugar a un conjunto de configuraciones de dicho program que minimizan su tiempo de ejecución y maximizan su o eficiencia. Dichas configuraciones se representan en forma de vector. RS-GDE3 trabajará con dichos vectores. Cada vez que un nuevo vector es generado, dicha vector se utiliza para instanciar el bucle anterior con los valores de dicha configuración. Dicha instanciación tiene lugar mediante el uso del compilador Insieme. Una vez instanciada, el bucle se ejecuta y se calculan los valores $t_p(x)$ y $x \times p(x)$, los cuales serán usados como fitness en RS-GDE3 para minizimar el tiempo de ejecución y maximizar la eficiencia.

En el caso de matrices de tamaño 1200 x 1200 y una máquina con hasta 40 cores o hilos de ejecución, el número posibles configuraciones es $1200 \times 1200 \times 40 = 57600000$, donde 1200×1200 define el las dimensiones máximas del bloque usado para multiplicar las matrices.

3.2 GDE3

La evolución diferencial es una de las técnicas de optimización más populares en el campo de los algoritmos evolutivos. En el campo de la optimización multi-objetivo, GDE3 [9] es uno de los métodos de evolución diferencial más conocidos. Una de las razones para seleccionar dicho método como motor de búsqueda dentro de nuestro compilador es su robustez y su rapidez convergiendo a un conjunto de soluciones Pareto óptimas [5].

GDE3 utiliza una población de tamaño N de posibles soluciones, que en este trabajo representan diferentes configuraciones de un programa. Dicha población se inicializa aleatoriamente. En concreto, dichas configuraciones consisten en el mismo código transformado, pero con diferentes valores para los parámetros que gobiernan su funcionamiento. En cada iteración GDE3 genera una nueva población de descendientes. Para cada configuración a en nuestra población inicial, GDE3 selecciona otras tres configuraciones b, c y d para generar un posible descendiente r. El método por el cual se genera dicha configuración se describe en el Pseudocódigo 1 [4]. En dicho código los parámetros CR y F guían la forma en la que nuevas configuraciones son generadas. En nuestro caso hemos usado $CR = 0.5$ y $F = 0.5$.

[4] El pseudocodigo muestra el operador de cruce usado en evolución diferencial, no el algoritmo completo.

Algorithm 1 Generando una nueva configuración mediante evolución diferencial

```
1: Entrada: a, b, c, d cuatro configurations diferentes, B la frontera actual
2: Salida: r
3: index =floor(rand() · |a|) + 1
4: for i = 1 .. |a| do
5:     if (rand() < CR ∨ i = index) then
6:         r(i) = b(i) + F · (c(i) − d(i))
7:     else
8:         r(i) = a(i)
9:     end if
10: end for
11: return B.getClosestTo(r)
```

Una vez generada la nueva configuración r, ésta se compara con a. Si r es dominada por a, entonces dicha solución es descartada como posible descendiente y a se inserta en la población de descendientes. Si r domina a la solución a entonces r se inserta en la población de descendientes. Finalmente, si a y r son no-dominadas, ambas son insertadas en la población de descendientes. Una vez que un nuevo individuo se ha generado por cada solución en la población inicial, la población de descendientes se convierte en la población actual de GDE3 tras limitar su tamaño a un máximo N seleccionando las mejores soluciones por ranking y crowding (ver [9] para más detalles). Antes de seguir con la siguiente generación, los límites del espacio de búsqueda son actualizados usando como referencia las soluciones dominadas y no-dominadas en la población actual y la teoría de conjuntos difusos explicada en la siguiente sección.

Como se ha comentado anteriormente el algoritmo acaba cuando durante 5 iteraciones seguidas ninguna de las soluciones generadas pasa a formar parte de la población de descendientes.

3.3 Teoría de conjuntos difusos

En cada iteración de GDE3 los límites del espacio de búsqueda se actualizan. La idea de esta actualización es reducir el espacio de búsqueda de forma que se acelere la convergencia de nuestro método a un conjunto de soluciones satisfactorias. Nuestro método usa la teoría de los conjuntos difusos para tal fin.

Esta teoría es un enfoque matemático para modelar conocimiento imperfecto acerca de un conjunto de soluciones. En nuestro caso, queremos modelar qué área del espacio de búsqueda contiene soluciones Pareto óptimas y qué área del espacio de búsqueda podemos ignorar porque sólo contiene soluciones que son dominadas. El enfoque usado en nuestro trabajo es similar al enfoque usado en [13]. Dicho enfoque se describe brevemente en la figura 3 (b).

El ejemplo mostrado en la figura consiste en una configuración con dos parámetros p_1 y p_2 que optimizan dos objetivos distintos. Para reducir el espacio de búsqueda nuestro algoritmo explora las configuraciones contenidas en la población actual de GDE3. Dicha población contendrá un conjunto de soluciones no-dominadas y un conjunto de soluciones dominadas (cuadrados y triángulos en nuestra figura). El objetivo es modelar el área que contiene únicamente las soluciones no-dominada (los cuadrados). Para calcular los límites de dicha región,

Table 1. Detalles de los problemas evaluados

Problema	Tamaño	Complejidad Computacional	Complejidad Espacial
mm	1400^2	$\mathcal{O}(N^3)$	$\mathcal{O}(N^2)$
dsyrk	1400^2	$\mathcal{O}(N^3)$	$\mathcal{O}(N^2)$
jacobi-2d	10000^2	$\mathcal{O}(N^2)$	$\mathcal{O}(N^2)$
3d-stencil	600^3	$\mathcal{O}(N^3)$	$\mathcal{O}(N^3)$
n-body	500000	$\mathcal{O}(N^2)$	$\mathcal{O}(N)$

nuestro mecanismo basado en conjuntos difusos usa las coordenadas de las soluciones dominadas que encierran las soluciones no-dominadas. En concreto el área mínima que encierra a todas las soluciones no-dominadas es usada como nuevo espacio de búsqueda para GDE3.

Una desventaja de este método es que el nuevo espacio de búsqueda puede no contener todas las soluciones no-dominadas incluidas en el espacio de búsqueda. Para mitigar dicho efecto, nuestro enfoque consiste en actualizar continuamente los límites del espacio de búsqueda considerando la posibilidad de mover y ampliar los límites del espacio de búsqueda en cada generación en contraposición al trabajo descrito en [13]. Dicha modificación del espacio de búsqueda se realiza mediante la posibilidad de generar soluciones fuera del espacio de búsqueda reducido mediante el uso de un operador de mutación. La mutación polynomial [4] es el operador usado en nuestro trabajo.

4 Evaluación

Esta sección evalúa el rendimiento de RS-GDE3. Primero se define los problemas que se han evaluado. Seguidamente se describen las técnicas con las cuales se ha comparado. Finalmente se presentan los resultados obtenidos.

4.1 Problemas

En este trabajo hemos considerado la optimización de 5 aplicaciones de álgebra lineal. En particular dichas aplicaciones son la multiplicación de matrices (mm), dysrk (dadas las matrices A y B calcula $B = A * A^T + B$), dos aplicaciones conocidas como jacobi y 3d-stencil y una implementación de la simulación del programa $n - body$. La tabla 1 define las características de dichos programas.

4.2 Técnicas usadas

Nuestro método RS-GDE3 se compara con un mecanismo de búsqueda aleatoria y un mecanismo de búsqueda enumerativa o jerárquica [5]. El método de búsqueda

[5] El lector no debe confundir busuqeda jerarquíca con fuerza bruta. En el primer caso, se pretende analizar el espacio de búsqueda de manera organizada, pero no se asume la evaluación de todas las posibles soluciones. En el segundo caso, se evaluan todas las posibles soluciones dentro del espacio de búsqueda. Sólo en el caso de la multiplicación de matrices, un algoritmo de fuerza bruta tardaría años en evaluar todas las posiblidades.

jerárquica hace un *sampling* del espacio de búsqueda evaluando para cada problema un número diferente de soluciones. La búsqueda aleatoria evalúa en total tantas soluciones como RS-GDE3. Debido a esta última condición primero se evalúa RS-GDE3 y después la búsqueda aleatoria. Un total de 10 ejecuciones de cada problema se realizan. RS-GDE3 utiliza un tamaño de población $N = 30$.

4.3 Resultados

En esta sección se describen los resultados obtenidos por los diferentes métodos evaluados. En concreto prestamos atención a la métrica hipervolumen *Hypervolume* [19] [6] y al número de evaluaciones requeridos por cada método. Los resultados presentados han sido validados mediante el test estadístico *Wilcoxon*. Dichos datos se omiten en este documento por razones de espacio. Para evaluar los problemas hemos considerados dos arquitectural diferentes: Intel *Westmere*, con un total de hasta 40 hilos de ejecución y AMD *Barcelona* con un total de hasta 32 hilos de ejecución. La tabla 2 muestra los resultados obtenidos. En dicha tabla, E representa el número de evaluaciones del problema realizadas por cada método y $HV(S)$ el hipervolumen medio de las diez ejecuciones realizadas.

En el caso de la arquitectura Intel, es posible observar como nuestro método calcula soluciones de mayor hipervolumen que ambos, el método jerárquico y aleatorio. En el caso del método jerárquico, RS-GDE3 sólamente requiere una pequeña fracción del número de evaluaciones usado por éste. La figura 2 muestra un ejemplo de frente de Pareto calculados por cada una de las tres técnicas evaluadas para el problema de la multiplicación de matrices.

En el caso de la arquitectura AMD, es posible observar como los resultados de RS-GDE3 son bastantes competitivos con respecto a la búsqueda jerárquica, aunque este último encuentra frentes con un mayor hipervolumen (en algunos casos sólo con una ligera diferencia). En este caso, RS-GDE3 utiliza también un pequeño porcentaje de las evaluaciones requeridas por la búsqueda jerarquica (menos del 5% en todos los problemas). Nuestra hipótesis para esto es que las soluciones óptimas en este caso se encuentran dispersa en áreas muy diferentes del espacio de búsqueda debido al uso de la *victim cache* usada por la arquitectura AMD. En la comparativa con la búsqueda aleatoria, RS-GDE3 siempre obtiene mejores resultados usando el mismo esfuerzo computacional.

5 Conclusiones

En este trabajo se presenta un método para optimizar el tiempo de ejecución y la eficiencia de un programa paralelo usando el compilador Insieme. Dicho método, llamado RS-GDE3, consiste en la hibridación entre el mecanismo de evolución diferencial GDE3 y la teoría de conjuntos difusos. Con el objetivo de compara el rendimiento de RS-GDE3 en este trabajo se compara con una búsqueda numerativa jerárquica y una búsqueda aleatoria para optimizar un conjunto de 5 problemas. Los resultados obtenidos muestran que RS-GDE3 es capaz de obtener

[6] Valores más altos de esta métrica indican frentes con mejor convergencia y diversidad.

Table 2. Número de evaluaciones (E) e hipervolumen medio ($HV(S)$) obtenido por las tres técnicas analizadas:

Intel Westmere

	Búsqueda Jerarquica		Búsqueda Aleatoria		RS-DGE3	
Benchmark	E	$HV(S)$	$\overline{E}$	$\overline{HV(S)}$	$\overline{E}$	$\overline{HV(S)}$
mm	71290	0.84	780	0.03	**724**	**0.88**
dsyrk	71290	0.75	1200	0.00	**1186**	**0.81**
jacobi-2d	23805	0.69	**1025**	0.73	**1027**	**0.83**
3d-stencil	10580	0.77	1000	0.52	**852**	**0.86**
n-body	26136	0.86	1350	0.65	**1334**	**0.95**

Intel Barcelona

	Búsqueda Jerarquica		Búsqueda Aleatoria		RS-DGE3	
Benchmark	E	$V(S)$	$\overline{E}$	$\overline{HV(S)}$	$\overline{E}$	$\overline{HV(S)}$
mm	85548	**0.83**	800	0.01	**740**	0.76
dsyrk	85548	**0.87**	1200	0.00	**1152**	0.78
jacobi-2d	28566	0.73	675	0.81	**674**	**0.88**
3d-stencil	12696	**0.87**	850	0.78	**822**	0.85
n-body	21780	**0.78**	1100	0.51	**1063.6**	0.68

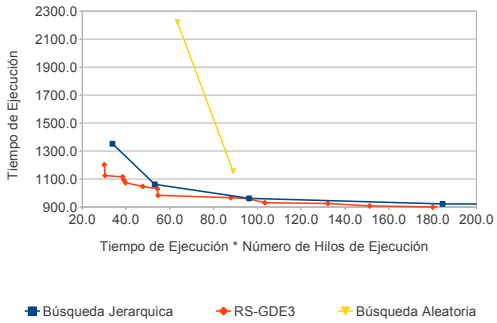


Fig. 2. Frentes encontrados para Intel Westmere.

resultados mejores o comparables a la búsqueda aleatoria usando un total de menos del 5% del esfuerzo computacional requerido por la búsqueda enumerativa. Como trabajo futuro se plantea la inclusión de más técnicas de optimización dentro del compilador Insieme y uso de otras técnicas como *machine learning*.

Acknowledgments. The authors acknowledge the Austrian Research Promotion Agency under contract nr. 834307 (AutoCore)

References

1. Saman P. Amarasinghe. Petabricks: a language and compiler based on autotuning. In Manolis Katevenis, Margaret Martonosi, Christos Kozyrakis, and Olivier Temam, editors, *HiPEAC*, page 3. ACM, 2011.
2. J. Ansel, C. Chan, Y.L. Wong, M. Olszewski, Q. Zhao, A. Edelman, and S. Amarasinghe. *PetaBricks: a language and compiler for algorithmic choice*, volume 44. ACM, 2009.

3. Matthias Christen, Olaf Schenk, and Helmar Burkhart. Patus: A code generation and autotuning framework for parallel iterative stencil computations on modern microarchitectures. In *IPDPS*, pages 676–687. IEEE, 2011.
4. Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
5. Juan J. Durillo, Antonio J. Nebro, Francisco Luna, Carlos A. Coello Coello, and Enrique Alba. Convergence speed in multi-objective metaheuristics: Efficiency criteria and empirical study. *International Journal for Numerical Methods in Engineering*, 84(11):1344 – 1375, December 2010.
6. K. Fatahalian, T.J. Knight, M. Houston, M. Erez, D.R. Horn, L. Leem, J.Y. Park, M. Ren, A. Aiken, W.J. Dally, et al. Sequoia: programming the memory hierarchy. In *SC 2006 Conference, Proceedings of the ACM/IEEE*, pages 4–4. IEEE, 2006.
7. M. Frigo. A fast fourier transform compiler. In *Acm Sigplan Notices*, volume 34, pages 169–180. ACM, 1999.
8. Shoaib Kamil, Cy Chan, Leonid Oliker, John Shalf, Samuel Williams, and Samuel Williams. An auto-tuning framework for parallel multicore stencil computations. In *IPDPS*, pages 1–12. IEEE, 2010.
9. Saku Kukkonen and Jouni Lampinen. Gde3: the third evolution step of generalized differential evolution. In *IEEE Congress on Evolutionary Computation*, pages 443–450. IEEE, 2005.
10. J. A. Nelder and R. Mead. A Simplex Method for Function Minimization. *The Computer Journal*, 7(4):308–313, January 1965.
11. Zdzisaw Pawlak. Rough sets. *International Journal of Parallel Programming*, 11(5):341–356, 1982.
12. M. Puschel, J.M.F. Moura, J.R. Johnson, D. Padua, M.M. Veloso, B.W. Singer, J. Xiong, F. Franchetti, A. Gacic, Y. Voronenko, et al. Spiral: Code generation for dsp transforms. *Proceedings of the IEEE*, 93(2):232–275, 2005.
13. Luis V. Santana-Quintero, Alfredo García Hernández-Díaz, Julián Molina Luque, Carlos A. Coello Coello, and Rafael Caballero. Demors: A hybrid multi-objective optimization algorithm using differential evolution and rough set theory for constrained problems. *Computers & OR*, 37(3):470–480, 2010.
14. Jun Shirako, Kamal Sharma, Naznin Fauzia, Louis-Noël Pouchet, J. Ramanujam, P. Sadayappan, and Vivek Sarkar. Analytical bounds for optimal tile size selection. In *ETAPS International Conference on Compiler Construction (CC'12)*, Tallinn, Estonia, March 2012. Springer Verlag.
15. C. Tapus, I.H. Chung, and J.K. Hollingsworth. Active harmony: Towards automated performance tuning. In *Supercomputing, ACM/IEEE 2002 Conference*, pages 44–44. IEEE, 2002.
16. Ananta Tiwari and Jeffrey K. Hollingsworth. Online adaptive code generation and tuning. In *Proceedings of the 2011 IEEE International Parallel & Distributed Processing Symposium*, pages 879–892. IEEE Computer Society, 2011.
17. R. Vuduc, J.W. Demmel, and K.A. Yelick. Oski: A library of automatically tuned sparse matrix kernels. In *Journal of Physics: Conference Series*, volume 16, page 521. IOP Publishing, 2005.
18. R.C. Whaley and J.J. Dongarra. Automatically tuned linear algebra software. In *Proceedings of the 1998 ACM/IEEE conference on Supercomputing (CDROM)*, pages 1–27. IEEE Computer Society, 1998.
19. E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.

Algoritmos evolutivos multi-objetivo para el diseño de plantas fotovoltaicas con seguimiento solar

Daniel Gómez Lorente[1], Isaac Triguero[2], Consolación Gil[3], Antonio Espín Estrella[1]

[1]Dept. de Ingeniería Civil, Sección de Ingeniería Eléctrica, ETSICCP, Universidad de Granada, Campus Fuentenueva, Granada. 18071, España
[2]Dept. Computer Science and Artificial Intelligence, CITIC-UGR (Research Center on Infomation and Communications Technology). Universidad de Granada, 18071 Granada, España
[3]Dept.Informatics, CITE III. Universidad de Almería, La Cañada de San Urbano s/n, Almería, España

Resumen. La descentralización de la producción de energía eléctrica ha propiciado un uso más eficaz y armonioso de los recursos energéticos. Por esta razón, las plantas fotovoltaicas de conexión a red (PFVCRs), así como otras fuentes de energía renovables han cobrado un gran interés en los últimos años ya que mejoran el suministro de energía eléctrica de la red. La optimización del diseño de una PFVCR ha sido previamente tratada en términos de pérdidas eléctricas con resultados exitosos. Sin embargo, estos resultados pueden mejorarse aún más si se toman en consideración, además, otros aspectos como la potencia instalada. Este trabajo se centra en la optimización del diseño de plantas fotovoltaicas con seguimiento solar. La investigación descrita tiene dos objetivos: (i) la maximización de la potencia instalada en el terreno; (ii) la minimización de las pérdidas eléctricas por efecto Joule en los conductores. El problema ha sido abordado con algoritmos evolutivos multi-objetivo, que han demostrado ser técnicas de optimización de gran utilidad para problemas con varios objetivos. Este artículo se centra en NSGA-II y SPEA2, dos algoritmos multi-objetivo ampliamente conocidos, y describe cómo se han usado para optimizar el diseño de la PFVCR. Los conjuntos de soluciones obtenidos como resultado proporcionan la flexibilidad y la adaptabilidad necesarias para construir una PFVCR. Por lo tanto estos algoritmos han resultado ser una herramienta eficaz para mejorar el rendimiento de una PFVCR.

1 Introducción

El progresivo agotamiento de las reservas de combustibles fósiles así como el constante aumento de las emisiones de CO_2 y la creciente demanda de energía han conllevado un creciente uso y una creciente explotación de las fuentes de energía renovables. Una de estas fuentes es la energía solar obtenida de los sistemas fotovoltaicos (PV). Por otro lado, la descentralización de la producción de energía eléctrica en la cual la electricidad es generada localmente cerca de zonas urbanas se ha convertido también en un importante objetivo [1], ya que esto reduce las pérdidas energéticas en el transporte. Es por ello que las plantas fotovoltaicas conectadas a red (PFVCRs) se

utilizan a menudo para abastecer consumos locales con la energía producida por los módulos fotovoltaicos.

Antes de comenzar la instalación de un PFVCR, deben considerarse diversos parámetros, como la irradiación, temperatura y viento en la ubicación propuesta. Esto es necesario para evaluar la viabilidad y rentabilidad de la PFVCR. Sin embargo, incluso después de que finalicen estos pasos, el diseño de la distribución y el tamaño de una planta fotovoltaica con seguidores solares está lejos de ser una tarea sencilla [2] ya que hay un gran número de variables que deben tenerse en cuenta. Estas variables determinan la eficiencia y eficacia de la instalación FV. Dependiendo de la ubicación potencial de la instalación FV, sería mejor optar por un menor número de seguidores solares grandes o por el contrario, un mayor número de seguidores pequeños. Cualquiera que sea la solución, afectará a la distancia entre los seguidores, la longitud de los conductores eléctricos y la elección de los inversores. Así, el diseño de los sistemas fotovoltaicos puede verse como un problema de optimización continua, que puede resolverse mediante algoritmos evolutivos [3, 4, 5].

Este trabajo propone el uso de algoritmos evolutivos multi-objetivo (MOEAs), que son capaces de mantener un equilibrio entre dos o más objetivos y han demostrado ser una técnica de optimización muy útil cuando disponemos de más de un objetivo. Concretamente, nuestra investigación se centra en dos algoritmos multi-objetivo bien conocidos, llamados NSGA-II [6] y SPEA2 [7], analizando su capacidad para optimizar la PFVCR. En este caso el problema va a consistir en la optimización de dos objetivos; i) maximizar la potencia instalada en el terreno; ii) minimizar las pérdidas eléctricas por efecto Joule en la instalación. Las soluciones resultantes proporcionan la flexibilidad y la adaptabilidad necesarias para construir una PFVCR, por lo que estos algoritmos han demostrado ser una herramienta eficaz para mejorar el rendimiento de los sistemas fotovoltaicos conectados a red.

Para analizar la efectividad de estas técnicas en el diseño de PFVCRs, se han estudiado varios terrenos así como la respuesta de cada uno de los MOEAs aplicados para su optimización. Los resultados están representados por curvas o frentes en ejes cartesianos que muestran los valores numéricos de los dos objetivos (potencia instalada y pérdidas eléctricas). Estos conjuntos de resultados óptimos nos proporcionan un abanico flexible de soluciones que va en consonancia con las necesidades del proyectista a la hora de valorar que opción es la más adecuada para el terreno objetivo.

2 MOEAs para optimizar el diseño de PFVCRs con seguidores solares

Esta sección explica la estructura del problema, así como cada una de las soluciones que se han adoptado para su resolución. Dado un terreno rectangular con unas dimensiones y una latitud específicas, se han usado MOEAs para encontrar la configuración óptima en la que las pérdidas por efecto Joule en los conductores eléctricos desde los seguidores a los inversores sean mínimas, manteniendo la potencia instalada en ese campo tan alta como sea posible. Por lo tanto, los dos objetivos de optimización han

sido la minimización de pérdidas eléctricas por efecto Joule y la maximización de la potencia instalada.

2.1 Codificación de los parámetros y estructura de los individuos

En primer lugar, en nuestra investigación, ha sido necesario definir una codificación de las soluciones. En los MOEAs propuestos, las variables que componen la configuración de la planta fotovoltaica con seguidores solares son los atributos que forman los individuos [3]. Por lo tanto, cada individuo en la población es una solución completa codificada. En otras palabras, todas las variables que componen el diseño de una instalación fotovoltaica se han codificado secuencialmente en cada individuo. Así, un individuo está compuesto por ocho variables diferentes, como se muestra en la Figura 1. Puesto que los valores máximos y mínimos de estas variables son especificados al inicio del algoritmo, éste no va más allá de estos límites en su búsqueda, para de esta manera, no generar soluciones físicamente irrealizables. Por lo tanto, para resolver nuestro problema, se han establecido los valores de los catálogos de los principales fabricantes de módulos fotovoltaicos e inversores como valores límites para nuestros individuos. Como se indica en [8], los valores de cada individuo son inicialmente generados de forma aleatoria.

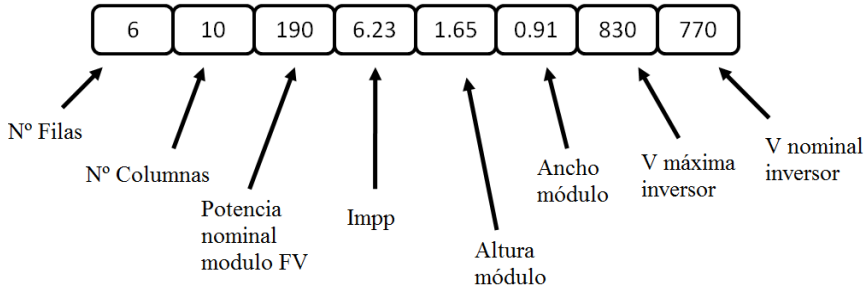


Fig. 1. Ejemplo de un individuo para una PFVCR

2.2 Generación de nuevas configuraciones

Tras el proceso de inicialización, cada MOEA entra en un bucle iterativo para realizar diferentes operaciones con el fin de definir nuevos individuos $U_{i,G}$. Los operadores de selección, cruce y mutación generan nuevas configuraciones en cada generación. Después de aplicar estos operadores, es necesario verificar que cada individuo $U_{i,G}$ ha sido generado con los valores correctos teniendo en cuenta las características de los prototipos, es decir, los valores son revisados para ver si se encuentran en el rango adecuado (ver Tabla 1).

Tabla 1. Rangos de diseño del problema

Parámetro	Mínimo	Máximo
Nº filas	5	9
Nº columnas	8	14
Potencia módulo (W)	150	290
I_{mpp} (A)	4.490	10.230
Altura módulo (m)	1.324	2.000
Ancho módulo (m)	0.8000	1.061
Tensión máxima inversor (V)	700	880
Tensión nominal inversor (V)	600	800

2.3 Funciones de aptitud

Como se ha mencionado anteriormente, nuestra investigación tiene dos objetivos. Consecuentemente, existe una función objetivo y un valor de aptitud para cada uno de los dos objetivos, los cuales van a ser representados en forma de frentes en unos ejes cartesianos.

Pérdidas eléctricas.

En este caso, el valor de aptitud consiste en las pérdidas eléctricas por efecto Joule obtenidas por la configuración, como se muestra en la ecuación (1). Esta primera función de aptitud está basada en las pérdidas por efecto Joule porque, como se ha citado anteriormente, es uno de los factores más importantes en el diseño del sistema FV. De hecho, hay otras pérdidas en una instalación fotovoltaica, pero nuestra decisión inicial ha sido enfocar el problema de la optimización de una planta fotovoltaica de conexión a red sólo considerando las pérdidas por efecto Joule que son eléctricamente más susceptibles de optimizar (debido al margen de optimización) y están influenciadas por la topología de la conexión serie-paralelo de los módulos FV. Además, el algoritmo elimina las pérdidas por auto-sombreado ya que, para cada solución obtenida, estima el valor de la superficie de la estructura de seguimiento y por lo tanto, se hace una distribución los seguidores de forma que no existan pérdidas por auto-sombreado. El algoritmo realiza el cálculo de las pérdidas eléctricas por efecto Joule mediante la siguiente ecuación:

$$P(\%) = \frac{\sum_{i=1}^{N} 2 \cdot \rho \cdot L_i \cdot I_i^2}{N.filas \cdot N.columnas \cdot Pw \cdot N} \quad (1)$$

donde P es el porcentaje de pérdidas por efecto Joule en todos los conductores de la planta fotovoltaica, sobre la potencia instalada, N es el número de seguidores solares instalados en el terreno; ρ es la resistividad del conductor de cobre; N. filas y N. columnas, son el número de filas y de columnas de módulos fotovoltaicos situados sobre cada estructura de seguimiento, y por último, Pw es la potencia nominal de cada módulo fotovoltaico.

Por lo tanto, una vez que la función de aptitud ha sido definida, los nuevos individuos $U_{i,G}$ con valores de aptitud más pequeños serán seleccionados como los más aptos para que cedan sus atributos a la próxima generación $X_{i,G+1}$:

$$X_{i,G+1} = \begin{cases} U_{i,G} \; si \; Joule\,losses\left(U_{i,G}\right) \leq Joule\,losses\left(X_{i,G}\right) \\ X_{i,G} \; en\,cualquier\,otro\,caso \end{cases} \quad (2)$$

Potencia instalada

El valor de la aptitud referente a la potencia instalada en el terreno es calculado por el algoritmo según la siguiente expresión (3):

$$Potencia\,instalada\,(kW) = N.\,filas \cdot N.columnas \cdot Pw \cdot N.seguidores \quad (3)$$

El resultado obtenido será mejor cuanto mayor sea la potencia instalada en el terreno. Sin embargo, para una mejor visualización de los resultados y una mejor representación de los frentes en los ejes cartesianos, se ha decidido representar el valor inverso de la potencia instalada, para de esta forma, tener dos objetivos de minimización. Por lo tanto, minimizando el inverso de la potencia instalada, estamos maximizando ésta.

3 Marco experimental y resultados

En esta sección se explica la estructura del problema, así como las herramientas adoptadas para su resolución. Se muestran los resultados obtenidos por los algoritmos, así como una comparación entre ellos y un algoritmo de evolución diferencial de un solo objetivo, utilizado en un trabajo previo.

3.1 Marco experimental

El rendimiento de los algoritmos utilizados ha sido analizado con 20 terrenos diferentes, que han sido seleccionados aleatoriamente de campos con dimensiones estándar donde se han instalado PFVCRs. La Tabla 2 muestra las dimensiones de los terrenos seleccionados, donde el tamaño de cada campo viene determinado por las dimensiones X e Y. Para comparar terrenos en la misma área geográfica, se ha elegido el valor de la latitud en 37º para todos los terrenos. Por otro lado, como ya se ha comentado, la Tabla 1 muestra los valores límites que pueden tomar todas las variables de nuestros individuos.

La Tabla 3 muestra los parámetros de los algoritmos. Se han elegido esos valores para hacer una comparación real de los algoritmos. De esta manera, el número de evaluaciones ha sido seleccionado empíricamente en 15000 para ambos algoritmos. Además, tanto SPEA2 como NSGA-II se han iniciado con el mismo número de individuos. El resto de los parámetros han sido seleccionados como especifican los respectivos autores de los algoritmos. Para cada terreno, cada algoritmo ha sido ejecuta-

do diez veces con diferentes semillas, de tal forma que, los resultados son la media y la desviación típica de las diez simulaciones.

Tabla 2. Terrenos elegidos para nuestro problema

Terreno	Dimensión X (m)	Dimensión Y (m)
1	200	100
2	210	170
3	280	180
4	300	150
5	300	200
6	335	290
7	380	300
8	400	300
9	400	375
10	400	400
11	420	350
12	450	390
13	500	400
14	500	430
15	500	500
16	600	525
17	700	580
18	750	610
19	825	680
20	900	900

Tabla 3. Especificación de parámetros para los dos algoritmos utilizados en la simulación

Algoritmo	Parámetros
SPEA2	Tamaño población = 100, Evaluaciones = 15000, Cruce binario simulado (SBX) ρ valor (SBX) = 5, Prob. cruce = 0.9, Prob. mutación.= 0.2
NSGA-II	Tamaño población = 100, Evaluaciones = 15000, Cruce binario simulado (SBX) ρ valor (SBX) = 5, Prob. cruce = 0.9, Prob. mutación.= 0.2

3.2 Medida del rendimiento de los MOEAs

El objetivo final de la optimización multi-objetivo es encontrar el frente óptimo de Pareto para un problema. Sin embargo, en la práctica, la optimización multi-objetivo sólo puede encontrar un subconjunto representativo. Este subconjunto normalmente contiene peores soluciones que las del frente óptimo de Pareto, o no contiene todas las soluciones posibles (que pueden ser innumerables problemas continuos). Es por ello que es necesario medir la calidad de estos frentes. Para este propósito se han diseñado muchos indicadores, también conocidos como métricas. La medida de la calidad de

estos frentes aproximados es un problema difícil y cualquier evaluación debería tener en cuenta la distancia entre el frente obtenido y el frente óptimo de Pareto [9].

Uno de los indicadores más populares para medir la calidad de las soluciones del MOEA, es el híper-volumen. Para nuestro problema, hemos calculado el hiper-volumen como el área encerrada bajo el frente obtenido.

En este trabajo, donde existe un problema de minimización, nos interesa que el híper-volumen sea tan bajo como sea posible, lo que significará que el frente obtenido se acercará a los ejes tanto como sea posible.

3.3 Resultados

Esta sección analiza los resultados obtenidos con los dos MOEAs utilizados, en términos de hiper-volumen. La Tabla 4 muestra la media y la desviación típica obtenidas en los veinte terrenos considerados. La primera columna, identifica el número del terreno, la segunda nos da la media de las diez ejecuciones realizadas con diferentes semillas; y la tercera columna, la desviación típica.

La Figura 2 nos muestra los frentes obtenidos por los MOEAs. El inverso de la potencia instalada en el terreno se encuentra sobre el eje de ordenadas, mientras que las pérdidas eléctricas por efecto Joule están sobre el eje de abscisas. Cada uno de los frentes está formado por 100 soluciones para nuestro problema.

Tabla 4. Resultados de hiper-volumen para SPEA2 y NSGA-II

	SPEA2		NSGA-II	
Terreno	Media	Des. Tip.	Media	Des. Tip
1	0.05360	0.00653	**0.04580**	0.00098
2	0.03011	0.00179	**0.02869**	0.00020
3	0.02727	0.00295	**0.02301**	0.00021
4	0.02848	0.00184	**0.02618**	0.00017
5	0.02285	0.00189	**0.02011**	0.00027
6	0.01788	0.00339	**0.01477**	0.00007
7	0.01492	0.00095	**0.01375**	0.00022
8	0.01548	0.00118	**0.01335**	0.00005
9	0.01313	0.00143	**0.01140**	0.00005
10	0.01300	0.00166	**0.01087**	0.00005
11	0.01209	0.00016	**0.01172**	0.00005
12	0.01133	0.00039	**0.01051**	0.00005
13	0.01176	0.00162	**0.00990**	0.00019
14	0.01300	0.00166	**0.00952**	0.00014
15	0.01028	0.00159	**0.00866**	0.00011
16	0.00786	0.00014	**0.00774**	0.00013
17	0.00780	0.00107	**0.00676**	0.00005
18	0.00650	0.00005	**0.00629**	0.00005
19	0.00636	0.00105	**0.00563**	0.00005
20	0.00528	0.00096	**0.00467**	0.00005

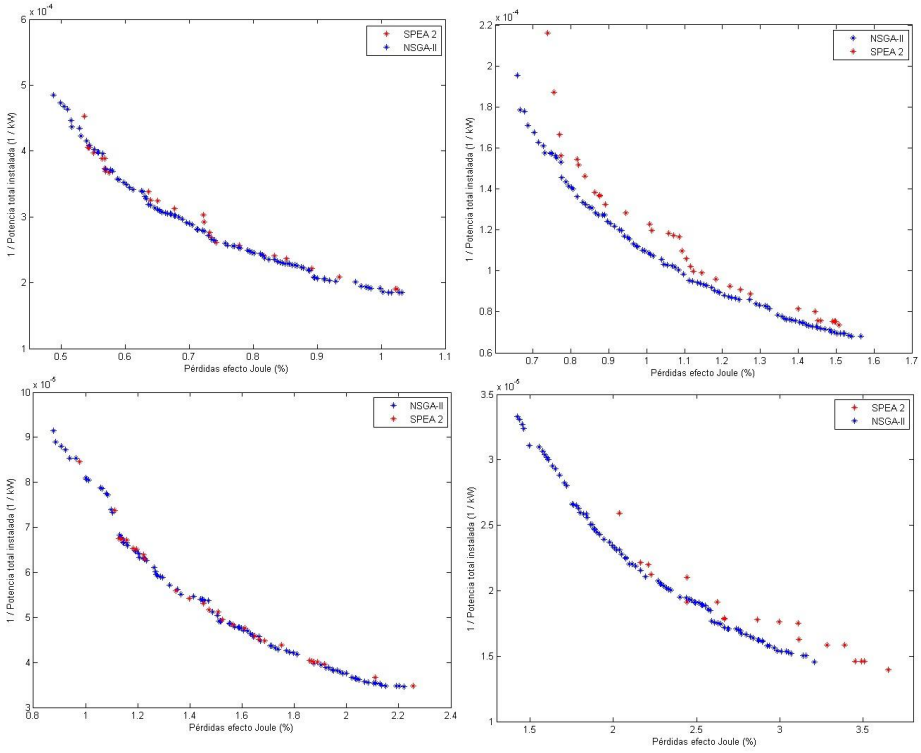


Fig. 2. Frentes obtenidos por SPEA2 y NSGA-II

Como se observa en la Tabla 4, NSGA-II supera a SPEA2 en todos los terrenos propuestos, ya que los híper-volúmenes son significativamente más bajos. La Figura 2 nos muestra los frentes obtenidos para los terrenos de dimensiones 300x200m, 400x400m, 600x525m y 900x900 m. En casi todos los casos, la figura muestra cómo los frentes obtenidos por NSGA-II superan a los frentes obtenidos por SPEA2. Como podemos observar, cuando el tamaño del terreno aumenta, el híper-volumen baja. La razón es que cuanto más grande es el terreno, la potencia instalada es mayor, por lo que, los puntos que representan al inverso de la potencia instalada en el eje vertical, se acercan al origen, lo cual hace que el área bajo ellos sea menor.

3.4 Comparación con técnicas mono-objetivo

Esta sección compara los resultados obtenidos por el MOEA que mejor se ha comportado en este problema (NSGA-II) con los resultados obtenidos por el mejor algoritmo evolutivo (DE) usado en [10]. Para ello, se han seleccionado terrenos del mismo tamaño y la misma latitud. La Tabla 5 nos muestra los valores de la máxima potencia instalada y de las pérdidas eléctricas por efecto Joule obtenidas por cada algoritmo evolutivo.

No es sencillo comparar un algoritmo mono-objetivo con un algoritmo multi-objetivo, ya que el resultado del primero es un valor numérico, mientras que el resultado del segundo es un frente compuesto por un vector de valores. La Tabla 5 nos muestra los valores extremos de los dos algoritmos y el valor central del frente obtenido por NSGA-II. La ventaja de tener un frente de valores es que nos hace posible movernos por el frente y elegir el punto que esté más en consonancia con nuestros requerimientos, lo cual nos facilita la elección de la mejor solución

Como se observa en la Tabla 5, las pérdidas eléctricas por efecto Joule obtenidas por el algoritmo DE son más bajas, y por lo tanto, mejores que las obtenidas por el MOEA. Sin embargo, los resultados del algoritmo DE vienen acompañados de un bajo rendimiento en términos de la potencia instalada en el terreno como podemos apreciar en la Tabla 5. Por el contrario, el MOEA hace un balance entre los dos objetivos. Si bien el valor de las pérdidas eléctricas es un poco mayor, tenemos la ventaja de que la potencia instalada en el terreno es mucho mayor, lo cual hace que el resultado global obtenido por el MOEA, sea significativamente mejor.

Tabla 5. Comparación entre los algoritmo DE y NSGA-II

	Potencia instalada (kW)				Pérdidas eléctricas (%)			
Terreno	DE	NSGA-II (max)	NSGA-II (min)	NSGA-II (med)	DE	NSGA-II (max)	NSGA-II (min)	NSGA-II (med)
1	610	1711	786	1183	0.3562	0.6960	0.4199	0.5196
2	897	2942	1156	1835	0.3847	0.8007	0.4069	0.5396
3	1633	4462	1761	2656	0.4369	0.9837	0.4662	0.6085
4	1282	3960	1543	2763	0.4513	0.9801	0.4676	0.6916
5	1816	5411	2064	3787	0.4631	1.0320	0.4878	0.7393
6	2307	8583	3226	6123	0.5211	1.2150	0.5462	0.8793
7	2478	10066	3973	6381	0.5456	1.3130	0.5978	0.8759
8	3114	10677	4196	6693	0.5878	1.3590	0.6395	0.8967
9	4927	13781	4824	8077	0.6238	1.5840	0.6270	0.9049
10	3945	14702	5599	8340	0.6269	1.5650	0.6664	0.9278
11	4170	13114	5485	8417	0.6020	1.4730	0.6831	0.9798
12	4093	15723	6964	10291	0.6501	1.6530	0.7710	1.0830
13	5594	18057	7668	11673	0.7132	1.7060	0.8019	1.1450
14	4242	19444	7917	12362	0.7265	1.8110	0.7975	1.1650
15	5760	22129	8841	13693	0.7327	1.8690	0.8009	1.1880
16	8375	28835	10928	17482	0.8148	2.2210	0.8774	1.3210
17	13535	35881	15094	22936	0.9669	2.3630	1.0430	1.5430
18	13098	42355	15451	27034	1.0020	2.6720	1.0890	1.6740
19	14766	51493	20964	34048	1.1120	2.8930	1.2070	1.8910
20	23058	68776	30030	47506	1.2709	3.2090	1.4260	2.1920

4 Conclusión

Este artículo ha presentado los resultados de un estudio de investigación en el cual se han utilizado dos conocidos algoritmos evolutivos multiobjetivo para encontrar la solución óptima para el diseño de PFVCRs con seguidores solares. El problema de optimización ha consistido en la selección de las variables que componen la planta fotovoltaica de tal forma que se minimicen las pérdidas eléctricas por efecto Joule en los conductores eléctricos que llevan la energía desde los módulos fotovoltaicos hasta los inversores y el transformador, y por otro lado se maximice la potencia instalada en el terreno.

Los dos MOEAs usados han sido SPEA2 y NSGA-II. Se han generado aleatoriamente veinte terrenos y se ha analizado la respuesta de cada algoritmo. La comparación de los dos MOEAs citados ha mostrado que para este tipo de problema, NSGA-II proporciona mejores resultados que SPEA2.

Finalmente, se han comparado los resultados de estos MOEAs con los resultados proporcionados por las técnicas usadas en [10]. Se ha encontrado que las soluciones obtenidas con los MOEAs citados son más flexibles que las proporcionadas por el algoritmo DE debido a que estas soluciones son frentes o curvas, que hacen posible elegir el punto de la curva que mejor se adapte a los requerimientos del proyectista.

Referencias

1. M. F. Demirbas, Decentralization of energy generation in turkey in the future, Energy Education Science and Technology Part A: Energy Science and Research 28 (1) (2011) 9–16.
2. G. Tsengenes, G. Adamidis, A multi-function grid connected pv system with three level npc inverter and voltage oriented control, Solar Energy 85 (11) (2011) 2595–2610.
3. A. Eiben, J. E. Smith, Introduction to Evolutionary Computing, Springer–Verlag, Berlin, 2003.
4. S. García, F. Herrera, Evolutionary under-sampling for classification with imbalanced data sets: Proposals and taxonomy, Evolutionary Computation 13 (3) (2009) 275–306.
5. A. Fernández, S. García, E. Luengo, J. Bernado-Mansilla, F. Herrera, Genetics-based machine learning for rule induction: State of the art, taxonomy and comparative study, IEEE Transactions on Evolutionary Computation 14 (6) (2010) 913–941.
6. K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjetive genetic algorithm: Nsga-II, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182–197
7. E. Zitzler, M. Laumanns, L. Thiele, Improving the strength pareto evolutionary algorithm, in: Proceedings of the evolutionary methods for design, 2000.
8. S. Ronald, Robust encodings in genetic algorithms: a survey of encoding issues, in: Proceedings of 1997 IEEE International Conference on Evolutionary Computation ICEC 97 Cat No 97TH8283, 1997, pp. 43–8.
9. E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, V. G. Fonseca, Performance assessment of multiobjective optimizers: An analysis and review, IEEE Transactions on Evolutionary Computation 7 (2) (2003) 117–132.
10. D. Gómez-Lorente, I. Triguero, C. Gil, A. Espín-Estrella, Evolutionary algorithms for the design of grid-connected pv systems, Expert Systems with Applications 39 (9) (2012) 8086–8094

Problemática γ de Ordenación con Algoritmos Evolutivos Multi-Objetivo y Programación Compromiso

M. Méndez[1], B. Galván[1], M. Frutos[2], A.L. Álamo[3]

[1] Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería (SIANI), Universidad de Las Palmas de Gran Canaria (ULPGC), España
mmendez@dis.ulpgc.es, bgalvan@step.es

[2] Departamento de Ingeniería, Universidad Nacional del Sur (UNS), Bahía Blanca, Argentina
mfrutos@uns.edu.ar

[3] Departamento de Matemáticas, Edificio Informática y Matemáticas, Universidad de Las Palmas de Gran Canaria (ULPGC), España
aalamo@dma.ulpgc.es

Abstract. La comunidad multi-objetivo ha mostrado recientemente un interés significativo por los algoritmos evolutivos multi-objetivo que introducen preferencias parciales del decisor. La intención, es dirigir la búsqueda de soluciones hacia la zona de predilección -Frente-óptimo Parcial Preferido de Pareto- para luego, elegir u ordenar el conjunto de soluciones encontrado en consonancia con las preferencias del decisor. Sin embargo este último aspecto no es trivial, pues aunque son numerosos los métodos existentes de Análisis de las Decisiones Multi-criterio, todos suponen Frentes-óptimos de Pareto completos. En este trabajo proponemos una idea sencilla de ayuda a un decisor para la ordenación (problemática γ) de manera automatizada del conjunto parcial preferido de soluciones encontrado, utilizando algoritmos evolutivos multi-objetivo que integran preferencias parciales y Programación Compromiso.

Keywords: Algoritmos Evolutivos Multi-objetivo, Programación Compromiso, Preferencias Parciales, Ayuda a la Decisión.

1 Introducción

Numerosos problemas de optimización de la vida real son de naturaleza difícil y multi-objetivos. En este contexto, los Algoritmos Evolutivos Multi-objetivo (AEMOs) son una herramienta muy popular en la comunidad multi-objetivo que ha demostrado [3] obtener excelentes resultados. Cuando se utilizan AEMOs, una consideración importante es en qué momento el decisor colabora con el algoritmo [8]. A posteriori, el decisor expresa sus preferencias al final, cuando el Frente-óptimo de Pareto (FOP) amplio y bien distribuido ha sido completamente determinado; la mayoría de los AEMOs son clasificados como a posteriori. Ejemplos clásicos que usan este enfoque son NSGAII [5] y SPEAII [18]. Los métodos

interactivos [13, 14] entre otros, alternan procesos de búsqueda con diálogos con el decisor. Por último, en un enfoque a priori [4, 7, 15] el decisor revela sus preferencias desde el principio del proceso de búsqueda. En esta categoría, algunos MOEAs resuelven un problema de optimización multi-objetivo (POM) el cual combina las diferentes funciones objetivo en una única función objetivo a optimizar. Más reciente, un cuarto enfoque -AEMOs bajo preferencias parciales- propuesto por Branke [2], considera la posibilidad de que un decisor no pueda especificar completamente sus preferencias pero sospecha que soluciones pueden ser preferidas. La idea es generar un pequeño conjunto de soluciones (Frente-óptimo Parcial Preferido de Pareto FOPPP) con las soluciones preferidas del decisor. Por ejemplo, suponiendo un POM de dos objetivos a minimizar, un decisor puede tener una idea de que soluciones podrían ser interesantes y la manifiesta por ejemplo mediante un punto g de referencia en el espacio de los objetivos (figura 1 izquierda); luego, un optimizador evolutivo multi-objetivo con preferencias parciales, véanse por ejemplo [1, 6, 10], obtiene el conjunto FOPPP que se visualiza en la figura 1 centro y sobre el cual, más tarde el decisor deberá hacer un análisis decisional.

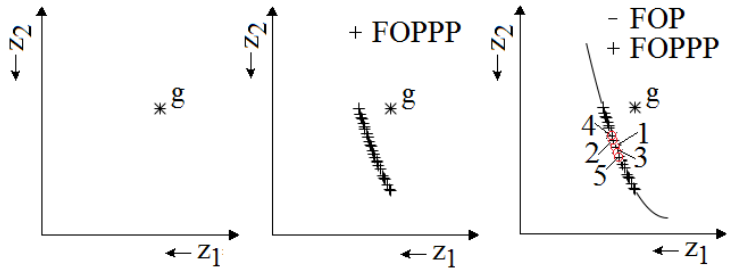


Fig. 1. Punto de referencia del decisor en espacio de la decisión (izq.). Frente-óptimo Parcial Preferido de Pareto (centro). Ordenación de soluciones (derecha).

Otra dirección de trabajo en optimización multi-objetivo es el Análisis de las Decisiones Multi-criterio (ADMC). Este punto de vista, se interesa fundamentalmente en ayudar a un decisor a elegir una solución (problemática α [11]) u ordenar (problemática γ [11]) un conjunto soluciones según varios objetivos en conflicto. En esta línea, numerosos tipos de métodos han sido desarrollados: MAUT [9], AHP [12], Compromise Programing (CP) [16], ELECTRE [11] etc. Por otra parte, y para concluir con el análisis decisional del ejemplo anterior, algún método de ADMC ordena las soluciónes según las preferencias del decisor, ver la figura 1 derecha donde se han representado las cinco mejores soluciones (etiquetadas del 1 al 5) y también el Frente-óptimo de Pareto del problema.

Se propone en este trabajo un procedimiento el cual, primero utiliza una metaheurística multi-objetivo para generar un conjunto parcial de soluciones (FOPPP). Insistiendo que en este trabajo sólo estamos interesados en este tipo

de Frentes de soluciones. Luego, a partir del conjunto FOPPP logrado antes, se obtiene según el modelo de la Programación Compromiso, la solución de mejores evaluaciones para cada una de las funciones objetivo. Esta solución es utilizada para, considerando las preferencias del decisor, ordenar (problemática γ) el conjunto de soluciones FOPPP, demostrandose que esta ordenación es la misma que la obtenida utilizando la solución ideal del problema.

El articulo esta organizado como sigue. En la siguiente sección, se explican brevemente algunos conceptos básicos. Una corta presentación del AEMO r-NSGAII y el método Programación Compromiso es realizada en las secciones 3 y 4. La sección 5 expone la metodología propuesta y su validación se hace en la sección 6. Por último, en la sección 7 se describen las conclusiones del trabajo.

2 Problema de Optimización Multi-objetivo

Un problema de optimización multi-objetivo se puede representar (en términos de minimización) de la siguiente forma:

$$\begin{cases} \textit{mín.} \quad F(x) = f_1(x), f_2(x), ..., f_m(x) \\ \quad s.a. \\ \quad\quad g_k(x) \leq 0 \qquad\qquad k \in (1, 2, ..., p) \\ \quad\quad h_l(x) = 0 \qquad\qquad l \in (1, 2, ..., q) \end{cases} \tag{1}$$

F(x) es el vector de objetivos a minimizar y m$\geq$ 2 representa el número de objetivos. Las restricciones $g_k(x) \leq 0$ y $h_l(x) = 0$ representan respectivamente p restricciones de desigualdad y q restricciones de igualdad. Los valores de x que satisfacen el conjunto de las (p+q) restricciones, definen el *espacio realizable S* (figura 2 izquierda). El vector x=$(x_1, x_2, ..., x_r) \in S$ es un vector solución de r variables de decisión.

En el marco de la Optimización Multi-objetivo, con frecuencia el decisor razona en términos de la evaluación de una solución según cada objetivo. Para representar el conjunto de soluciones realizables, en el espacio de los objetivos, es preciso obtener la imagen de cada solución realizable del espacio de decisión. Se obtiene así un conjunto $Z = f(S)$ (figura 2 centro) definido como:

$$\begin{cases} Z = \{z = (z_1, z_2, ..., z_m) \in R^m : \\ \quad z_1 = f_1(x), z_2 = f_2(x), ..., z_m = f_m(x), \ \forall x \in S\} \end{cases} \tag{2}$$

donde Z representa el conjunto de soluciones realizables en el espacio de los objetivos y z=$(z_1, z_2, ..., z_m)$ una solución realizable en el espacio de los objetivos.

Definición 1: Dominancia de Pareto. Una solución $z^t = (z_1^t, z_2^t, ..., z_m^t)$ domina una solución $z^u = (z_1^u, z_2^u, ..., z_m^u)$ si se verifican las siguientes condiciones:

1. $z_j^t \leq z_j^u \qquad \forall j \in (1, 2, ..., m)$
2. $\exists j \in (1, 2, ..., m) \ \ tal \ \ que \ \ z_j^t < z_j^u$

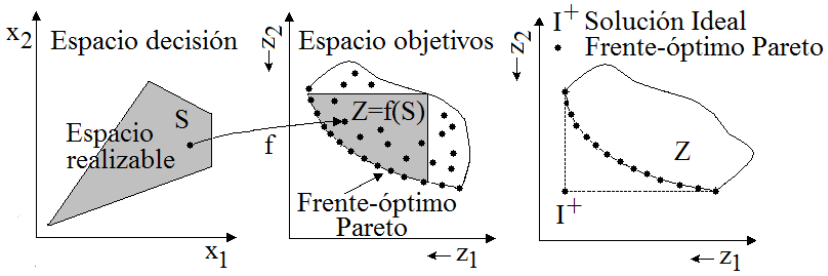


Fig. 2. Espacios realizable (izq.) y de los objetivos (centro). Solución ideal (derecha).

Definición 2: Solución Pareto óptima. Una solución $x^* \in S$ es Pareto-óptima si y sólo si no existe una solución $x \in S$ tal que $x \in S$ domine $x^* \in S$. Una solución Pareto-óptima también es conocida como solución eficaz.

Definición 3: Conjunto de Pareto. En general, en un POM no existe una única solución $x^* \in S$ Pareto-óptima, si más bien un conjunto de soluciones denominado conjunto de Pareto o conjunto eficiente.

Definición 4: Frente de Pareto-óptimo. El conjunto de soluciones en el espacio de los objetivos que son imágenes del conjunto de Pareto, se denomina frente de Pareto-óptimo (figura 2 centro).

Definición 5: Solución ideal I^+. La solución ideal de un POM es un vector solución $I^+ = (I_1^+, I_2^+, ..., I_m^+)$, construido con los mejores valores función objetivo de todas las soluciones del espacio de los objetivos (figura 2 derecha).

3 Non-r-Dominated Sorting Genetic Algorithm

El algoritmo evolutivo Non-r-Dominated Sorting Genetic Algorithm (r-NSGAII) propuesto en [1] modifica el algoritmo NSGAII de la siguiente manera: primero, la ordenación por dominancia de Pareto es sustituida por la relación r-dominance y segundo, para mantener la diversidad de la población el algoritmo realiza un ajuste durante la evolución del algoritmo del umbral δ que utiliza la r-dominance. La idea de la r-dominance es preferir aquellas soluciones más próximas al punto de referencia preservando la dominancia de Pareto. Más concretamente, dada una población P de soluciones y un punto de referencia g, una solución x r-domina una solución y, si se verifican las siguientes condiciones:

1. x Pareto-domina y
2. x,y son Pareto-equivalentes y $D(x, y, g) < -\delta$ donde:
   $\delta \in [0, 1]$
   $D(x,y,g)=\frac{d(x,g)-d(y,g)}{d_{\max}-d_{\min}}$

$d_{max} = Máx_{z \in P} d(z, g)$
$d_{min} = Mín_{z \in P} d(z, g)$
$d(x,g) = \sqrt{\sum_{j=1}^{m} w_j (\frac{f_j(x) - f_j(g)}{f_j^{max} - f_j^{min}})^2}$
f_j^{max} máx valor del objetivo j-ésimo
f_j^{min} mín valor del objetivo j-ésimo
w_j peso asociado con el j-ésimo objetivo
$w_j \in]0, 1[$, $\sum_{j=1}^{m} w_j = 1$

Los autores demuestran que ajustando el valor del parámetro δ, resulta muy fácil controlar la extensión del FOPPP.

4 Programación Compromiso

La Programación Compromiso fue propuesta en el año 1973 por Zeleny [16]. El método primero obtiene todas las soluciones no-dominadas del problema y luego define la solución compromiso como aquella que está más cerca de la solución ideal según la ecuación (3) (suponiendo minimización).

$$Min.\ d_p^{I^+}(z) = \left[\sum_{j=1}^{m} w_j^p \left| z_j - I^+ \right|^p \right]^{1/p} \tag{3}$$

En la ecuación (3), $d_p^{I^+}(z)$ es una función de distancias para cada valor de p entre una solución z y el vector solución ideal I^+, w_j es la ponderación asociada a cada objetivo j. Por último p es un parámetro o métrica comprendida entre los valores [1, ∞], si p=1 la ecuación (3) adopta la forma (4).

$$Min.\ d_1^{I^+}(z) = \sum_{j=1}^{m} w_j \left| z_j - I^+ \right| \tag{4}$$

5 Método Propuesto

Suponiendo el POM modelado según (1), el método propuesto sigue los siguientes pasos: primero, el decisor indica los parámetros de la metaheurística multi-objetivo utilizada (en este trabajo hemos usado el AEMO r-NSGAII [1] aunque otra metaheurística puede ser empleada) y sus preferencias (en este trabajo se utilizó un punto de referencia g (figura 1 izquierda) aunque otra forma de indicar las preferencias [2] es posible); segundo, el AEMO es ejecutado para obtener un conjunto FOPPP discretizado (figura 1 centro); tercero y ya en el análisis de la decisión, de nuevo el decisor señala sus preferencias (pesos w_j según ecuación (4)) para a continuación y cuarto, obtener la solución I^+_{FOPPP} (figura 3). Finalmente, en el quinto y último paso las soluciones del FOPPP son ordenadas utilizando la métrica 1 en el método de la Programación Compromiso.

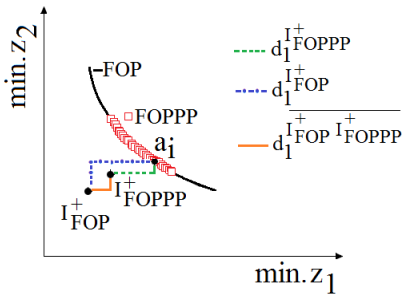


Fig. 3. Distancias d_1.

Proposición. $\forall$ solución $a_i \in FOPPP$, utlizando la métrica d_1, el método de la minimización de la distancia a la solución I^+_{FOPPP} y el método de la minimización de la distancia a la solución ideal I^+_{FOP} obtienen la misma ordenación.

Demostración. Suponiendo conocida la solución ideal del FOP del problema se tiene que $\forall a_i \in FOPPP$: $\text{Min}_i\ d_1^{I^+_{FOP}}(a_i) = d_1^{\overline{I^+_{FOP} I^+_{FOPPP}}} + Min_i\ d_1^{I^+_{FOPPP}}(a_i)$. De otra parte la distancia $d_1^{\overline{I^+_{FOP} I^+_{FOPP}}}$ es una constante C por tratarse de la distancia entre la solución ideal I^+_{FOP} y la solución I^+_{FOPPP}. Por tanto, $\forall a_i \in FOPPP$: $\text{Min}_i\ d_1^{I^+_{FOP}}(a_i) = C + Min_i\ d_1^{I^+_{FOPPP}}(a_i)$ y por consiguiente, $\forall a_i \in FOPPP$ el método de la minimización de la distancia a la solución I^+_{FOPPP} y el método de la minimización de la distancia a la solución ideal I^+_{FOP} obtienen la misma ordenación.

6 Ejemplo Numérico

El método propuesto fue aplicado al problema test ZDT1 [17], el problema tiene dos funciones a minimizar (5) y 30 variables de decisión. Los parámetros del algoritmo fueron: código binario, codificación variables de decisión 30 bits, talla población=50, cruce uniforme, prob. cruce=0.8, prob. mutación=1/n (n es el ancho del cromosoma 30*30), número máximo de generaciones=250, $w_1 = w_2$=0.5 y δ=0.1. En cuanto a los valores de las preferencias del decisor adoptados fueron: punto de referencia de coordenadas (0.3, 0.6), $w_1 = w_2 = 0.5$ y luego $w_1 = 0.9, w_2 = 0.1$ en la ecuación de distancias (4); punto de referencia de coordenadas (0.45, 0.25), $w_1 = w_2 = 0.5$ en la ecuación de distancias (4).

$$\begin{cases} Min. \quad f_1(\mathbf{x}) = x_1 \\ Min. \quad f_2(\mathbf{x}) = g(x)[1 - \sqrt{x_1/g(x)}] \\ Min. \quad g(\mathbf{x}) = 1 + 9(\sum_{i=2}^{n} x_i)/(n-1) \\ 0 \leq \ x_i \leq 1 \quad i = 1, 2, ...n \quad n = 30 \end{cases} \tag{5}$$

En la figura 4 izquierda se ilustran las preferencias del decisor reflejadas según el punto g de coordenadas (0.3, 0.6) y el conjunto FOPPP logrado por r-NSGAII. La figura 4 derecha es un zoom de la figura 4 izquierda y recoge la ordenación de las seis primeras soluciones cuando las preferencias del decisor en el modelo de la Programación Compromiso (métrica 1) son $w_1 = w_2 = 0.5$. Remarcamos que la ordenación obtenida de soluciones es con respecto a la solución ideal I^+_{FOP} del problema (incluso siendo desconocida).

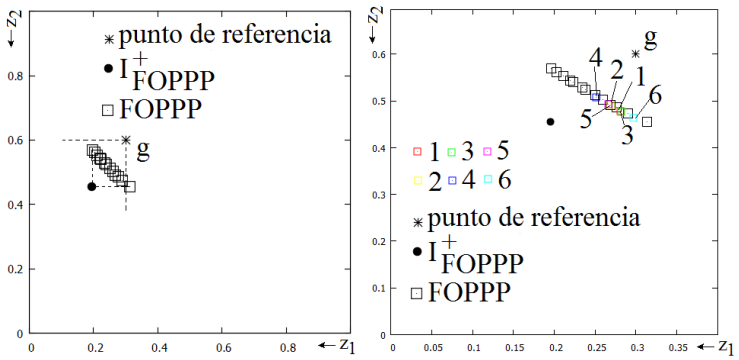


Fig. 4. FOPPP logrado por r-NSGAII con g=(0.3, 0.6) (izquierda). Ordenación de soluciones de FOPPP con $w_1 = w_2 = 0.5$ (derecha).

La figura 5 es un duplicado de la figura 4, pero en la figura 5 derecha las preferencias del decisor en el modelo de la Programación Compromiso (métrica 1) se modificaron a los valores $w_1 = 0.9$ y $w_2 = 0.1$. Observese el desplazamiento lógico de las soluciones preferidas hacia esquina izquierda superior pues el objetivo Z_1 es ahora prioritario.

Por último en la figura 6 izquierda, las preferencias del decisor fueron cambiadas por el punto g=(0.45, 0.25); la figura 6 derecha es un zoom de la figura 6 izquierda y muestra la ordenación de las seis primeras soluciones cuando las preferencias del decisor en el modelo de la Programación Compromiso (métrica 1) son $w_1 = w_2 = 0.5$.

7 Conclusiones

Se propone en este artículo un método que utiliza Algoritmos Evolutivos Multi-objetivo (AEMOs) que introducen preferencias del decisor y Programación Compromiso. Los AEMOs que incluyen preferencias han demostrado obtener buenos resultados en problemas con elevado número de objetivos [6] y la Programación Compromiso es bien conocida y ampliamente empleada en el Análisis de las Decisiones Multi-criterio. La idea que se presenta en este trabajo es encontrar un

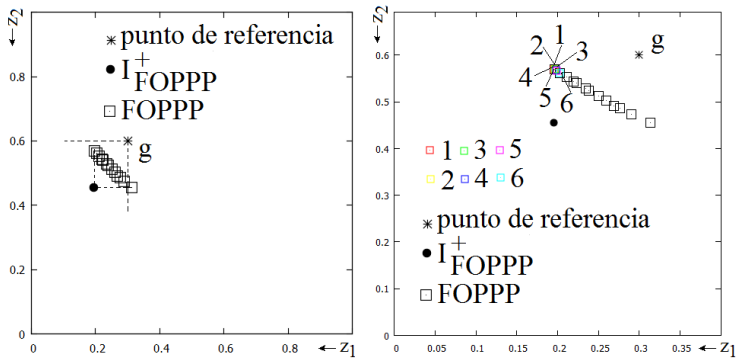


Fig. 5. FOPPP logrado por r-NSGAII con g=(0.3, 0.6) (izquierda). Ordenación de soluciones de FOPPP con $w_1 = 0.9, w_2 = 0.1$ (derecha).

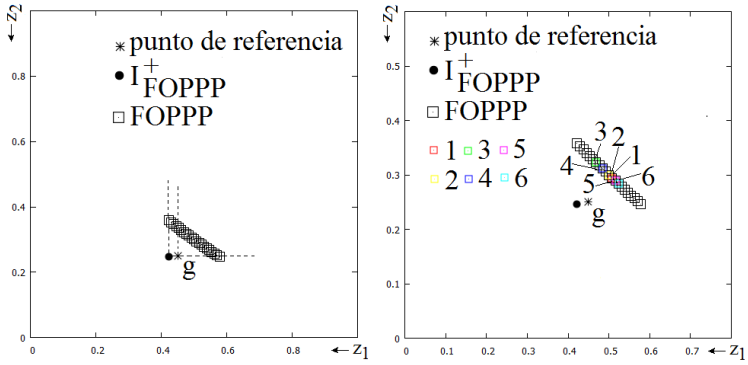


Fig. 6. FOPPP logrado por r-NSGAII con g=(0.45, 0.25) (izquierda). Ordenación de soluciones de FOPPP con $w_1 = w_2 = 0.5$ (derecha).

Frente-óptimo Parcial Preferido de Pareto (FOPPP) acorde con las preferencias del decisor para luego ordenar (problemática γ de Roy [11]) las soluciones encontradas utilizando la métrica 1 según el modelo de la Programación Compromiso. Se demuestra que la ordenación lograda usando la solución del FOFFF que tiene las mejores evaluaciones para cada una de las funciones objetivo, es la misma que la ordenación de soluciones obtenida utilizando la solución ideal del Frente-optimo de Pareto (FOP).

Agradecimientos

Agradecemos el soporte económico del Departamento de Informática y Sistemas de la Universidad de Las Palmas de Gran Canaria.

References

1. L. Ben Said, S. Bechikh, K. Ghedira (2010) The r-Dominance: A New Dominance Relation for Interactive Evolutionary Multicriteria Decision Making. IEEE Transactions on Evolutionary Computation, 14(5), 801-818.
2. J. Branke (2008) Consideration of Partial User Preferences in Evolutionary Multiobjective Optimization. In Branke et al. (Eds.) : Multiobjective Optimization, Lecture Notes in Computer Science, 5252, 157-178.
3. C.A. Coello, G.B. Lamont, D.A. Van Veldhuizen (2007) Evolutionary Algorithms for Solving Multi-Objective Problems. Springer.
4. C.A. Coello (1998) Using the Min-Max Method to Solve Multiobjective Optimization Problems with Genetic Algorithms. In Progress in Artificial Intelligence, IBERAMIA 98, LNCS, Springer-Verlag, 1484, 303-313.
5. K. Deb, A. Pratap, S. Agrawal, T. Meyarivan (2002) A Fast and Elitist Multiobjective Genetic Algorithm : NSGA-II. IEEE Transactions on Evolutionary Computation, 6(2),182-197.
6. K. Deb, J. Sundar, R. Bhaskara, S. Chaudhuri (2006) Reference Point Based Multi-Objective Optimization Using Evolutionary Algorithms. International Journal of Computational Intelligence Research, 2(3), 273-286.
7. P. Hajela, C.-Y. Lin (1992) Genetic search strategies in multicriterion optimal design. Structural optimization, Springer-Verlag, 4(2), 99-107.
8. J. Horn (1997) Multicriterion Decision Making and Evolutionary Computation. In Thomas Bck et al. (Eds.) : Handbook of Evolutionary Computation, 1: F1.9:1-F1.9:15.
9. R.L. Keeney, H. Raiffa (1976) Decisions with multiple objectives: preferences and value tradeoffs. John Wiley & Sons.
10. J. Molina, L.V. Santana, A.G. Hernndez-Daz, C.A. Coello Coello, R. Caballero (2009) g-dominance: Reference point based dominance for multiobjective metaheuristics. European Journal of Operational Research, 19(2), 685-692.
11. B. Roy (1996) Multicriteria Methodology for Decision Aiding. Kluwer Academic Publishers.
12. T.L. Saaty (1980) The Analytic Hierarchy Process. McGraw-Hill.
13. K. Sindhya, A.B. Ruiz, K. Miettinen (2011) A Preference Based Interactive Evolutionary Algorithm for Multi-objective Optimization: PIE. Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science, 6576, 212-225.
14. A. Sinha, P. Korhonen, J. Wallenius, K. Deb (2010) An Interactive Evolutionary Multi-objective Optimization Method Based on Polyhedral Cones. Learning and Intelligent Optimization, Lecture Notes in Computer Science, 6073, 318-332.
15. D.A. Van Veldhuizen, B.S. Sandlin, R.E. Marmelstein, G.B. Lamont, A.J. Terzuoli (1998) Finding improved wire-antenna geometries with genetic algorithms. In the 1998 IEEE International Conference on Evolutionary Computation Proceedings, 102-107.
16. M. Zeleny (19739 Compromise Programming. In: Multiple Criteria Decision Making, J.L. Cochrane and M. Zeleny (Eds.) : University of South Carolina Press, Columbia, 262-301.
17. E. Zitzler, K. Deb, L.Thiele (2000) Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. IEEE Transactions on Automatic Control, 8(1), 59-60.
18. E. Zitzler, M. Laumanns, L. Thiele (2001) SPEA2: Improving the Strengh Pareto Evolutionary Algoritm. In Proc. EUROGEN 2001, (K. Giannakoglou et al. (Eds.), Athens, Greece.

Un nuevo Algoritmo Evolutivo de Optimización Multiobjetivo basado en Preferencias: WASF-GA

Rubén Saborido, Ana Belén Ruiz y Mariano Luque

Departamento de Economía Aplicada (Matemáticas)
Facultad de Ciencias Económicas y Empresariales
Universidad de Málaga
.
rsain@uma.es, abruiz@uma.es, mluque@uma.es

Resumen Para resolver problemas de optimización multiobjetivo, los *algoritmos evolutivos basados en preferencias* incorporan información preferencial en el proceso evolutivo, con el objetivo de encontrar un conjunto de soluciones Pareto óptimas acordes a las preferencias del decisor (DM). En este trabajo, se propone un nuevo algoritmo evolutivo basado en preferencias que aproxima la región de la frontera Pareto óptima que mejor se ajusta a los valores de aspiración que el decisor desea alcanzar en cada objetivo. Estos valores constituyen las componentes de lo que se conoce como *punto de referencia*. El algoritmo hace uso de la conocida función escalarizada de logro de Wierzbicki para clasificar los individuos de cada generación en diferentes fronteras. Dicha clasificación se realiza en base a los valores que la función de logro alcanza para el punto de referencia proporcionado, considerando un conjunto de vectores de pesos. Los experimentos realizados sobre conocidos problemas test muestran unos resultados prometedores en cuanto a la calidad de las soluciones obtenidas, en comparación a otros algoritmos existentes basados en puntos de referencia.

Keywords Optimización multiobjetivo, algoritmo evolutivo basado en preferencias, punto de referencia, función escalarizada de logro.

1. Introducción

En la vida real, muchos problemas consisten en optimizar conjuntamente varios criterios, que normalmente se encuentran en conflicto. En general, un *problema de optimización multiobjetivo* puede formularse de la siguiente forma:

$$\begin{array}{rl} \text{minimizar} & \{f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})\} \\ \text{sujeto a} & \mathbf{x} \in S, \end{array} \tag{1}$$

siendo $f_i : S \to \mathbf{R}$, para cada $i = 1, \ldots, k$, las k ($k \geq 2$) funciones objetivo a minimizar y S el conjunto factible de vectores de decisión $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$.

La imagen del conjunto factible $Z = \mathbf{f}(S)$ está compuesta por vectores criterio $\mathbf{z} = \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x}))^T$, para algún $\mathbf{x} \in S$.

Por lo general, el conflicto existente entre los objetivos imposibilita encontrar una solución en la que todos los objetivos alcancen sus óptimos individuales simultáneamente. Sin embargo, hay un conjunto de soluciones en las que no se puede mejorar ningún objetivo sin empeorar algún otro: las denominadas *soluciones eficientes*, *no dominadas* o *Pareto óptimas*. Dados dos vectores $\mathbf{z}, \mathbf{z}' \in \mathbf{R}^k$, decimos que $\mathbf{z}$ *domina a* $\mathbf{z}'$ si y solo si $z_i \leq z'_i$ para todo $i = 1, \ldots, k$, y existe algún $j \in \{1, \ldots, k\}$ tal que $z_j < z'_j$. Se considera que $\mathbf{x} \in S$ es una solución *Pareto óptima* del problema (1) si no existe ningún $\mathbf{x}' \in S$ tal que $\mathbf{f}(\mathbf{x}')$ domine a $\mathbf{f}(\mathbf{x})$. El conjunto de todas las soluciones Pareto óptimas es conocido como el *conjunto eficiente* o *conjunto óptimo de Pareto*, denotado por E. Su imagen en el espacio de objetivos es denominado *frontera eficiente* o *frente óptimo de Pareto*, denotado por $\mathbf{f}(E)$.

Desde el punto de vista matemático todas las soluciones eficientes son equivalentes, por lo que es necesaria cierta información preferencial de algún decisor (DM) para encontrar la solución final. Una forma natural de expresar preferencias consiste en indicar valores deseables para cada función objetivo que constituyen las componentes del denominado punto de referencia. En concreto, un *punto de referencia* viene dado por $\mathbf{q} = (q_1, \ldots, q_k)^T$, donde q_i es un valor de aspiración para el objetivo f_i proporcionado por el DM, para cada $i = 1, \ldots, k$. Se dice que un punto de referencia es *alcanzable* si los correspondientes valores de aspiración pueden ser simultáneamente alcanzados o mejorados por alguna solución factible; en caso contrario, se dice que el punto de referencia es *inalcanzable*.

A lo largo de los años, se han desarrollado una gran variedad de métodos para resolver problemas de optimización multicriterio. Entre ellos, pueden destacarse la *Toma de Decisiones Multicriterio (Multiple Criteria Decision Making, MCDM)* [13, 17] y la *Optimización Multiobjetivo Evolutiva (Evolutionary Multiobjective Optimization, EMO)* [3, 5, 10]. El principal objetivo de MCDM es ayudar al DM a elegir la solución más adecuada considerando sus preferencias. Por el contrario, EMO pretende obtener un conjunto bien distribuido de soluciones que aproximen el frente óptimo de Pareto, haciendo uso de algoritmos basados en población que emulan el proceso de evolución natural de las especies.

Aunque los métodos MCDM muestran al DM únicamente soluciones que se ajustan a sus preferencias, en algunos casos no son capaces de resolver problemas con variables de decisión enteras o binarias, o con funciones objetivo no convexas, discontinuas o no diferenciables. Por el contrario, aunque los algoritmos EMO pueden tratar problemas de complicada naturaleza, no consideran información preferencial alguna en el proceso de resolución. Según esto, el DM tiene que analizar un número elevado de soluciones para encontrar la solución final, lo que supone un gran esfuerzo cognitivo. Además, dado que las preferencias del DM acotan al conjunto de soluciones eficientes, no es necesaria la aproximación completa del frente óptimo de Pareto para encontrar la solución final al problema.

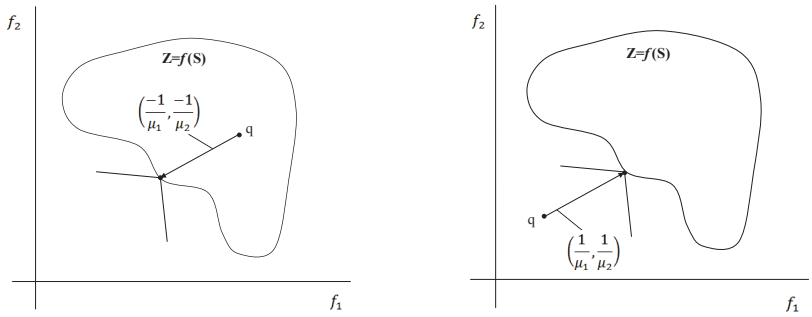
Durante los últimos años, ha surgido una nueva vertiente de algoritmos evolutivos basados en preferencias [1, 4, 14]. Éstos, se basan en la evolución de una población de soluciones considerando las preferencias del DM en el proceso evolutivo, con el fin de guiar la búsqueda de nuevas soluciones hacia la zona interés del frente óptimo de Pareto. De esta forma, se reduce la carga cognitiva del DM ya que se reduce el número de soluciones que tiene que revisar, las cuales, a su vez, se ajustan a sus preferencias.

Entre los algoritmos evolutivos existentes basados en preferencias se encuentra el método *Reference-Point-Based NSGA-II* (R-NSGA-II) [8]. A partir de uno o varios puntos de referencia definidos por el DM, R-NSGA-II reemplaza la crowding distance del NSGA-II [7] por una distancia preferencial, basada en la distancia euclídea de cada solución a los puntos de referencia. En [15], se propone un nuevo criterio de dominancia, denominado *r-dominancia*, que distingue soluciones no dominadas entre sí en función de sus distancias euclídeas a un punto de referencia proporcionado por el DM, dando mayor prioridad a aquéllas más cercanas a dicho punto. Los autores reemplazan el criterio de dominancia usual en el NSGA-II por la r-dominancia, denominando al algoritmo resultante r-NSGA-II. Otros algoritmos evolutivos basados en preferencias son *Guided Multi-Objective Evolutionary Algorithm* (G-MOEA) [2], *Reference Direction Based NSGA-II* (RD-NSGA-II) [6], *Preference Based Evolutionary Algorithm* (PBEA) [18], *Preference-based Interactive Evolutionary Algorithm* (PIE) [16] o *interactive MOEA/D* (iMOEA/D) [12].

En este artículo, se propone un nuevo algoritmo evolutivo basado en preferencias denominado WASF-GA (*Weighting Achievement Scalarizing Function Genetic Algorithm*). A partir de un punto de referencia determinado por el DM, WASF-GA aproxima la región de la frontera eficiente proyectable desde el punto de referencia. Para ello, hace uso de la función escalarizada de logro de Wierzbicki [19], clasificando por fronteras los individuos de cada generación. Una función escalarizada de logro es una función real que combina las funciones objetivo originales con cierta información preferencial, expresada, por ejemplo, mediante un punto de referencia. Al minimizar una función de logro sobre S, se obtiene la solución Pareto óptima que 'mejor' se ajusta al punto de referencia empleado, proyectándolo en una dirección determinada por un vector de pesos.

En cada generación de WASF-GA, la clasificación de individuos en fronteras se realiza atendiendo a los valores de la función de logro propuesta en [19] para cada solución, teniendo en cuenta el punto de referencia proporcionado por el DM y un conjunto de vectores de pesos. Para obtener una buena distribución de soluciones, los vectores de pesos empleados deben ser equidistribuídos en el espacio de pesos $(0, 1)^k$. Esto, junto con el uso de la función de logro, permite que la población final de soluciones de WASF-GA se ajuste al punto de referencia dado.

En este trabajo se comparan WASF-GA, R-NSGA-II y r-NSGA-II, haciendo uso de una métrica basada en hipervolumen [11, 21] adaptada a la región de interés proyectable desde el punto de referencia. Para ello se han usado puntos de referencia alcanzables e inalcanzables, generados de forma aleatoria.



(a) Punto de referencia alcanzable (b) Punto de referencia inalcanzable

Figura 1. Idea gráfica de la minimización de (2) para un problema con dos objetivos.

El artículo se organiza como sigue. En la sección 2, se describe WASF-GA detalladamente. Algunos experimentos sobre diferentes problemas test se muestran en la sección 3 y, finalmente, se concluye con la sección 4.

2. El Algoritmo WASF-GA

WASF-GA aproxima la región del frente óptimo de Pareto proyectable desde un punto de referencia $\mathbf{q}$, proporcionado por el DM. Se basa en el uso de la función escalarizada de logro propuesta en [19], dada por la expresión siguiente:

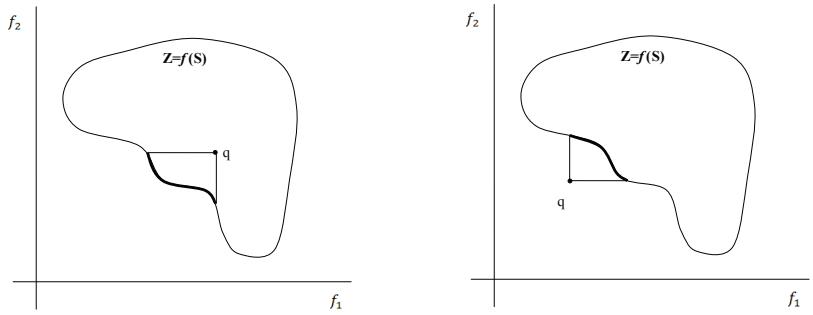
$$s(\mathbf{q}, \mathbf{f}(\mathbf{x}), \mu) = \max_{i=1,\ldots,k} \{ \mu_i (f_i(\mathbf{x}) - q_i) \} + \rho \sum_{i=1}^{k} (f_i(\mathbf{x}) - q_i), \tag{2}$$

donde $\mu = (\mu_1, \ldots, \mu_k)$ es un vector de pesos, con $\mu_i > 0$ para todo $i = 1, \ldots, k$ y $\rho > 0$ un valor real suficientemente pequeño (normalmente 0,001). La solución factible que minimiza (2) es una solución Pareto óptima de (1)[13].

En la práctica, si $\mathbf{q}$ es inalcanzable, minimizar (2) equivale a la distancia minmax. Si $\mathbf{q}$ es alcanzable, la solución que minimiza (2) mejora los valores de aspiración de $\mathbf{q}$ tanto como sea posible. En caso de que los pesos considerados sean estrictamente positivos, minimizar (2) significa proyectar $\mathbf{q}$ sobre el conjunto eficiente en la dirección determinada por la inversa de los vectores de pesos. La Figura 1 da una idea de lo que significa dicha minimización para un problema como (1) con dos funciones objetivos, tanto para puntos de referencia alcanzables como inalcanzables.

WASF-GA hace uso del concepto de región proyectable desde un punto de referencia. Consideramos que la *región proyectable de la frontera eficiente desde un punto de referencia* $\mathbf{q}$ es el conjunto de soluciones Pareto óptimas que se obtienen al minimizar (2) sobre S, variando el vector de pesos μ en todo $(0, 1)^k$. De esta forma, si $\mathbf{q}$ es alcanzable, la región proyectable estará formada por el conjunto de soluciones eficientes que dominan $\mathbf{q}$ (esto es, los $\mathbf{x} \in E$ tales que

$f_i(\mathbf{x}) \leq q_i$, para todo $i = 1, \ldots, k$). En caso contrario, si $\mathbf{q}$ es inalcanzable, la región proyectable estará formada por las soluciones eficientes más próximas a $\mathbf{q}$ respecto a la distancia minmax. Esto significa que, en el caso inalcanzable, la región proyectable podrá estar formada por los $\mathbf{x} \in E$ tales que $f_i(\mathbf{x}) \geq q_i$, para todo $i = 1, \ldots, k$, aunque esto no tiene por qué ser así en todos los casos. En la Figura 2, se muestran dos gráficas correspondientes a un problema con dos objetivos como el dado en (1), en las que se ha resaltado la región proyectable, tanto en el caso alcanzable como en el inalcanzable.



(a) Punto de referencia alcanzable (b) Punto de referencia inalcanzable

Figura 2. Región proyectable para un problema con dos objetivos.

Cualquier solución Pareto óptima de (1) puede obtenerse minimizando (2) variando el punto de referencia y/o el vector de pesos [13]. Según esto, cualquier solución eficiente dentro de la región proyectable del punto de referencia $\mathbf{q}$ se puede obtener minimizando (2) sobre S variando el vector de pesos en todo $(0, 1)^k$. Basándose en este hecho, WASF-GA aproxima la región proyectable mediante la proyección de $\mathbf{q}$ en el frente óptimo de Pareto, considerando una muestra equidistribuída de vectores de pesos.

En lo siguiente, N es el tamaño de la población, $N_\mu > 0$ el número de vectores de pesos considerados, h la generación actual y P_h la población de individuos en la generación h. A continuación, se describe el funcionamiento de WASF-GA:

- **Paso 1. Inicialización.** Definir una muestra equidistribuida de N_μ vectores de pesos $W = \{\mu^j \in (0, 1)^k : j = 1, \ldots, N_\mu\}$, inicializar $h = 0$ y obtener P_0 mediante la generación aleatoria de N individuos.

- **Paso 2. Recombinación y mutación.** Aplicar los operadores de recombinación y mutación a P_h (población de padres) para obtener una población de N hijos. Combinar las poblaciones de padres e hijos en una nueva población P de tamaño $2N$.

- **Paso 3. Clasificación por fronteras.** Para cada solución factible $\mathbf{x} \in P$, calcular $s(\mathbf{q}, \mathbf{f}(\mathbf{x}), \mu^j)$ para cada $j = 1, \ldots, N_\mu$. La primera frontera, F_1^h, la componen las soluciones factibles de P que alcanzan el menor valor de $s(\mathbf{q}, \mathbf{f}(\mathbf{x}), \mu^j)$ para cada $j = 1, \ldots, N_\mu$ (se permite repetir soluciones si fuera necesario). Se eliminan temporalmente de P los individuos de F_1^h. La segunda frontera, F_2^h, la forman las soluciones factibles de P que alcanzan el (segundo) menor valor de $s(\mathbf{q}, \mathbf{f}(\mathbf{x}), \mu^j)$ para cada $j = 1, \ldots, N_\mu$. Nuevamente, se eliminan temporalmente de P los individuos de F_2^h. Este proceso continúa hasta que todas las soluciones factibles de P han sido clasificadas. Posteriormente, se clasifican las soluciones infactibles de P atendiendo a la violación acumulada de las restricciones.

- **Paso 4. Selección.** Actualizar $h = h + 1$, $P_h = \emptyset$ y sea $n = 1$. Si el $\#(P_h \cup F_n^{h-1}) \leq N$, entonces $P_h = P_h \cup F_n^{h-1}$. Actualizar $n = n + 1$ y repetir este proceso hasta que $\#(P_h \cup F_n^{h-1}) > N$, en cuyo caso se incluirán en P^h los individuos de F_n^{h-1} que alcancen los menores valores de la función de logro hasta tener N individuos en P_h.

- **Paso 5. Criterio de parada.** Si se verifica el criterio de parada, los individuos de F_1^{h-1} forman la salida final de WASF-GA. En caso contrario, ir al paso 2.

Con este procedimiento, se da más importancia a aquellas soluciones con menor valor de la función de logro para cada uno de los N_μ vectores de pesos.

Si todas las soluciones de una generación son factibles, cada frontera se compone de N_μ soluciones, a excepción de la última, que puede contener menos. Por ello, se considera que $N_\mu \leq N$ si se desea tener, al menos, una solución correspondiente a cada vector de pesos en la siguiente generación. Asimismo, como las soluciones que se muestran al DM son las de la primera frontera de la última generación, el DM tiene que revisar exactamente N_μ soluciones para determinar la solución final. Por defecto se considera que $N_\mu = \frac{N}{2}$, aunque el DM puede determinar el valor de N_μ indicando el número de soluciones que desea obtener.

En cada generación, la clasificación por fronteras realizada por WASF-GA define un orden parcial estricto entre las soluciones, que además es completo con la Pareto dominancia[1], debido a que, si $\mathbf{x}_1$ domina a $\mathbf{x}_2$, se verifica que $s(\mathbf{q}, \mathbf{f}(\mathbf{x}_1), \mu) < s(\mathbf{q}, \mathbf{f}(\mathbf{x}_2), \mu)$, para cualquier vector de pesos μ estrictamente positivo [13]. De esta forma, se puede asegurar que si una solución domina a otra, la primera pertenecerá a una frontera de orden inferior a la de la segunda.

Como se indicó en el Paso 3, una solución puede aparecer varias veces en una misma frontera si dicha solución minimiza simultáneamente (2) para varios vectores de pesos. Esto asegura que dentro de una frontera no existan soluciones dominadas entre sí.

[1] Si $\rhd$ es una relación binaria donde $\mathbf{x} \rhd \mathbf{y}$ implica que $\mathbf{x}$ es preferido a $\mathbf{y}$, se dice que $\rhd$ es *completo con la Pareto dominancia* si y solo si $\mathbf{x}$ Pareto domina a $\mathbf{y}$ implica que $\mathbf{x} \rhd \mathbf{y}$.

3. Experimentos

Para evaluar el funcionamiento de WASF-GA se han realizado una serie de experimentos sobre varios problemas test y se han comparado los resultados obtenidos con los generados por otros algoritmos evolutivos basados en punto de referencia, r-NSGA-II [15] y R-NSGA-II [8]. Los tres métodos han sido implementados en lenguaje C++ usando la Standard Template Library (STL).

Los experimentos se han realizado sobre problemas de las familias ZDT [20] (con dos funciones objetivo y 30 variables) y DTLZ [9] (con tres objetivos y 22 variables).

El número de generaciones se ha fijado en 300, y el tamaño de población en 120 individuos para los problemas de 2 objetivos y en 200 individuos para los problemas con 3 objetivos. Los índices de distribución de los operadores de recombinación SBX [5] y de mutación polinomial [5] se han definido como 10 y 20, respectivamente. La probabilidad de recombinación se ha establecido a 0.9, mientras que la de mutación se ha fijado a $1/n$, siendo n el número de variables. Se han llevado a cabo 30 ejecuciones independientes de cada algoritmo.

r-NSGA-II hace uso de un parámetro $\delta \in [0, 1]$ que controla la presión de selección de la r-dominancia. En los experimentos realizados, el valor de δ se ha fijado usando el procedimiento adaptativo propuesto por los autores en [15], donde δ evoluciona de forma progresiva desde cero hasta un valor δ_0, prefijado de antemano. Por el contrario, R-NSGA-II controla la diversidad de soluciones mediante una estrategia de selección basada en distancias a través de un parámetro ϵ. Puesto que la densidad o distribución de soluciones de las poblaciones finales de r-NSGA-II y de R-NSGA-II dependen de estos dos parámetros, sus valores han sido ajustados, por ensayo y error, para cada problema test y punto de referencia. Así, las poblaciones finales son comparables a la generada por WASF-GA, ya que las aproximaciones del frente óptimo de Pareto se ajustan a la región proyectable desde el punto de referencia considerado. La elección de los puntos de referencia alcanzables e inalcanzables, generados de forma aleatoria, reflejan el interés del decisor en una región concreta de la frontera eficiente.

La Tabla 1 muestra, para cada problema test, los puntos de referencia considerados y los valores de los parámetros δ_0 and ϵ empleados en r-NSGA-II y en R-NSGA-II, respectivamente.

Tabla 1. Parámetros δ_0 y ϵ en r-NSGA-II y R-NSGA-II, respectivamente.

Número de objetivos	**Problema test**	**Alcanzable (A)**			**Inalcanzable (I)**		
		q	δ_0	ϵ	**q**	δ_0	ϵ
2	ZDT1	(0.8, 0.6)	0.91	0.0064	(0.2, 0.4)	0.73	0.0016
	ZDT2	(0.8, 0.8)	0.83	0.0045	(0.5, 0.3)	0.93	0.0049
	ZDT3	(0.3, 0.8)	0.8405	0.0018	(0.2, 0)	0.71	0.0016
3	DTLZ2	(0.6, 0.7, 0.7)	0.65	0.07	(0.4, 0.4, 0.4)	0.6	0.078
	DTLZ7	(0.1, 0.8, 6)	0.8	0.07	(0.1, 0.1, 4)	0.8	0.06

Tabla 2. Media y desv. típica de la métrica basada en hipervolumen en 30 ejecuciones.

Problema	q	WASF-GA	r-NSGA-II	R-NSGA-II
ZDT1	A	$\mathbf{1,76e-01_{1,95e-05}}$	$1,76e-01_{3,85e-04}$	$1,76e-01_{3,26e-04}$
	I	$\mathbf{4,67e-01_{6,53e-05}}$	$4,65e-01_{2,58e-03}$	$4,67e-01_{6,10e-04}$
ZDT2	A	$6,90e-02_{1,12e-05}$	$6,91e-02_{9,20e-05}$	$\mathbf{6,93e-02_{4,26e-05}}$
	I	$\mathbf{2,65e-01_{4,42e-05}}$	$2,65e-01_{6,93e-04}$	$2,64e-01_{1,61e-03}$
ZDT3	A	$\mathbf{6,88e-02_{7,30e-07}}$	$6,80e-02_{2,07e-03}$	$6,55e-02_{1,77e-03}$
	I	$\mathbf{7,36e-01_{6,59e-05}}$	$7,36e-01_{2,92e-03}$	$7,31e-01_{1,71e-03}$
DTLZ2	A	$\mathbf{3,05e-03_{8,85e-07}}$	$2,03e-03_{1,95e-04}$	$1,73e-03_{1,10e-04}$
	I	$1,80e-01_{2,71e-04}$	$\mathbf{1,82e-01_{1,88e-03}}$	$1,74e-01_{2,89e-03}$
DTLZ7	A	$\mathbf{4,15e-02_{1,52e-03}}$	$3,68e-02_{1,50e-03}$	$3,72e-02_{1,23e-03}$
	I	$7,54e-01_{3,81e-04}$	$\mathbf{7,85e-01_{5,47e-03}}$	$7,84e-01_{3,98e-03}$

Para determinar la calidad de la aproximación realizada por cada algoritmo, se ha hecho uso de un indicador basado en hipervolumen [11] adaptado a la región de interés. Esta métrica estima el volumen del espacio de objetivos (acotado por un punto de referencia alcanzable, denotado por **r**) dominado por un conjunto de soluciones. Cuanto mayor es su valor, mejor es la aproximación. En los experimentos realizados, **r** se ha igualado al punto de referencia, si éste era alcanzable. En el caso inalcanzable, se considera que **r** es el vector formado por los peores valores de cada objetivo (*vector nadir*). En cualquier caso, para el cálculo del hipervolumen se tienen en cuenta únicamente las soluciones de la población final que pertenecen a la región proyectable. La Tabla 2 muestra el valor medio y la desviación típica del indicador en 30 ejecuciones de cada algoritmo para cada problema (considerando los párametros y puntos de referencia descritos en la Tabla 1), resaltando en negrita los mejores valores. Como puede observarse, las soluciones finales generadas por WASF-GA poseen un mayor valor medio de esta métrica, según lo cual puede considerarse que realiza una mejor aproximación de la región proyectable en comparación a r-NSGA-II y R-NSGA-II. Atendiendo a los valores de la desviación típica es posible puntualizar que el algoritmo propuesto es más estable, ya que genera aproximaciones similares en diferentes ejecuciones. Además, la complejidad computacional de WASF-GA es $O(k \cdot N \cdot N_\mu)$ y la de r-NSGA-II y R-NSGA-II $O(k \cdot N^2)$. Dado que hemos considerado $N_\mu = \frac{N}{2}$, la complejidad del primero es menor que la de los otros dos algoritmos.

4. Conclusiones

El principal propósito de los algoritmos evolutivos basados en preferencias es aproximar la región del frente óptimo de Pareto que mejor se ajusta a las preferencias del DM. Considerando un punto de referencia dado por el DM, el algoritmo WASF-GA propuesto en este artículo tiene como objetivo aproximar

la región del frente óptimo de Pareto que puede ser proyectada desde dicho punto de referencia. Cuando el punto de referencia es alcanzable, esta región contiene todas las soluciones eficientes que lo dominan. Para un punto de referencia inalcanzable, la región proyectable está formada por las soluciones eficientes más cercanas a este punto, con respecto a la distancia minmax. En este caso, consideramos que estas soluciones son las de mayor interés para el DM puesto que empeoran en menor medida los valores de referencia.

WASF-GA clasifica los individuos en cada generación por fronteras teniendo en cuenta los valores de la función escalarizada de logro de Wierzbicki para un conjunto de vectores de pesos. Para aproximar la región proyectable desde un punto de referencia con una buena distribución de soluciones, la muestra de vectores de pesos debe estar equidistribuida.

Una característica importante de WASF-GA desde el punto de vista del DM, es que éste únicamente debe proporcionar un punto de referencia. Por el contrario, tanto r-NSGA-II como R-NSGA-II requieren de un parámetro adicional, que debe ser ajustado en cada problema.

WASF-GA aproxima la región proyectable desde el punto de referencia con N_μ soluciones. El DM puede indicar de antemano el número de soluciones deseadas, aunque, por defecto, puede establecerse que $N_\mu = \frac{N}{2}$.

Se han realizado una serie de experimentos sobre varios problemas test, comparando los resultados obtenidos por WASF-GA con los generados por otros dos métodos evolutivos basados en punto de referencia, r-NSGA-II y R-NSGA-II. Según la complejidad computacional y la calidad de las aproximaciones de los tres algoritmos, consideramos que WASF-GA es más eficiente y consigue una mejor aproximación de la región proyectable desde el punto de referencia, según la métrica empleada basada en hipervolumen. No obstante, parece sugerente realizar nuevos experimentos en problemas test más complejos, como pueden ser las familias WFG y UF. Asimismo, un análisis estadístico de los hipervolúmenes obtenidos, por cada método en cada una de la ejecuciones independientes, permitiría determinar el nivel de confianza estadística de los resultados presentados en los experimentos realizados.

Desde el punto de vista preferencial, WASF-GA sería susceptible de ser adaptado para considerar varios puntos de referencia al mismo tiempo, lo que permitiría aproximar simultáneamente diferentes regiones del frente óptimo de Pareto.

Referencias

[1] J. Branke, K. Deb, K. Miettinen, and R. Slowinski, editors. *Multiobjective Optimization. Interactive and Evolutionary Approaches*, volume 5252 of *Lecture Notes in Computer Science*. Springer, 2008.

[2] J. Branke, T. Kaussler, and H. Schemeck. Guidance in evolutionary multiobjective optimization. *Advances in Engineering Software*, 32:499–507, 2001.

[3] C. A. C. Coello, G. B. Lamont, and D.A.V. Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems. Second Edition*. Springer, New York, 2007.

[4] C.A.C. Coello. Handling Preferences in Evolutionary Multiobjective Optimization: A Survey. In *IEEE Congress on Evolutionary Computation*, pages 30–37, 2000.

[5] K. Deb. *Multi-objective Optimization using Evolutionary Algorithms*. Wiley, Chichester, 2001.

[6] K. Deb and A. Kumar. Interactive Evolutionary Multi-Objective Optimization and Decision-Making using Reference Direction Method. In *9th Annual Conference on Genetic and Evolutionary Computation (GECCO-2007)*, pages 781–788, 2007.

[7] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[8] K. Deb, J. Sundar, B. Ubay, and S. Chaudhuri. Reference point based multi-objective optimization using evolutionary algorithm. *International Journal of Computational Intelligence Research*, 2(6):273–286, 2006.

[9] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable multi-objective optimization test problems. In *Congress on Evolutionary Computation (CEC-2002)*, pages 825–830, 2002.

[10] M. Ehrgott and X. Gandibleux. *Multiple Criteria Optimization: State of the Art. Annotated Bibliographic Surveys*. Springer-Verlag, Berlin, 2002.

[11] C.M. Fonseca, L. Paquete, and M. Lopez-Ibanez. An improved dimension-sweep algorithm for the hypervolume indicator. In *IEEE Congress on Evolutionary Computation (CEC-2006)*, pages 1157 – 1163, 2006.

[12] M. Gong, F. Liu, W. Zhang, L. Jiao, and Q. Zhang. Interactive MOEA/D for multi-objective decision making. In *13th annual conference on Genetic and evolutionary computation*, GECCO '11, pages 721–728, 2011.

[13] K. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston, 1999.

[14] L. Rachmawati and D. Srinivasan. Preference Incorporation in Multiobjective Evolutionary Algorithms. In *IEEE Congress on Evolutionary Computation*, pages 3385–3391, 2006.

[15] L. Ben Said, S. Bechikh, and K. Ghedira. The r-dominance: A new dominance relation for interactive evolutionary multicriteria decision making. *IEEE Transactions on Evolutionary Computation*, 14(5):801–818, 2010.

[16] K. Sindhya, A.B. Ruiz, and K. Miettinen. A Preference Based Interactive Evolutionary Algorithm for Multi-objective Optimization: PIE. In R. Takahashi, K. Deb, E. Wanner, and S. Greco, editors, *Evolutionary Multi-Criterion Optimization*, volume 6576 of *Lecture Notes in Computer Science*, pages 212–225, 2011.

[17] R. E. Steuer. *Multiple Criteria Optimization: Theory, Computation and Application*. John Wiley, New York, 1986.

[18] L. Thiele, K. Miettinen, P. Korhonen, and J. Molina. A preference-based evolutionary algorithm for multi-objective optimization. *Evolutionary Computation*, 17(3):411–436, 2009.

[19] A. P. Wierzbicki. The use of reference objectives in multiobjective optimization. In G. Fandel and T. Gal, editors, *Multiple Criteria Decision Making, Theory and Applications*, pages 468–486. Springer-Verlag, Berlin, 1980.

[20] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195, 2000.

[21] E. Zitzler and L. Thiele. Multiobjective optimization using evolutionary algorithms - A comparative case study. In A. Eiben, T. Bäck, M. Schoenauer, and H.P. Schwefel, editors, *Parallel Problem Solving from Nature*, volume 1498 of *Lecture Notes in Computer Science*, pages 292–301. Springer-Verlag, Berlin, 1998.

Paralelización de nuevas meta-heurísticas para resolver problemas de enrutado de vehículos

R. Baños†, J. Ortega†, C. Gil‡

† Dpt. Arquitectura y Tecnología de Computadores, Centro de Investigación en Tecnologías de la Información y las Comunicaciones (CITIC-UGR), Universidad de Granada, Periodista Daniel Saucedo s/n, E-18071 Granada, {rbanos,jortega}@ugr.es
‡ Dpt. Informática, ceiA3, Universidad de Almería, Carretera de Sacramento s/n, La Cañada de San Urbano s/n, E-04120 Almería, cgilm@ual.es

Resumen El problema del enrutado de vehículos, y sus múltiples variantes, tienen una gran influencia económica y organizativa en los ámbitos de la logística, el transporte, y la distribución. En las últimas décadas se han propuesto diferentes métodos de optimización computacional para su resolución, incluyendo meta-heurísticas. No obstante, el tiempo requerido para obtener soluciones de calidad en instancias complejas suele ser elevado, especialmente en el caso de resolver problemas multi-objetivo. En este trabajo se analiza la paralelización de meta-heurísticas en procesadores multi-núcleo para resolver el problema de enrutado de vehículos con ventanas temporales. Los resultados obtenidos muestran que las versiones paralelas mejoran significativamente el rendimiento de las versiones secuenciales ejecutadas en un solo núcleo del procesador, tanto en la formulación mono-objetivo como la multi-objetivo.

Keywords: Meta-heurísticas, optimización multi-objetivo, procesamiento paralelo, procesador multi-núcleo, enrutado de vehículos.

1. Introducción

La resolución de problemas de optimización, en especial los NP-completos [1], requieren del diseño de algoritmos eficientes. Uno de estos problemas es el del enrutado de vehículos (Vehicle Routing Problem, VRP), que tiene implicaciones económicas y organizativas, y que ha sido tratado mediante técnicas deterministas y aproximadas [2,3]. En los últimos años, se han planteado formulaciones multi-objetivo de VRPs cuya complejidad es aun mayor, ya que es necesario encontrar, no una, sino varias soluciones que constituyan una aproximación al frente óptimo de Pareto [4]. En ambos casos, la ejecución secuencial en un solo procesador para resolver determinadas instancias puede conllevar tiempos de ejecución elevados. Esta situación puede afrontarse mediante estrategias de procesamiento paralelo [5,6], que permiten reducir el tiempo de ejecución y/o mejorar la calidad de las soluciones obtenidas por las versiones secuenciales sin incrementar el tiempo de cómputo. Taillard [7] propuso uno de los primeros algoritmos paralelos para resolver el VRP, que consistía en descomponer el problema

en regiones polares incluyendo un elevado número de vehículos en subproblemas que son resueltos independientemente. El paradigma maestro-trabajador ha sido aplicado por diversos autores para tratar VRPs [8]. Dado que en determinados casos se pueden producir cuellos de botella en las comunicaciones a través de la red haciendo uso del modelo maestro-trabajador, se han propuesto estrategias como el uso de archivos donde se lee y escribe la información sobre las rutas encontradas [9]. Otros autores han propuesto la paralelización de estrategias que combinan búsqueda local y global utilizando un número de comunicaciones reducidas que se limitan a enviar el fitness calculado por cada nodo [10]. El diseño de algoritmos paralelos para resolver VRPs multi-objetivo en procesadores multi-núcleo es, hoy día, una cuestión abierta.

La Sección 2 describe el VRPTW y se incluye una descripción de las formulaciones mono-objetivo y multi-objetivo del problema de enrutado de vehículos con ventanas temporales (VRPTW). En la Sección 3 se presentan los algoritmos secuenciales y paralelos implementados para su resolución, mientras que los resultados obtenidos en un conjunto de benchmarks se presentan en la Sección 4. Finalmente, la Sección 5 ofrece las conclusiones y el trabajo futuro.

2. Enrutado de vehículos con ventanas temporales

El problema de enrutado de vehículos con ventanas temporales (Vehicle Routing Problem with Time Windows, VRPTW) [11] consiste en determinar el conjunto de rutas óptimas que debe seguir un determinado número de vehículos, con capacidad de carga limitada, estacionados en un almacén (centro logístico) que operan dentro de una franja temporal, y que son utilizados para visitar y suministrar las mercancías a los clientes. Las rutas asignadas a todos los vehículos parten del centro logístico para finalizar de nuevo en el tras realizar el servicio. La cantidad de mercancías suministradas por cada vehículo no puede exceder su capacidad. Los clientes, cuyas demandas pueden ser suministradas únicamente por un solo vehículo, están localizados en posiciones geográficamente diversas y tienen requisitos pre-establecidos en lo que respecta a las mercancías requeridas y al tiempo de servicio necesario para la descarga. La ventana temporal de cada cliente implica que el vehículo que le suministra puede llegar antes de que la ventana temporal abra (esperando a su apertura), pero no después de que esta haya cerrado. La distancia total recorrida por todos los vehículos definen el tiempo total de viaje. La versión mono-objetivo de este trabajo considera como objetivo a minimizar la distancia total recorrida por los vehículos utililzados.

2.1. VRPTW: Formulación mono-objetivo y multi-objetivo

El problema VRPTW puede ser modelado mediante un grafo completo no dirigido $G(V,E)$, donde los vértices $V=\{1, .., N\}$ corresponden al almacén central y a los clientes, y las aristas $e \in E\{(i,j) : i,j \in V\}$ a los enlaces entre ellos [11].

$$minimizar:\ TD = \sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{j=1}^{N} X_{ij}^{k} C_{ij} \quad (1)$$

$$sujeto\,a: \begin{cases} X_{ii}^{k} = 0 & (\forall i \in \{1,...,N\},\ \forall k \in \{1,...,K\}) \quad (2) \\ X_{ij}^{k} \in \{0,1\} & (\forall i,j \in \{1,...,N\}, \forall k \in \{1,...,K\}) \quad (3) \\ \sum_{k=1}^{K}\sum_{i=1}^{N} X_{ij}^{k} = 1 & (\forall j \in \{2,..,N\}) \quad (4) \\ \sum_{i=1}^{N}\sum_{j=2}^{N} X_{ij}^{k} d_j \leq Q & (\forall k \in \{1,..,K\}) \quad (5) \\ \sum_{k=1}^{K}\sum_{j=2}^{N} X_{1j}^{k} \leq K & (6) \\ \sum_{j=2}^{N} X_{1j}^{k} - \sum_{j=2}^{N} X_{j1}^{k} = 0 & (\forall k \in \{1,...,K\}) \quad (7) \\ a_j \leq s_{kj} \leq b_j & (\forall i,j \in [1,..,N], \forall k \in \{1,...,K\}) \quad (8) \\ s_{ki} + C_{ij} - L(1 - X_{ij}^{k}) \leq s_{kj} & (\forall i,j \in \{1,...,N\}, \\ & \forall k \in \{1,...,K\}) \quad (9) \end{cases}$$

La variable de decisión es X_{ij}^{k}, que es igual a 1 si el vehículo k viaja desde el nodo i hacia el nodo j, y 0 en caso contrario. Los parámetros son: a_j, que es el tiempo inicial en el que el cliente j permite iniciar el servicio; b_j es el último instante de tiempo para el cliente j en iniciar el servicio; C_{ij} es el coste (distancia o tiempo) de viajar desde el nodo i al nodo j; d_j es la demanda del cliente j; K es el máximo número de vehículos que pueden ser utilizados; N es el número de clientes más el almacén (el almacén se denota con el número 1, y los clientes con el 2,...N); Q es la capacidad de carga de cada vehículo.

Algunas formulaciones multi-objetivo del VRPTW añaden como segundo objetivo el balanceo de la carga de trabajo asociada a los vehículos [13], que suele estar expresada en términos de la distancia recorrida por los vehículos. La formulación multi-objetivo aquí considerada minimiza la distancia total recorrida (travelling distance, TD) y el desbalanceo en términos de la distancia recorrida por los diferentes vehículos (distance imbalance, DI) de forma simultánea. Este segundo objetivo se describe matemáticamente como:

$$minimizar:\ DI = max(\sum_{i=1}^{N}\sum_{j=2}^{N} X_{ij}^{k} C_{ij}) - min(\sum_{i=1}^{N}\sum_{j=2}^{N} X_{ij}^{k} C_{ij})\ \ (\forall k \in K) \quad (10)$$

3. Algoritmos implementados

3.1. Versiones secuenciales mono-objetivo y multi-objetivo

A continuación se describen las principales características del algoritmo secuencial mono-objetivo (Multiple Temperature Simulated Annealing, MT-SA).

- *La representación de las soluciones* se realiza mediante una codificación entera. La población P consta de p individuos (soluciones), P=$\{I_1, I_2, \ldots, I_p\}$,

cada una de las cuales representa las rutas recorridas por los K vehículos para suministrar a los N clientes. Así, cada individuo, I_i, se representa mediante cromosomas, C_{ik}, que constan en un conjunto variable de genes, $C_{ik} = \{0, G^1_{ik}, G^2_{ik}, ..., G^l_{ik}, 0\}$, que representan la ruta del k-ésimo vehículo en el i-ésimo individuo, con $1 \leq G^j_{ik} \leq N$. Así, $C_{4,9} = \{0, 11, 9, 74, 28, 0\}$ indica que el noveno vehículo del cuarto individuo parte del almacén central, y visita a los clientes 11, 9, 74, y 28, antes de volver a dicho almacén central.

- *Las soluciones iniciales* se crean mediante una heurística basada en las ventanas temporales (Time Window based Insertion Heuristic, TWIH), que consiste en visitar a los clientes según su orden de apertura de ventana temporal, que son insertados en los vehículos siempre y cuando se cumplan las restricciones del problema. Este procedimiento es muy rápido e intrínsecamente reduce el número de vehículos utilizados inicialmente.
- *Los operadores de búsqueda* implementados han sido diez operadores de mutación tomados de la literatura [11,14]. Algunos de ellos se basan en elegir un cliente y relocalizarlo en un orden diferente en el mismo vehículo, otras se basan en modificar el vehículo asignado a los clientes, mientras que otros operadores dividen, crean, o eliminan una determinada ruta.
- *El Enfriamiento Simulado (Simulated Annealing, SA)* [15] es una meta-heurística que utiliza la variable temperatura, t, que es inicializada a un valor elevado Ti, y que va reduciéndose al avanzar el proceso se búsqueda según un factor de enfriamiento, T_{factor}, hasta caer por debajo de un determinado umbral, T_{umbral}. Así, aquellas soluciones del vecindario, generadas a través de un operador de variación (mutación), que mejoren a la solución de partida serán directamente aceptadas, mientras que si la empeoran se aceptará con una probabilidad que decrece conforme se reduce la temperatura actual, t. La variable t se incluye dentro de la denominada función de Metrópolis [16] y actúa como factor de probabilidad para aceptar o rechazar una determinada solución. La meta-heurística secuencial mono-objetivo utilizada, Multiple Temperature Simulated Annealing (MT-SA), optimiza una población de p individuos utilizando los operadores de mutación. Siguiendo estudios previos [17], y a diferencia de otras estrategias que usan los mismos parámetros de enfriamiento en todas las soluciones [10], se establece un intervalo de temperaturas iniciales $[Ti_{min}, Ti_{max}]$, tal que la temperatura inicial del individuo I_1 es Ti_{min}, la del individuo I_p es Ti_{max}, mientras que el resto se distribuyen uniformemente a lo largo de este intervalo.

La versión multi-objetivo de MT-SA (MT-PSA), utiliza el concepto de Pareto-dominancia, y una variación del criterio de Metrópolis [16] a la hora de aceptar o rechazar las soluciones obtenidas tras aplicar los operadores de mutación. Así, la nueva solución obtenida tras la mutación se acepta si no es dominada por su solución padre, mientras que si la solución padre domina a la solución hijo, esta última es aceptada de acuerdo al criterio de Metrópolis, cuyos parámetros serán ahora la temperatura actual y la suma de las variaciones del fitness en los objetivos considerados. Las soluciones no dominadas encontradas por MT-PSA en el proceso de búsqueda son almacenadas en un archivo externo (ND). Si el

número de soluciones no dominadas excede el tamaño de dicho archivo, se aplica un operador de truncamiento preservando las soluciones extremas.

3.2. Versiones paralelas mono-objetivo y multi-objetivo

Los algoritmos secuenciales mono-objetivo y multi-objetivo (MT-SA y MT-PSA, respectivamente) han sido paralelizados utilizando el paradigma basado en islas, cuya elección viene determinada por la aspiración de obtener soluciones de mayor calidad que los algoritmos secuenciales sin incrementar el tiempo de ejecución. El Algoritmo 1 describe el funcionamiento de pMT-SA. La población de soluciones secuencial es distribuida entre los diferentes procesos, de forma que dichas soluciones iniciales son optimizadas aplicando el proceso descrito para MT-SA. Además, con el objetivo de equilibrar la diversificación y la intensificación en la búsqueda, esta aproximación paralela permite migrar (copiar) periódicamente el mejor individuo encontrado hasta el momento. Así, cada subpoblación envía periódicamente la mejor solución encontrada a un proceso central, que es responsable de evalúar las soluciones recibidas, además de la mejor propia, y distribuir la mejor global al resto de islas. Tras ello, el proceso de búsqueda continua utilizando su población actualizada con sus propios parámetros y el mismo esquema de enfriamiento de la solución previamente reemplazada.

La paralelización de la formulación multi-objetivo utiliza el mismo esquema pero con algunas diferencias. Por un lado, la versión multi-objetivo modificaría las líneas 9 a 11 del Algoritmo 1 para evaluar los diversos objetivos, así como para indicar la aceptación o rechazo de las mutaciones haciendo uso de la variante multi-objetivo del criterio de Metrópolis descrita. Antes de la línea 15 se insertaría una llamada a un procedimiento que gestione el archivo externo de soluciones no dominadas (ND). La última diferencia radica en que, aun manteniendo la topología de comunicaciones descrita en las líneas 15-27, en el caso multi-objetivo se migran frentes de soluciones en lugar de soluciones particulares, que son agrupadas y distribuidas por un proceso central.

Algoritmo 1: MT-SA paralelo (pMT-SA)

1. Parámetros entrada: $NP, MR, p, TC, Pr_mut, T_min, T_max, T_factor, T_umbral$;
2. Inicializar MPI en cada núcleo C_i, $i \in [1...NP]$
3. Para cada individuo $P_j \in C_i[P]$
4. $C_i[P_j] \leftarrow$TWIH();
5. $C_i[P_j].t \leftarrow T_min + j$*$((T_max - T_min)/p)$;
6. Mientras no se cumpla la condición de parada TC
7. Para cada $C_i[P_j] \in C_i[P]$
8. $C'_i[P_j] \leftarrow$Mutación$(C_i[P_j], Pr_mut, Rand(1,10))$;
9. $C_i[P_j] \leftarrow$Metropolis$(C_i[P_j], C'_i[P_j]), C_i[P_j].t)$;
10. If $(C_i[P_j] < C_i.Mejor_C)$ then
11. $C_i.Mejor_C \leftarrow C_i[P_j]$;
12. $C_i[P_j].t \leftarrow C_i[P_j].t * T_factor$;
13. If$(C_i[P_j].t < T_umbral)$ then
14. $C_i[P_j].t \leftarrow T_min + j * ((T_max - T_min)/p)$;
15. Si (Migrar($iteración_actual$,$instante_actual$,MR))
16. Si (i=1)
17. Para cada proceso C_r, $\forall r \in [2....NP]$
18. Recibir$(C_r.Mejor_C, C_r)$;
19. Mejor_global$\leftarrow$Mejor$(C_j.Mejor_C)$,$\forall$j$\in$[1..NP];
20. Para cada proceso C_r, $\forall r \in [2....NP]$
21. Enviar(Mejor_global,C_r);
22. Si (i>1) then
23. Enviar$(C_i.Mejor_C, C_1)$;
24. Recibir(Mejor_global,C_1);
25. Para cada $C_i[P_j] \in C_i[P]$
26. Si (Rand(1,100)<25)
27. $C_i[P_j] \leftarrow$ Mejor_global;
28. Retornar(Mejor_global);

4. Experimentos

4.1. Benchmarks y configuración paramétrica

La evaluación de pMT-SA en la resolución del VRPTW se lleva a cabo utilizando el conjunto de benchmarks propuesto por Solomon [18], que se compone de 56 instancias divididas en seis categorías, todas ellas con 25 vehículos y 100 clientes. Estos 56 benchmarks se dividen en seis categorías son C1, C2, R1, R2, RC1, y RC2, en función de la distribución geográfica del almacén y clientes, y de las características de las ventanas temporales.

Las poblaciones secuenciales, que constan de $|P|$=200 individuos, se distribuyen proporcionalmente entre las islas, cada una de las cuales es gestionada por un proceso ejecutado en un núcleo. La versión multi-objetivo, MT-PSA, utiliza un archivo externo ND del mismo tamaño. La probabilidad de aplicar un operador de mutación en cada iteración para un determinado individuo es Pr_{mut}=50 %, en cuyo caso se elige uno de los diez operadores con una probabilidad de entre el 5 % y el 15 %. Los parámetros de enfriamiento simulado son: Ti=$[T_{min}, T_{max}]$=[1,100], T_{factor}=0.995, T_{umbral}=0.001, de forma que si cuando la temperatura cae por debajo de T_{umbral}, se reinicializa (t=Ti) y el proceso de búsqueda continúa con la solución actual. Se ha establecido como condición de parada un tiempo de ejecución fijo. En el caso mono-objetivo, se han lanzado 15 ejecuciones independientes de pMT-SA usando NP={1,2,3,4} procesos/núcleos

para cada benchmark, retornando del mejor resultado y la mediana de dicha tanda de ejecuciones. El tiempo de ejecución fijo ha sido de 60 segundos, a la vez que se realiza una comunicación cada 6 segundos. La mejor solución global, calculada por un proceso central, es distribuida al resto de procesos, se copiándose en el 25 % de los individuos de las islas, que continúan el proceso de búsqueda con la misma configuración paramétrica. En el caso multi-objetivo se han realizado 25 ejecuciones independientes para cada benchmark durante un tiempo fijo de 30 segundos, también realizando 10 comunicaciones (una cada 3 segundos), aunque en este caso se envían frentes y no soluciones individuales.

Existen diversas librerias que permiten implementar de forma semi-automática algoritmos poblacionales para problemas reales [19,20]. No obstante, y con el objetivo de implementar un algoritmo tan específico como MT-SA/MT-PSA al VRPTW, hemos implementado totalmente los códigos secuenciales haciendo uso de C++. Existen diferentes librerías paralelas (MPI, OpenMP, Pthreads, OpenCL, TBB, Erlang, etc.). Aquí, hemos utilizado MPI (MPICH2 versión 1.2.1p1), ya que es una librería de paso de mensajes que se adapta bien al esquema de paralelización basado en islas, además de que permite una gran portabilidad y flexibilidad. El computador utilizado consta de un procesador Intel Quad Core (4 núcleos), 2.27 GHz, y 4 GB de memoria RAM.

4.2. Resultados

La Tabla 1 muestra los resultados obtenidos por las versiones secuencial (NP=1 proceso/núcleo) y paralelas (NP=2, NP=3, y NP=4 procesos/núcleos), en términos del mejor resultado y de la mediana de las 15 ejecuciones independientes lanzadas en las seis categorías de problemas de Solomón. Se resalta en negrita el mejor resultado obtenido.

Cuadro 1. Resumen de los resultados obtenidos por MT-SA y pMT-SA en los seis tipos de problemas de Solomon (tiempo de parada 60 segundos).

	1 proceso		2 procesos		3 procesos		4 procesos	
	mejor	mediana	mejor	mediana	mejor	mediana	mejor	mediana
R1	1485.0	1537.5	1269.7	1321.0	1264.5	1316.1	**1256.6**	**1303.7**
R2	1338.0	1400.3	1011.0	1091.6	1005.5	1087.2	**999.8**	**1069.2**
C1	1336.7	1427.0	900.7	1006.1	882.6	985.5	**869.0**	**953.2**
C2	1039.6	1119.2	683.6	795.7	674.2	783.0	**673.9**	**765.8**
RC1	1703.2	1776.8	1458.3	1518.5	1451.9	1512.5	**1432.4**	**1493.4**
RC2	1584.4	1685.5	1173.6	1290.6	1171.8	1277.4	**1154.7**	**1258.0**
MEDIA	1414.5	1489.8	1082.8	1170.6	1075.1	1160.3	**1064.4**	**1140.5**

Dichos resultados muestran claramente como las versiones paralelas mejoran a la versión secuencial, y que el incremento del número de procesos/núcleos utilizados permiten mejorar la calidad de las soluciones. En concreto, el coste medio de la ejecución mediana correspondiente a la versión secuencial (NP=1 proceso) es 1489.8, mientras que en el caso de las ejecuciones secuenciales se obtiene un resultado de 1170.6 para NP=2, de 1160.3 para NP=3, y de 1140.5 para NP=4. Por tanto, se observa como es posible mejorar la calidad de las

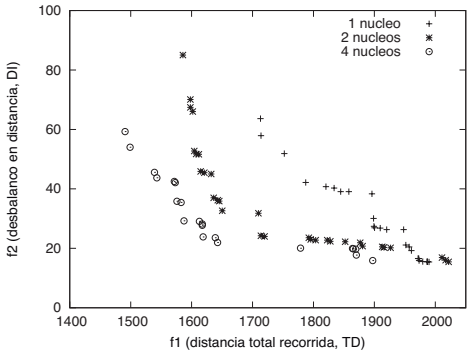


Figura 1. Frente mediano obtenido por MT-PSA y pMT-PSA (benchmark RC108).

soluciones obtenidas haciendo uso de procesamiento paralelo utilizando el modelo de islas. Además, los resultados desglosados denotan que haciendo uso de cuatro núcleos es posible obtener soluciones muy próximas a las mejores soluciones conocidas para estos problemas [12].

Cuadro 2. Resultados obtenidos siguiendo la métrica del hipervolumen (HV).

	1 proceso		2 procesos		4 procesos	
	HV_{mejor}	$HV_{mediana}$	HV_{mejor}	$HV_{mediana}$	HV_{mejor}	$HV_{mediana}$
R103	88.33	84.37	88.97	87.15	**90.94**	**88.09**
R108	88.50	85.02	89.99	86.80	**93.81**	**87.52**
R203	90.69	89.10	92.05	90.37	**93.49**	**90.78**
R208	92.96	88.67	93.76	89.21	**93.79**	**90.05**
C103	91.38	88.04	92.63	88.09	**92.87**	**88.37**
C108	87.06	82.42	86.19	84.45	**89.15**	**85.49**
C203	88.43	87.22	88.78	87.54	**89.23**	**87.59**
C208	92.37	88.27	**93.34**	90.16	93.02	**90.35**
RC103	84.05	77.12	85.19	81.51	**88.07**	**82.03**
RC108	88.81	85.70	91.49	87.44	**92.38**	**89.61**
RC203	90.99	88.41	91.34	88.62	**91.48**	**88.86**
RC208	91.47	89.44	91.40	89.34	**91.91**	**90.01**
MEDIA	89.59	86.15	90.43	87.56	**91.68**	**88.23**

En cuanto a la implementación multi-objetivo, la Tabla 2 muestra los resultados obtenidos por la versión secuencial (NP=1 proceso/núcleo) y paralela (NP=2 y NP=4 procesos/cores) en términos del hipervolumen (HV) cuando el criterio de parada es un tiempo de ejecución fijo de 30 segundos, realizando 10 comunicaciones (una cada 3 segundos). Se han lanzado ejecuciones en un subconjunto de problemas de Solomon. El frente mediano corresponde al frente con el decimotercer valor de hipervolumen más elevado. Dichos resultados numéricos, que se corroboran gráficamente en la Figura 1, muestran que la calidad de las soluciones tiende a mejorar claramente cuando el número de núcleos crece.

La Figura 2 muestra los tiempos de ejecución requeridos por pMT-PSA en obtener un frente cuyo hipervolumen sea de, al menos, el 95 % del hipervolu-

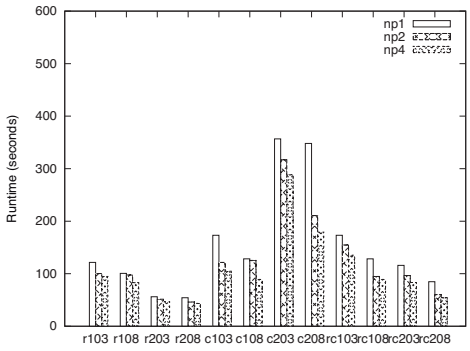


Figura 2. Tiempos de ejecución requeridos para obtener un frente con hipervolumen del 95 % respecto al frente de referencia.

men obtenido por este algoritmo tras ser ejecutado durante tres horas. Como se observa, el tiempo requerido por pMT-PSA haciendo uso de NP=4 núcleos es inferior al requerido por dicho algoritmo utilizando un menor número de núcleos.

5. Conclusiones

Los problemas de enrutado de vehículos constituyen un área primordial en el campo de la logística, transporte y distribución. Dado que dichos problemas son NP-duros, resulta aconsejable la aplicación de meta-heurísticas. Sin embargo, en determinadas aplicaciones es necesario abordar instancias de gran tamaño u obtener soluciones de alta calidad en un tiempo reducido, por lo que el uso del procesamiento paralelo se convierte en una herramienta conveniente. En el presente trabajo se analiza la mejora del rendimiento de una meta-heurística basada en enfriamiento simulado mediante la paralelización basada en el modelo de islas utilizando MPI. Los resultados obtenidos en la resolución de un conjunto de benchmarks asociados al problema VRPTW muestran que dicha paralelización es capaz de mejorar la calidad de las versiones secuenciales, tanto mono-objetivo como multi-objetivo, sin incrementar el tiempo de cómputo. Como trabajo futuro se plantea la paralelización haciendo uso de esquemas híbridos MPI/OpenMP.

Agradecimientos

Este trabajo ha sido financiado por el Ministerio de Economía y Competitividad a través de los proyectos TIN2008-01117 y TIN2012-32039. R. Baños agradece la finaciación del contrato postdoctoral Juan de la Cierva.

Referencias

1. Garey, M.R., Johnson, D.S.: *Computers and intractability: A guide to the theory of NP-completeness*, W.H. Freeman & Company, San Francisco, 1979.
2. Bräysy, O., Gendreau, M.: *Vehicle routing problem with time windows, part II: metaheuristics*, Transportation Science 39 (2005) 119-139.
3. Eksioglu, B., Vural, A.V., Reisman, A.: *The vehicle routing problem: A taxonomic review*, Computers & Industrial Engineering 57/4 (2009) 1472-1483.
4. Goldbergm D.E.: *Genetic algorithms in search, optimization and machine learning*, Addison-Wesley Publisher, 1989.
5. Alba, E.: *Parallel metaheuristics: A new class of algorithms*, Wiley, 2005.
6. Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: *Evolutionary algorithms for solving multi-objective problems*, Genetic and Evolutionary Computation Series, Springer (2007).
7. Taillard, É.: *Parallel iterative search methods for vehicle routing problems*, Networks 23/8 (1993) 661-673.
8. Gehring, H., Homberger, J.: *Parallelization of a two-phase metaheuristic for routing problems with time windows*, Journal of Heuristics 8/3 (2002) 251-276.
9. Gröer, C., Golden, B., Wasil, E.: *A Parallel Algorithm for the Vehicle Routing Problem*, INFORMS Journal on Computing 23/2 (2011) 315-330.
10. Arbelaitz, O., Rodriguez, C.: *A High Eficiency Parallel Algorithm for the VRPTW based on Simulated Annealing*, In Proceedings of the Joint Conference on Information Sciences, JCIS-2000, 411-416.
11. El-Sherbeny, N.A.: *Vehicle routing with time windows: an overview of exact, heuristic and metaheuristic methods*, Journal of King Saud University (Science) 22/3 (2010) 123-131.
12. Nagata, Y., Bräysy, O., Dullaert, W.: *A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows*, Computers & Operations Research 37 (2010) 724-737.
13. Jozefowiez, N., Semet, F., Talbi, E-G.: *Multi-objective vehicle routing problems*, European Journal of Operational Research 189/2 (2008) 293-309.
14. Alvarenga, G.B., Mateus, G.R., de Tomic, G.: *A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows*, Computers & Operations Research 34/6 (2007) 1561-1584.
15. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: *Optimization by Simulated Annealing*, Science 220/4598 (1983) 671-680.
16. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A., Teller, E.: *Equation of state calculations by fast computing machines*, The Journal of Chemical Physics 21/6 (1953) 1087-1092.
17. Baños, R., Gil, C., Ortega, J., Montoya, F.G.: *A parallel multilevel metaheuristic for graph partitioning*, Journal of Heuristics 10/3 (2004) 315-336.
18. Solomon, M.M.: *Algorithms for the vehicle routing and scheduling problems with time window constraints*, Operations Research 35/2 (1987) 254-265.
19. Wall, M.: *GAlib: a C++ library of genetic algorithm components, (ver. 2.4.)*, Massachusetts Institute of Technology, 1996.
20. Cahon, S., Melab, N. Talbi, E.G.: *ParadisEO: A framework for the reusable design of parallel and distributed metaheuristics*, Journal of Heuristics, 10/3 (2004) 357-380.

Programación Automática con Colonias de Hormigas Multi-Objetivo en GPUs

Alberto Cano, Juan Luis Olmo, y Sebastián Ventura

Departamento de Informática y Análisis Numérico,
Universidad de Córdoba, España
{acano,jlolmo,sventura}@uco.es

Abstract. La tarea de clasificación ha sido abordada recientemente con éxito mediante el paradigma de programación automática con hormigas. Esta tarea de minería de datos demanda grandes recursos computacionales, especialmente cuando se abordan conjuntos de elevada dimensionalidad. Este trabajo presenta un modelo de paralelización de un algoritmo de programación automática con hormigas multi-objetivo para clasificación, usando tarjetas gráficas (GPUs) y el modelo de programación NVIDIA CUDA. El modelo es escalable a múltiples GPUs y a conjuntos de gran tamaño. Se realiza un estudio experimental donde se evalúa el rendimiento y la eficiencia del modelo comparando los tiempos de ejecución de la implementación en CPU y en GPU. Los resultados experimentales nos permiten analizar la escalabilidad y eficiencia del modelo con respecto a la complejidad del conjunto de datos y el tamaño de la población del algoritmo. El modelo en GPU demuestra un gran rendimiento y escalabilidad en conjuntos de datos con un elevado número de instancias, logrando una aceleración de alrededor de 250 $\times$ cuando se comparan los tiempos de ejecución del algoritmo en 4 GPUs con la versión multi-hilo en CPU. El gran rendimiento obtenido en GPU nos permite ampliar la aplicabilidad del algoritmo a conjuntos de datos sensiblemente más complejos, que previamente eran inmanejables en un tiempo razonable.

Palabras clave: Minería de datos, programación automática con colonias de hormigas, paralelización en GPU

1 Introducción

La tarea de clasificación es una tarea de aprendizaje automático supervisado consistente en predecir la clase de ejemplos no etiquetados, cuya clase es desconocida, utilizando las propiedades de los ejemplos de entrenamiento empleados para aprender previamente un modelo, y cuya etiqueta de clase era conocida.

El problema de la dimensionalidad es uno de los mayores retos en la aplicación de algoritmos basados en optimización mediante colonias de hormigas (ACO) [1] debido al elevado tiempo computacional que requieren. Además, centrándonos en la tarea de clasificación, la inducción de clasificadores resulta cada vez más complicada, debido al hecho de que las capacidades de generación y recolección de datos en dominios reales crecen a un ritmo exponencial. De hecho, puede que no sea posible ejecutar ciertos algoritmos secuenciales sobre determinados conjuntos de datos. Por este motivo, resulta

crucial diseñar algoritmos paralelos capaces de manejar estas ingentes cantidades de datos [2, 3]. Recientemente se ha proporcionado una taxonomía de las estrategias de paralelización de los algoritmos de ACO, que distingue entre modelos maestro-esclavo, celular, de ejecuciones independientes paralelas, multicolonia e híbrido [4]. Por otro lado, en lo que respecta a las arquitecturas que se han utilizado hasta la fecha para paralelizar algoritmos de ACO, destacan las plataformas de clúster como opción más empleada, seguidas por los multiprocesadores y los computadores masivamente paralelas [4]. Recientemente, otras opciones van ganando atención, como la computación en grid, los servidores multinúcleo y las unidades de procesamiento gráfico (GPUs) [5, 6].

Las GPUs son dispositivos con unidades de procesamiento masivamente paralelo, que proporcionan hardware paralelo muy rápido por un precio mucho más económico que los sistemas paralelos tradicionales. De hecho, desde la introducción de la arquitectura CUDA en 2007, multitud de investigadores han aprovechado el poder de la GPU para computación de propósito general (GPGPU) [7, 8]. Concretamente, se ha estudiado la aplicación de GPGPU para acelerar algoritmos de computación evolutiva [9, 10] y metaheurísticas [11]. Debido a las ventajas que ofrece la GPGPU, en este trabajo pretendemos explorar el rendimiento de un algoritmo multiobjetivo de clasificación basado en programación con hormigas, denominado MOGBAP [12], paralelizado mediante GPUs [13], comparándolo con su versión paralela multihilo. El estudio experimental analiza el rendimiento y escalabilidad del algoritmo desde problemas más simples a más complejos, incrementando el número de instancias y atributos. Los resultados demuestran que su paralelización permite extender la aplicación de MOGBAP a dominios que presentan conjuntos de datos con enormes cantidades de instancias y atributos, donde antes era extremadamente complicado ejecutar el algoritmo.

El artículo se organiza como sigue. En la siguiente sección se introducirá el algoritmo multiobjetivo para clasificación en que se centra este trabajo, para pasar a discutir la paralelización del mismo y finalizar presentando la implementación GPU. La Sección 3 describe el estudio experimental. Los resultados obtenidos se presentan en la Sección 4. Por último, en la Sección 5, se presentan las conclusiones del trabajo.

2 Algoritmo GPU-MOGBAP (GPU Multi-Objective Grammar Based Ant Programming)

El algoritmo MOGBAP, acrónimo de *Multi-Objective Grammar-Based Ant Programming*, es un algoritmo de programación automática multiobjetivo para el problema de clasificación multiclase [12]. Como entrada, recibe un conjunto de datos de entrenamiento, y construye un clasificador compuesto de reglas *IF-THEN* y que actúa como lista de decisión donde las reglas descubiertas se ordenan de forma descendente en base a su *fitness*. Así, a la hora de predecir la clase para una nueva instancia del conjunto de test, la clase asignada corresponderá con el consecuente de la primera regla del clasificador que se dispare, que será aquella cuyo antecedente cubra la nueva instancia.

El algoritmo MOGBAP se basa en el uso de una gramática de contexto libre que restringe el espacio de búsqueda y que asegura que cualquier individuo generado sea sintácticamente válido [14]. En lo concerniente a la codificación de los individuos, cada hormiga representará una regla de clasificación. La creación de una nueva hormiga

seguirá la aplicación de la regla de transición desde el estado inicial del entorno hasta alcanzar un estado final, entorno que adopta la forma de un árbol de derivación.

El algoritmo MOGBAP presenta también una estrategia multiobjetivo novedosa para la tarea de clasificación en el sentido de que descubre un frente de Pareto separado para cada clase del conjunto de datos, presuponiendo que determinadas clases van a ser más difíciles de predecir que otras. De hecho, si en lugar de separar los individuos por clase, se les asignase un rango a todos juntos de acuerdo a la dominancia de Pareto, es muy posible que ocurriera solapamiento. La estrategia se puede resumir como sigue. Una vez que los individuos de la generación actual se han creado y evaluado por cada objetivo considerado, se dividen en k grupos, uno por cada clase del conjunto de entrenamiento. Entonces, cada grupo de individuos se combina con las soluciones almacenadas previamente en el frente de Pareto correspondiente a esa clase en la generación anterior, para ordenarlos de acuerdo a la dominancia de Pareto, encontrando así un nuevo frente de Pareto para cada clase. Por tanto, existirán k frentes de Pareto, y sólo las soluciones no dominadas participarán en el refuerzo de feromonas sobre el entorno. Por último, el clasificador final se compondrá a partir de los individuos no dominados que se han almacenado en cada uno de los k frentes de Pareto descubiertos en la última generación del algoritmo. Para seleccionar las reglas del clasificador apropiadamente, se lleva a cabo sobre cada frente un enfoque de nichos para seleccionar los individuos del mismo que se incluirán en el clasificador.

2.1 Versión paralelizada con multihilo

En esta sección comentaremos las fases del algoritmo MOGBAP que se han paralelizado utilizando multihilo. Esta versión paralela en CPU del algoritmo se ha diseñado para ejecutar los hilos utilizando el *Java ExecutorService*, que define un *pool* de hilos de forma dinámica teniendo en cuenta el número de procesadores disponibles. Así, la escalabilidad del algoritmo con respecto al número de procesadores se lleva a cabo automáticamente sin supervisión por parte del usuario.

El esquema de esta versión paralela de MOGBAP se ajusta a la estructura jerárquica maestro-esclavo definida por Pedemonte et al. [4], dado que un proceso maestro gestiona las estructuras de información globales y delega tareas en los procesos esclavos. Dichas tareas son la generación de individuos, su evaluación, el descubrimiento de los frentes de Pareto y el procedimiento de nichos, comunicando los resultados de vuelta al proceso maestro. A continuación se justifica la paralelización de cada fase.

La fase de inicialización de MOGBAP se encarga de inicializar la gramática a utilizar a partir de los metadatos existentes en el conjunto de entrenamiento, inicializando, asimismo, la estructura de datos que albergará el espacio de estados. Dicha estructura sigue un enfoque de construcción incremental, de manera que cada hormiga que se cree en cualquier iteración almacenará sus estados visitados en dicha estructura sólo en caso de que no hayan sido insertados previamente. La fase de creación de hormigas se puede paralelizar mediante multihilo, ya que las hormigas se pueden crear simultáneamente, al no influir el camino seguido por una hormiga en el que vaya a seguir otra en la misma generación. Sin embargo, hay que tener en cuenta que el almacenamiento de los estados visitados en la estructura de datos del espacio de estados no se puede delegar en los hilos directamente, al tratarse de una estructura común que no se puede modificar

concurrentemente. Por este motivo, la actualización de dicha estructura no se puede hacer en paralelo y debe ser efectuada por el proceso maestro, que recibirá los caminos seguidos por cada hormiga generada.

La fase de evaluación comprueba las instancias que cubre cada hormiga y calcula los valores de *fitness* de cada objetivo basándose en la matriz de confusión. La implementación paralela con multihilo es elemental.

En cuanto a la estrategia multiobjetivo, el descubrimiento de cada frente de Pareto a partir de un conjunto de individuos no se puede paralelizar, ya que es necesario seguir un procedimiento secuencial para descubrir las relaciones de dominancia entre los mismos. No obstante, ya que la dominancia de Pareto se evalúa por grupos de individuos, uno por clase, es posible utilizar hilos, de forma que se emplee un hilo para calcular la dominancia en un grupo de individuos determinado.

Las actividades de refuerzo, evaporación y normalización las realiza el proceso maestro, al requerir el acceso a la estructura de datos del espacio de estados y, por tanto, no se puede paralelizar (los accesos para escritura no pueden ser concurrentes).

El enfoque de nichos que se lleva a cabo sobre los individuos de cada frente de Pareto se puede paralelizar utilizando múltiples hilos, y los individuos resultantes se devuelven al proceso maestro, que los ordena para construir el clasificador.

2.2 Versión paralelizada con GPU

En esta sección se presenta la implementación en GPU del algoritmo MOGBAP. Antes de centrarnos en la paralelización mediante GPUs de la fase de evaluación, pasamos a justificar el motivo por el cual el resto de fases no se paralelizan también en GPU.

La inicialización de la gramática y el espacio de estados no se paralelizó utilizando multihilo y, del mismo modo, no es posible hacerlo con GPU. La paralelización de la fase de creación de los individuos tampoco se efectuó en GPU, manteniéndola en multihilo, ya que la transferencia de la estructura de datos utilizada por el espacio de estados a la GPU no es apropiada. De hecho, no es posible conseguir una paralelización interna más allá de la obtenida mediante la creación de cada hormiga en su hilo correspondiente, ya que no consiste más que en seleccionar una serie de transiciones desde el estado inicial a uno final, lo cual se hace secuencialmente. Así pues, los costes de transferencia sobrepasarían claramente los beneficios de implementar esta fase en GPU, además de la complejidad inherente a dicha implementación, que está controlada por una gramática y por el número de derivaciones disponibles en cada momento. La implementación paralela en CPU utilizando multihilo es, en cambio, simple y eficiente.

La estrategia multiobjetivo se paralelizó utilizando multihilo, como se explicó en la Sección 2.1. No es posible paralelizar este paso más allá, debido al hecho de que localizar las relaciones de dominancia de Pareto en un conjunto de individuos es un proceso secuencial. Para el procedimiento de nichos se puede argumentar la misma justificación, ya que internamente procede ordenando los individuos de acuerdo a su *fitness* y efectuando una secuencia serial de operaciones siguiendo dicho orden. Por último, el proceso maestro es el encargado de llevar a cabo las fases de refuerzo, evaporación, normalización y la construcción del clasificador, como se explicó con anterioridad, por lo que no se pueden paralelizar.

Por otro lado, se ha demostrado que la paralelización de la fase de evaluación en algoritmos bioinspirados mediante GPUs obtiene un gran rendimiento [15, 16]. En nuestro caso, la evaluación de una hormiga no depende de la evaluación del resto de hormigas. Es más, el proceso de interpretar la regla codificada por una hormiga y comprobar si cubre una determinada instancia o no es también independente de las reglas y las instancias. Por todo ello, cada hilo del modelo en GPU se encarga de interpretar la regla de una hormiga sobre una instancia. La Figura 1 muestra el diagrama de flujo del modelo de GPU donde la fase de evaluación en la GPU se lleva a cabo empleando dos funciones de núcleo o *kernel*.

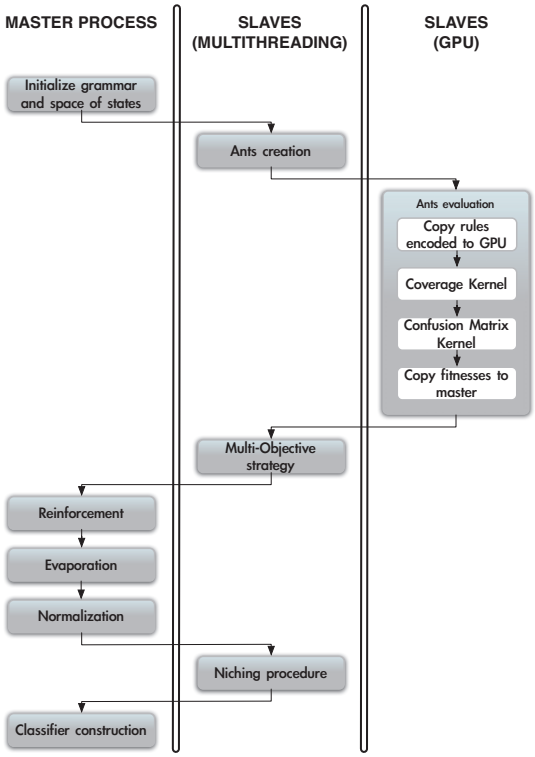


Figura 1. Computational flow chart of the GPU version

***Kernel* de cubrimiento** El *kernel* de cubrimiento interpreta las reglas, que se expresan en notación polaca inversa, sobre las instancias del conjunto de datos. El *kernel* se muestra en la parte superior de la Figura 2 y se ejecuta utilizando un grid de bloques de hilos en 2D. La longitud de la primera dimensión viene dada por el número de reglas. La longitud de la segunda depende del número de instancias y del número de hilos por bloque. Por tanto, el total de hilos en el grid será el producto de ambas dimensiones.

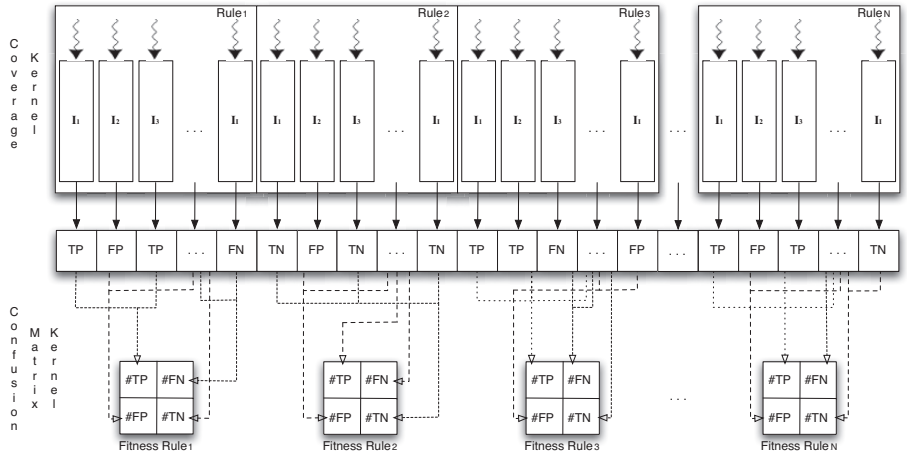


Figura 2. *Kernels* en la GPU

Los kernels se diseñan de modo que los hilos acceden a posiciones de memoria global alineadas para alcanzar un rendimiento óptimo. Así, los hilos solicitan direcciones de memoria consecutivas que se pueden servir en un menor número de transacciones de memoria. Las reglas se alojan en la memoria de tipo constante a modo de multidifusión para todos los hilos en un grupo. Todos los hilos del grupo evalúan la misma regla, pero sobre instancias diferentes. Por este motivo se sigue el nivel de paralelismo de una instrucción, múltiples datos, que se consigue cuando cada procesador realiza la misma tarea sobre instancias diferentes en datos distribuidos [17].

La interpretación de las reglas se efectúa utilizando una pila, insertando operandos en ella y, cuando una operación se realiza, extrayendo los operandos de la pila e insertando de vuelta el resultado de operar con ellos. El intérprete comprueba si la instancia es cubierta por la regla, teniendo en cuenta la clase predicha y la real de la instancia, pudiendo producirse uno de los siguientes valores: verdadero positivo (T_P), falso positivo (F_P), verdadero negativo (T_N), y falso negativo (F_N).

Kernel de la matriz de confusión El *kernel* de la matriz de confusión, que se muestra en la parte inferior de la Figura 2, realiza el conteo del número de T_P, F_P, T_N, y F_N que han sido previamente calculados para cada regla. Estos valores se utilizan para construir la matriz de confusión que nos permite calcular los valores de *fitness*. El proceso de sumar todos los valores de un array se conoce como reducción [18].

Los hilos se organizan para realizar una reducción en dos niveles y proporcionar accesos de memoria alineados que puedan servirse en menos transacciones de memoria, evitando así conflictos de bancos en memoria compartida. Cada hilo lleva a cabo la reducción parcial de los resultados de cubrimiento de una hormiga, almacenando la contabilización parcial en la memoria compartida. Una vez todos los elementos han sido contabilizados, se efectúa una llamada a una barrera de sincronización. La barrera

de sincronización se usa para coordinar la comunicación entre los hilos de un mismo bloque. Cuando algunos hilos dentro de un bloque acceden las mismas direcciones en memoria global o compartida, existen riesgos potenciales de lectura-tras-escritura, escritura-tras-lectura y escritura-tras-escritura para algunos accesos a memoria.

A continuación, sólo cuatro hilos realizan la suma final de las cuatro posiciones de la matriz de confusión. En este punto, no es necesario llamar a la barrera de sincronización dado que los hilos se encuentran en el mismo grupo y, ya que un grupo ejecuta una instrucción común a la vez, los hilos dentro de un grupo están sincronizados implícitamente.

Por último, se calcula los valores de *fitness* para los diferentes objetivos y se escriben de vuelta en la memoria global. Todos los valores de *fitness* se devuelven en un único array para ahorrar múltiples transacciones de memoria segmentadas.

3 Configuración del estudio experimental

En esta sección se presenta la configuración del estudio experimental. Los experimentos se ejecutaron en un PC equipado con un procesador de cuatro núcleos Intel Core i7 a 2.66 GHz, 12 GB de DDR3-1600 y 2 tarjetas gráficas NVIDIA GTX 690 de doble GPU (en total existen 4 GPUs operativas). El sistema operativo fue GNU/Linux Ubuntu 12.04 64 bit, junto con CUDA runtime 5.0, controladores NVIDIA 310.40, Eclipse 3.7.0, Java OpenJDK 1.6-23 y compilador GCC 4.6.4 (nivel de optimización O2).

En cuanto a la parametrización de la experimentación hay que definir por un lado los parámetros de los kernels, y por otro lado los parámetros del algoritmo MOGBAP. La configuración de los kernels requiere definir el número de hilos por bloque, habitualmente siendo 128, 256 o 512 hilos por bloque. De acuerdo a la hoja de cálculo de NVIDIA de utilización de la GPU, el número óptimo de hilos por bloque para nuestro problema es de 256 hilos por bloque. Este número garantiza la máxima utilización de los recursos de la GPU, mantiendo la carga de trabajo de la GPU lo más ocupada posible, a la vez que proporciona un gran número de bloques en el grid que permitan facilitar la escalabilidad a futuras GPUs con un mayor número de procesadores. Por otro lado, el algoritmo MOGBAP se ejecutó usando los parámetros por defecto recomendados por los autores del algoritmo en [12].

4 Resultados

En esta sección se presentan y analizan los resultados experimentales procedentes la ejecución del modelo sobre 11 conjuntos de datos del repositorio UCI [19]. Los conjuntos de datos poseen una gran variedad en su grado de complejidad. El número de clases varía entre 2 y 17, el número de atributos varía entre 4 y 44, y el número de instancias varía entre 150 y 1 millón. Los experimentos evalúan diferentes tamaños de población para analizar la escalabilidad del modelo, especialmente sobre conjuntos de datos completos con un elevado número de instancias. Los resultados experimentales mostrados representan la media del tiempo de ejecución y la aceleración sobre 100 ejecuciones independientes.

Tabla 1. Resultados en conjuntos de datos UCI

					Tiempo de ejecución (ms)				Aceleración vs 4 CPUs		
Conjunto de datos	Instancias	A	C	R	4 CPUs	1 690	2 690	4 690	1 690	2 690	4 690
iris	150	4	3	10	62.80	35.01	20.91	15.14	1.79	3.00	4.15
				20	90.70	37.44	25.25	16.45	2.42	3.59	5.51
				50	156.00	49.31	29.34	21.47	3.16	5.32	7.27
				100	235.30	57.92	35.09	25.78	4.06	6.71	9.13
				200	434.30	81.41	58.75	37.64	5.33	7.39	11.54
ionosphere	351	33	2	10	282.70	38.21	23.30	12.62	7.40	12.13	22.40
				20	418.70	42.46	25.75	15.23	9.86	16.26	27.49
				50	770.00	53.02	38.04	29.83	14.52	20.24	25.81
				100	1,411.90	85.07	51.85	34.62	16.60	27.23	40.78
				200	2,691.50	122.21	77.74	58.34	22.02	34.62	46.13
australian	690	14	2	10	208.50	33.56	23.01	17.58	6.21	9.06	11.86
				20	315.30	41.94	30.49	19.66	7.52	10.34	16.04
				50	429.37	51.57	30.72	23.51	8.33	13.98	18.26
				100	847.62	70.94	45.08	28.86	11.95	18.80	29.37
				200	1,732.65	73.02	45.76	37.37	23.73	37.86	46.36
vowel	990	13	11	10	308.00	54.90	35.43	28.18	5.61	8.69	10.93
				20	487.40	68.70	43.51	30.26	7.09	11.20	16.11
				50	761.05	82.46	56.34	38.47	9.23	13.51	19.78
				100	1,458.00	100.86	63.84	41.58	14.46	22.84	35.06
				200	3,262.87	154.51	91.48	62.96	21.12	35.67	51.82
segment	2,310	19	7	10	722.40	67.45	49.38	41.12	10.71	14.63	17.57
				20	1,215.70	76.33	53.46	44.26	15.93	22.74	27.47
				50	2,117.47	113.36	71.90	45.29	18.68	29.45	46.75
				100	3,962.32	159.06	103.02	80.06	24.91	38.46	49.49
				200	7,630.10	222.55	137.10	98.81	34.28	55.65	77.22
mushroom	8,124	22	2	10	2,074.40	94.46	57.98	49.20	21.96	35.78	42.16
				20	3,723.90	117.48	70.86	52.89	31.70	52.55	70.41
				50	7,141.74	152.41	95.64	55.00	46.86	74.67	129.85
				100	15,581.70	245.83	145.94	75.02	63.38	106.77	207.70
				200	29,596.69	424.43	231.88	142.33	69.73	127.64	207.94
kr-vs-kp	28,056	6	17	10	1,320.30	70.28	46.91	34.55	18.79	28.15	38.21
				20	2,394.70	71.24	49.63	42.15	33.61	48.25	56.81
				50	4,179.39	97.42	69.59	52.96	42.90	60.06	78.92
				100	8,058.31	159.90	80.38	57.69	50.40	100.25	139.68
				200	15,899.56	185.08	123.26	82.04	85.91	128.99	193.80
connect-4	67,557	42	3	10	24,665.15	794.75	636.26	473.05	31.04	38.77	52.14
				20	42,737.95	886.92	652.52	524.94	48.19	65.50	81.41
				50	110,240.25	1,351.64	1,086.06	836.53	81.56	101.50	131.78
				100	212,262.88	2,027.63	1,559.34	1,201.56	104.69	136.12	176.66
				200	412,462.96	3,191.51	2,485.45	1,843.21	129.24	165.95	223.77
fars	100,968	29	8	10	28,265.81	954.79	584.11	417.38	29.60	48.39	67.72
				20	51,754.57	1,110.29	657.12	452.73	46.61	78.76	114.32
				50	121,558.74	1,880.54	998.31	638.77	64.64	121.76	190.30
				100	251,519.89	2,727.52	1,332.41	994.23	92.22	188.77	252.98
				200	491,768.62	4,902.58	2,574.66	1,833.58	100.31	191.00	268.20
kddcup	494,020	41	10	10	162,285.80	7,547.98	4,042.97	2,357.78	21.50	40.14	68.83
				20	323,639.32	8,217.94	5,433.55	3,727.78	39.38	59.56	86.82
				50	775,597.05	12,802.72	7,942.96	4,816.93	60.58	97.65	161.01
				100	1,667,554.00	19,620.57	14,207.00	9,116.12	84.99	117.38	182.92
				200	3,283,987.91	29,580.09	17,324.18	13,272.88	111.02	189.56	247.42
poker	1,025,010	10	10	10	118,820.52	5,389.48	2,968.95	1,627.00	22.05	40.02	73.03
				20	245,588.40	7,994.84	5,250.21	3,291.00	30.72	46.78	74.62
				50	591,440.37	12,464.92	6,815.25	5,157.01	47.45	86.78	114.69
				100	1,219,990.65	20,725.03	10,011.94	6,737.62	58.87	121.85	181.07
				200	2,508,965.42	28,032.03	16,520.27	10,429.78	89.50	151.87	240.56

La Tabla 1 muestra los conjuntos de datos (instancias, attributos (A), clases (C)), los tiempos de ejecución y la aceleración obtenida. El modelo demuestra alcanzar un gran rendimiento y eficiencia, que se incrementa conforme el número de instancias y el tamaño de la población (número de reglas (R)) se incrementan. Concretamente, la aceleración máxima se alcanza cuando se emplean 4 GPUs sobre los tres conjuntos de datos con mayor número de instancias y con los mayores tamaños de población. Por otro lado, resultados menos significativos se obtienen en los conjuntos más pequeños. No obstante, pese a su menor número de instancias, se obtiene en todo caso aceleraciones superiores a 1, es decir, se demuestra que independientemente del tamaño de los datos y del número de reglas, siempre es recomendable realizar la evaluación en GPU. Además, se observa una buena escalabilidad del modelo respecto al aumento del número de GPUs, tamaño de la población y del tamaño de los datos. La Figura 3 muestra la tendencia de la aceleración con 1, 2 y 4 GPUs con respecto al número de instancias.

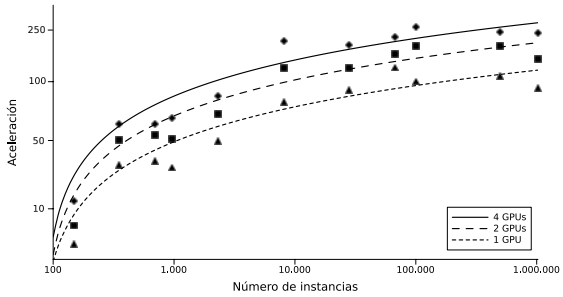


Figura 3. Aceleración con respecto al número de instancias

5 Conclusiones

En este trabajo se ha presentado una implementación paralela en GPU del algoritmo MOGBAP. El proceso de creación de hormigas, la estrategia multiobjetivo y el enfoque de nichos para seleccionar las reglas del clasificador se han paralelizado usando multihilo, debido a sus requerimientos sequenciales internos. Por otro lado, la evaluación de las hormigas, que es el proceso que demanda más tiempo, se realiza en GPU, alcanzando una gran aceleración especialmente sobre conjuntos de datos de alta dimensionalidad. El estudio experimental ha analizado el rendimiento y eficiencia del modelo atendiendo a diferentes tamaños de población y conjuntos de datos con variado número de instancias y atributos. El modelo ha demostrado una buena escalabilidad a conjuntos de datos de gran tamaño y a múltiples GPUs. Los resultados experimentales demuestran la gran aceleración que supone la implementación del modelo de evaluación en GPU, alcanzando aceleraciones de alrededor de 250 $\times$ cuando se compara el tiempo de evaluación en 4 GPUs con el multi-hilo en CPU. Además, se nos permite abordar conjuntos de datos mucho más grandes que anteriormente no era posible en un tiempo razonable.

Agradecimientos

Este trabajo ha sido financiado por la Junta de Andalucía y el Ministerio de Ciencia e Innovación bajo los proyectos P08-TIC-3720 y TIN-2011-22408, los fondos FEDER, y la beca FPU AP2010-0042.

Referencias

1. Michelakos, I., Mallios, N., Papageorgiou, E., Vassilakopoulos, M.: Ant colony optimization and data mining. Studies in Computational Intelligence **352** (2011) 31–60
2. Alba, E., Tomassini, M.: Parallelism and evolutionary algorithms. IEEE Transactions on Evolutionary Computation **6**(5) (2002) 443–462
3. Luque, G., Alba, E.: Parallel genetic algorithms: Theory and realworld applications. Studies in Computational Intelligence **367** (2011) 1–183
4. Pedemonte, M., Nesmachnow, S., Cancela, H.: A survey on parallel ant colony optimization. Applied Soft Computing **11** (2011) 5181–5197
5. Cecilia, J.M., García, J.M., Nisbet, A., Amos, M., Ujaldón, M.: Enhancing data parallelism for Ant Colony Optimization on GPUs. Journal of Parallel and Distributed Computing **73**(1) (2013) 42–51
6. Delévacq, A., Delisle, P., Gravel, M., Krajecki, M.: Parallel Ant Colony Optimization on GPUs. Journal of Parallel and Distributed Computing **73**(1) (2013) 52–61
7. Che, S., Boyer, M., Meng, J., Tarjan, D., Sheaffer, J.W., Skadron, K.: A performance study of general-purpose applications on graphics processors using CUDA. Journal of Parallel and Distributed Computing **68**(10) (2008) 1370–1380
8. Owens, J.D., Luebke, D., Govindaraju, N., Harris, M., Krger, J., Lefohn, A.E., Purcell, T.J.: A survey of general-purpose computation on graphics hardware. Computer Graphics Forum **26**(1) (2007) 80–113
9. Fok, K.L., Wong, T.T., Wong, M.L.: Evolutionary computing on consumer graphics hardware. IEEE Intelligent Systems **22**(2) (2007) 69–78
10. Vidal, P., Luna, F., Alba, E.: Systolic neighborhood search on graphics processing units. Soft Computing **In press** (2013)
11. Alba, E., Luque, G., Nesmachnow, S.: Parallel metaheuristics: Recent advances and new trends. International Transactions in Operational Research **20**(1) (2013) 1–48
12. Olmo, J.L., Romero, J.R., Ventura, S.: Classification rule mining using ant programming guided by grammar with multiple Pareto fronts. Soft Computing **16** (2012) 2143–2163
13. Cano, A., Olmo, J., Ventura, S.: Parallel Multi-Objective Ant Programming for Classification Using GPUs. Journal of Parallel and Distributed Computing **73**(6) (2013) 713–728
14. Wong, M.L., Leung, K.S.: Data Mining Using Grammar-Based Genetic Programming and Applications. Kluwer Academic Publishers, Norwell, MA, USA (2000)
15. Cano, A., Zafra, A., Ventura, S.: Speeding up the evaluation phase of GP classification algorithms on GPUs. Soft Computing **16**(2) (2012) 187–202
16. Franco, M.A., Krasnogor, N., Bacardit, J.: Speeding up the evaluation of evolutionary learning systems using GPGPUs. In: Genetic and Evolutionary Computation Conference (GECCO) (2010). (2010) 1039–1046
17. Leist, A., Playne, D., Hawick, K.: Exploiting graphical processing units for data-parallel scientific applications. Concurrency Computation Practice and Experience **21**(18) (2009) 2400–2437
18. Hwu, W.: Illinois ECE 498AL: Programming Massively Parallel Processors, Lecture 13: Reductions and their Implementation (2009)
19. Newman, D.J., A. Asuncion: UCI machine learning repository (2007)

Un Modelo Formal para Analizar Propiedades de Algoritmos Genéticos Paralelos *

Natalia López, Pablo Rabanal, Ismael Rodríguez, Fernando Rubio

Dpto. Sistemas Informáticos y Computación
Facultad Informática, Universidad Complutense de Madrid, Spain

Resumen En este trabajo presentamos un modelo formal que nos permite analizar propiedades no triviales del comportamiento de algoritmos genéticos paralelos implementados sobre múltiples islas. El modelo está basado en un sistema de transiciones etiquetadas probabilísticas, que representa la evolución de la población de cada una de las islas así como las interacciones entre las distintas islas. Gracias al estudio de las trazas que pueden producirse con estos sistemas, el modelo resultante permite comparar formalmente el comportamiento de distintos algoritmos.

1. Introducción

Los métodos formales proveen métodos sistemáticos para verificar la corrección de los sistemas computacionales, así como para analizar diversas de sus propiedades, tales como su rendimiento temporal o su comportamiento probabilístico. Este tipo de métodos resulta más provechoso si cabe en el estudio de sistemas paralelos y distribuidos [8,11]. Un sistema distribuido debe evitar enviar demasiados mensajes de coordinación entre los procesos para ser eficiente. Debido a la necesaria escasez de dichos mensajes, en cada instante de ejecución cada proceso paralelo podría haber avanzado una cantidad diferente de pasos en la realización de sus respectivas tareas. Por ello, el orden en que los procesos finalizan sus tareas es inherentemente indeterminista, lo que hace que en este tipo de sistemas aparezcan, con cierta frecuencia, errores debidos a que el orden causal esperado entre las actividades de los procesos no se haya preservado.

Para poder detectar estos problemas en las fases tempranas de desarrollo, los métodos formales permiten crear *modelos* donde se representan únicamente los aspectos críticos que se desean analizar (por ejemplo, la coordinación entre procesos), así como analizar sistemáticamente dichos modelos en busca de posibles errores. Es más, diversos métodos formales permiten incorporar en dichos modelos detalles que no solo conciernen a los objetivos funcionales del programa (qué debe hacer), sino también otros no funcionales (cómo debe hacerlo). Por ejemplo, existen modelos formales para representar el comportamiento temporal de los sistemas [18], el probabilístico [10], el temporal-probabilístico [2] (donde se cuantifica la probabilidad de tardar cada tiempo posible) o incluso el económico [14,15], entre otros.

* Trabajo apoyado parcialmente por el proyecto TIN2012-39391-C04-04.

Un tipo de algoritmos que se están parelizando con éxito durante los últimos años son los algoritmos evolutivos [6,20,1,12,5]. En ellos, un conjunto de entidades sencillas interactúan entre sí de manera local y conforme a un conjunto de reglas simples, de forma que dicha interacción conduce al conjunto al descubrimiento de soluciones eficientes para cierto problema considerado. Dado que la evolución de cada una de dichas entidades es semi-independiente (y su relación con las demás entidades es sencilla o puede acotarse fácilmente), estos algoritmos son susceptibles de ser paralelizados eficientemente.

A pesar de que este tipo de sistemas paralelos o distribuidos encajan en el tipo de análisis provistos por los modelos formales mencionados antes, ninguno de dichos modelos se ajusta específicamente al estudio del tipo de propiedades que nos interesan sobre los algoritmos evolutivos. Aunque hay muchas evidencias empíricas de las propiedades de optimización de los algoritmos genéticos (GAs), durante mucho tiempo hubo muy pocos resultados teóricos que probaran su éxito. El primer modelo matemático de los GAs fue propuesto por J.H. Holland en [9]. Sin embargo, este modelo no era capaz de dar una explicación completa de lo que sucede en una búsqueda empleando GAs según algunos investigadores. Años más tarde, se comenzó a modelizar el comportamiento de los GAs usando cadenas de Markov [13,19,17,21]. Por ejemplo, en [13] se modeló un GA con una cadena de Markov para calcular la matriz de transiciones de un algoritmo genético simple. Más tarde, en [17] se usaron cadenas de Markov para demostrar que variantes del GA canónico que usan un tipo de estrategia elitista siempre convergen al óptimo global. De este modo, quedó probada la utilidad de este modelo para modelizar GAs.

En [4,3] se presentó un modelo basado en cadenas de Markov para predecir los efectos que tendrán los parámetros de los GAs paralelos en la calidad de las soluciones encontradas. En el algoritmo considerado en el estudio presentado, la migración se realiza cuando la población ha convergido a la misma solución, provocando la reinicialización del algoritmo; y acaba cuando todas las poblaciones convergen a la misma solución. Los resultados obtenidos muestran que ejecuciones suficientemente largas de GAs paralelos obtienen soluciones de la misma calidad que GAs simples. Además, se exponen observaciones que ayudan a entender cómo afectan al algoritmo diferentes topologías y ratios de migración.

En el presente artículo desarrollamos una primera aproximación para definir un *sistema de transiciones etiquetadas probabilistas*, con el objetivo de permitir especificar y estudiar sistemáticamente la paralelización de los algoritmos genéticos (AG) [7]. Estos algoritmos se paralelizan eficientemente dividiendo la población total de individuos (cromosomas) en distintos subconjuntos y asignando cada subconjunto a un procesador diferente, en donde el subconjunto evolucionará de manera independiente durante cierto número de turnos de ejecución. Cada una de estas divisiones es una *isla* paralela. No obstante, cada cierto número de iteraciones los cromosomas de las diversas islas se influencian entre sí, lo que permite que, a largo plazo, todas las islas converjan hacia la mejora común de la solución. El método formal probabilístico presentado en este artículo tendrá como objetivo la modelización de dicho tipo de algoritmo evolutivo paralelo.

2. Sistema Básico

En esta sección presentamos nuestro modelo formal para representar la evolución de algoritmos genéticos con múltiples islas. Utilizaremos para ello un sistema de transiciones etiquetadas probabilista.

Definición 1 *Un sistema de transiciones etiquetadas probabilista (STEP) es una terna* $(S, \mathrm{Act}, \rightarrow)$ *donde cada una de las componentes representa lo siguiente:*

- S *es un conjunto de estados,*
- Act *es un onjunto de acciones,*
- $\rightarrow \subseteq S \times (\mathrm{Act} \times [0,1]) \times S$ *es el onjunto de transiciones.* □

Por comodidad, utilizaremos la notación $s \xrightarrow{\alpha}_p s'$, en lugar de $(s, (\alpha, p), s')$, para expresar que, con probabilidad p, el estado s evoluciona hacia al estado s' tras realizar la acción α. En un AG, cada estado de un STEP representará una población (un conjunto de cromosomas), la acciones representarán las maneras de evolucionar el AG, y la probabilidad asociada con la transición denotará la probabilidad de que el AG tome, desde el estado origen de la transición, dicha transición en lugar de las demás disponibles. Por simplicidad, las transiciones que tengan una probabilidad igual a 1 serán a veces denotadas sin mostrar dicha probabilidad.

La funciones de adecuación (*fitness*), inicialización, cruce y mutación utilizadas en un AG se definen específicamente para cada problema en función de las características de dicho problema. Por ese motivo, en nuestro esquema de formalización de AG dichas funciones estarán representadas de manera genérica. Existen diversas maneras de comenzar un AG con la acción de inicialización. Por ejemplo, podemos partir de una población aleatoria o bien comenzar con una solución obtenida aplicando otro método (e.g. un método voraz) y quizás también con un conjunto de variaciones de la misma. Además, para la función de cruce también existen diversas definiciones posibles, por ejemplo en función de cuántos puntos de cruce se consideren a la hora de mezclar los cromosomas. Incluso en el caso de la función de mutación, tenemos que decidir cuántos inidividuos mutaremos y de qué forma. Es más, en general dichas funciones se comportan de manera no determinista: cada ejecución con una misma entrada puede dar un resultado diferente. Desde el punto de vista de nuestro esquema general, únicamente requeriremos que las funciones a considerar se adapten a los tipos propuestos, y que estén definidas de manera probabilista, es decir, que cuantifiquen la probabilidad de cada resultado posible a partir de una misma entrada.

Definición 2 *Un cromosoma (o individuo) se define como una m-tupla de enteros* $(c_1, c_2, \cdots, c_m) \in \mathbb{Z}^m$ *que representa la información genética de un individuo. Denotamos con* Q *al conjunto de todos los individuos posibles. Usaremos* $x_1, x_2 \cdots$ *para denotar elementos dentro del conjunto* Q*.*

Una población se define como un multiconjunto $\{\!\{ x_1, \cdots, x_n \}\!\}$ *de individuos. Asumimos que* Pop *denota el conjunto de todas las poblaciones posibles.* □

Utilizamos multiconjuntos para denotar poblaciones, pues las poblaciones podrían contener varias instancias del mismo individuo, lo cual no podría representarse utilizando conjuntos.

Definición 3 *Un estado de un algoritmo genético se define por medio de una tupla de elementos* $GA = (\text{pop}, \text{nS}, \text{nG})$ *donde:*

- $\text{pop} = \{\!\!\{ x_1, \cdots, x_n \}\!\!\} \in \text{Pop}$ *es la población actual del AG.*
- nS *es el número de individuos de la población que serán seleccionados en cada fase de selección.*
- nG *es el número de generaciones (iteraciones) restantes para finalizar la ejecución del AG.*

Denotaremos como S *el conjunto de estados de un AG.* □

Ahora que ya tenemos los estados de un STEP, debemos definir sus acciones. El conjunto de acciones consistirá en:

- La *acción de inicialización*, denotada como init, que crea una población inicial.
- La *acción de evolución*, denotada como evol, que produce una nueva generación (una nueva población) basándose en dos acciones consecutivas:
  - La función de *selección*, denotada como $fitSel$, que, aplicada a una población, devuelve un conjunto de pares. Cada uno de estos pares consiste en una población de individuos ordenada y una probabilidad de que dicha población ordenada sea seleccionada. La ordenación será utilizada para escoger la manera en que los individuos de la población se cruzarán en la reproducción.
  - La función de *reproducción*, denotada como *reprod*, que, aplicada a cada par de individuos, devuelve un multiconjunto de individuos donde están incluidos el par inicial (los progenitores) y otros pares (su progenie).
- La acción de *parada*, denotada como stop. Esta accción se ejecutará para terminar el proceso de evolución, cuando no se puedan realizar más acciones. Cuando sea ejecutada, el estado alcanzado será un estado especial en el que lo único relevante de la terna que representa el estado será su multiconjunto de individuos (que representará la población de soluciones alcanzada por el AG). En dicha terna, el número de generaciones restantes será 0.

Por tanto el conjunto de acciones de un AG será $\text{Act} = \{\text{init}, \text{evol}, \text{stop}\}$.

Antes de definir las reglas de evolución de un AG, necesitamos definir un STEP más sencillo para definir el comportamiento de la acción de *reproducción*. Esta acción también necesita dos funciones auxiliares, *cruce* y *mutación*, definidas a continuación.

Definición 4 *Sea* $Pop_2 \subseteq Pop$ *el conjunto de multiconjuntos de individuos con multiplicidad 2 (es decir, cada multiconjunto contiene dos individuos diferentes o bien dos instancias del mismo individuo) y sea* $act = \{reprod\}$ *un conjunto de acciones. Definimos el STEP de la reproducción para pares de individuos como la terna* $(Pop, act, \longrightarrow)$ *donde* $\longrightarrow \subseteq Pop_2 \times (act \times (0, 1]) \times Pop$ *se describe en la Tabla 1.* □

$$\frac{(\{\!|\ x_1,x_2\ |\!\},p)\in crossOver(x_1,x_2)}{\{\!|\ x_1,x_2\ |\!\} \xrightarrow{reprod}_p \{\!|\ x_1,x_2\ |\!\}} \quad \text{(Reprod1)}$$

$$\frac{(\{\!|\ y_1,\cdots,y_r\ |\!\},p)\in crossOver(x_1,x_2),\ (z_i,q_i)\in mutation(y_i),1\leq i\leq r}{\{\!|\ x_1,x_2\ |\!\} \xrightarrow{reprod}_{p\cdot\Pi_{1\leq i\leq r} q_i} \{\!|\ x_1,x_2,z_1,\cdots,z_r\ |\!\}} \quad \text{(Reprod2)}$$

Tabla 1. Reglas Reprod1 y Reprod2

El STEP así definido hará uso de cierta función $crossOver : Q \times Q \longrightarrow \mathcal{P}(Pop \times [0,1])$ que definirá ómo deseamos que se realice el cruce en el AG. Dependiendo de cómo se defina esta función, la función podrá devolver un número diferente de hijos. Además, utilizará cierta función $mutation : Q \longrightarrow \mathcal{P}(Q \times [0,1])$ que definirá cómo deseamos que el AG realice la mutación. Dicha función se aplica a un individuo y devuelve un conjunto de pares del tipo $Q \times [0,1]$ que denotan la probabilidad de que el resultado de la mutación sea cada individuo posible. En particular, consideraremos que $(y,p) \in mutation(x)$ si y solo si x muta a y con probabilidad p.

Las reglas mostradas en la Tabla 1 describen la reproducción de dos individuos, donde se realiza el cruce y la mutación. Cada par de invididuos tiene dos opciones: cruzarse o no. En el último caso, no hay nada más que hacer. Sin embargo, si hay cruce, entonces su descendencia puede, a su vez, sufrir una mutación o no. La primera regla de la tabla representa cuándo no hay reproducción (no hay cruce). En este caso, los individuos permanecen inalterados. La segunda regla representa la reproducción de dos individuos, donde primero se realiza el cruce y después, posiblemente, se introducen ciertas mutaciones sobre la progenie.

La evolución de una población representará la evolución de todos los individuos de la población. El STEP utilizado para describir el comportamiento de una población completa se define de la siguiente manera:

Definición 5 *La evolución de un algoritmo genético es un STEP* $(S, \text{Act}, \rightarrow)$ *tal que* S *es el onjunto de estados del algoritmo genético,* Act *es el onjunto de acciones* $\{\text{init}, \text{evol}, \text{stop}\}$*, y la transición* $\rightarrow \subseteq S \times (\text{Act} \times (0,1]) \times S$ *se describe conforme a las reglas mostradas en la Tabla 2.* □

El valor $nMaxGen$, que aparece en la primera regla, representa el máximo número de generaciones permitidas. En las dos últimas reglas, el predicado termination(s, nS, nG) devolverá `cierto` si el estado satisface el criterio de optimalidad en las soluciones alcanzadas o bien si el número de generaciones restantes es 0, y será `falso` en caso contrario. Este predicado, $nMaxGen$, y las funciones $mutation : Q \longrightarrow \mathcal{P}(Q \times (0,1])$ y $crossOver : Q \times Q \longrightarrow \mathcal{P}(Pop \times (0,1])$, ya mencionadas antes, se definirán en función del AG concreto que deseemos modelizar.

$$\frac{(\text{pop},p)\in\text{init}}{(\emptyset,nS,0)\xrightarrow{\text{init}}_p(\text{pop},nS,nMaxGen)}$$

$$\frac{\neg termination(\{\!|\, x_1,\cdots,x_n \,|\!\},nS,nG)\ \wedge\ ([z_1,\cdots,z_{nS}],p)\in \text{fitSel}(\{\!|\, x_1,\cdots,x_n \,|\!\},nS)\ \wedge\ 0<i\leq nS\ div\ 2, \{\!|\, z_{2i-1},z_{2i} \,|\!\} \xrightarrow{reprod}_{p_i} \{\!|\, y_{i1},\cdots,y_{in_i} \,|\!\}}{(\{\!|\, x_1,\cdots,x_n \,|\!\},nS,nG)\xrightarrow{evol}_{(\Pi p_i)\cdot p}(\{\!|\, y_{11},\cdots,y_{1n_1},\cdots,y_{r1},\cdots,y_{rnr} \,|\!\},nS,nG-1)}$$

$$\frac{\text{termination}(\text{pop},nS,nG)}{(\text{pop},nS,nG)\xrightarrow{\text{stop}}(\text{pop},nS,0)}$$

Tabla 2. Reglas para la evolución del AG

En la Tabla 2 la primera regla representa la inicialización del AG. La función $\text{init} : \mathcal{P}(\text{Pop}\times(0,1])$ creará un conjunto de pares de población y probabilidad

$$\{(\text{pop}_1,p_1),\cdots,(\text{pop}_n,p_n)\}$$

donde $\text{pop}_i\in\text{Pop}$, $p_i\in(0,1]$, y $\sum_{i=1}^n p_i=1$. Cada par (pop,p) del conjunto representa que la población pop será creada con probabilidad p. Por tanto, la primera regla representa que, desde el estado inicial, el proceso puede evolucionar a la población inicial *pop* dada por la función init con probabilidad p, suponiendo que $(pop,p)\in\text{init}$. Además, el número de generaciones restantes se establece al máximo, es decir, $nMaxGen$. Después de cada evolución (siguiente regla), este número se decrementa en 1.

El estado inicial evolucionará a un estado s en su primer paso con la probabilidad que indique la primera regla. La segunda regla representa la evolución del AG. Esta evolución es posible solo si el número de generaciones restantes es mayor que 0 y no se ha alcanzado un estado donde la optimalidad requerida ya se haya obtenido (cosa que comprueba la función *termination*). En cada proceso de evolución se realizará una selección de los mejores individuos en términos de su adecuación, un cruce entre individuos y la mutación de varios de ellos. Asumiendo que $lists(A)$ denota el conjunto de todas las listas ordenadas que pueden formarse con individuos de A (en las que cada individuo puede aparecer cualquier número de veces), la función $\text{fitSel} : \text{Pop}\times\mathbf{N}\longrightarrow\mathcal{P}(lists(\text{Pop})\times(0,1])$ asignará una probabilidad a cada forma de ordenar los individuos de una población dada considerando sus valores de adecuación. En particular, el algoritmo considerará que los pares de progenitores a cruzar serán los pares de individuos consecutivos en esa lista: primero con segundo, tercero con cuarto, etc. Algunos individuos podrán aparecer más de una vez si son elegidos como progenitores más de una vez. Por tanto, tenemos

$$\text{fitSel}(\{\!|\, x_1,\cdots,x_n \,|\!\},nS)=\{(listpop_1,p_1),\cdots,(listpop_m,p_m)\}$$

donde $\sum_{i=1}^m p_i=1$. Todas las listas contendrán nS individuos. Por tanto, la función devolverá un conjunto de pares de lista de individuos y probabilidad,

$(c_1, \cdots, c_m) \in \mathbb{Z}^m$	: cromosoma o individuo
Q	: conjunto de todos los individuos
$x_1, x_2, \cdots \in Q$	: representan individuos
$\{\!\!\{ x_1, \cdots, x_n \}\!\!\}$	: población (multiconjunto de individuos)
Pop	: conjunto de todas las poblaciones
$pop, pop_1 \in Pop$	: representan poblaciones
$GA = (pop, nS, nG)$	: estado de un AG
nS	: número de individuos de la población
nG	: número de generaciones restantes
$(S, \mathrm{Act}, \longrightarrow)$	: evolución de un AG
S	: conjunto de todos los estados de un AG posibles
$s_1, s_2, \cdots \in S$	: representan estados de un AG
$\mathrm{Act} = \{init, evol, stop\}$	: conjunto de acciones
$nMaxGen$	: número máximo de generaciones permitidas

Tabla 3. Resumen de notación utilizada

donde

$$(listpop, p) \in \mathrm{fitSel}(\{\!\!\{ x_1, \cdots, x_n \}\!\!\}, nS)$$

representa que con probabilidad p se escogerá la lista $listpop$ (que contendrá nS instancias de individuos).

La última regla de la Tabla 2 representa la situación en la que no pueden realizarse más evoluciones porque el estado satisface el criterio requerido o bien porque el número de generaciones restantes es 0. En ambos casos, la única acción que puede realizarse es stop.

Una vez definida la evolución de un AG, formalizamos un proceso de evolución análogo para poblaciones independientes que evolucionan en *paralelo*. Por simplicidad, comenzamos presentando el caso en el que simplemente existen dos islas independientes. Inicialmente dichas islas evolucionan en paralelo sin ninguna comunicación, hasta que alcanzan un estado de terminación local. Cuando eso ocurre, ambas poblaciones intercambian algunos individuos, y entonces la evolución independiente de ambas islas se reanuda.

El resultado de la composición paralela de dos sistemas definidos por los STEP $A_1 = (S_1, \mathrm{Act}_1, \rightarrow_1)$ y $A_2 = (S_2, \mathrm{Act}_2, \rightarrow_2)$ es un nuevo STEP $A_1 \| A_2 = (S_1 \times S_2, \mathrm{Act}, \rightarrow)$. En lugar de representar sus estados compuestos de su forma de par $(s_1, s_2) \in S_1 \times S_2$, escribiremos $s_1 \| s_2$ para representarlos. Las transiciones en $\rightarrow$ son las que se infieren de las reglas detalladas en la Tabla 4.

La primera regla de la Tabla 4 define la evolución en paralelo de dos poblaciones. Las dos reglas siguientes representan la situación en la que una de las poblaciones puede evolucionar de manera independiente, mientras la otra ya ha satisfecho su condición de terminación local. En este caso, la isla que aún puede evolucionar lo hace, mientras que la otra isla espera hasta que ambas se encuentren en condición de terminación local. La cuarta regla representa la situación en la que ambas poblaciones han terminado sus evoluciones independientes. En tal situación, ambas podrán proseguir su evolución intercambiando parte de sus in-

$$\frac{s_1 \xrightarrow{evol}_p s_1',\ s_2 \xrightarrow{evol}_q s_2'}{s_1 \| s_2 \xrightarrow{evol}_{p \cdot q} s_1' \| s_2'}$$

$$\frac{s_1 \xrightarrow{evol}_p s_1',\ \text{termination}(s_2)}{s_1 \| s_2 \xrightarrow{evol}_p s_1' \| s_2}$$

$$\frac{\text{termination}(s_1),\ s_2 \xrightarrow{evol}_q s_2'}{s_1 \| s_2 \xrightarrow{evol}_p s_1 \| s_2'}$$

$$\frac{\text{termination}(s_1),\ \text{termination}(s_2),\ \text{iteration}(s_1,s_2)}{s_1 \| s_2 \xrightarrow{exchange}_{p \cdot q} s_1' \| s_2'}$$

$$\frac{\text{termination}(s_1),\ \text{termination}(s_2),\ \neg\text{iteration}(s_1,s_2)}{s_1 \| s_2 \xrightarrow{stop} s_1 \| s_2}$$

donde $s_1' = (exch_1(p_1, p_2), nS, nMaxGen)$ y $s_2' = (exch_2(p_1, p_2), nS, nMaxGen)$.

Tabla 4. Reglas para la evolución paralela de un AG

dividuos y recomenzando el proceso de evolución independiente de nuevo. Ahora bien, dicho intercambio y reinicio solo podrá realizarse si se cumple el predicado $iteration(s_1, s_2)$ que controla el momento de finalización de la evolución paralela. Si no se cumpliera el predicado, se aplicaría la última regla de la tabla, y el proceso paralelo terminaría.

Sean $p_1 = \{\!|\ x_1, \cdots, x_n\ |\!\}$ y $p_2 = \{\!|\ y_1, \cdots, y_m\ |\!\}$ dos poblaciones, sean $m', n' \in \mathbb{N}$ tales que $m' < m$ y $n' < n$, y sean $\{\!|\ x_{i_1}, \cdots, x_{i_{n'}}, \cdots, x_{i_n}\ |\!\}$ y $\{\!|\ y_{j_1}, \cdots, y_{j_{m'}}, \cdots, y_{j_m}\ |\!\}$ dos permutaciones aleatorias de p_1 y p_2 respectivamente. Entonces, la función $exch : Pop \times Pop \longrightarrow Pop \times Pop$ se define como $exch(p_1, p_2) = (p_1', p_2')$ donde $p_1' = \{\!|\ x_{i_1}, \cdots, x_{i_{n'}}, y_{j_{m'+1}}, \cdots, y_{j_m}\ |\!\}$ y $p_2' = \{\!|\ y_{j_1}, \cdots, y_{j_{m'}}, x_{i_{n'+1}}, \cdots, x_{i_n}\ |\!\}$.

Destaquemos que el método que hemos presentado para combinar dos islas paralelas puede aplicarse varias veces para combinar sistemas con un mayor número de islas. Por ejemplo, dadas tres islas A_1, A_2 y A_3, el sistema que combina todas ellas es $(A_1 \| A_2) \| A_3$.

A continuación definimos el comportamiento de un AG a varios pasos de ejecución vista. Una secuencia de pasos de ejecución será llamada *traza*.

Definición 6 *Consideremos el STEP de evolución de un algoritmo genético* $E = (\mathrm{S}, \mathrm{Act}, \longrightarrow)$. *Sea* $s' \in \mathrm{S}$ *y sea* $\rho \in \mathrm{Act}^*$ *la secuencia de acciones* $\rho = \alpha_1 \cdots \alpha_r$ *con* $\alpha_1, \cdots, \alpha_r \in \mathrm{Act}$. *Decimos que* E *produce la traza* $(\emptyset, nS, 0) \overset{\rho}{\Longrightarrow}_p s'$ *si existen* $s_1, \cdots, s_r \in \mathrm{S}$ *y* $p_1, \cdots, p_r \in (0, 1]$ *tales que* $\Pi_{i=1}^r p_i = p$ *y*

$$(\emptyset, nS, 0) \xrightarrow{\alpha_1}_{p_1} s_1 \xrightarrow{\alpha_2}_{p_2} \cdots s_{r-1} \xrightarrow{\alpha_r}_{p_r} s_r = s'.$$

Una traza es ompleta si $\alpha_r = \text{stop}$ *y* $s' = (pop, nS, 0)$ *on* $pop \in \text{Pop}$. □

Dado que la probabilidad de una traza es la multiplicación de la probabilidad de todos los pasos en la traza, nuestro modelo permite calcular la probabilidad de que un AG obtenga una población dada por un camino considerado. Es más, sumando la probabilidad de todas las trazas que dan lugar a la misma población, podemos obtener la probabilidad de que un AG finalice su ejecución con una población dada. Utilizando la misma idea, podemos sumar la probabilidad de todas las poblaciones en las cuales el mejor individuo sea uno determinado, y de esta forma calcular la probabilidad de que un AG devuelva, como resultado final, un determinado individuo.

Definición 7 *Consideremos el STEP de evolución de un algoritmo genético* $E = (\text{S}, \text{Act}, \longrightarrow)$. *Definimos la probabilidad de que* E *produzca al final de su ejecución una población* $pop \in Pop$ *como:*

$$prob_{pop} = \sum \{\!|\ p \mid \exists\ \rho : (\emptyset, nS, 0) \stackrel{\rho}{\Longrightarrow}_p\ (pop, nS, 0)\ |\!\}$$

Sea best : Pop $\longrightarrow Q$ *una función tal que, para toda* $pop \in Pop$, $best(pop)$ *es el individuo con mayor valor de adecuación en pop. Definimos el conjunto de poblaciones finales en las que un individuo* $q \in Q$ *es la mejor solución como:*

$$pops_q = \{pop \mid q = best(pop)\ \wedge\ \exists\ \rho, p : (\emptyset, nS, 0) \stackrel{\rho}{\Longrightarrow}_p\ (pop, nS, 0)\}$$

Definimos la probabilidad de que el STEP E *devuelva al final de su ejecución la solución* q *omo:*

$$\sum_{pop \in pops_q} prob_{pop}$$

□

3. Conclusiones y Trabajo Futuro

En este artículo hemos presentado un modelo formal basado en sistemas de transiciones etiquetadas probabilistas para modelizar AG y su evolución, tanto en entornos secuenciales como en entornos paralelos basados en la utilización de islas paralelas. El modelo presentado es solo una primera aproximación de un trabajo en curso en el que nuestro objetivo es utilizar el sistema desarrollado para demostrar ciertas propiedades de los AG. Por ejemplo, estamos utilizando el modelo para demostrar que cierto conjunto de condiciones es suficiente para garantizar que un algoritmo evolutivo alcanzará la misma optimalidad en la resolución de dos problemas NP-completos diferentes, lo que supone la continuación formal de nuestros estudios empíricos anteriores al respecto [16]. Dichas propiedades comparan el comportamiento del algoritmo para los dos problemas que se deseen considerar aplicando una *reducción polinómica* entre ambos, la cual existe necesariamente por ser ambos NP-completos.

Referencias

1. E. Alba. Parallel evolutionary algorithms can achieve super-linear performance. *Information Processing Letters*, 82(1):7 – 13, 2002.
2. S. Andova. Time and probability in process algebra. In *8th Int. Conf. AMAST, LNCS 1816*, pages 323–338, 2000.
3. E. Cantú-Paz. *Efficient and Accurate Parallel Genetic Algorithms*. Kluwer Academic Publishers, 2000.
4. E. Cantú-Paz. Markov chain models of parallel genetic algorithms. *IEEE Transactions on Evolutionary Computation*, 4(3):216–226, 2000.
5. A. de la Encina Encina, M. Hidalgo-Herrero, P. Rabanal, and F. Rubio. A parallel skeleton for genetic algorithms. In *Int. Work-Conference on Artificial Neural Networks, IWANN 2011*, volume 6692 of *LNCS*, pages 388–395. Springer, 2011.
6. A.E. Eiben and J.E. Smith. *Introduction to Evolutionary Computing*. Springer, 2003.
7. D.E. Goldberg. *Genetic Algorithms in Search, Optimisation and Machine Learning*. Addison-Wesley, 1989.
8. C.A.R. Hoare. *Communicating Sequential Processes*. Prentice Hall, 1985.
9. J.H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.
10. B. Jonsson, W. Yi, and K.G. Larsen. Probabilistic extensions of process algebras. In *Handbook of process algebra*, chapter 11. North Holland, 2001.
11. R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.
12. N. Nedjah, L. de Macedo-Mourelle, and E. Alba, editors. *Parallel Evolutionary Computations*, volume 22 of *Studies in Computational Intelligence*. Springer, 2006.
13. A.E. Nix and M.D. Vose. Modeling genetic algorithms with Markov chains. *Annals of Mathematics and Artificial Intelligence*, 5(1):79–88, 1992.
14. M. Núñez and I. Rodríguez. `PAMR`: A process algebra for the management of resources in concurrent systems. In *FORTE 2001*, pages 169–185. Kluwer Academic Publishers, 2001.
15. M. Núñez, I. Rodríguez, and F. Rubio. Specification and testing of autonomous agents in e-commerce systems. *Software Testing, Verification & Reliability*, 15(4):211–233, 2005.
16. P. Rabanal and I. Rodríguez. Using polynomial reductions to test the suitability of metaheuristics for solving np-complete problems. In *ACM Symposium on Applied Computing, SAC 2013*, pages 194–199. ACM Press, 2013.
17. G. Rudolph. Convergence analysis of canonical genetic algorithms. *IEEE Transactions on Neural Networks*, 5(1):96–101, 1994.
18. J. Springintveld, F. Vaandrager, and P.R. D'Argenio. Testing timed automata. *Theoretical Computer Science*, 254(1-2):225–257, 2001.
19. J. Suzuki. A markov hain analysis on a genetic algorithm. In *International Conference on Genetic Algorithms*, pages 146–154. Morgan Kaufmann, 1993.
20. K. van de Jong. Evolutionary computation: a unified approach. In *Genetic and Evolutionary Computation Conference, GECCO'08*, pages 2245–2258. ACM, 2008.
21. A.H. Wright and Y. Zhao. Markov chain models of genetic algorithms. In *Genetic and Evolutionary Computation (GECCO)*, pages 734–741. Morgan Kaufmann, 1999.

Experimentación con algoritmos distribuidos usando herramientas libres y gratuitas

Juan Julián Merelo Guervós, Maribel García Arenas, Pedro A. Castillo Valdivieso

Grupo GeNeura
Departamento de Arquitectura y Tecnología de Computadores
Universidad de Granada
(jmerelo,maribel,pedro)@geneura.ugr.es

Resumen En un entorno de restricción de costes para grandes instalaciones computacionales, acompañado de la existencia de herramientas en la nube y ordenadores de sobremesa de altas capacidades, la experimentación con algoritmos distribuidos se puede hacer fácilmente combinando ambas cosas. En este trabajo se presenta una metodología de experimentación con algoritmos genéticos distribuidos usando servicios de almacenamiento en la nube tales como Dropbox (o alternativas libres auto-instalables) y aplicaciones para gestión de máquinas virtuales tales como VirtualBox. Usando el almacenamiento en la nube como un sistema de intercambio de soluciones entre los diferentes nodos, se tratará de probar la aplicabilidad de esta metodología así como probar las capacidades de estos nuevos algoritmos evolutivos distribuidos.

1. Introducción

La computación paralela no necesita ser complicada y prever escenarios complejos o grandes variaciones de estructura en los programas secuenciales. La mayor parte de los ordenadores actuales pueden trabajar cómodamente con muchos procesos ejecutándose simultáneamente y poseen sistemas de almacenamiento rápido que pueden usarse para intercambiar información. Implementar un algoritmo que funcione de forma concurrente es, por lo tanto, tan simple como ejecutar varios procesos simultáneamente y que intercambien información a través de un directorio especialmente designado para ello. La eficiencia de la implementación no tiene por qué ser grande (y dependerá sobre todo del tipo de procesador, del número de núcleos que posea y de la velocidad y eficiencia del sistema de ficheros) pero, sin embargo, la simplicidad en la programación es tal que puede compensar la menor ganancia en velocidad obtenida de esta forma. Además, añadir nuevos nodos de computación al sistema paralelo distribuido resultará tan simple como ejecutar tantos procesos simultáneos como la CPU sea capaz de soportar.

Simultáneamente, está cada vez más vigente el uso de infraestructuras *nube* que permiten usar desde un ordenador conectado a la red diferentes recursos tales

como CPUs virtuales o discos duros virtuales. El hecho de que sean *virtuales* implica que aparezcan, desde el punto de vista del interfaz de programación, como si se tratara de recursos disponibles desde el sistema operativo. En la práctica, podemos usar un disco duro remoto situado en la nube como si se tratara de un disco duro local. De esta forma, también simple y transparente al programador podemos paralelizar un algoritmo, simplemente usando una infraestructura de almacenamiento virtual, siempre que ese almacenamiento virtual sea visible desde varios ordenadores y tengamos alguna forma de iniciar la ejecución de los procesos en las diferentes máquinas de forma más o menos simultánea. A la vez, en algunos casos estas infraestructuras son gratuitas, bien por el hecho de que formen parte de la misma organización (discos conectados a la red, NAS o bien infraestructuras creadas con OpenStack u OpenNebula [5]) o bien porque se trate de productos comerciales que poseen una versión gratuita, como se trata de Dropbox, Ubuntu One u otros.

En este trabajo mostramos la primera aproximación al uso de infraestructuras virtuales para la creación de experimentos de computación distribuida de forma gratuita; en este caso, para evitar problemas de sincronización, suponemos que desde un ordenador *central* se tiene acceso a todos los nodos. Los nodos, en realidad varias máquinas virtuales dentro de un solo anfitrión (lo que, a efectos prácticos, no tiene más importancia que el hecho de que se trate de nodos heterogéneos) comparten directorios mediante Dropbox. Estos experimentos los aplicaremos a un algoritmo genético canónico.

Desde el punto de vista de este trabajo, lo interesante de los algoritmos evolutivos es que se prestan a una fácil paralelización: simplemente dividiendo la población en grupos denominados *islas* [3] y creando algún mecanismo de intercambio de individuos entre ellas se pueden paralelizar. A este mecanismo se le suele denominar *migración* y puede hacerse de diferentes maneras: síncrona o asíncronamente y usando diferentes topologías, es decir, conexiones entre las islas. No es el objetivo de este trabajo revisar estos aspectos y su influencia sobre el resultado de la ejecución del algoritmo; simplemente adoptaremos el método que suponga un cambio menor sobre el código de un algoritmo genético secuencial clásico.

Esto es precisamente lo que vamos a hacer en este trabajo. Usaremos un algoritmo genético que, dividiendo su población en varias *islas*, intercambiará información a través de un directorio compartido. Ese directorio compartido podría estar sincronizado en un servicio tal como Dropbox, pero en este caso no haremos experimentos relacionados con esto, sino que nuestro objetivo es establecer el comportamiento de un algoritmo base con todos los procesos ejecutándose en el mismo ordenador. Trabajaremos además usando un sistema de fuentes abiertas en el que tanto el código como la experimentación como este mismo trabajo están a disposición de la comunidad científica desde el momento de su creación. Todos ellos se pueden descargar de `un.url.anoni.mo`.

Con este trabajo tratamos de demostrar que, usando un mecanismo de intercambio a través de almacenamiento, se pueden conseguir mejoras de velocidad incluso en un sólo ordenador; para ello probaremos desde uno a cuatro proce-

sos. Además, haremos ciertos experimentos preliminares que nos permitan saber qué políticas de migración son las más adecuadas.

El resto del trabajo se organiza como sigue: a continuación exponemos los resultados más sobresalientes en este área. Posteriormente explicamos el algoritmo y la metodología de experimentación usada en la Sección 3. Finalmente expondremos los resultados obtenidos y las conclusiones derivadas de los mismos para terminar con algunas notas de trabajo futuro en la Sección 4.

2. Estado del arte

Los principales resultados en este área han sido publicados recientemente [1,2,9]. En estos trabajos los autores usaron un algoritmo evolutivo implementado en Java para probar diferentes sistemas gratuitos de almacenamiento en nube tales como Dropbox u otros. Como principal resultado se obtuvo el hecho de que se conseguían escalados interesantes, pero sólo con unas pocas máquinas; el sistema se saturaba a partir de un número determinado de nodos. Además, el retraso en la propagación de los resultados de unas máquinas a otras implicaba que el problema de optimización debía de ser de cierto tamaño para que los resultados fueran significativos. El principal problema era que, debido a este retraso en la propagación, la migración introducía también un retraso en la ejecución del algoritmo para que esta fuera capaz de propagarse.

En este trabajo principalmente tratamos de reproducir los resultados obtenidos en el mismo con otro tipo de implementación usando un lenguaje de programación diferente y sin introducir retrasos cuando se hace migración y, además, usando otra implementación diferente que tiene como principal diferencia el hecho de no tener en cuenta si el algoritmo se está ejecutando secuencialmente o conjuntamente con otros nodos. Veremos en la sección siguiente estos detalles de implementación.

3. Detalles de implementación del algoritmo evolutivo y experimentos

Para hacer los experimentos se ha usado la librería `Algorithm::Evolutionary::Simple`, un módulo en Perl que permite crear un algoritmo genético rápidamente y en pocos pasos. Este módulo está optimizado para alcanzar velocidades aceptables [8] a pesar de tratarse de un lenguaje interpretado como el Perl. Este lenguaje, por otro lado, resulta un lenguaje bastante adecuado para trabajar, en general, con algoritmos evolutivos teniendo una variedad de herramientas ya desarrolladas [7] y resultando fácil la implementación de nuevas funciones, estructuras de datos u operadores. El hecho de trabajar con un lenguaje diferente, en todo caso, alterará las prestaciones de base, pero no tendría que tener tanta influencia sobre el escalado.

Para hacer los experimentos se ha buscado un problema que represente cierto reto para un algoritmo genético y cuya evaluación también requiera cier-

Cuadro 1: Valores de los parámetros del algoritmo genético y de la función P-Peaks usada.

Parámetro	Valor
Selección	Rueda de ruleta
Mutación	1-bit
Entrecruzamiento	2 puntos
P (número de picos)	256
b (bits del cromosoma)	512
Población base	1024

to tiempo, de forma que el algoritmo necesite un número de evaluaciones alto que pueda mejorarse a base de la paralelización. Por eso la función elegida ha sido P-Peaks [4]. En esta función se genera aleatoriamente un conjunto de p cadenas binarias de longitud b. P-Peaks devuelve la distancia de Hamming (es decir, número de bits diferentes) a la cadena *más cercana*, es decir, el mínimo de las distancias medida a todas las cadenas. La función resulta *pesada* porque hay que medir distancias a un número determinado de cadenas y resulta complicada para un algoritmo evolutivo al tener un número alto de máximos globales (correspondientes a cada una de las cadenas que se han generado). La implementación de esta función es también libre, está escrita en Perl y forma parte del módulo `Algorithm::Evolutionary` denominándose `Algorithm::Evolutionary::Fitness::P_Peaks`; este módulo es estándar y está incluido en la librería `Algorithm::Evolutionary` mencionada anteriormente: `http://search.cpan.org/~jmerelo/Algorithm-Evolutionary-0.78/lib/Algorithm/Evolutionary/Fitness/P_Peaks.pm`

Los parámetros base usados en el algoritmo evolutivo se muestran en la tabla 1. En general, son los valores por omisión de la librería. Sin embargo, como se ha comentado, P y b han sido elegidos para hacer el trabajo suficientemente difícil como para que cada ejecución del algoritmo dure un tiempo considerable. Por otro lado, tratándose de un problema difícil, se ha escogido un tamaño grande de población para que el algoritmo sea capaz de encontrar la solución; con tamaños más pequeños de población (o subdivisiones del tamaño) se vio que en muchos casos el algoritmo se quedaba estancado y era incapaz de encontrar la solución.

Para probar las prestaciones en paralelo del algoritmo se dividió la población entre dos y entre cuatro y se probó con un número igual de procesos. Los procesos se lanzaban desde un programa de línea de órdenes de Linux de forma que el inicio de los procesos es simultáneo (salvo tiempo de inicio de un proceso, que es muy bajo en todo caso). Había un proceso *principal* y otros *secundarios*; la principal diferencia es que este proceso *principal* decide cuando comienza y termina cada algoritmo y se usa también para medir la duración de los mismos. Esto puede significar que, si la solución se encuentra por primera vez en alguno de los procesos secundarios, el programa puede continuar durante un tiempo adicional; sin embargo, no suele demorarse mucho dado que se intercambian cro-

mosomas entre unos procesos y otros. El hacerlo así, además, evita que hagan falta mecanismos adicionales de comunicación del final que tengan que propagarse y simplifica la programación que, en realidad, es exactamente la misma para el programa secuencial y el paralelo. El tiempo de ejecución se mide simplemente a través de la diferencia en segundos entre el tiempo de creación de dos ficheros que se crean, precisamente, al comenzar y terminar este programa.

Para la versión paralela hace falta una *política de migración*. En nuestro caso se ha elegido guardar un cromosoma aleatorio elegido entre el 50 % con más fitness y tomar, a la vez, uno aleatorio del directorio en el que lo han depositado el resto. El elegir uno aleatorio entre los mejores coincide con la política que suele obtener mejores resultados en modelos isla, según hemos podido establecer en el pasado [6]. De hecho, algunas pruebas hechas depositando el mejor en cada generación ha dado peores resultados, provocando que en la mayor parte de los casos no termine el algoritmo. La estrategia aleatoria, aparte de rápida (no necesita ordenar ni hacer ninguna otra operación, salvo la necesaria para extraer los 50 mejores, para lo que no hace falta ordenar la población) tiene la ventaja de la estrategia de depósito de cromosomas: no necesita tampoco llevar a cabo ningún cambio en el código para el caso paralelo.

Tanto el código como el resultado de los experimentos (que analizaremos en la siguiente sección) están disponibles de forma abierta en la siguiente dirección: `https://code.launchpad.net/~jjmerelo/simplea/trunk`. El objetivo es que la comunidad científica se beneficie de esta ciencia abierta no sólo en los resultados, sino también en los datos que podrán ser, en caso de desearlo, analizados de forma independiente.

4. Resultados, conclusiones y trabajo futuro

Los experimentos se ejecutaron 10 veces. En todos los casos se encontró la solución, salvo en el caso en el que se hizo con cuatro nodos, en el que acabó sólo en 6 ocasiones. En todo caso, los resultados se muestran sobre los que efectivamente terminaron.

El primer resultado es que efectivamente, a pesar de ejecutarse en un sólo ordenador y cargar la tabla de procesos del mismo, se consigue una mejora en la velocidad con el número de *nodos*. Esto se muestra en las Figuras 1 y 2; en el primer caso los tiempos han sido tomados en un ordenador AMD con séxtuple núcleo ejecutando Ubuntu 12.04; en el segundo es un portátil Toshiba Portegé ejecutando el mismo sistema operativo y con un Intel i5 con cuatro núcleos de procesador; en este caso el disco duro es un SSD lo que lo hace, teóricamente, más rápido que en el primer caso.

Como se puede ver en ambas gráficas, el añadir nodos consigue rebajar el tiempo necesario para hallar la solución. De hecho, parte de esta mejora se debe al hecho de que se usen menos individuos en la población; pero una parte también se debe a que se están evaluando simultáneamente más individuos. De hecho, la mejora para cuatro nodos es de un 70 % en la velocidad en el primer caso, mientras que la mejora para dos nodos es de un 40 % en el segundo caso, algo

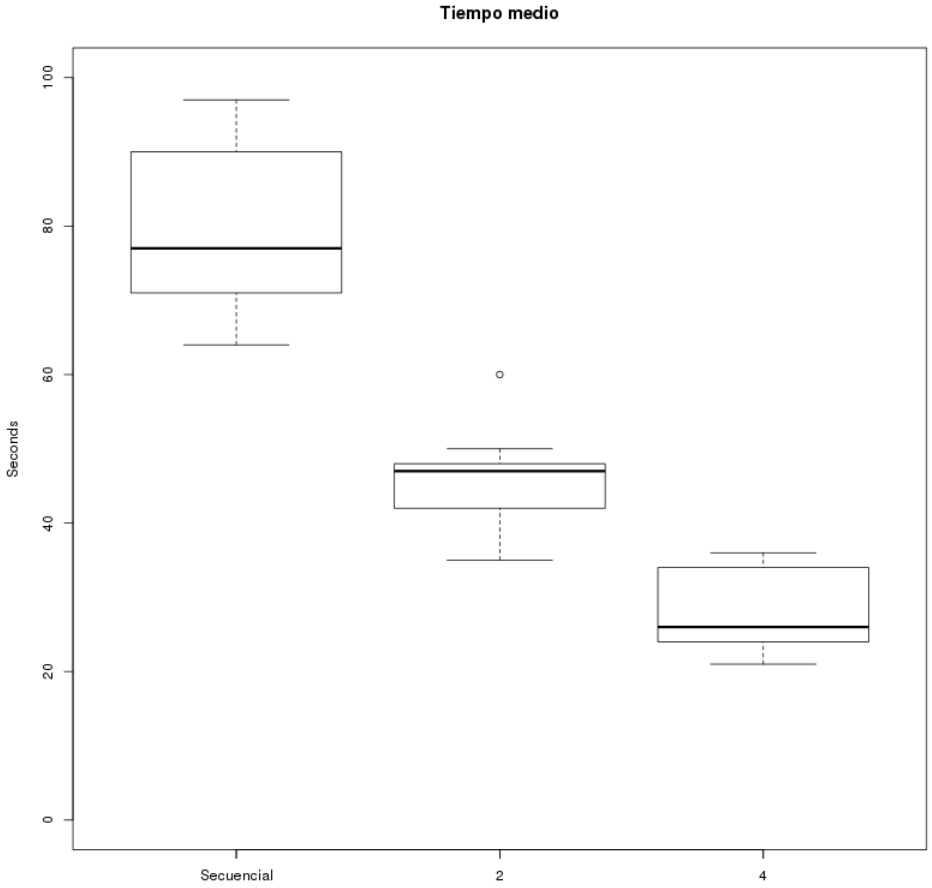


Figura 1: Boxplot del tiempo medio (en segundos) necesario para terminar el algoritmo usando un programa secuencial y dos y cuatro procesos simultáneos. Los tiempos han sido tomados en un ordenador de sobremesa.

mejor en el primer caso. Esto, en parte, puede deberse al hecho de que realmente no nos preocupamos de cuando halla la solución cualquier nodo, sino sólo cuando la halla uno de ellos; pero en parte también a que se nota la carga del sistema al ejecutar varios procesos simultáneamente, resultando excesiva tanto para la CPU como para la entrada/salida. De hecho, en el primer caso las tres diferencias son significativas usando el test no paramétrico de Wilcoxon, mientras que en el segundo caso la diferencia no es significativa. Esto puede significar que la mayor velocidad del disco duro haga que la diferencia entre uno y otro sea menor al dedicar menos tiempo al escribir en disco; esto podría indicar que uno de los factores que influyen en el tiempo es el tiempo empleado en leer del sistema

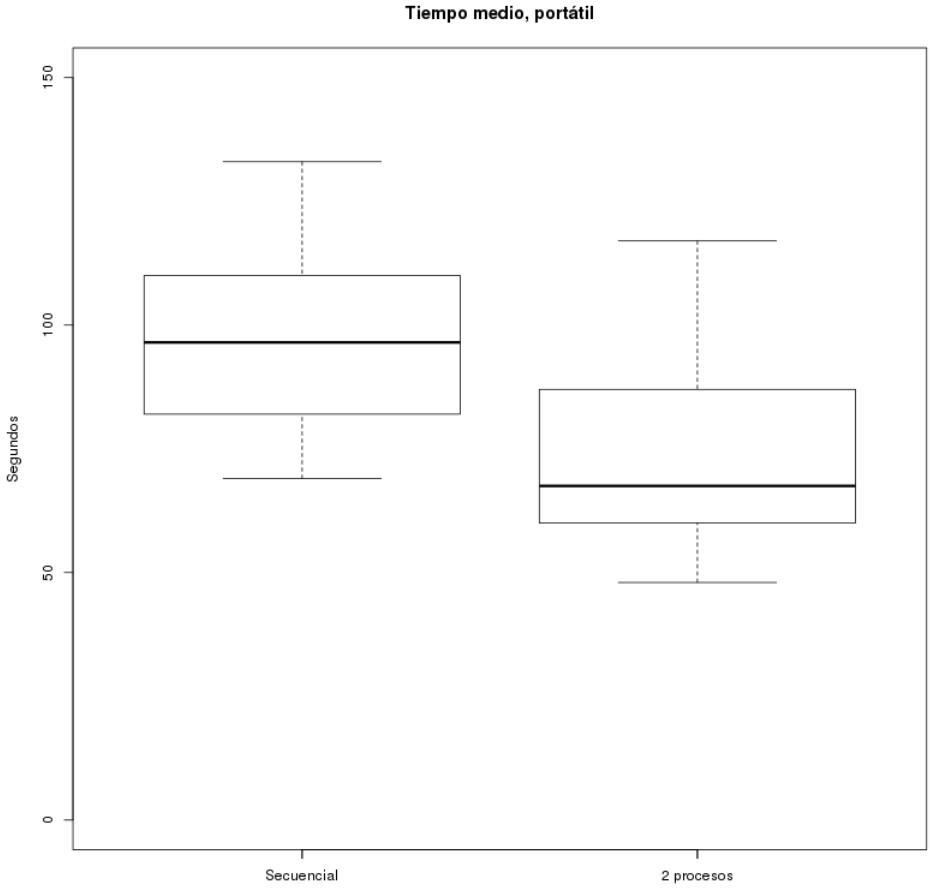


Figura 2: Boxplot del tiempo medio (en segundos) necesario para terminar el algoritmo usando un programa secuencial y dos procesos simultáneos en un ordenador portátil.

de ficheros. Habrá que evaluar mediante un profiler este tipo de hipótesis para probar si es cierta o no, aunque una evaluación preliminar indica que el tiempo invertido en leer y escribir del disco duro es tres órdenes de magnitud inferior al tiempo total del programa.

Por otro lado, en el primer caso se puede ver en la figura 3 la evolución procede de forma bien diferente dependiendo del número de procesos. En este caso lo que se traza para cada número de procesos es el tiempo de creación del fichero con el individuo que se está migrando tomado directamente del sistema de ficheros; el tiempo tiene resolución de segundos; esto es una ventaja adicional de usar este sistema, que te permite ver la evolución, con el tiempo, del fitness.

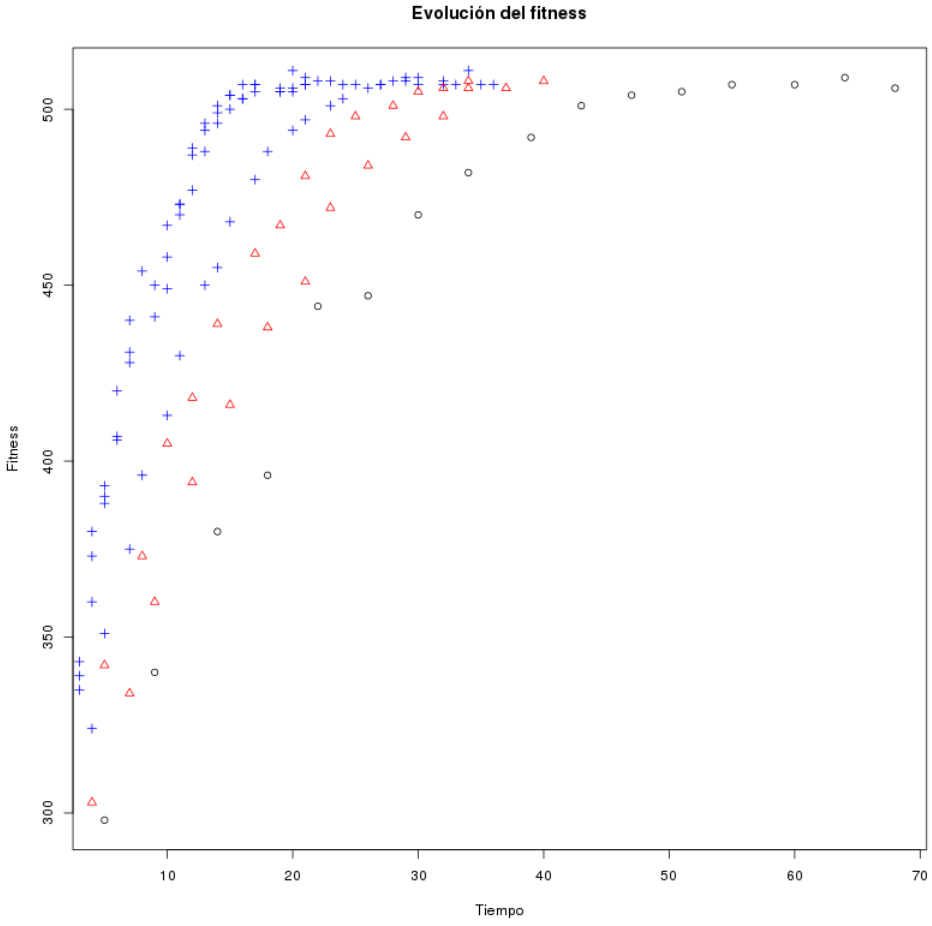


Figura 3: Evolución del fitness de una instancia de cada uno de los tres experimentos; los círculos indican el experimento con un solo proceso, los triángulos con dos y las cruces con 4 procesos. El tiempo en el eje de abscisas es el tiempo real; el fitness máximo es 512.

Si recordamos que en realidad el que se graba es un individuo aleatorio entre los 50 % mejores, vemos que, en todo caso y para un tiempo determinado, por ejemplo 10 segundos, cuando hay 4 nodos se ha avanzado mucho más que cuando se usan dos o un solo nodo; esto prueba que el algoritmo evolutivo se puede paralelizar usando este simple mecanismo que es, también, extensible a sistemas que permitan almacenamiento transparente en la nube como Dropbox, tal como queríamos probar.

Adicionalmente hemos hecho alguna prueba con directorios compartidos a través de Dropbox y alojados en diferentes máquinas virtuales dentro del mismo

ordenador. Esto presenta una serie de retos, el principal que la velocidad de tales máquinas virtuales va a ser muy diferente y continuando con el problema del retraso en la aparición de los ficheros individuales en el resto de los nodos. Sin embargo, algunas pruebas iniciales (que se pueden ver en el repositorio de código y datos indicados) indican que, aunque se consiguen ciertas mejoras al añadir un nuevo nodo, no está claro que sean significativas, por lo que hay que avanzar haciendo experimentos en este sentido, probando con diferentes configuraciones, máquinas virtuales y parametrización de las mismas. Esto es algo que se propone como trabajo futuro.

Este trabajo ha sido el comienzo de una línea de investigación en la que se pretenden montar máquinas paralelas con la infraestructura de la que se dispone en cualquier despacho o laboratorio. Utilizándo un multiprocesador (varias máquinas con espacio de direcciones distribuido unidas por una red de interconexión) y un espacio de almacenamiento en disco al que todos los procesadores pueden acceder, se ha reproducido el sistema de comunicación de un multiprocesador con espacio de direcciones compartido, probando que efectivamente la arquitectura es posible y es posible ejecutar un algoritmo evolutivo utilizando esta infraestructura, contando sólo con los recursos disponibles y un esquema de programación SPMD.

Los tiempos empleados en la comunicación van retardados por la necesidad de escribir en disco para que los procesos se comuniquen, y esa será nuestro siguiente objetivo, intentar simplificar al máximo la programación intentando evitar al máximo este tipo de operaciones, acercando más la arquitectura propuesta a una más eficiente.

Aunque los resultados obtenidos no son sorprendentes, nuestro objetivo no es superar la velocidad de las máquinas diseñadas para computación paralela, sino simplemente probar que existe la posibilidad de ejecutar en paralelo con los recursos mínimos una aplicación cualquiera.

Por otro lado, una vez más habrá que hacer un profiling extensivo de la aplicación para detectar los cuellos de botella y comprobar qué provoca retardos y en qué caso. También se comprobará con diferentes tipos de configuraciones que incluyan directorios sincronizados usando almacenamiento en la nube o directorios montados remotamente usando otro tipo de mecanismos. Eventualmente, lo que se pretende es crear un sistema que pueda distribuirse fácilmente entre diferentes ordenadores y crear experimentos distribuidos sin necesidad de ninguna compra ni de hardware ni de servicios.

5. Agradecimientos

Este trabajo está apoyado por los proyectos TIN2011-28627-C04-02 del Ministerio español de Ciencia y Competitividad y por el P08-TIC-03903 del gobierno regional andaluz, así como el proyecto CEI2013-P-14 (CANUBE) concedido por el CEI-BioTIC UGR (`http://biotic.ugr.es`).

Referencias

1. Maribel García Arenas, Juan J. Merelo Guervós, Antonio Miguel Mora, Pedro A. Castillo, Gustavo Romero, and Juan Luís Jiménez Laredo. Assessing speed-ups in commodity cloud storage services for distributed evolutionary algorithms. In *IEEE Congress on Evolutionary Computation*, pages 304–311. IEEE, 2011.
2. Maribel García Arenas, Juan Julián Merelo Guervós, Pedro A Castillo, Juan Luís Jiménez Laredo, Gustavo Romero, and Antonio Miguel Mora. Using free cloud storage services for distributed evolutionary algorithms. In Natalio Krasnogor and Pier Luca Lanzi, editors, *GECCO*, pages 1603–1610. ACM, 2011.
3. Juan J. Merelo Guervós, Antonio Miguel Mora, Carlos M. Fernandes, Anna Isabel Esparcia-Alcázar, and Juan Luís Jiménez Laredo. Pool vs. island based evolutionary algorithms: An initial exploration. In Fatos Xhafa, Leonard Barolli, and Kin Fun Li, editors, *3PGCIC*, pages 19–24. IEEE, 2012.
4. K.A. De Jong, M.A. Potter, and W.M. Spears. Using problem generators to explore the effects of epistasis. In *ICGA*, pages 338–345. Morgan Kaufmann, 1997.
5. Francisco Magaz Villaverde. Opennebula y hadoop: Cloud computing con herramientas open source, 2012.
6. Juan J. Merelo, Antonio M. Mora, Pedro A. Castillo, Juan L. J. Laredo, Lourdes Araujo, Ken C. Sharman, Anna I. Esparcia-Alcázar, Eva Alfaro-Cid, and Carlos Cotta. Testing the intermediate disturbance hypothesis: Effect of asynchronous population incorporation on multi-deme evolutionary algorithms. In Gunter Rudolph, Thomas Jansen, Simon Lucas, Carlo Poloni, and Nicola Beume, editors, *Parallel Problem Solving from Nature - PPSN X*, volume 5199 of *LNCS*, pages 266–275, Dortmund, 13-17 September 2008. Springer.
7. Juan-Julián Merelo. A Perl primer for evolutionary algorithm practitioners. *SIGEVOlution*, 4(4):12–19, 2010.
8. Juan-Julián Merelo-Guervós, Gustavo Romero, Maribel García-Arenas, Pedro A. Castillo, Antonio-Miguel Mora, and Juan-Luís Jiménez-Laredo. Implementation matters: Programming best practices for evolutionary algorithms. In Joan Cabestany, Ignacio Rojas, and Gonzalo Joya Caparrós, editors, *IWANN (2)*, volume 6692 of *Lecture Notes in Computer Science*, pages 333–340. Springer, 2011.
9. K Meri, MG Arenas, AM Mora, JJ Merelo, PA Castillo, P García-Sánchez, and JLJ Laredo. Cloud-based evolutionary algorithms: An algorithmic study. *Natural Computing*, 2013.

Relación entre la Estructura de un Problema y la Topología de Islas en Algoritmos Evolutivos Paralelos

Ignacio Arnaldo, Iván Contreras, David Millán-Ruiz,
Javier Nobel, J. Ignacio Hidalgo, and Natalio Krasnogor

[1] I. Arnaldo, I. Contreras, D. Millán-Ruiz, J. Nobel and J.I. Hidalgo
Parallel Architectures and Bioinspired Algorithms (PABA) Research Group
Universidad Complutense de Madrid, Spain
web:http:\\bioinspired.dacya.ucm.es
[2] N. Krasnogor
Interdisciplinary Computing and Complex Systems (ICOS) Research Group
School of Computer Science, University of Nottingham
web:http:\\icos.cs.nott.ac.uk
twitter:@NKrasnogor

Resumen Se ha demostrado que la estructura de la población de un Algoritmo Evolutivo Paralelo tiene un gran impacto en el proceso búsqueda. Sin embargo, ningún trabajo ha estudiado la correlación entre la estructura de la población utilizada y la estructura del problema que se intenta resolver. En este trabajo, se demuestra esta correlación mediante el estudio de un gran conjunto de topologías de islas (basadas en β-graphs) y una serie de problemas parametrizables NK-Landscape.

1. Introducción

El análisis de redes constituye una herramienta fundamental para entender las propiedades de procesos complejos presentes en campos como la biología, las ciencias sociales, sistemas tecnológicos o incluso la resolución de problemas [1]. En particular, las redes *small-world* descritas por Watts y Strogatz [2] generaron una gran interés ya que aparecen en un amplio abanico de áreas de investigación. En estas estructuras, aparecen grupos o clusters de nodos y, sin embargo, la información se filtra rápidamente a través de la red. Posteriormente, Barabási y Albert [3] sugirieron que la distribución de vértices con alta conectividad en redes tales como la WWW o las citaciones de publicaciones científicas no es aleatoria. De hecho, en estas redes denominadas *scale-free* la conectividad de los vértices sigue una distribución exponencial negativa, lo que implica que un conjunto reducido de vértices domina la conectividad de la red. Esta característica surge a consecuencia de los siguientes mecanismos: las redes se expanden continuamente mediante la anexión de nuevos vértices, y los nodos nuevos tienen mayor probabilidad de conectarse a nodos que ya tienen una alta conectividad.

Los avances en el estudio de redes también han tenido un gran impacto en el campo de la computación evolutiva, en concreto aplicados a los Algoritmos Evolutivos Paralelos (AEPs) y a los Algoritmos Evolutivos Celulares (AECs). En éstos, el intercambio de información entre individuos (a través de recombinación genética, políticas de migración o transmisiones meméticas) se rige por una determinada estructura de la población. Se ha demostrado que, en determinadas situaciones, el uso de poblaciones descentralizadas permite una mejor exploración del espacio de búsqueda mejorando tanto el resultado obtenido como el tiempo de ejecución. Por ejemplo, Cantú-Paz [4] establece una relación entre la topología de la red en un modelo de islas, el número de subpoblaciones, el ratio de migración, y la eficacia de un Algoritmo Evolutivo Paralelo. En efecto, estos parámetros permiten ajustar la presión selectiva y pueden acelerar la convergencia de la población. Posteriormente, en [5] se realizó un estudio de AECs con poblaciones estructuradas como grafos *small-world* y *scale-free*. Sus resultados mostraron que las topologías *small-world* presentan un buen balance entre robustez y velocidad de búsqueda. Sin embargo, las topologías *scale-free* no resultan apropiadas por problemas de convergencia prematura. De la misma manera, el trabajo presentado en [6] presenta un estudio sistemático del rendimiento de un AEP bajo diferentes topologías (población única, subpoblaciones inconexas, anillo, y subpoblaciones totalmente conexas). En este caso, la topología de anillo muestra el mejor comportamiento.

El estudio de la red subyacente en el proceso de optimización también se ha realizado en el contexto de otras metaheurísticas. Por ejemplo, Kennedy *et al.* [7] analizan el efecto de diferentes topologías en la optimización mediante PSO, demostrando que algunas redes aleatorias son capaces de lograr muy buenos resultados, mientras que las estructuras usadas comúnmente (totalmente conexa y de anillo) son soluciones subóptimas. En efecto, el proceso de búsqueda puede beneficiarse de algunas propiedades topológicas de esas estructuras aleatorias, dando como resultado un buen balance entre exploración y explotación. Li *et al.* proponen un extenso estudio del espacio de topologías [8], analizando el efecto de 1200 topologías de red en el contexto de la generación espontánea de programas. El trabajo citado estudia topologías de malla, redes *small-world* y redes completamente aleatorias. Se demuestra que tanto el *Clustering Coefficient* (Coeficiente de Clustering) como el *Average Path Length* (Camino Mínimo Medio) de la red tienen una gran influencia tanto en la calidad como en la diversidad de las soluciones obtenidas.

Esta sección recoge una serie de trabajos que proponen un análisis de redes en un contexto de optimización heurística. Queremos resaltar lo siguiente (a) la diversidad y la calidad de las contribuciones, y (b) aunque está fuera de toda duda que diferentes estructuras de la población inducen distintos rendimientos en la búsqueda, ningún trabajo ha evidenciado que *exista una correlación entre la estructura de población usada para resolver el problema y la estructura del problema*. Esta correlación es la que queremos verificar a través de un conjunto de experimentos simples y claros, basados en β-graphs (como topologías de islas

en un AEP), y problemas NK-landscapes ya que éstos presentan una red interna de interacciones epistáticas.

El resto del artículo se estructura como sigue: la Sección 2 explica la metodología de los experimentos. Detallamos la Configuración Experimental la Sección 3 mientras que los resultados se exponen en la Sección 4. Por último, la Sección 5 concluye el artículo.

2. Metodología

En esta sección describimos los benchmarks utilizados para estudiar la potencial correlación entre la estructura del problema y la topología usada en un Algoritmo Evolutivo Paralelo siguiendo un modelo de islas. Para ello utilizamos distintos problemas que presentan estructuras sin interacciones (Onemax) y con interacciones parametrizables (NK-Landscape).

2.1. ONEMAX Benchmark

El problema OneMax es un problema simple que consiste en encontrar la cadena de bits de un determinado tamaño que maximice el número de unos. Este problema se ha utilizado para evaluar el rendimiento tanto de algoritmos como de operadores genéticos (ver [9]). Desde un punto de vista formal, se trata de encontrar la cadena $\boldsymbol{x} = \{x_1, x_2, \ldots, x_N\}$ con $x_i \in \{0, 1\}$, que maximice la siguiente función:

$$F(\boldsymbol{x}) = \sum_{i=1}^{N} x_i \tag{1}$$

2.2. Problemas NK-Landscape

Los problemas NK-Lankscape se han utilizado de manera recurrente en el campo de la búsqueda evolutiva [10]. Este dominio de problemas es adecuado para tales labores ya que es posible obtener instancias de dificultad ajustable partiendo un modelo simple. Cada instancia se caracteriza por un número N de genes y un número K de interacciones entre genes tales que $0 < N, 0 \leq K \leq N-1$. Es importante recalcar que el problema del Onemax es equivalente a un problema NK-Landscape con K=0. Las interacciones epistáticas o de vecindad del bit i pueden ser elegidas aleatoriamente o de una manera regular seleccionando los K bits mas cercanos. La matriz E de tamaño $N \times 2^{K+1}$, almacena los resultados de las interacciones epistáticas de K bits. El objetivo del problema es maximizar el número de unos en una cadena de bits. Así, una solución candidata S es una cadena binaria de longitud N cuyo valor de fitness se calcula de la siguiente manera:

$$Fitness(S) = \frac{\sum_{i=1}^{N} f_i(S_i, S_{i_1}, \ldots, S_{i_K})}{N} \tag{2}$$

donde $f_i(.)$ es una casilla de E, S_i el valor del string S en la posición i y S_{i_j} es el valor de la cadena S en el vecino j del bit i.

En el caso de los NK-Landscape, la red de interacciones epistáticas determina la complejidad del problema. De hecho, el problema es resoluble en tiempo polinomial si se consideran solamente interacciones entre bits adyacentes. En cambio, si las interacciones se eligen de forma aleatoria, el problema es NP-duro [11]. Además, es posible ajustar la rugosidad del *landscape* o *paisaje* mediante el parámetro K. En el contexto de los Algoritmos Meméticos, Krasnogor y Gustafson [12,13] permitieron la evolución de buscadores locales y demostraron que la estructura de éstos se ajusta a la del problema a resolver. En este artículo queremos evaluar si se pueden relacionar distintas topologías de islas con problemas NK-Landscapes diferentes. Así, estudiamos cuatro escenarios diferentes: epistasia baja y resoluble en tiempo polinomial, epistasia alta y resoluble en tiempo polinomial, epistasia baja y problema NP-duro, y epistasia alta e instancia NP-dura del problema NK-Landscape.

2.3. Topologías $\beta-$graphs

Utilizamos la familia de los $\beta-$graphs como fuente sistemática de topologías con el fin de evaluar de manera exhaustiva si distintas configuraciones de islas son más adecuadas para determinadas estructuras del problema (es decir, distintos valores de K en los NK-Landscape e interacciones regulares o aleatorias).

La familia de los $\beta-$graphs engloba una gran variedad de estructuras de red, desde una topología ordenada, hasta un grafo aleatorio. Las propiedades del grafo generado bajo este modelo se determinan mediante tres parámetros: el número de vértices n, el número de conexiones por nodo k y el grado de aleatoriedad β. Por otra parte, las dos siguientes métricas se utilizan frecuentemente para analizar las propiedades de estos grafos:

- *Characteristic Path Length (CPL)* o Camino Mínimo Medio: esta métrica determina la distancia media entre dos nodos cualesquiera del grafo. Por lo tanto, es una medida apropiada para evaluar la velocidad de la difusión de la información en la red analizada.
- *Clustering Coefficient (CC)* o Coeficiente de Clustering: es la probabilidad de que dos nodos conectados mediante una arista tengan un vecino común. A grandes rasgos, proporciona una medida de la presencia de *clusters* o grupos de nodos altamente conectados entre sí mismos.

Hemos elegido los problemas $NK-$landscape y la familia de los $\beta-$graph porque su estructura interna guarda una notable similitud (ver Figura 2.3). Si *no encontramos una correlación* entre las topologías de islas obtenidas mediante $\beta-$graphs y la estructura de los problemas $NK-$Landscape, entonces será difícil justificar que *determinadas topologías de islas* son mejores para *determinadas estructuras de problemas*. Por otra parte, si estas dos estructuras pueden ser clara-

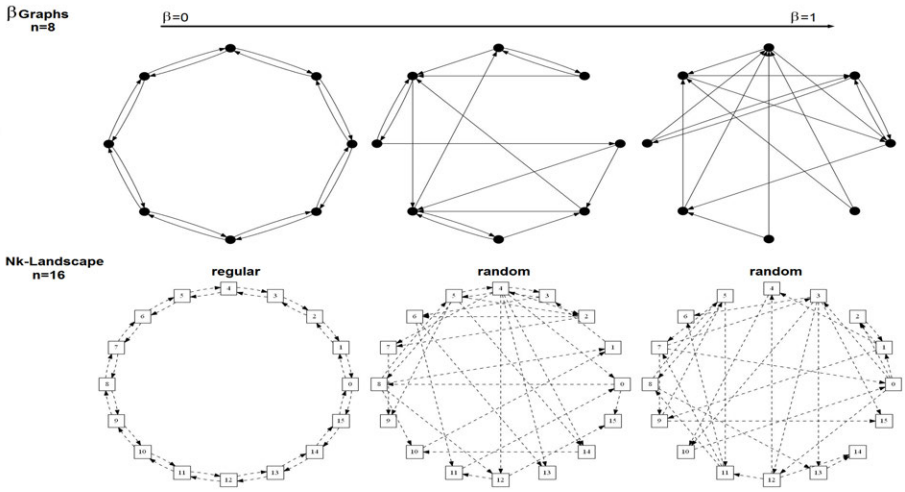


Figura 1. Representación gráfica de problemas $NK-$landscapes y $\beta-$graphs. La imagen superior muestra tres ejemplos de $\beta-$graphs con $n = 8$ y con $\beta = 0; 0{,}5; 1$. En la imagen inferior, se pueden apreciar tres problemas $NK-$landscape. Todas las instancias tienen $N = 16$ y $K = 2$. El problema de la izquierda tiene una estructura epistática regular, mientras que los otros dos presentan interacciones epistáticas aleatorias.

mente relacionadas, entonces se habrá iniciado una nueva línea de investigación para mejorar los procesos de optimización.

3. Configuración Experimental

Cada topología se define por una tupla (n,k,β) en los siguientes rangos:

$$n \in [1, 2, 4, 8, 16, 20]$$

$$k \in [0, \ldots, N-1]$$

$$\beta \in [0{,}05, 0{,}10, 0{,}15, \ldots, 0{,}90, 0{,}95, 1{,}00].$$

Por cada topología o tupla, generamos 20 grafos. Con estos parámetros, y eliminando las combinaciones redundantes, obtenemos un total de *851* topologías. Cada uno de estos grafos sirve de topología de islas para un AEP. Dicho algoritmo trata de resolver problemas NK-landscape fijando $N = 100$ y variando el valor de K y del grado de aleatoriedad de las interacciones R, tal que $K \in [2, 4, 8, 14, 28, 56]$ y $R \in [0, 1]$. En el caso del problema OneMax, usamos una instancia de 5000 bits (experimentos anteriores con instancias más pequeñas son fácilmente resueltas por un Algoritmo Genético simple). En resumen, analizamos 12 problemas NK-landscape y una instancia del problema del OneMax.

En ambos casos, se ejecuta un AEP 20 veces por topología (una por cada β-graph generado). En cada ejecución, la topología de islas se mantiene fija durante el proceso evolutivo. Los principales parámetros del algoritmo son:

1. *Población Inicial*: La población inicial del algoritmo se genera aleatoriamente.
2. *Selección*: Se emplea un Torneo Binario para seleccionar los padres. Los individuos seleccionados siempre sobreviven hasta la siguiente generación.
3. *Cruce*: La descendencia se genera mediante un operador de Cruce en un Punto. La probabilidad de cruce se fija en 0.9.
4. *Mutación*: No se considera ningún operador de mutación con el fin de evitar otra variable.
5. *Política de Migración*: Se fija una política de reemplazamiento en la que se traspasa el 10 % de la población de una isla a otra. Así, los mejores individuos de la isla de origen remplazan individuos elegidos al azar de la isla receptora (siempre preservando el elitismo).

4. Resultados

Como se explica en la sección anterior, realizamos un barrido paramétrico tanto en los problemas como en las topologías analizadas, realizando un total de 221260 experimentos, esto es, 20 réplicas por cada una de las 851 topologías y cada uno de los 13 problemas propuestos.

4.1. 851 topologías vs. 13 problemas

Para cada topología y problema, calculamos el valor medio del fitness del mejor individuo a partir de las 20 ejecuciones realizadas. Este valor se utiliza para clasificar las topologías (desde el mejor al peor rendimiento en términos relativos) para cada problema. Estas posiciones se almacenan en una matriz de 13 filas (problemas) y 851 columnas (topologías). Para analizar de forma concisa y clara los datos obtenidos, realizamos un biclustering de la matriz de resultados. La Figura 2 muestra el resultado del proceso de clustering.

En la figura se puede observar que determinados grupos de topologías de islas benefician (o perjudican) por igual a problemas similares. Por ejemplo, los problemas NK-Landscape regulares y con un valor de K bajo (NK2,NK4,NK8,NK14) han sido agrupados correctamente. Por otra parte, los problemas NK2R, NK4R y NK8R con interacciones aleatorias muestran un comportamiento similar con las 851 topologías analizadas. Cabe destacar que el NK28 ha sido agrupado con estos últimos debido a su mayor dificultad. Los problemas NK14R y NK28R muestran un comportamiento similar y presentan patrones verdes y rojos que no aparecían para problemas más sencillos. Esto significa que algunas topologías que funcionaban mal en los problemas anteriores obtienen buenos resultados en estos problemas y viceversa. Por último, se puede apreciar un grupo claramente

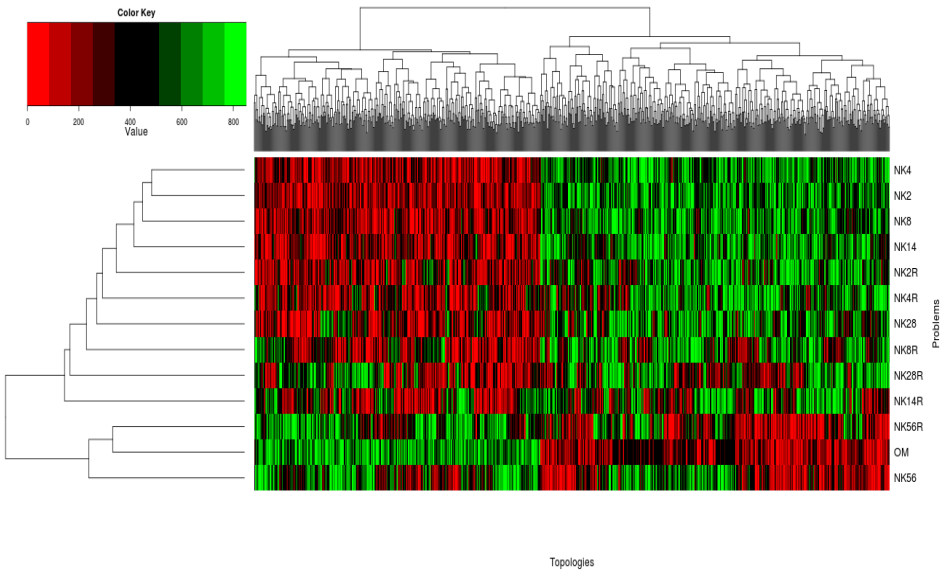


Figura 2. Biclustering de las 851 topologías (tuplas (n, k, β)) y problemas NK-Landscape con N=100 y OneMax con tamaño de cadena de 5000 bits. Para cada tupla se generan 20 grafos y se calcula el fitness medio de la resolución de un determinado problema con los grafos generados. A continuación, se ordenan todas las tuplas función de su puntuación en los diferentes problemas y se realiza el proceso de biclustering. Las casillas verdes indican la idoneidad de una topología para resolver un determinado problema mientras que las rojas indican un rendimiento pobre.

diferenciado compuesto por los problemas NK56, NK56R y OneMax. La asociación entre los dos primeros es intuitiva ya que ambos consideran un elevado número de interacciones epistáticas (56 en una cadena de 100 bits) anulando el efecto de la aleatoriedad.

4.2. Clustering de las topologías

A continuación proponemos un análisis de la relación entre las propiedades de las topologías y los problemas estudiados. Para ello, reordenamos las 851 topologías basándonos en sus valores de *Clustering Coefficient* y de *Characteristic Path Length*. Para realizar el clustering de las topologías, definimos la distancia entre dos β-Graphs g_1 y g_2 como:

$$\begin{aligned} D(g_1, g_2) = & \sqrt{|CC_{g_2} - CC_{g_1}|^2 + |CPL_{g_2} - CPL_{g_1}|^2} \\ & + max\{|CC_{g_2} - CC_{g_1}|, |CPL_{g_2} - CPL_{g_1}|\} \end{aligned} \tag{3}$$

Como se puede apreciar en la ecuación, calculamos la distancia euclídea basada en el CC y el CPL de las topologías comparadas y añadimos una penalización para valores muy dispares. Con esta distancia, distribuimos las topologías en 25

clusters o grupos del mismo tamaño. Cada grupo de topologías se caracteriza por la media del *Clustering Coefficient* ($\overline{CC}$) y la media de los *Characteristic Path Length* ($\overline{CPL}$) de las topologías que contiene. Los grupos obtenidos se muestran en la Tabla 1. El lector puede consultar más detalles de este procedimiento en [14].

Bin	*CC*	*CPL*
0	*0.9445*	*0.0031*
1	*0.8986*	*0.0056*
2	*0.8381*	*0.0091*
3	*0.7932*	*0.0117*
4	*0.7491*	*0.0145*
5	*0.7014*	*0.0179*
6	*0.6618*	*0.0203*
7	*0.7203*	*0.0236*
8	*0.6638*	*0.0244*
9	*0.6022*	*0.0277*
10	*0.4439*	*0.0309*
11	*0.4598*	*0.0343*
12	*0.4183*	*0.0391*
13	*0.2671*	*0.0451*
14	*0.4727*	*0.0491*
15	*0.2037*	*0.0577*
16	*0.2902*	*0.0710*
17	*0.1733*	*0.0821*
18	*0.0914*	*0.1143*
19	*0.0548*	*0.1446*
20	*0.0227*	*0.2100*
21	*0.0014*	*0.2833*
22	*0.0005*	*0.4937*
23	*0.0000*	*0.7160*
24	*0.0000*	*0.8301*

Cuadro 1. Distribuimos las 851 topologías en 25 grupos en función del *Clustering Coefficient* y del *Characteristic Path Length*. En esta tabla se muestra el *Clustering Coefficient* y *Characteristic Path Length* promedio de los distintos clusters.

A continuación volvemos a realizar el proceso de *biclustering*, pero esta vez entre problemas y grupos de topologías. Para ello, asignamos a cada grupo, la media del fitness obtenido por cada una de las topologías que lo componen en cada problema. Así, obtenemos una matriz de 13 filas (problemas) y 25 columnas (grupos de topologías). Por último, ejecutamos el algoritmo de biclustering jerárquico. La Figura 3 muestra el resultado obtenido.

Como se puede ver en la figura, el dendograma de los grupos de topologías muestra tres regiones distintas claramente identificadas. En la izquierda, se sitúan topologías que tienen un rendimiento global pobre excepto para los

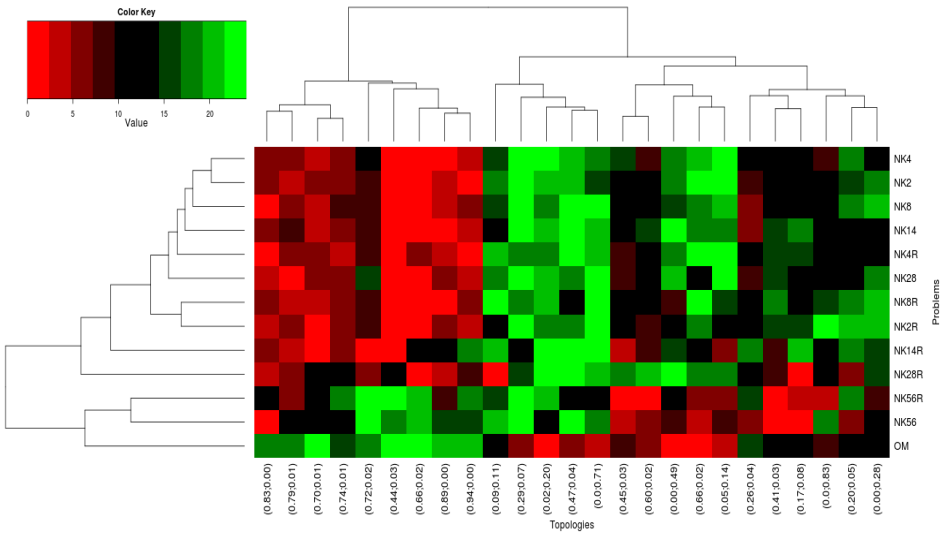


Figura 3. Distribuimos las 851 topologías en 25 grupos en función del *Clustering Coefficient* y del *Characteristic Path Length*. Realizamos el biclustering de los 13 problemas y los 25 grupos de acuerdo con el valor de fitness asignado a cada grupo.

problemas más difíciles. En el centro se hallan las familias de topologías con un buen rendimiento en los problemas sencillos (hasta el problema NK14R) mientras que las topologías cuyo ranking es mayoritariamente intermedio (valores en color negro) se pueden ver en la parte derecha.

5. Conclusión

En este artículo hemos realizado un análisis de la correlación entre la topología de islas en un Algoritmo Evolutivo Paralelo y la estructura de los problemas resueltos. Hemos sido capaces de establecer claramente que diferentes estructuras de problemas requieren topologías de islas distintas a través del análisis de los problemas OneMax y NK-Landscape. El proceso experimental compuesto por miles de simulaciones ha mostrado que no existe una topología ideal que sea mejor para todos los problemas y que la conexión entre las configuraciones de islas y la estructura de los problemas es compleja. Sin embargo, el análisis basado en el Clustering Coefficient y Characteristic Path Length de las topologías sí que ha demostrado la correlación entre estas características topológicas y la estructura de del problema. En efecto, la utilización de técnicas de biclustering permite asegurar que ciertas configuraciones de islas son beneficiosas para determinadas estructuras del problema.

Referencias

1. Michell, A.: The limits of economy of materials in frame structures. Philosophical Magazine (1904) 589 – 597
2. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. Nature **393**(6684) (jun 1998) 440–442
3. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. Science **286** (1999) 509–512
4. Cantú-Paz, E.: Topologies, migration rates, and multi-population parallel genetic algorithms (1999)
5. Giacobini, M., Preuss, M., Tomassini, M.: Effects of scale-free and small-world topologies on binary coded self-adaptive cea. In Gottlieb, J., Raidl, G., eds.: Evolutionary Computation in Combinatorial Optimization. Volume 3906 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2006) 86–98
6. Wang, G., Wu, D., Szeto, K.: Quasi-parallel genetic algorithms with different communication topologies. In: Evolutionary Computation (CEC), 2011 IEEE Congress on. (june 2011) 721 –727
7. Kennedy, J., Mendes, R.: Population structure and particle swarm performance. In: Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on. Volume 2. (2002) 1671 –1676
8. Li, L., Garibaldi, J.M., Krasnogor, N.: Automated self-assembly programming paradigm: The impact of network topology. International Journal of Intelligent Systems **24**(7) (2009) 793–817
9. Fialho, A., Costa, L., Schoenauer, M., Sebag, M.: Extreme value based adaptive operator selection. In: Proceedings of the 10th international conference on Parallel Problem Solving from Nature: PPSN X, Berlin, Heidelberg, Springer-Verlag (2008) 175–184
10. Verel, S., Ochoa, G., Tomassini, M.: Local optima networks of nk landscapes with neutrality. Evolutionary Computation, IEEE Transactions on **15**(6) (dec. 2011) 783 –797
11. Weinberger, E.D.: NP completeness of kauffman's n-k model, a tuneable rugged fitness landscape. Working papers, Santa Fe Institute (1996)
12. Krasnogor, N., Gustafson, S.: A study on the use of "self-generation" in memetic algorithms. Natural Computing **3** (2004) 53–76 10.1023/B:NACO.0000023419.83147.67.
13. Krasnogor, N.: Self-generating metaheuristics in bioinformatics: the protein structure comparison case. Genetic Programming and Evolvable Machines **5**(2) (2004) 181–201
14. Arnaldo, I., Contreras, I., Millán-Ruiz, D., Hidalgo, J., Krasnogor, N.: Matching island topologies to problem structure in parallel evolutionary algorithms. Soft Computing (2013) 1–17

Obtención de los parámetros de un modelo de redes mediante un AGP con intercambio de individuos en Dropbox

J.Gabriel García Caro, Javier Villanueva-Oller, J.Ignacio Hidalgo, and Rafael J. Villanueva

Universidad Politécnica de Valencia, Universidad Complutense de Madrid
jggarciacaro@gmail.com,jvillanueva@pdi.ucm.es,hidalgo@dacya.ucm.es,
rjvillan@imm.upv.es

Resumen Normalmente, al tratar con modelos de redes aleatorios, nos encontramos con que la búsqueda de los parámetros que mejor hacen evolucionar al modelo es una tarea computacionalmente muy costosa. Esto es así porque la manera habitual de afrontarlos es mediante la evaluación exhaustiva de todas las soluciones. Esto da pie a la búsqueda de una optimización de este proceso que permita disminuir el coste en recursos y tiempo necesarios. En este artículo se presenta una alternativa que combina algoritmos evolutivos, computación distribuida y almacenamiento en la nube que nos ha permitido hacer trabajar juntos a elementos creados de manera independiente (sistema de computación, modelo de redes y generador genético) y para distintas plataformas sin tener que modificarlos.

Keywords: Algoritmos Genéticos Paralelos, Virus Respiratorio Sincitial, Modelo de Redes

1. Introducción

Los modelos de redes se han convertido en un paradigma de vital importancia en el análisis de sistemas complejos. Estos sistemas abarcan desde la biología evolutiva [1] a redes neuronales [4], pasando por redes sociales [2], de transporte o económicas [3]. Por ejemplo, estos modelos pueden usarse para estudiar enfermedades infectocontagiosas [6,7,8]. En los últimos tiempos las redes aleatorias se han popularizado como medio para simular los patrones de difusión de enfermedades en redes de gran tamaño [9,10,11]. Además, este tipo de redes proporcionan una alternativa a los esquemas tradicionales basados en ecuaciones diferenciales cuyo origen está en los trabajos de Kermack y Mckendrick [12,13,14]. Es cierto que las ecuaciones diferenciales son un mecanismo conocido y potente [14,15] que permite estudiar la dinámica de muchos sistemas, pero cuyo inconveniente es que, cuando se utilizan en el entorno de un modelo epidemiológico, tiene muchas limitaciones pues no puede distinguir entre individuos concretos, por lo que introducir en el modelo elementos tales como edad, sexo, enfermedades previas,

etc., se convierte en algo muy complejo. Por un lado, los modelos de redes se gestionan como un grafo en la que cada nodo es un individuo, con una serie de atributos concretos que le pertenecen, como por ejemplo edad, sexo, estado respecto a la enfermedad (susceptible, infectado, recuperado, en latencia, etc.), y con una serie de aristas que los unen a otros nodos y que representan las relaciones (sociales o de otro tipo) que definen la estructura de la red misma y las vías por las que se contagia la enfermedad. Hay distintas maneras de implementar este tipo de redes. La más tradicional es la de Erdos y Renyi [17], pero han ido surgiendo otras alternativas como las scale-free networks [18] o las de Watts y Strogatz [19].

Hasta ahora, los estudios llevados a cabo usando este tipo de redes se han restringido a un número relativamente pequeño de individuos, normalmente no más de 10000 [5], pero en muchos casos (por ejemplo una pandemia), el número de sujetos involucrados es del orden de millones. Esto tiene un alto precio en términos de coste computacional, ya que, salvo casos concretos como las redes con distribución de probabilidad potencial (que permiten un ajuste similar al de las ecuaciones diferenciales), su ajuste implica una búsqueda exhaustiva, haciéndolo inviable utilizando medios tradicionales u obligando a grandes limitaciones, como tamaños de la red reducidos (en nodos y/o en relaciones) o restricciones en la exploración de los parámetros de ajuste.

Modelos de este tipo ya se han analizado y resuelto con buenos resultados [25,26,27], incluso usando computacón distribuida como es el caso del Virus Respiratorio Sincitial (VRS) [25,26] o los impulsos cerebrales [28].Sin embargo los tiempos de ejecución son extremadamente altos. Por ello, en este artículo se presenta un sistema capaz de ajustar estos modelos a un coste computacional mucho menor. Para ello se sustituye la búsqueda exhaustiva por un algoritmo genético (AG), el cuál se combina con un sistema de computación paralela. Además para mantener un bajo acoplamiento entre ambos, los dos sistemas se apoyan en un servicio de almacenamiento de archivos en la nube como Dropbox.

Este artículo está organizado como sigue: en la sección 2 se van a detallar los principios del modelo de la red neuronal aleatoria, en la sección 3 se explica cómo se generan los individuos mediante lógica genética para su posterior resolución a través del modelo anteriormente descrito, en la sección 4 se muestran los resultados obtenidos en los experimentos realizados y por último en la sección 5 se detallan las conclusiones obtenidas tras los cálculos además de las experiencias y líneas futuras.

2. Modelo VRS

El VRS (Virus Respiratorio Sincitial) es el principal causante de afecciones respiratorias en bebés y niños pequeños con epidemias anuales de neumonía y bronquiolitis, así como de traqueobronquitis e infecciones de vías aéreas altas en niños mayores y adultos [20]. Su impacto sobre los sistemas de salud crece conforme aumenta el número de niños hospitalizados por bronquiolitis [21] (más de 15.000 visitas/año al pediatra de atención primaria en España), y no ha sido

hasta hace poco que se estudia su efecto en adultos, siendo causa de más del 18 % de las hospitalizaciones por neumonía en mayores de 65 años [22].

Por ello resulta de especial interés la simulación de esta enfermedad, y poder estudiar su evolución, entender qué parámetros influyen en su contagio, predecir cómo se va a extender entre la población, y poder tomar decisiones tales como qué tipo de estrategia de vacunación seguir. En este proceso de simulación es donde entran en juego los modelos, cuya elaboración (a grandes rasgos) lleva asociados los siguientes pasos:

1. Elaborar el modelo en función de una serie de parámetros que lo caractericen (población, tasa de infección, tasa de mortalidad, tiempo de inmunidad, tiempo de convalecencia, etc.).
2. Ajustar el modelo acorde a resultados conocidos. Si el modelo calcula, p. ej. el número de víctimas mortales en un periodo de tiempo, buscamos la combinación de parámetros que hagan que dicho cálculo se asemeje más a las tasas de mortalidad de años pasados.
3. Usar el modelo. Una vez encontrados los valores de los parámetros de mejor ajuste, hacemos evolucionar el modelo para que prediga los resultados objeto de interés en el futuro.

En nuestro caso, el modelo a utilizar ha sido contrastado en [25,27], solo que en nuestro caso el parámetro a ajustar va a ser únicamente la probabilidad de contagio ante un contacto entre un nodo de la red infectado y uno susceptible con el que esté unido mediante una arista. El grado medio de los nodos de la red aleatoria es de 48 y el número de nodos es de 1000000. Se eligen estos valores concretos como punto de partida porque conocemos la mejor solución obtenida por medio de búsqueda exhaustiva, y queremos ver si somos capaces de converger a algo parecido y a qué coste. Las transiciones entre estados de los nodos siguen una evolución estándar SIRS (susceptible a infectado a resistente a susceptible). La evaluación de red se hace durante un tiempo simulado de 7 años. Con estas características, cada red tarda unos 150 minutos en evaluarse y comprobar la calidad de la solución a la que conduce, consistente en lo bien que predice el número de infectados de la población en cada instante dado.

3. AG y su implementación

Para evitar la necesidad de usar una búsqueda exhaustiva, se pone en juego reemplazar éste método por un algoritmo genético capaz de explorar el espacio de soluciones reduciendo el tiempo de cálculo. En concreto, para el modelo descrito en la sección 2 se ha elaborado un algoritmo capaz de, en base a los parámetros de construcción de la red (antes explicados), encontrar una solución al menos igual de buena que la obtenida mediante búsqueda exhaustiva con un coste computacional mucho más bajo.

3.1. Algoritmo Genético (AG)

El primer paso del AG es la generación de una población inicial aleatoria de N individuos. No confundir los individuos de esta población (cada individuo

es una red completa con sus parámetros), con los individuos de la red (nodos o neuronas cada una de las cuales representa una persona infectada o no). En este caso cada individuo modela una red neuronal aleatoria dónde en base a sus parámetros se dotará al VRS de un comportamiento único que desembocará en una cantidad mayor o menor de infectados a lo largo del tiempo.

El parámetro a estudiar es la componente constante de la probabilidad de contagio de un individuo infectado sobre uno susceptible, llamado por el modelo VRS, `b0`. El valor inicial de `b0` es un número aleatorio que varía entre 0 y 1 codificado de forma binaria y de una longitud de 12 bits.

La selección se realiza mediante torneo binario y la fase de cruzado se basa en el cruce uniforme [31].

El operador de mutación niega el valor binario de como mucho 1 bit elegido aleatoriamente de la codificación de cada individuo. Además, se utiliza un esquema elitista en el que los dos mejores individuos de cada generación forman parte de la nueva generación sin verse afectados por las mutaciones.

Cuando el modelo del VRS calcula un individuo (red neuronal) genera un fichero que contiene el número de infectados para cada instante de tiempo. Estos datos son ajustados con un error cuadrático medio (ECM) que es la media cuadrática para una colección de N valores $\{x_1, x_2, ..., x_N\}$ de una variable discreta x (en nuestro caso los infectados por el VRS), que viene dada por la siguiente fórmula:

$$x_{RMS} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}$$

Este ciclo se repite hasta que se alcanza la condición de terminación, delimitada por un número n de iteraciones. El individuo mejor es el que presente el menor valor en su ECM.

3.2. Paralelización

Para resolver un modelo de las características del VRS mediante búsqueda exhaustiva se necesitan unos 20 años de tiempo de CPU en un ordenador personal de tipo medio [25]. Estos tiempos son tan elevados debido al alto coste computacional que lleva a calcular y simular cada red neuronal aleatoria (unos 90 minutos). El introducir como método de búsqueda y optimización un algoritmo genético, mejora notablemente el tiempo de cómputo (pasamos de tener que resolver cientos de miles de redes a resolver solo unos pocos miles), pero aún así para un solo ordenador necesitaríamos 525 días para procesarlos todos. Para solventar esto optamos por calcular los modelos de manera paralela y distribuida, y para ello utilizaremos *Sísifo*[1], un gestor de cálculo distribuido de fácil configuración y manejo que conocemos bien por haber usado en ocasiones anteriores.

Así, pasamos de una arquitectura de algoritmo genético común a una adaptada al entorno distribuido donde se evalúan todos los individuos en paralelo,

[1] Para un mayor nivel de detalle consultar http://sisifo.imm.upv.es/

pudiendo entonces clasificar la combinación de Sísifo más AG como algoritmo genético paralelo (AGP) maestro-esclavo síncrono [23].

Sísifo es un sistema cliente-servidor desarrollado por la Universidad Politécnica de Valencia que permite que un problema sea resuelto utilizando computación distribuida según la filosofía Desktop Grid, permitiendo el uso conjunto de ordenadores heterogéneos. De manera similar a BOINC[2], asigna tareas a un conjunto de PCs, espera a que las tareas terminen y recoge los resultados para su posterior análisis. Sin embargo, y en contraste con BOINC (el cual requiere un equipo de especialistas trabajando para instalar y configurar este), Sísifo está pensado para poder instalarse y configurarse en una red local en apenas minutos y apenas necesita mantenimiento, a cambio de ofrecer limitado control de acceso y seguridad, por lo que es apto en la práctica solo en entornos controlados como una intranet. Sísifo está gobernado por un servidor, el cual se encarga de distribuir las tareas pendientes entre sus clientes. Éstos al terminar con sus respectivas tareas las devuelven al servidor y es este el que se encarga de almacenarlas debidamente.

Sísifo está compuesto por 15 equipos de las siguientes características:

- 10 equipos XEON X3230 a 2.66 GHz con 6.6 GB de RAM
- 5 equipos XEON X3430 a 2.4 GHz con 16 GB de RAM

3.3. Adaptación del generador genético al entorno distribuido

Como primera aproximación, se introdujo el **generador genético** en el propio servidor de **Sísifo**, y éste era el encargado de repartir los individuos a sus clientes para ser resueltos. El problema de esta arquitectura y método de trabajo es que, cada vez que se realizaban cambios y/o ajustes en el generador genético, la introducción de nuevo del mismo en el entorno era costosa, ya que Sísifo corre bajo **Linux** y el generador genético esta implementado en **Windows**.

Hoy en día es de uso común la llamada *nube*, cuyo principio básico es permitir compartir ficheros entre ordenadores heterogéneos a través de Internet. Esa característica se amolda bien a nuestras necesidades, pues en el proceso de cálculo que tenemos que procesar individuos y soluciones en forma de ficheros compartidos entre Sísifo y el generador genético. Por ello, se decide usar uno de estos servicios en la nube, en concreto *Dropbox*[3].

Drobpbox es un servicio de alojamiento de archivos multiplataforma en la nube, operado por la compañía del mismo nombre. Este servicio permite a sus usuarios almacenar y sincronizar archivos en línea y compartir éstos archivos y carpetas con otros usuarios. Dropbox es un software que enlaza todas las computadoras mediante una sola carpeta, lo cual constituye una manera fácil de respaldar y sincronizar los archivos. El servicio *cliente de Dropbox* permite a los usuarios dejar cualquier archivo en una carpeta designada. Ese archivo

[2] Las características y detalles de éste pueden ser consultadas en http://boinc.berkeley.edu/trac/wiki/BoincIntro

[3] Para más detalle visitar www.dropbox.com

es sincronizado en la nube y en todas las demás computadoras del cliente de Dropbox.

De esta manera, uniendo el generador genético y Sísifo a través de Dropbox, se consigue que cada elemento del sistema de cálculo colabore de forma conjunta y sincronizada con los otros a pesar de encontrarse en *redes* físicamente separadas y en ordenadores con sistemas operativos diferentes. Otra ventaja es que la única visión que tiene cada elemento del sistema son los archivos compartidos, lo cual el acoplamiento que presenta el sistema es muy bajo.

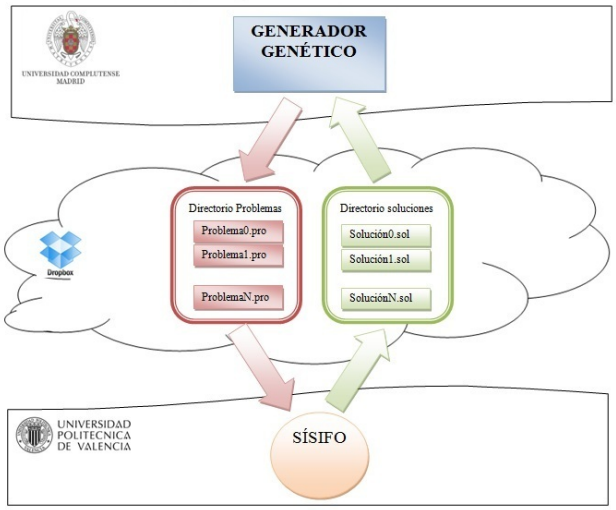


Figura 1. Entorno de cálculo distribuido

En trabajos recientes, se ha estudiado y comprobado que es posible hacer uso de herramientas como Dropbox para compartir ficheros entre ordenadores combinando estos con AGs y computación distribuida. Aunque se está utilizando Dropbox para compartir información entre ordenadores como en [29,30], como principal diferencia podemos observar es la arquitectura que se presenta en la figura 1 y que está compuesta por:

Generador Genético: Detallado en la sección 3 está alojado en la Universidad Complutense de Madrid y se encarga de generar los individuos en un fichero con un formato determinado para su posterior resolución con la ayuda de Sísifo, además de operar acorde a la lógica genética.

Sísifo: Servidor y clientes de cálculo alojados en la Universidad Politécnica de Valencia, se encarga de recoger los individos generados por el generador genético para resolverlos.

Dropbox: Es el medio por el cual Sísifo y el AG comparten individuos y soluciones de manera sincronizada.

3.4. Funcionamiento del sistema

Este sistema (figura 1) trabaja de la siguiente manera:

- En cada iteración el **generador genético** crea en una carpeta compartida en **Dropbox** un determinado número N de individuos y acto seguido empieza a monitorizar dicha carpeta hasta que todos los individuos han sido resueltos y sus soluciones generadas y compartidas en otra carpeta al uso por Sísifo;
- **Sísifo** detecta que hay nuevos individuos por resolver y reparte éstos entre sus clientes para su procesado;
- Una vez el individuo ha sido procesado, el cliente que lo ha resuelto devuelve el resultado de dicha simulación al **servidor de Sísifo**, y es éste quien se encarga de validar y copiar los resultados en el directorio al uso en Dropbox;
- **El generador genético**, una vez detecta que están disponibles todas las soluciones de los individuos a procesar, las lee, calcula mediante la lógica genética descrita en la sección 3 su mejor o peor ajuste, y repite todo este proceso hasta alcanzar la condición de terminación fijada.

4. Resultados experimentales

En estos cálculos se han definido los parámetros expuestos en el cuadro 1, ajustados de forma empírica.

Cuadro 1. Parámetros del AG

Número total de individuos	100
Número total de iteraciones	100
Probabilidad de mutación	2 %
Probabilidad de cruce	80 %

En la figura 2 se muestra la mejor solución obtenida en los cálculos. Los datos representados en color rojo son los datos reales de hospitalizados por semana en comunidad valenciana. Los datos en azul son los resultados calculados por el modelo VRS acorde a los datos proporcionados por el AG.

La evolución que hemos calculado se ha compuesto de 60 generaciones para un total de tiempo de CPU acumulado de 344 días. En la práctica, y dado que hemos usado Sísifo con 15 ordenadores con cuatro núcleos cada uno trabajando en paralelo, eso se traduce en tener la mejor solución en poco menos de 6 días. Es más, la mejor solución se obtiene ya en la iteración 19, con lo que el tiempo es en realidad mucho meno de unos 2 días.

La solución que hemos obtenido es al menos igual de buena que la encontrada por medios mucho más costosos en [25], si bien es cierto que hemos limitado la

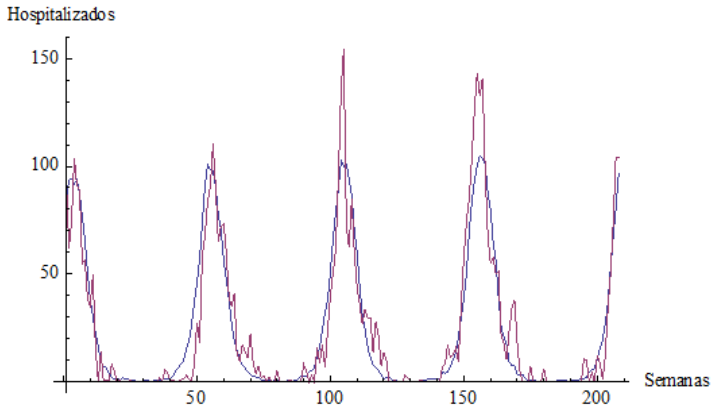


Figura 2. La mejor solución obtenida con un ECM igual a 12.47

búsqueda en esta primera experiencia a un solo parámetro (b0) y que tenemos que elaborar un cara a cara en condiciones más realistas, donde los parámetros a buscar son dos más (grado medio de conectividad de la red y tiempo de recuperación de las personas infectadas). Esto explica porqué en la figura 2 entorno a la semana 100 y 150 el ajuste no es especialmente bueno, ya que estamos limitados por el resto de parámetros fijos de la red.

5. Conclusiones y futuras líneas

5.1. Conclusiones

Según nuestras experiencias podemos sacar las siguientes conclusiones

- Los AGs son capaces de encontrar soluciones similares a las encontradas mediante búsqueda exhaustiva a mucho menor coste.
- Para algunos problemas muy complejos, incluso usando AG es necesaria su combinación con cálculo distribuido a fin de tener tiempos de solución aceptables, dando lugar a un AGP.
- El incorporar un servicio en la nube como Dropbox, facilita la integración de ambos sistemas y su monitorización, pues todo se basa en el uso compartido de carpetas y archivos.

5.2. Futuras líneas

Dado que el AGP ha funcionado correctamente en la búsqueda de un sólo parámetro, estamos adaptando éste para la búsqueda de no solo el parámetro descrito con anterioridad sino que, queremos estudiar otros parámetros como pueden ser el grado medio de conectividad entre los nodos de la red neuronal y el tiempo de recuperación de las personas infectadas.

Además queremos adaptar la búsqueda mediante AGs a otros modelos de virus, como son el meningococo y el VPH.

Referencias

1. S. R. Proulx, D. E. L. Promislow, P. C. Phillips, *Network thinking in ecology and evolution, TRENDS in Ecology and Evolution*, Vol. 20 N. 6(2005) 345-353.
2. A. L. Traud, P. J. Mucha, M. A. Porter, *Social structure of Facebook networks*, Physica A 391 (2012) 4165-4180.
3. M. J. van der Leij, *The Economics of Networks: Theory and Empirics, Thesis dissertation, Tinbergen Institute Research Series*, ISBN:905170973 0, Available Online at http://repub.eur.nl/res/pub/8212/
4. Y. Bar-Yam, *Dynamics of Complex Systems*, Addison-Wesley, Reading, Massachusetts, 1997.
5. N. A. Christakis, J. H. Fowler, *The collective dynamics of smoking in a large social network*, New England Journal of Medicine 358 (21) (2008) 2249-2258. doi:10.1056/NEJMsa0706154.
6. E. Ahmed, H. N. Agiza, *On modeling epidemics including latency, incubation and variable susceptibility*, Physica A: Statistical and Theoretical Physics 253 (1-4) (1998) 347-352. doi:10.1016/S0378-4371(97)00665-1.
7. R. M. Z. dos Santos, *Immune responses: Getting close to experimental results with cellular automata models*, in: D. Stauer (Ed.), Annual Reviews of Computational Physics VI, 1999.
8. U. Hershberg, Y. Louzoun, H. Atlan, S. Solomon, *HIV time hierarchy: winning the war while, losing all the battles*, Physica A: Statistical Mechanics and its Applications 289 (1-2) (2001) 178-190. doi:10.1016/S0378-4371(00)00466-0.
9. G. Witten, G. Poulter, *Simulations of infectious diseases on networks*, Computers in Biology and Medicine 37 (2) (2007) 195-205. doi:10.1016/j.compbiomed.2005.12.002.
10. L. Acedo, J.-A. Moraño, J. D ez-Domingo, *Cost analysis of a vaccination strategy for respiratory syncytial virus (RSV) in a network model*, Mathematical and Computer Modelling 52 (7-8) (2010) 1016-1022. doi:10.1016/j.mcm.2010.02.041.
11. C. L. Barrett, K. R. Bisset, S. G. Eubank, X. Feng, M. V. Marathe, *Episimdemics: ancient algorithm for simulating the spread of infectious disease over large realistic social networks*, in: Proceedings of the 2008 ACM/IEEE conference on Supercomputing, SC'08, IEEE Press, Piscataway, NJ, USA, 2008, pp. 37:1-37:12. URL http://portal.acm.org/citation.cfm?id=1413370.1413408
12. W. O. Kermack, A. G. McKendrick, Contributions to the mathematical theory of epidemics - Part I, Proc. Roy. Soc. 115 (1927) 33-55.
13. L. Edelstein-Keshet, *Mathematical models in Biology*, SIAM, 2005.
14. J. D. Murray, *Mathematical Biology: I. An Introduction*, Springer-Verlag, Berlin, 2002.
15. H. W. Hethcote, *The mathematics of infectious diseases*, SIAM Rev. 42(2000) 599-653.
16. A.Weber, M.Weber, P.Milligan, *Modeling epidemics caused by respiratory syncytial virus (RSV)*, Mathematical Biosciences 172 (2) (2001) 95-113. doi:10.1016/S0025-5564(01)00066-9.
17. B. Bollob as, *Random graphs*, Cambridge University Press, 2nd edition, 2001.
18. A.-L. Barab asi, R. Albert, Emergence of scaling in random networks, Science 286 (5439) (1999) 509-512. doi:10.1126/science.286.5439.509.

19. D. J. Watts, Small worlds: *The dynamics of networks between order and randomness*, Princeton University Press, 2003.
20. Glezen, W.P., Taber, L.H., Frank, A.L., Kasel, J.A., R*isk of primary infection and reinfection with respiratory syncytial virus*, Am. Jour. Dis. Ch., 140:441-456, 1986.
21. Langley, J.M., Leblanc, J.C., Smith, B., Wang, E.E.L., *Increasing incidence of hospitalization for bronchiolitis among canadian children 1980-2000*, J. Inf. Dis., 118:1764-1767, 2003.
22. Han, L., Alexander, J., Anderson, L., *Respiratory syncitial virus, pneumonia among the elderly: An assessment of disease burden*, J. Inf. Dis., 179:25-30, 2003.
23. A. la Torre de la Fuente,*Algoritmos genéticos paralelos*, 2005.
24. M. Mitchell, *An introduction to genetic algorithms*. ISBN 0-262-13316-4(HB),0-262-63185-7(PB), 1996.
25. L. Acevo, J.A. Moraño, R.J. Villanueva, J. Villanueva-Oller, *Using random networks to study the dynamics of respiratory syncitial virus (RSV) in the Spanish region of Valencia*, Mathematical and Computer Modelling nº 54, ISSN 0895-7177, pp. 1650-1654 (año 2011) doi:10.1016/j.mcm.2010.11.068.
26. I.C. Lombana, M. Rubio, E. Sánchez, F.J. Santonja, J. Villanueva-Oller, *A network model for the short-term prediction of the evolution of cocaine consumption in Spain in the next few years*, Mathematical and Computer Modelling nº 52, ISSN 0895-7177, pp. 1023-1029 (año 2010) doi:10.1016/j.mcm.2010.02.032.
27. J. Díez-Domingo, J. Villanueva-Oller, L. Acevo, R.J. Villanueva, J.A. Moraño, *Seasonal Respiratory Syncytial Virus Epidemic in a Random Social Network*, Modelling for addictive behaviour, medicine and engineering 2010, ISBN 978-84-693-9537-0, pp. 1-4 (año 2010). Ed. Instituto Universitario de Matemática Multidisciplinar.
28. L. Acedo, J.A. Moraño, R.J. Villanueva, J. Villanueva-Oller, *The Neurona@Home project: Simulating a large-scale cellular automata brain*, 12th Granada Seminar on Computational and Statistical Physics Universidad de Granada, Instituto Carlos I, Granada, España, 17-21/09/2012 doi: 10.1063/1.4776528.
29. Maribel García-Arenas, Juan Julián Merelo Guervós, Pedro Castillo,Juan Luis Jiménez Laredo, Gustavo Romero, y Antonio M. Mora, *Using free cloud storage services for distributed evolutionary algorithms*, Proceedings of the 13th annual conference on Genetic and evolutionary computation (GECCO '11), Natalio Krasnogor (Ed.). ACM, New York, NY, USA, 1603-1610, 2011.
30. K. Meri, M. G. Arenas, A. M. Mora, J. J. Merelo, P. A. Castillo, P. García-Sánchez, J. L. J. Laredo, *Cloud-based evolutionary algorithms:An algorithmic study. Natural Computing*, Volume 12, Issue 2, pp 135-147, 2013.
31. J. Arranz de la Peña, A. Parra Truyol, *Algoritmos Genéticos*, Universidad Carlos III, 2006.

Mejorando el Comportamiento de las Metaheurísticas Cooperativas basadas en Trayectoria mediante Path Relinking

Gabriel Luque and Enrique Alba

ETSI Informática, Universidad de Málaga, 29071 Málaga, Spain
{gabriel,eat}@lcc.uma.es

Abstract. En este artículo proponemos la utilización de la técnica path relinking para mejorar el comportamiento de las metaheurísticas paralelas cooperativas basadas en trayectorias. Los algoritmos basados en la explotación del vecindario de una única solución como el enfriamiento simulado (SA) han ofrecido resultados muy precisos para un gran número de problemas de índole real. En particular, algunos modelos avanzados como el paralelo cooperativo, donde múltiples instancias de los métodos (subalgoritmos) se ejecutan en paralelo cooperando mediante el intercambio de soluciones, han permitido resolver problemas muy complejos. Pero pese a ello, aún se necesitan versiones más avanzadas que nos permitan tratar la alta complejidad temporal y numérica de los problemas planteados por la industria. En este artículo, utilizaremos path relinking para mejorar el rendimiento de algoritmos basados en este tipo de técnicas. En concreto, estudiamos diferentes estrategias de diseño para incorporar este algoritmo en el esquema de las metaheurísticas basadas en trayectoria y probamos su rendimiento en dos problemas bien conocidos y muy diferentes como son MAXSAT y QAP. En los resultados mostramos cómo nuestro esquema nos permite mejorar el rendimiento numérico (eficacia) sin afectar o incluso reduciendo su coste computacional (eficiencia).

1 Introducción

Las metaheurísticas son técnicas generales que proporcionan buenas soluciones (e incluso óptimas) en un tiempo razonable para un gran número de problemas de optimización [3]. Dependiendo del número de soluciones manejadas simultáneamente durante el proceso de optimización, se pueden clasificar en dos categorías: técnicas basadas en trayectoria y técnicas basadas en población. Estas últimas hacen uso de población (conjunto) de soluciones que se van mejorando mediante un mecanismo de modificación/reemplazo de las existentes. Por otro lado, en los algoritmos basados en trayectoria partimos de una única solución que, en cada paso de la búsqueda, es reemplazada por otra solución (generalmente mejor) encontrada en el vecindario de la solución actual.

Aunque el uso de las técnicas metaheurísticas nos permiten reducir de forma significativa la complejidad temporal del proceso de búsqueda, aún puede resultar demasiado costoso para problemas de alta complejidad como los que se encuentra en el mundo real. En este contexto, el paralelismo surge como una estrategia muy útil para reducir

los tiempos de cálculo y poder abordar este tipo de problemas [1]. Además las versiones paralelas no sólo permiten acelerar el cómputo, sino que también pueden permitir mejorar la calidad de las soluciones obtenidas. En la literatura se han propuesto múltiples modelos de paralelismo que pueden aplicarse a las metaheurísticas. Pero esos modelos en general se han enfocado más en los algoritmos basados en población, y existe un vacío bastante importante en este tipo de estudio para las basadas en trayectoria.

Por ese motivo, en este artículo nos enfocamos en las metaheurísticas paralelas basadas en trayectoria. Para este tipo de técnicas se suelen definir tres tipos de modelos paralelos: (a) exploración del vecindario en paralelo, (b) la evaluación en paralelo de cada solución y (c) la exploración de varias trayectorias en paralelo. Las dos primeras únicamente permiten acelerar el cómputo pero su comportamiento es exactamente igual al de la versión secuencial. Quizás desde un punto de vista algorítmico es más interesante el último modelo de paralelismo, ya que además de acelerar el proceso de búsqueda, cambia la semántica del algoritmo respecto al secuencial. Este modelo ejecuta en paralelo varias instancias del algoritmo (que pueden ser iguales o no). Los diferentes subalgoritmos pueden cooperar entre sí o ser ejecutadas independientemente. En el modelo cooperativo durante la ejecución cada subalgoritmo intercambia información (soluciones) con otros subalgoritmos. Cuando un subalgoritmo recibe una nueva solución puede continuar su proceso de búsqueda o bien con la nueva recibida o bien con la que ya tenía (de acuerdo a un mecanismo de selección). El problema de este modelo cooperativo es que siempre pierde información: si descarta la nueva solución (y continúa con la actual) no se incorpora nueva información y cuando se acepta, se pierde la información histórica contenida en la solución actual del algoritmo.

En este trabajo describimos un nuevo esquema para resolver este problema. En concreto, cuando dos subalgoritmos cooperan, utilizamos la técnica path relinking entre la nueva solución recibida y la ya existente, para generar soluciones que contengan información de ambas. En la integración del path relinking con la técnica basada en trayectoria existen múltiples alternativas de diseño: entre qué subalgoritmos se produce la cooperación, cómo seleccionar la solución entre todas las generadas por path relinking, ... Entonces el objetivo de este artículo es múltiple: proponer el nuevo esquema cooperativo paralelo, el estudio de las diferentes alternativas de diseño y finalmente la validación el algoritmo resultante en dos problemas de alta complejidad y bien conocidas como son MAXSAT y QAP (ambos clasificados como NP-duros [9]).

El artículo se organiza de la siguiente forma: en las siguientes secciones presentaremos información introductoria para entender correctamente el resto del trabajo. En la Sección 4 describiremos nuestra propuesta algorítmica y sus diferentes alternativas de diseño. Tras enunciar el diseño experimental (Sección 5), en la Sección 6 analizaremos los resultados. En la última sección resumiremos las principales conclusiones que se extraen del trabajo y posibles líneas de trabajo que pensamos explorar en un futuro próximo.

2 Algoritmos Basados en Trayectoria

Los métodos basados en trayectoria son técnicas basadas en la manipulación de una única solución que se va mejorando constantemente mediante la exploración de su

vecindario en el espacio de búsqueda. Esta técnica empieza el proceso de búsqueda a partir de una solución tentativa inicial generada aleatoriamente o proporcionada por otro algoritmo. En cada paso, reemplazamos la solución actual con otra (generalmente mejor) de su vecindario. Existen múltiples técnicas que se ajustan a ese esquema general, como el enfriamiento simulado (o *simulated annealing*, SA) [6], búsqueda tabú (o *tabu search*, TS) [4] o la búsqueda local iterada (o *iterated local search*, ILS) [7], que se diferencian principalmente en como realizan cada uno de los pasos.

Aunque estas técnicas permiten obtener resultados muy precisos para un gran número de problemas, aún se necesita avanzar más para tratar problemas más complejos. Una de las posibles mejoras que se pueden realizar es utilizar versiones paralelas. En la literatura habitualmente se identifican tres modelos de paralelismo:

- **Modelo paralelo con reinicio múltiple**: Este modelo consiste en ejecutar varias instancias del algoritmo en paralelo. Cada instancia es inicializada con una solución diferente. Las instancias pueden funcionar de forma independiente o cooperar entre sí mediante el intercambio de información.
- **Modelo de exploración del vecindario en paralelo**: Este modelo sigue un modelo de maestro-esclavo que permite explorar el vecindario en paralelo.
- **Modelo de evaluación de las soluciones en paralelo**: Este modelo es similar al anterior, pero en este caso lo que hace es calcular el valor de adecuación de una solución en paralelo. Este mecanismo es muy eficiente si la evaluación de una solución es muy costosa computacionalmente.

Mientras los dos últimos modelos lo que nos permite es acelerar la búsqueda, el primero (si se utiliza la versión cooperativa) cambia también el comportamiento del algoritmo respecto al modelo secuencial, lo cual lo hace muy atractivo para muchos problemas. Por ese motivo en este artículo nos centraremos en el modelo con reinicio múltiple y cooperativo.

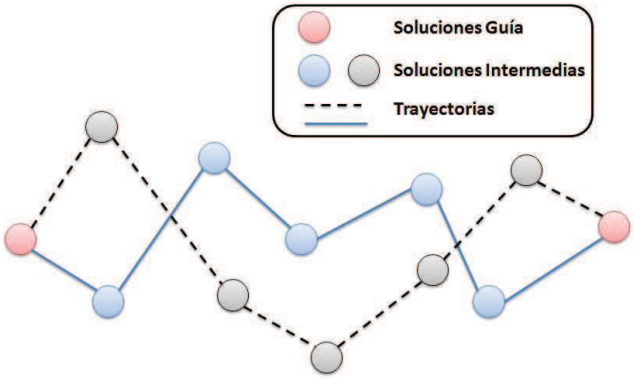


Fig. 1. Esquema de Path Relinking

3 Path Relinking

Path relinking (PR) [5] fue propuesto originalmente dentro del contexto de la metaheurística búsqueda tabú, donde uno de los principales objetivos es realizar un balance entre la búsqueda en intensificación y diversificación. PR se basa en la generación de nuevas soluciones mediante la exploración de trayectorias que conectan soluciones de calidad elevada (soluciones guía), iniciando la búsqueda desde una de estas soluciones y generando un camino en el espacio de vecindario que dirige la búsqueda hacia las otras soluciones. En la Fig. 1 se muestra el esquema básico de este mecanismo de obtención de nuevas soluciones.

4 Propuesta Algorítmica

Nuestro objetivo es diseñar un nuevo modelo para las metaheurísticas basadas en trayectoria que permita reducir el tiempo de ejecución global pero, al mismo tiempo, permita mejorar la eficacia con la que se explora el espacio de búsqueda. Aunque este tema ha sido ampliamente tratado en las metaheurísticas poblacionales, en las basadas en trayectoria tienen una cantidad de trabajos significativamente menor.

Puesto que nuestra meta es mejorar la eficacia del algoritmo paralelo resultante, nos enfocaremos en el paradigma basado en ejecutar múltiples instancias del algoritmo en paralelo que cooperan, ya que es el único que modifica la dinámica respecto a la versión secuencial. Como se comentó en la introdución, el problema de los acercamientos clásicos a ese modelo de paralelismo es que el algoritmo continuamente va eliminando información potencialmente útil. Eso se produce porque cuando un subalgoritmo recibe una solución de otro subalgoritmo, tiene que elegir si continúa con su solución actual o la nueva recibida, y siempre debe descartar la otra.

Nuestro nuevo esquema que proponemos es utilizar ambas soluciones (la actual y la nueva) para generar soluciones que tengan características de ambas. Se podrían utilizar diferentes mecanismos para generar nuevas soluciones, por ejemplo siguiendo un método similar a las técnicas de recombinación de los algoritmos poblacionales [8]. En este caso proponemos la utilización de otro mecanismo más avanzado como es path relinking. Para ello utilizamos la solución actual y la nueva que recibe desde otro subalgoritmo como soluciones guías y generamos una trayectoria entre ambas. Esto generará un conjunto de soluciones, de las que hay que seleccionar una para reemplazar a la actual.

En ese esquema que proponemos, hay dos importantes aspectos de diseño que pueden afectar al comportamiento del algoritmo:

- **Tipo de cooperación**: que indica qué subalgoritmos cooperan entre sí y cómo.
- **Tipo de selección**: que se refiere al proceso de elección de una solución de entre todas las existentes en la trayectoria generada por PR.

Para cada una de esas decisiones de diseño hemos propuesto diferentes alternativas. En concreto para la elección de qué subalgoritmo aporta la nueva solución guía tenemos las siguientes opciones:

- **Predefinida**: en este caso, cada subalgoritmo recibe una única solución (el subalgoritmo origen de esa solución vendrá definido por una topología predefinida). Por lo tanto, no hay elección, simplemente se elige como solución guía a la única que recibe cada subalgoritmo.
- **Mejor**: en este caso, cada subalgoritmo recibe una solución de cada subalgoritmo que componen el método global. Una vez recibidas todas, el subalgoritmo selecciona aquella que tenga un mejor valor de adecuación al problema.
- **Más distante**: al igual que en el caso previo, cada subalgoritmo recibe soluciones del resto de subalgoritmos, pero ahora selecciona aquella solución que está más lejos (considerando la distancia genotípica) a la solución del subalgoritmo receptor. Con esto permitimos una mayor exploración del espacio de búsqueda.
- **Aleatoria**: como en los dos casos previos, el subalgoritmo receptor recibe una solución de cada subalgoritmo del método, pero esta estrategia elige entre todas ellas una aleatoria como solución guía.

Utilizando alguna de las alternativas anteriores conseguimos la segunda solución guía (la otra siempre es la actual del subalgoritmo) y se puede aplicar la técnica path relinking para generar una trayectoria. Una vez tenemos la trayectoria, hay que seleccionar una única solución de la misma para reemplazar a la actual y continuar ejecutando el método basado en trayectoria. Para esta elección también podemos considerar múltiples alternativas:

- **Mejor**: elegimos la solución con mejor valor de adecuación, es decir:

$$max_{s \in S}(f(s)), \tag{1}$$

  donde S es el conjunto de soluciones generadas en la trayectoria, y f es la función de adecuación (suponiendo maximización).
- **Máxima información compartida**: en este caso selecciona aquella solución que comparte más información con las soluciones guía. Para conocer cuanta información comparte una solución con otra calculamos la distancia genotípica entre ellas (esto dependerá de la representación) y utilizaremos la siguiente ecuación:

$$min_{s \in S}(max(dist(s,s'), dist(s,s''))), \tag{2}$$

  donde S es el conjunto de soluciones generadas en la trayectoria, s' y s'' son las soluciones guía y $dist$ es la distancia genotípica entre dos soluciones. Con esta ecuación minimizamos la distancia con ambas soluciones guía simultáneamente.
- **Aleatoria**: En este caso se seleciona aleatoriamente una solución de la trayectoria independientemente de su calidad o distancia al resto de soluciones.

En la parte experimental analizaremos el comportamiento de cada una de estas posibles estrategias.

5 Diseño Experimental

En esta sección concretaremos los detalles del algoritmo empleado en los experimentos, sus parámetros y los problemas empleados para estudiar su comportamiento.

5.1 Problemas de prueba

Para probar nuestra propuesta utilizaremos dos problemas diferentes y muy bien conocidos en la literatura: MAXSAT y QAP.
MAXSAT: El problema de la satifacibilidad (SAT) es reconocido como un problema fundamental en Informática. El MAXSAT es una variante de dicho problema en que se intenta satisfacer el máximo número de cláusulas en una expresión lógica.

En nuestros experimentos usamos 5 instancias extraídas de SATLIB[1]. En concreto usamos las instancias: *uf250-01.cnf* - *uf250-05.cnf*. Cada instancia consta de 250 variables y 1065 cláusulas, y todas ellas están en la transición de fase.
QAP: El problema de asignación cuadrática (o *quadratic assignment problem*, QAP) es un problema combinatorio NP-duro muy estudiado ya que muchos problemas combinatorios pueden expresarse como casos específicos de él.

En este caso hemos utilizado también cinco instancias extraídas de QAPLIB[2]. En concreto hemos elegido las cinco instancias más complejas atendiendo a la clasificación ofrecida en [2].

5.2 Algoritmo utilizado y detalles de implementación

Como algoritmo basado en trayectoria para nuestros experimentos hemos seleccionado uno de los más estudiados y usados en la literatura, el enfriamiento simulado (SA) [6]. Este método heurístico de búsqueda local se basa en recorrer el espacio de soluciones del problema a partir de una solución inicial con el fin de encontrar un óptimo. Para ello utiliza una función de vecindad. Evita caer en óptimos locales mediante la aceptación con una cierta probabilidad de aquellos movimientos a soluciones vecinas que empeoren el valor de la función objetivo respecto de la solución actual. La probabilidad de aceptar un movimiento desfavorable viene dado por un heurístico parametrizado por una temperatura T, de manera que un movimiento que produzca una variación δ de la función objetivo, en caso de empeorarla, será aceptada en función de una probabilidad igual a $\exp(-\delta/T)$. La temperatura T es un parámetro que se actualiza (disminuyendo) durante la ejecución del algoritmo.

Para definir como utilizar SA para resolver los problema de nuestro banco de prueba, nos falta definir cómo generar el vecindario. En el caso del MAXSAT, una solución vecina se genera cambiando en la solución actual el valor de cada variable lógica con cierta probabilidad. En el caso de QAP, esta generación se produce intercambiando las localizaciones asignadas a los servicios (es decir, se intercambian dos posiciones de la permutación).

En nuestro acercamiento ejecutaremos 8 instancias de SA independientes (subalgorithmos) que cooperan de forma asíncrona cada 50 iteraciones. Como mecanismo de cooperación utilizamos las variantes presentadas en el Apartado 4, es decir, utilizando una topología fija (anillo unidireccional en este caso), o bien comunicando todas con todas y seleccionando la mejor solución o la más distante de la solución manejada en el subalgoritmo destino. Una vez producida la cooperación y seleccionada la solución,

[1] `http://www.cs.ubc.ca/~hoos/SATLIB/benchm.html`
[2] `http://www.seas.upenn.edu/qaplib/`

se aplica PR. Para generar la trayectoria, se analizan qué componentes de la solución nueva difieren de la solución actual del subalgoritmo destinatario y cada elemento de la trayectoria, se generan cambiando uno de esos componentes diferentes para ajustarlo al valor existente en la solución actual (es el objetivo final de todos los cambios).

Esta forma de generar la trayectoria, nos permite hacer implementaciones eficientes para los esquemas de selección. En concreto, es posible conocer a priori qué conjunto de soluciones maximizan la similitud con la soluciones guía y se puede generar una solución de ese conjunto sin necesidad de generar toda la trayectoria. Igualmente ocurre con la selección aleatoria. En cambio para la selección basada en el valor la adecuación sí hay que generar todas las soluciones de la trayectoria y evaluarlas. Para reducir el coste de este acercamiento, en nuestra implementación utilizaremos evaluaciones parciales en vez de completas (que aunque no es obligatorio es muy recomendable). Esto permite reducir el coste computacional de forma importante. Por ejemplo, en el MAXSAT, modificar el valor de una variable lógica sólo afecta al 2-5% de las cláusulas de la instancia, por lo que sólo es necesario calcular el cambio en esas instancias. En el QAP, podemos reducir el orden de complejidad de $O(n^2)$ que requiere la evaluación completa de una solución a aproximadamente $O(n)$ que requiere una evaluación parcial (suponiendo que se intercambiaron dos posiciones en la permutación que representa la solución).

5.3 Metodología

En los experimentos, hemos analizado doce posible variantes de nuestra propuesta (3 mecanismos de selección por 4 mecanismos de cooperación) para los que utilizaremos la siguiente nomenclatura: **SA_X_Y**, donde **X** es el mecanismo de selección e **Y** es el mecanismo de cooperación. Los posibles valores de **X** son: **ale** para aleatorio, **mej** para mejor e **inf** para el que intenta maximizar la información compartida. Los posibles valores para **Y** son: **pre** para el que usa una topología predefinida, **ale** para aleatorio, **mej** para mejor y **dis** para el que intenta maximizar la distancia entre las soluciones guía. Además esas doce variantes la hemos comparado un SA paralelo no cooperativo (**SA_noCo**) y otro cooperativo pero que no tiene el procedimiento avanzado para combinar las soluciones (**SA_co**). Para poder hacer comparaciones justas, la condición de parada de todos los algoritmos será alcanzar la solución óptima (salvo que se detecte que es incapaz de hacerlo debido a que se quedó estancado en un óptimo local).

Debido a la naturaleza estocástica de los algoritmos se han realizado 30 ejecuciones independientes por problema y algoritmo. Para comprobar que nuestras conclusiones son correctas hemos aplicado un análisis estadístico a los resultados para asegurar que las diferencias son significativas. En concreto tras aplicar el test de normalidad de Shapiro-Wilks, comprobamos las diferencias con el test ANOVA o el test de Kruskal-Wallis, dependiendo del resultado del test de normalidad.

6 Análisis de Resultados

En esta sección analizaremos los resultados de las diferentes variantes de nuestra propuesta. En primer lugar analizaremos la precisión de los algoritmos y posteriormente pasaremos a analizar el coste computacional de los mismos.

6.1 Precisión

Como la condición de parada es alcanzar el óptimo (si es posible), en la Tabla 1 se dan dos valores relacionados con cuántas instancias es capaz de resolver cada variante. En concreto el primer valor indica el número de instancias de los problemas que es capaz de resolver el algoritmo en al menos en una de las ejecuciones independientes. El segundo indica el número de instancias que es capaz de resolver de forma estable, considerando que esto se produce si al menos en 25 de las 30 ejecuciones independientes la técnica alcanza el óptimo. Como nuestro banco de pruebas está formado por cinco instancias de MAXSAT y cinco de QAP, estos valores pueden oscilar entre 0 (nunca alcanza el óptimo) y 10 (siempre alcanza el óptimo).

Tipo de Cooperación	Tipo de selección		
	Mejor	Aleatorio	Información
Predefinido	9 - 5	7 - 3	8 - 3
Mejor	10 - 10	10 - 6	10 - 8
Aleatorio	10 - 6	8 - 3	10 - 4
Distancia	10 - 10	9 - 4	10 - 8
No cooperativo	5 - 0		
Cooperativo sin PR	7 - 2		

Table 1. Precisión de los algoritmos.

De la Tabla 1 se pueden extraer varias conclusiones interesantes. En primer lugar podemos observar que la cooperación permite mejorar de forma significativa la calidad de las soluciones encontradas, ya que mientras el algoritmo no cooperativo, **SA_noCo**, aunque es capaz de encontrar algunas soluciones óptimas, para ninguna instancia es capaz de resolverla de forma constante, el resto de algoritmos que ya cooperan sí son capaces. La segunda conclusión es que la inclusión de un mecanismo para tratar de forma inteligente la cooperación (en este caso la inclusión de PR) es muy beneficial como se puede ver en que todas las versiones de nuestra propuesta mejoran al modelo clásico (algoritmo **SA_co**).

Comparando las diferentes estrategias de diseño de nuestra propuesta se puede observar que aquellas que consideran las características de las soluciones (estrategias mejor, distancia e información) resuelve de forma más robusta un mayor número de instancias que aquellas que se basan en otros aspectos (aleatorio o usando un esquema predefinido a priori). Este resultado era previsible ya que utilizar información del problema durante el proceso de búsqueda aporta mayores elementos a la hora de guiar dicho proceso.

En concreto podemos observar que los tipos de cooperación que usan la mejor solución o la solución más diferente a la actual son los mecanismos que proporcionan mejores resultados. Esto es bastante sorprendente, ya que ambas técnicas proporcionan búsquedas muy diferentes (una favorece la intensificación y la otra la diversificación) pero obtienen soluciones de muy alta calidad. Un comportamiento similar se puede observar cuando se analizan los tipos de selección, donde tanto el que favorece la inten-

sificación (mejor) como el que ofrece más diversidad (información) permiten obtener resultados muy parecidos. Pero pese a esta similitud de los resultados, se puede ver que hay una tendencia beneficiosa clara hacia las técnicas que favorecen más la intensificación. De hecho, **SA_mej_mej** y **SA_dis_mej** son las únicas variantes que consiguen resolver de forma robusta todas las instancias probadas.

6.2 Coste Computacional

Ahora nos centraremos en el coste computacional, tanto numérico (número de evaluaciones parciales) como temporal (en segundos). Para hacer una comparación justa en este aspecto todos los algoritmos tienen que ser capaz de alcanzar soluciones de la misma calidad, que como vimos en la sección anterior no ocurre. Por ello, este análisis lo restringiremos a las mejores variantes obtenidas (**SA_mej_mej**, **SA_mej_inf**, **SA_dis_mej** y **SA_dis_inf**) y usaremos únicamente las instancias que todos resuelven de forma consistente. En la Fig. 2 se muestra el rendimiento numérico (figura de la izquierda) y el rendimiento temporal (figura de la derecha). Ambos valores están normalizados por cada instancia respecto a los valores obtenidos con **SA_dis_inf**.

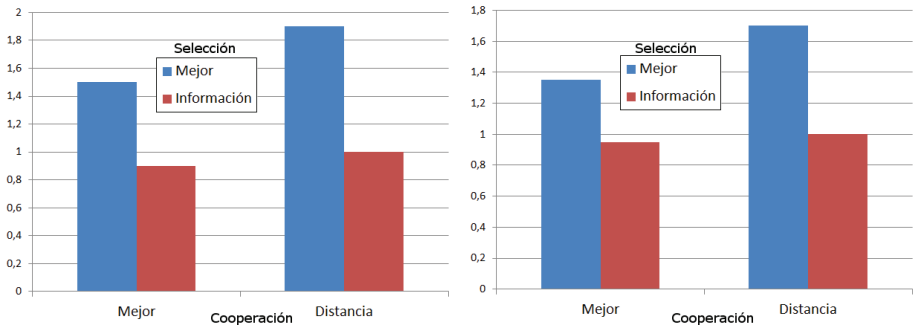


Fig. 2. Rendimiento: (a) Numérico (número de evaluaciones) y (b) temporal (segundos).

Los rendimientos podemos clasificarlos en tres grupos: (i) **SA_dis_mej** es la variante con un mayor coste computacional, esto era esperable ya que el tipo de cooperación (seleccionar soluciones muy diferentes) provoca trayectorias muy largas y la estrategia de seleccionar el mejor requiere que se evalúen todas las soluciones de la trayectoria; (ii) **SA_mej_mej** es el segundo con mayor coste, principalmente debido a que haya que evaluar todas las soluciones de la trayectoria (que es necesario para elegir la mejor) pero es menor que el caso anterior porque las trayectorias son mas cortas; (iii) las estrategias **SA_X_inf** necesitan una cantidad significativamente menor de tiempo para conseguir alcanzar los óptimos ya que sólo generan un punto de la trayectoria en vez de todas las posibles soluciones.

Como resumen final podemos observar que las estrategias que seleccionan la mejor solución de la trayectoria son las que mejores resultados ofrecen pero a costa de requerir un esfuerzo computacional significativamente mayor que las estrategias basadas en generar soluciones que maximicen la información conservada de ambas soluciones, pero en este caso sacrifican un poco la calidad de las soluciones que proporcionan.

7 Conclusiones

En este artículo hemos presentado un nuevo esquema para generar metaheurísticas basadas en trayectoria cooperativas que mejore al esquema tradicional. La idea era tener un procedimiento que permitiese combinar la información contenida en la solución actual del subagoritmo y la información que nos facilitaban el resto de subalgoritmos. La propuesta de este artículo es utilizar la técnica path relinking como procedimiento de combinación.

En los resultados se ha observado que este nuevo esquema proporciona muy buenos resultados, permitiendo resolver instancias que el acercamiento tradicional no permitía resolver. También hemos analizado diferentes alternativas de diseño para el esquema general, como por ejemplo el mecanismo de cooperación entre los subalgoritmos o el tipo de selección de soluciones dentro de la trayectoria generada por PR. Cada una de las variantes tiene sus ventajas e incovenientes atendiendo a calidad de soluciones producidas y coste computacional requerido.

Como trabajo futuro planteamos completar este trabajo añadiendo problemas con diferentes características para analizar la generalidad de las conclusiones aquí presentadas. Como hemos visto la estrategía que selecciona la mejor solución de la trayectoria ofrece resultados muy precisos pero a costa de un mayor esfuerzo computacional, por ello es importante estudiar diferentes alternativas a la hora de generar la trayectoria y estudiar si es posible emplear algún mecanismo que facilite este aspecto (como puede ser estudios teóricos que nos ofrezcan información del espacio de búsqueda).

8 Acknowledgments

La realizacin de este trabajo fue posible gracias a la financiacin obtenida en los proyectos TIN2008-06491-C04-01 (`http://mstar.lcc.uma.es`) y P07-TIC-03044 (`http://diricom.lcc.uma.es`).

References

1. E. Alba, editor. *Parallel Metaheuristics: A New Class of Algorithms*. Wiley, 2005.
2. F. Chicano, G. Luque, and E. Alba. Autocorrelation measures for the quadratic assignment problem. *Applied Mathematics Letters*, 25(4):698–705, 2012.
3. M. Gendreau. *Handbook of metaheuristics*, volume 146. Springer, 2010.
4. F. Glover. Tabu Search, part I. *ORSA, Journal of Computing*, (1):190–206, 1989.
5. F. Glover, M. Laguna, and R. Martí. Fundamentals of scatter search and path relinking. *Control and cybernetics*, 39(3):653–684, 2000.
6. S. Kirkpatrick, C. Gellatt, and M. Vecchi. Optimization by Simulated Annealing. *Science*, 220(4598):671–680, 1983.
7. H. R. Lourenço, O. C. Martin, and T. Stützle. Iterated local search. *International series in operations research and management science*, pages 321–354, 2003.
8. G. Luque, F. Luna, and E. Alba. A new parallel cooperative model for trajectory based metaheuristics. In *Distributed Computing and Artificial Intelligence*, pages 559–567. Springer, 2010.
9. C. Papadimitriou. *The Complexity of Combinatorial Optimization Problems*. Master's thesis, Princeton University, 1976.

Búsqueda dispersa cooperativa para problemas a gran escala en el contexto de biología de sistemas

Alejandro F. Villaverde[1], Jose A. Egea[2], and Julio R. Banga[1]

[1] Grupo de Ingeniería de Procesos, IIM-CSIC
afvillaverde@iim.csic.es, julio@iim.csic.es
[2] Dpto. Matemática Aplicada y Estadística, Universidad Politécnica de Cartagena
josea.egea@upct.es

Resumen. En este trabajo se presenta un algoritmo cooperativo basado en búsqueda dispersa que ha demostrado su superioridad frente a la versión secuencial. Este algoritmo ha sido implementado en los lenguajes Matlab y R, cuyos códigos están disponibles en la web. Se ilustra su aplicación en casos de estudios de funciones "benchmark" para evaluar este tipo de algoritmos, así como en ejemplos de estimación de parámetros en biología de sistemas. Su uso puede ser extendido a cualquier otro campo de aplicación.

Keywords: metaheurísticas cooperativas, biología de sistemas, paralelización, optimización a gran escala

1 Introducción

Muchos problemas de optimización en ciencias e ingeniería conllevan elevados tiempos de cálculo por el coste computacional que implica evaluar las funciones objetivo y/o restricciones, por manejar un elevado número de variables de decisión (hasta miles en algunos casos), o por ambos motivos a la vez. La necesidad de aplicar metaheurísticas como métodos más eficientes de resolución de dichos problemas así como de intentar reducir los tiempos de computación, lleva a diseñar estrategias de paralelización donde se permite cooperar a varios hilos (o "threads") que intercambian información cada cierto tiempo [1, 2]. El uso de cooperación, entendida como intercambio de información, modifica el comportamiento sistémico del algoritmo de optimización y acelera su convergencia [3]. La aplicación de dichas estrategias cooperativas es especialmente atractiva en el caso de metaheurísticas poblacionales, donde diversos elementos de las poblaciones, que evolucionan en cada hilo, pueden ser intercambiados para aumentar la eficiencia y/o la diversidad de la búsqueda. Es posible distinguir entre métodos homogéneos, en los que un mismo algoritmo con distintas configuraciones de parámetros y/o soluciones inciales trabaja en diferentes hilos , y los heterogéneos, en los que metaheurísticas de distinta naturaleza comparten información entre sí [4]. En este trabajo se presenta un método homogéneo basado en búsqueda dispersa.

La biología de sistemas es un área de investigación aparecida en las últimas décadas [5], que se sitúa en la frontera de la ingeniería de sistemas con otras disciplinas. Su objetivo es entender cómo se organizan los sistemas biológicos complejos, para lo cual combina el uso técnicas experimentales con el modelado matemático y técnicas computacionales avanzadas. La creciente disponibilidad de cantidades masivas de datos "ómicos", esto es, procedentes de medidas genómicas, proteómicas o metabolómicas, ha conducido a la necesidad de técnicas de modelado a gran escala. La construcción de un modelo es un proceso iterativo que se puede dividir en varias subtareas, una de los cuales es la estimación de parámetros o calibración de modelos, imprescindible para dotar al modelo de capacidad predictiva.

La estimación de parámetros es una de las principales aplicaciones de la optimización en el ámbito de la biología de sistemas [6]. Consiste en encontrar el vector de parámetros $p-$dimensional para minimizar una función de coste que mide la bondad del ajuste de las predicciones del modelo con respecto a los datos. Matemáticamente se formula como un problema de programación no lineal.

Para modelos de biología de sistemas, este problema es frecuentemente multimodal, debido a la no linealidad de los modelos, y con restricciones no relajables derivadas de la naturaleza dinámica de los sistemas. Éstos son modelizados mediante ecuaciones diferenciales, cuya resolución numérica puede ser un proceso computacionalmente costoso, e introducir ruido en la función de coste y restricciones. Además, habitualmente el número de parámetros a estimar es muy elevado, y el contenido de información de los datos disponibles es escaso, lo que conduce a problemas de identificabilidad.

Por todas estas razones, el uso de metaheurísticas es la opción mas eficiente pues éstas son capaces de resolver problemas de estas características de formas más eficientes que otro tipo de métodos de optimización, siendo en algunos casos la única opción para alcanzar las soluciones globales. El uso cooperativo de dichas metaheurísticas puede ayudar a incrementar esta eficiencia, no sólo por la reducción en los tiempos de computación al usar diferentes procesos en paralelo, sino porque el intercambio de información entre ellas potencia su eficacia.

En las siguientes secciones se presenta una metaheurística cooperativa basada en búsqueda dispersa para la resolución de problemas de optimización global. Este método ha sido implementado en los lenguajes Matlab y R, y los códigos se encuentran disponibles en internet[1]. Aunque ha sido diseñado para su aplicación en el campo de biología de sistemas, su uso puede ser extendido a cualquier otro campo científico o ingenieril que presente problemas con las mismas características.

2 Búsqueda dispersa aumentada (*eSS*)

La búsqueda dispersa aumentada (o *eSS*, enhanced Scatter Search) es una metaheurística de tipo evolutivo desarrollada recientemente y cuyas prestaciones se

[1] `http://www.iim.csic.es/~gingproc/meigo.html`

comparan favorablemente a otras técnicas del estado del arte [7, 8]. Se trata de un método basado en poblaciones, que mantiene en todo momento un conjunto de soluciones de referencia, *RefSet*. Estas soluciones se escogen en función de su calidad y diversidad. La calidad de una solución se mide por el valor de su función de coste; la diversidad de soluciones implica que éstas cubran las diferentes áreas del espacio de parámetros.

Una característica crucial de esta técnica es que orienta de forma sistemática, no aleatoria, la exploración del espacio de búsqueda. Así, explotando la información recogida por cada solución, se conduce la búsqueda hacia áreas prometedoras. Para lograr una mejora adicional del rendimiento esta técnica evolutiva se combina con búsquedas locales, las cuales pueden acelerar la convergencia hacia soluciones óptimas cuando se emplean desde puntos cercanos a ellas. El pseudocódigo correspondiente se muestra en el Algoritmo 1.

3 Cooperación

En esta sección se muestra cómo las prestaciones del algoritmo *eSS* se pueden mejorar al usarlo en un entorno paralelo con un mecanismo de cooperación.

En un entorno multiprocesador se lanzan, en paralelo, varios hilos o "threads", todos los cuales implementan el método *eSS*. Los hilos cuentan con diferentes ajustes y/o inicializaciones aleatorias, y se les permite intercambiar informacin en determinados instantes. Siguiendo la clasificación propuesta en [3], esta metodología se caracteriza de la siguiente forma:

1. La información compartida es la mejor solución encontrada hasta el momento y el conjunto de puntos de referencia *RefSet*, que contiene información sobre la diversidad de las soluciones.
2. Todos los hilos intercambian información.
3. La cooperación tiene lugar a intervalos fijos, τ.
4. El número de programas que se ejecutan simultáneamente es η.

Con el fin de introducir un balance apropiado entre diversificación e intensificación, cada hilo tiene un diferente grado de "agresividad". Los hilos más "agresivos" se concentran en la intensificación y llevan a cabo más búsquedas locales, mientras que los más "conservadores" se ocupan de la diversificación (búsqueda global). Así, las anteriores consiguen acelerar la convergencia a los óptimos en las regiones más "suaves" o convexas del espacio de parámetros, mientras que las últimas aumentan la robustez del algoritmo en las áreas más "rugosas". Un esquema de este algoritmo se muestra en la Fig. 1, y su pseudo-código se resume en el Algoritmo 2. En éste, el texto sombreado en gris corresponde a la parte del código implementada en paralelo por cada uno de los procesadores esclavos; los cálculos restantes son llevados a cabo por el procesador maestro.

Nuestro algoritmo cooperativo de búsqueda dispersa aumentada ha sido implementado en dos plataformas diferentes: Matlab y R. Además, se dispone de una herramienta para definir y resolver problemas en la plataforma Python mediante llamada a la implementación en R. Estas implementaciones así como

Algoritmo 1 Pseudocódigo básico de *eSS*

Inicializar variables
Generar un conjunto de *ndiverse* soluciones
Generar *RefSet* inicial com *dim_refset* soluciones de alta calidad y elementos aleatorios
repeat
  Ordenar *RefSet* por calidad $\left[x^1, x^2, \ldots, x^{dim_refset}\right]$ de forma que $f(x^i) \leq f(x^j)$ donde $i, j \in [1, 2, \ldots, dim_refset]$ e $i < j$
  for $i = 1$ to *dim_refset* **do**
    Combinar x^i con los restantes miembros de la población para generar un conjunto de *dim_refset* soluciones nuevas, $\mathbf{offspring}^i$
    $x^i_{\text{off}} =$ mejor solución de $\mathbf{offspring}^i$
    if x^i_{off} mejora a x^i **then**
      x^i_{off} reempleza a x^i
    end if
  end for
  $\mathbf{offspring} = \begin{pmatrix} \mathbf{offspring}^1 \\ \mathbf{offspring}^2 \\ \ldots \\ \mathbf{offspring}^n \end{pmatrix}$ con $n = dim_refset$
  if Se aplica búsqueda local en la iteración actual **then**
    Escoger z como miembro de **offspring** balanceando calidad y diversidad
    Lanzar búsqueda local desde z para obtener z^*
    if z^* es un nuevo óptimo local **then**
      Actualizar lista de óptimos locales encontrados añadiendo z^*
    end if
    if $f(z^*) < fbest$ **then**
      Actualizar $xbest, fbest$
    end if
    $neval = 0$
  end if
  $neval = neval +$ evaluaciones de la función en la iteración actual
until satisfacer el criterio de parada
Aplicar búsqueda local final a *xbest* y actualizarlo

detalles técnicos y casos de estudio se pueden encontrar en la página web mantenida por nuestros grupos bajo el nombre genérico de *MEIGO*, acrónimo de "MEtaheuristics for systems bIology and bioinformatics Global Optimization". En Matlab, la gestión de la cooperación de los hilos se hace a través de la herramienta *jPar* [9], mientras que en R, esta gestión puede ser llevada a cabo por diferentes paquetes, como *snowfall* [10] o *Rsge*[2].

4 Aplicación a casos de estudio

En esta sección se ilustrará la aplicación de la búsqueda dispersa cooperativa descrita en este estudio a dos tipos diferentes de problemas. En el primer caso los

[2] `http://cran.r-project.org/src/contrib/Archive/Rsge/`

Algoritmo 2 Pseudo-código básico de la estrategia cooperativa

```
MAESTRO, PROCESADOR 0: Initialización
Fijar ajustes comunes a todos los hilos: τ, η, n_iters
Fijar matriz de ajustes de agresividades:
Ψ_{η×4} = { dim_refset_{η×1}, ndiverse_{η×1}, local.n2_{η×1}, balance_{η×1} }
Inicializar matriz global de conjuntos de referencia: Global_ref = [ ]
for i = 1 to n_iters do
  for j = 1 to η do
    ESCLAVO, PROCESADOR j (procesado concurrente): llevar a cabo
    optimización de acuerdo al Algoritmo 1
    (El hilo j usa el vector de ajustes de agresividad Ψ_{j,:} y puntos iniciales
    Global_ref)
  end for
  if CPUtime ≥ τ then
    for j = 1 to η do
      Leer resultados del hilo j
      if ref_j ∉ Global_ref then
        Global_ref = [Global_ref, ref_j]
      end if
    end for
  end if
end for
Solución final = mejor solución de Global_ref
```

casos de estudios corresponden a funciones matemáticas complejas de dimensión elevada usadas habitualmente para la evaluación de metodos de optimización global, mientras que en el apartado siguiente se analizarán dos casos de estimación de parámetros en el ámbito de la biología de sistemas. Todos los experimentos se realizaron en las estaciones de cálculo de nuestro grupo (procesador Xeon Quadcore, 2.50 GHz, 3 GB/core)

4.1 Funciones "benchmark" de a gran escala

En este apartado, y como paso preliminar a la aplicación del método a casos de biología de sistemas para verificar su adecuación a problemas a gran escala, se aplica la búsqueda dispersa cooperativa sobre una serie de funciones presentadas en la Sesión Especial de Computacion Evolutiva para Optimizacion Global a Gran Escala del *2012 IEEE World Congress on Computational Intelligence (CEC@WCCI-2012)*. Las funciones objetivo presentadas en dicha sesión son típicos "benchmarks" para la evaluación de métodos numéricos de optimización global a gran escala en el dominio continuo. Se puede obtener más información acerca de las funciones, así como obtener sus códigos fuentes en diferentes lenguages en la siguiente página web: `http://staff.ustc.edu.cn/~ketang/cec2012/lsgo_competition.htm`.

Aquí presentamos la aplicación del método sobre tres de estas funciones (**f10**, **f17** y **f20**) usando la versión en R de nuestro algoritmo cooperativo. En la Fig. 2

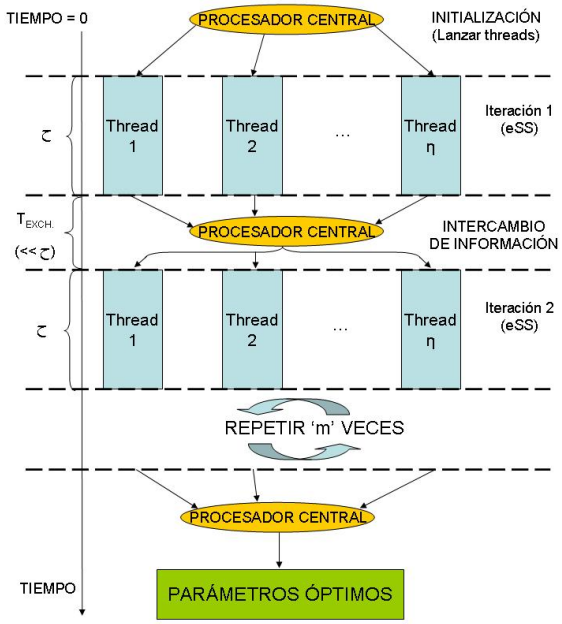
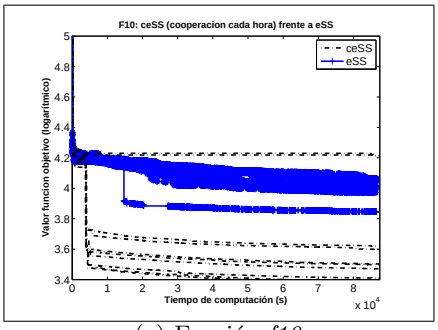


Fig. 1. Esquema del algoritmo cooperativo: η hilos paralelos intercambian información a intervalos fijos con periodicidad τ.

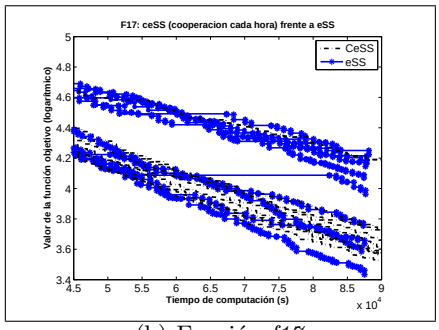
se presentan las curvas de convergencia obtenidas al optimizar las mencionadas funciones. Estas curvas representan la mejor solución obtenida frente al tiempo de cálculo. Este tiempo de cálculo es proporcional al número de evaluaciones de las funciones objetivo y, aunque podría considerarse el tiempo adicional necesario para el intercambio de información en el método cooperativo, éste es despreciable comparado con el tiempo total de ejecución dado el elevado número de evaluaciones realizadas. Se realizaron 10 ejecuciones para cada función, tanto para el método cooperativo como para el no cooperativo. Como se puede ver en la figura, la estrategia cooperativa mejora significativamente los resultados obtenidos con la estrategia no cooperativa, alcanzando mejores soluciones en el mismo horizonte de tiempo. Asimismo, estos resultados son competitivos con los presentados en el concurso de la Sesión Especial citada más arriba.

4.2 Estimación de parámetros en modelos de biología de sistemas.

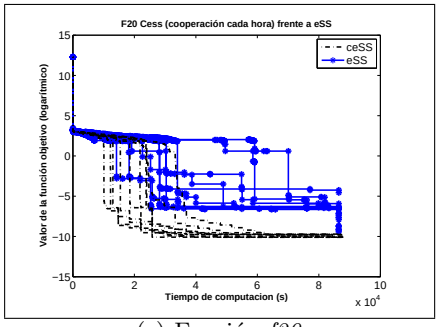
En esta sección se muestran los resultados de la aplicación del método cooperativo propuesto sobre dos problemas de estimación de parámetros en el campo de



(a) Función *f10*



(b) Función *f17*



(c) Función *f20*

Fig. 2. Curvas de convergencia para las funciones *f10, f17* y *f20*

la biología de sistemas. Los modelos matemáticos en este área suelen ser no lineales, dinámicos, multimodales, y, en muchos casos, de elevada dimensión, como es el caso de los ejemplos aquí presentados.

El tipo de problema que se pretende resolver es el de encontrar el vector de variables de decisión $\mathbf{p}$ – formado por los parámetros de un modelo cuyo valor es desconocido – que minimiza una función de coste que mide la bondad del ajuste de las predicciones del modelo con respecto a los datos. Matemáticamente se formula como un problema de programación no lineal, cuyo objetivo es encontrar el vector $\mathbf{p}$ que minimice

$$J = \sum_{\epsilon=1}^{n_\epsilon} \sum_{o=1}^{n_o^\epsilon} \sum_{s=1}^{n_s^{\epsilon,o}} \left(ym_s^{\epsilon,o} - y_s^{\epsilon,o}(\mathbf{p})\right)^T \times \\ \times W \times \left(ym_s^{\epsilon,o} - y_s^{\epsilon,o}(\mathbf{p})\right) \tag{1}$$

donde n_ϵ es el número de experimentos, n_o^ϵ el número de observables (variables de estado medidas) por experimento, y $n_s^{\epsilon,o}$ es el número de muestras por observable por experimento. Los datos medidos son $ym_s^{\epsilon,o}$, $y_s^{\epsilon,o}(\mathbf{p})$ son las predicciones del modelo, y W es una matriz de escalado que pondera las contribuciones de los observables. La minimización de J está sujeta a ciertas restricciones diferenciales

y algebraicas que pueden estar relacionadas con la dinámica del sistema, las condiciones iniciales, o los límites en los valores de los parámetros [11].

Los modelos testados corresponden al metabolismo de la bacteria *Escherichia Coli* y consideran diferentes niveles regulatorios: el *Modelo 1* describe su metabolismo de carbono central (CCM) modelado de acuerdo al formalismo "common modular rate law" [12], y el *Modelo 2* conecta el CCM con regulación enzimática y transcriptómica [13]. Se trata de modelos de gran escala con cientos de parámetros, de elevada complejidad y exigencia computacional. Para su calibración se generaron datos pseudo-experimentales. Las caractersticas de ambos modelos se recogen en la Tabla 1.

Tabla 1. Características de los modelos usados como casos de estudio.

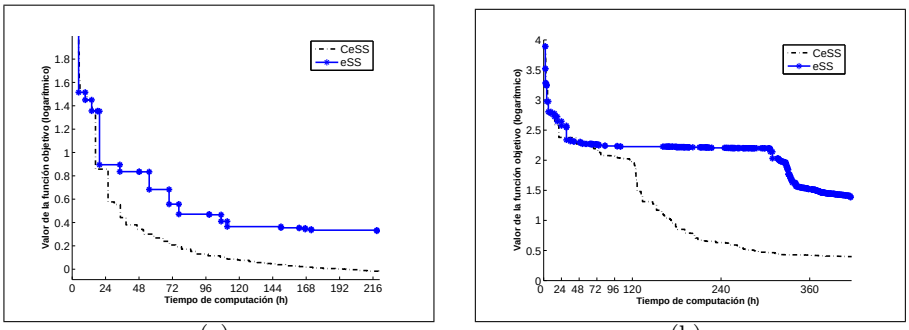
	Modelo 1	**Modelo 2**
Niveles	Metabólico	Metabólico y transcriptómico
Parámetros	918	193
Estados	115	47
Observables	115	47
Medidas	1150	16732
Estático / Dinámico	Estático	Dinámico

Como paso previo a la utilización del método propuesto, se probó la eficacia de varios métodos locales mediante un procedimiento "multistart", lanzando búsquedas desde varios puntos escogidos aleatoriamente entre el espacio de parámetros. Ninguna de las búsquedas alcanzó valores próximos al óptimo global, lo que revela el carácter multimodal de estos modelos.

Los resultados se muestran en la Fig. 3, donde las curvas de convergencia representan el logaritmo del menor valor de la función objetivo encontrado en cada instante en funcion del tiempo de computación. Para ambos modelos se compara el rendimiento de 10 hilos individuales, no cooperativos (línea negra) con el de 10 hilos cooperativos (línea verde), de forma que el esfuerzo computacional es equivalente, para garantizar una comparación justa. Las curvas presentadas en la figura corresponden a la mejor ejecución (en términos de valor final de función objetivo) para ambos métodos considerando las 10 ejecuciones realizadas con cada uno. Estos resultados muestran que la cooperación consigue acelerar la convergencia, reduciendo los tiempos de computación hasta en un 85%.

5 Conclusiones

En este trabajo se ha presentado un método de optimización global basado en la cooperación entre diferentes hilos de un algoritmo de búsqueda dispersa. Ha sido diseñado para resolver problemas a gran escala, explotando las capacidades



(a) (b)

Fig. 3. Curvas de convergencia para los modelos de estimación de parámetros en biología de sistemas. Hilos con *eSS* cooperativo (verde) frente a no cooperativo (negro), para Modelo 1 (a) y Modelo 2 (b). Se compara el rendimiento de 10 hilos individuales, no cooperativos, con el de $\eta = 10$ hilos cooperativos. La figura muestra el logaritmo del valor de la función objetivo en función del tiempo de computación.

de entornos de computación paralelos. Cada hilo corre en un procesador descentralizado e implementa la metaheurística, mientras que un procesador central gestiona el intercambio de información entre hilos. El método ha sido probado con distintas funciones que constituyen el estado del arte para la evaluación de métodos de optimización global a gran escala, y con dos casos de estimación de parámetros en biología de sistemas. En todos los casos el mecanismo cooperativo proporciona mejoras en el rendimiento. El método es de aplicación general, y puede ser empleado en otros problemas de optimización tanto de biología de sistemas como de otras áreas.

Agradecimientos

Este trabajo ha recibido financiación de la Unión Europea a través del proyecto "BioPreDyn" (EU FP7 289434), del MINECO a través del proyecto "MultiScales" (DPI2011-28112-C04-03 y DPI2011-28112-C04-04), y del CSIC a través del proyecto intramural "BioREDES" (PIE-201170E018). Los autores agradecen a Wolfram Liebermeister por proporcionar el *Modelo 1*.

Referencias

1. Crainic, T.G., Toulouse, M.: Introduction to the special issue on parallel metaheuristics. Journal of Heuristics **8**(3) (2002) 247–249
2. Alba, E., Luque, G., Nesmachnow, S.: Parallel metaheuristics: Recent advances and new trends. International Transactions in Operational Research **20**(1) (2013) 1–48
3. Toulouse, M., Crainic, T.G., B., S.: Systemic behavior of cooperative search algorithms. Parallel Comput **30** (2004) 57–79

4. Masegosa, A.D., Pelta, D., del Amo, I.G., Verdegay, J.L.: On the performance of homogeneous and heterogeneous cooperative search strategies. In: Nature Inspired Cooperative Strategies for Optimization (NICSO 2008). Volume 236 of Studies in Computational Intelligence. (2009) 287–300
5. Kitano, H.: Foundations of Systems Biology. The MIT Press (2001)
6. Banga, J.R.: Optimization in computational systems biology. BMC Systems Biology **2**(1) (2008) 47
7. Egea, J.A., Balsa-Canto, E., García, M.S.G., Banga, J.R.: Dynamic optimization of nonlinear processes with an enhanced scatter search method. Industrial & Engineering Chemistry Research **48**(9) (2009) 4388–4401
8. Egea, J.A., Martí, R., Banga, J.R.: An evolutionary algorithm for complex process optimization. Computers & Operations Research **37**(2) (2010) 315–324
9. Karbowski, A., Majchrowski, M., Trojanek, P.: jpar–a simple, free and lightweight tool for parallelizing matlab calculations on multicores and in clusters. 9th International Workshop on State-of-the-Art in Scientific and Parallel Computing (2008)
10. Knaus, J.: Developing parallel programs using snowfall (Available at http://cran.r-project.org/web/packages/snowfall/vignettes/snowfall.pdf). (2010)
11. Walter, E., Pronzato, L.: Identification of Parametric Models from Experimental Data. Springer, Berlin, Germany (1997)
12. Liebermeister, W., Uhlendorf, J., Klipp, E.: Modular rate laws for enzymatic reactions: thermodynamics, elasticities, and implementation. Bioinformatics **26**(12) (2010) 1528–1534
13. Kotte, O., Zaugg, J.B., Heinemann, M.: Bacterial adaptation through distributed sensing of metabolic fluxes. Molecular Systems Biology **6** (2010) 355

Búsqueda de Vecindad Variable secuencial y paralela: una aplicación al problema de la maximización del corte

D. Concha, D. Fernández-López, J. Sánchez-Oro, A.S. Montemayor, J.J. Pantrigo, and A. Duarte[1]

Dept. Ciencias de la Computación, Universidad Rey Juan Carlos (Spain)
{ david.concha, david.fernandez, jesus.sanchezoro, antonio.sanz, juanjose.pantrigo, abraham.duarte }@urjc.es

Resumen En este trabajo se propone un análisis comparativo de la implementación paralela y secuencial de un algoritmo basado en el esquema *Variable Neighborhood Search*. La paralelización se lleva a cabo utilizando NVIDIA CUDA sobre GPU. Para ilustrar el comportamiento de ambas propuestas se utiliza el problema de la maximización del corte (MaxCut Problem - MCP). Dado un grafo G no dirigido, se define un corte como la división del conjunto de vértices de G en dos subconjuntos S y S' de forma que $S' = V \setminus S$. El MCP consiste entonces en encontrar el corte de máximo valor en G de entre todos los posibles. Para resolver dicho problema se propone un esquema basado en la estrategia *Basic VNS*. La paralelización se aplica a la búsqueda local, analizando las diferencias entre la búsqueda local paralela y la secuencial. Los resultados obtenidos muestran como la paralelización de la búsqueda local en GPU consigue reducir en gran medida el tiempo de ejecución, proporcionando soluciones de mayor calidad.

Keywords: maxcut, paralelismo, NVIDIA CUDA, variable neighborhood search, metaheurísticas

1. Introducción

Se define un grafo $G = (V, E)$, con $n = |V|$ y $m = |E|$, como un conjunto de vértices V y el conjunto de aristas que los unen E. Siendo w_{ij} el peso asociado con la arista $(i, j) \in E$, se define un $corte(S, S')$ como la división de V en dos conjuntos S y $S' = V \setminus S$. El valor de dicho corte se calcula mediante la expresión

$$corte(S, S') = \sum_{u \in S, v \in S'} w_{uv}$$

La Figura 1 muestra un ejemplo de grafo G no dirigido (izquierda), así como uno de los posibles cortes para dicho grafo (derecha). El corte de la figura se calcula de la siguiente manera: $corte(S, S') = 1 + 0 + (-1) + (-1) + 0 = -1$.

El problema de la maximización del corte (MaxCut Problem - MCP) consiste en encontrar el corte de máximo valor en G de entre todos los posibles. En [10]

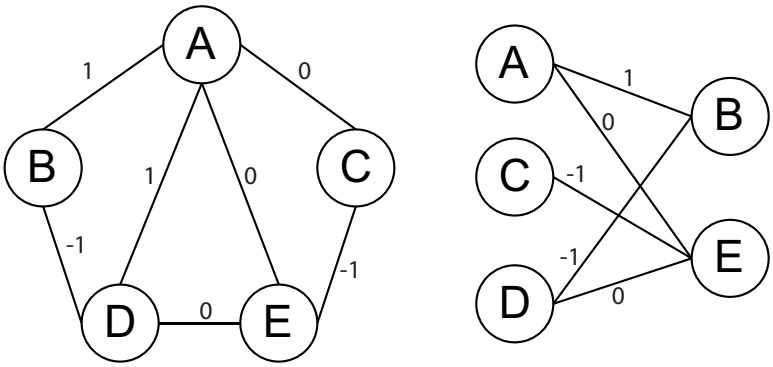


Figura 1. Ejemplo de grafo no dirigido con pesos en las aristas (izquierda) y uno de los posibles cortes (derecha)

se demuestra que el MCP es NP-difícil. El MCP es un problema de optimización con un gran número de aplicaciones prácticas ([4,5]) que ha sido ampliamente trabajado en los últimos años.

La primera solución propuesta para el MCP se encuentra en [18] y más tarde se presentan algunas variantes ([17,11]). En [8] se obtienen cotas superiores y un método heurístico con un rendimiento garantizado de 0.878. En [13] se propone una paralelización de dicho método. En cuanto a trabajos recientes, en [2] se propone un método basado en una relajación que genera soluciones mejores que los métodos anteriores en un menor intervalo de tiempo. En [6] se proponen seis algoritmos diferentes basados en metodologías *Variable Neighborhood Search* (VNS), *Greedy Randomized Adapted Search Procedures* (GRASP) y *Path Relinking* (PR). En dicho trabajo se muestra como la metodología VNS junto con PR obtiene soluciones de alta calidad pero necesitando largos períodos de tiempo. Finalmente, en [14] se propone un algoritmo basado en *Scatter Search* combinado con *Path Relinking* que genera soluciones de alta calidad en un menor período de tiempo.

La paralelización de algoritmos está siendo muy estudiada en los últimos años. En [16] aparece una de las primeras paralelizaciones sobre GPU (Graphics Processing Units) de algoritmos heurísticos, aplicado sobre el mismo problema que se trata en este trabajo. En dicho trabajo se propone una nueva representación del problema, modelando la solución basándose en las aristas en lugar de en los vértices. Dicho modelo era necesario en ese momento debido a las limitaciones técnicas que existían a la hora de trabajar con el hardware gráfico. Con la evolución de los paradigmas paralelos de programación hoy día es más sencillo aprovechar el potencial del hardware gráfico, pero cabe destacar la importancia de hacer una buena adaptación del modelo del problema para que sea eficiente en la plataforma donde se van a realizar las pruebas, ya que, de otra forma, es posible que el modelo paralelo llegue a ser más lento que el secuencial. Además, en [7,3] se proponen varias estrategias de paralelización de la metodología Varia-

ble Neighborhood Search aplicadas al problema de la p-mediana. Los sistemas paralelos se pueden clasificar mediante la taxonomía de arquitecturas paralelas de Flynn, la cual se basa en dos criterios: el número de flujos de instrucciones (FI) y el de flujos de datos (FD), definiendo cuatro modelos diferentes: SISD (1-FI, 1-FD), SIMD (1-FI, varios FD), MISD (varios FI, 1-FD) y MIMD (varios FI, varios FD). En el contexto de las arquitecturas gráficas (GPU), SIMD es el modelo más común, siendo el elegido para este trabajo. En la actualidad, las tecnologías más utilizadas para la paralelización de algoritmos sobre GPU son OpenCL y NVIDIA CUDA. OpenCL es un estándar multiplataforma de programación de procesadores modernos (tanto CPU como GPU) que permite tanto paralelismo de datos como de tareas. Por otro lado, NVIDIA CUDA es una plataforma y un modelo de programación orientado al paralelismo de datos sobre sobre GPU. En este trabajo se elige NVIDIA CUDA como tecnología de paralelización debido a que presenta una mayor estabilidad y rendimiento.

NVIDIA CUDA es un API (*Application Program Interface*) para la programación de propósito general de procesadores gráficos NVIDIA. Si bien el paradigma de programación es muy similar al tradicional, es necesario conocer ciertos aspectos internos del hardware gráfico para poder aprovechar el rendimiento que puede llegar a obtener.

La arquitectura unificada de las GPU de NVIDIA está formada por varios multiprocesadores de multitud núcleos cada uno, llegando en total a ofrecer cientos ó miles de núcleos CUDA (CUDA-cores), denominándose arquitectura masivamente paralela. Para aprovechar el paralelismo masivo de NVIDIA CUDA es necesario lanzar una gran cantidad de hilos de ejecución, que se agrupan en bloques. Un bloque es ejecutado por un multiprocesador y los distintos núcleos de dicho multiprocesador se encargan de ejecutar los hilos de ese bloque. Una tarea es llevada a cabo por varios bloques que se agrupan en un *grid*. Se denomina kernel a aquella función que es ejecutada en la GPU por un *grid*. Un kernel tiene el código que ejecuta un hilo, y como el modelo de ejecución de una tarjeta gráfica es SIMD, todos los hilos ejecutan el mismo código.

NVIDIA CUDA aplica una serie de instrucciones a un conjunto muy amplio de datos. Por lo tanto, para sacar el mayor rendimiento posible es necesario conocer la jerarquía de la memoria en NVIDIA CUDA para poder transmitir los datos suficientemente rápido.

Inicialmente los datos están en memoria principal (CPU). La primera memoria que muestra NVIDIA CUDA es la que denomina *memoria global*. Esta memoria es similar a la memoria principal de la CPU pero en la tarjeta gráfica. Todos los hilos del *grid* pueden leer y escribir a esta memoria pero su acceso es el más lento de todos (a pesar de que esta memoria tiene mayor ancho de banda que la principal). A continuación se puede encontrar la memoria constante, que mejora el rendimiento de la memoria global gracias a un sistema de *cacheo* a cambio de ser una memoria relativamente pequeña de solo lectura. Dentro de cada bloque se encuentra la memoria compartida, la cual es una memoria de lectura y escritura muy rápida, pero que solo existe dentro de cada bloque, por lo tanto solo permite compartir información entre los hilos de un mismo bloque,

no siendo posible la comunicación entre bloques. Por ultimo, los registros aparecen en cualquier procesador, pero que en el caso de las GPU es un factor a tener en cuenta, ya que su número es limitado y hacer un uso intensivo de ellos puede traducirse en movimientos de datos a través de la memoria global, lo que repercute en el rendimiento.

2. Búsqueda de Vecindad Variable

El objetivo de este trabajo es llevar a cabo una comparativa entre un algoritmo secuencial tradicional y su implementación paralela sobre GPU para ilustrar el comportamiento de la paralelización de algoritmos sobre GPU aplicado sobre el problema del MCP. Para ello se propone un algoritmo basado en la metodología *Variable Neighborhood Search* (VNS), propuesta en [15]. VNS es una metaheurística para resolver problemas de optimización basada en un cambio sistemático de estructuras de vecindad. Existen diversas variantes de VNS, entre las que destacan: *Variable Neighborhood Descent* (VND), *Reduced VNS* (RVNS), *Basic VNS* (BVNS), *Skewed VNS* (SVNS), *General VNS* (GVNS) y *Variable Neighborhood Decomposition Search* (VNDS), entre otras (ver [9] para más información). En este trabajo se propone una variante basada en BVNS que combina cambios de vecindad estocásticos y deterministas, descrita en el Algoritmo 1. En este problema, se define la vecindad k de una solución φ, $N_k(\varphi)$ como el conjunto de soluciones en los cuáles hay exactamente k nodos situados en diferentes posiciones del corte respecto a la solución de la que se parte (φ).

Algorithm 1 BVNS(k_{max}, t_{max})

1: $\varphi \rightarrow Construir()$
2: **repeat**
3: $k \rightarrow 1$
4: **repeat**
5: $\varphi' \rightarrow Shake(\varphi, k)$
6: $\varphi'' \rightarrow BusquedaLocal(\varphi'')$
7: $CambioVecindad(\varphi, \varphi'', k)$
8: **until** $k = k_{max}$
9: $t \rightarrow Time()$
10: **until** $t = t_{max}$
11: **return** φ

El algoritmo comienza a partir de una solución inicial (paso 1). En este trabajo se considera una solución inicial aleatoria. El algoritmo se ejecuta durante un tiempo de ejecución predefinido t_{max} (pasos 2-10). El procedimiento de búsqueda comienza en la primera vecindad (paso 3). El algoritmo entonces lleva a cabo una serie de cambios estocásticos en las estructuras de vecindad hasta alcanzar la mayor vecindad permitida (pasos 4-8). Dentro del bucle principal se pueden distinguir los tres métodos más importantes: $Mezcla$, $BusquedaLocal$ y

CambioVecindad. En el método $Mezcla$ se genera una solución en la vecindad k actual. A la solución generada se le aplica el procedimiento de búsqueda para encontrar un óptimo local. Por último, el cambio de vecindad comprueba si la nueva solución es mejor que la mejor solución encontrada hasta el momento. En caso afirmativo, se actualiza la mejor solución y se vuelve a la primera vecindad. En caso contrario, se pasa a la siguiente vecindad. El algoritmo finaliza cuando se ha alcanzado el tiempo máximo de ejecución permitido t_{max}.

La estructura de la solución utilizada es un vector de n elementos donde cada elemento representa a un nodo del grafo, que tendrá valor 0 si pertenece a un lado del corte y 1 si pertenece al otro lado. De esta forma se propone como movimiento el cambio de lado de corte de un nodo. Es decir, el movimiento $Cambio(v, \varphi, G)$ vendrá definido como:

$$Cambio(v, \varphi, G) = \begin{cases} \varphi(v) \to 1 \; si \; \varphi(v) = 0 \\ \varphi(v) \to 0 \; si \; \varphi(v) = 1 \end{cases}$$

El procedimiento de $Mezcla(\varphi, k)$ propuesto en este trabajo selecciona k vértices de manera aleatoria y realiza con ellos el movimiento *Cambio* anteriormente definido, alterando la solución de forma aleatoria para incrementar la diversidad del algoritmo.

Algorithm 2 $LS(\varphi, G))$

```
1: mejora → TRUE
2: while mejora do
3:    mejora → FALSE
4:    φ_best → φ
5:    for v ∈ V do
6:       φ' → Cambio(v, φ, G)
7:       if MCP(φ') > MCP(φ_best) then
8:          φ_best → φ'
9:          mejora = TRUE
10:      end if
11:   end for
12:   t → Time()
13: end while
14: return φ_best
```

La parte intensificadora del procedimiento se basa en las búsquedas locales. En este trabajo se presenta una búsqueda local con dos implementaciones, una de forma secuencial en CPU y la otra de forma paralela en GPU. Se trata de una búsqueda local exhaustiva en la que en cada iteración para cada vértice v de una solución φ sobre un grafo G se lleva a cabo el movimiento $Cambio(v, \varphi, G)$ y se comprueba si la solución resultante, φ' es mejor que la mejor solución encontrada hasta el momento (φ_{best}). En caso afirmativo se actualiza la mejor solución y se continúa con la siguiente iteración. La búsqueda local termina cuando no es

posible llevar a cabo el movimiento sobre ningún vértice del grafo. El Algoritmo 2 muestra el pseudocódigo de la búsqueda local propuesta.

La Figura 2 ilustra cómo se llevan a cabo los movimientos en la búsqueda local. Se define una solución al problema como un vector de dimensión igual al número de nodos, donde el conjunto al que pertenece el nodo i viene dado por el valor de la posición i del vector. Las aristas con línea continua en la Figura 2 son aquellas que afectan a la función objetivo, ya que unen dos nodos situados en diferentes conjuntos. Por otra parte, las aristas con línea discontinua son las que no contribuyen, por unir nodos del mismo conjunto. La zona sombreada separa los dos conjuntos de manera gráfica. El nodo A estará situado en la posición 1, el nodo B en la posición 2 y así sucesivamente hasta el nodo E en la posición 5. La Figura 2.(a) tiene como vector solución $\varphi_a = [1, 1, 0, 0, 0]$, mientras que el vector solución de la Figura 2.(b) sería $\varphi_b = [1, 1, 0, 1, 0]$.

Dada una solución φ sobre la que se aplicará un movimiento de cambio, se puede calcular el valor de la nueva solución únicamente atendiendo a las aristas del nodo que se mueve. El valor de la función objetivo para el caso de la Figura 2.(a) es la suma de los costes de las aristas 2, 3, 4 y 5; por otra parte, el valor para la Figura 2.(b) es la suma de los costes de las aristas 2, 4, 5, 6 y 7.

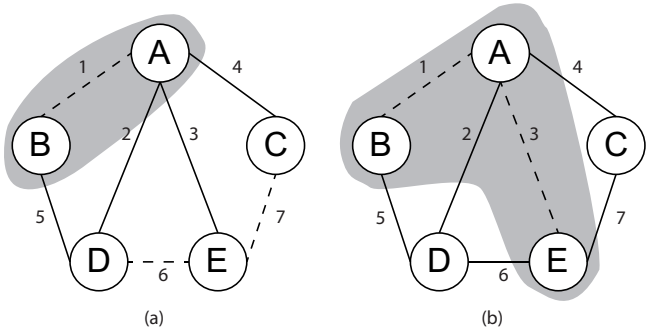


Figura 2. Ejemplo de movimiento del nodo D en la búsqueda local

Si partimos de φ_a, para calcular el valor de la solución φ_b sólo es necesario tener en cuenta las aristas del nodo a mover, en este caso el nodo D (aristas 3, 6 y 7). Si observamos la solución φ_a, las aristas cuyos nodos están en distinto conjunto, se restarán, como es el caso de la arista 3 entre los nodos A y B. Por el contrario las aristas de nodos que están en el mismo conjunto se sumarán, como son las aristas 6 y 7 entre el nodo D y los nodos C y E, respectivamente. Esto es así ya que al mover un nodo, las aristas que antes sumaban por estar en distinto conjunto, dejaran de hacerlo, y las que no sumaban pasaran a hacerlo.

Dada una pareja de nodos i y j, la operación en GPU que calcula el aporte que realiza a la función objetivo la arista entre i y j es la siguiente:

$$w_{(i,j)} = \begin{cases} M[i,j] \text{ si } S[i] = S[j] \\ -M[i,j] \text{ si } S[i] <> S[j] \end{cases}$$

siendo $w_{(i,j)}$ el peso que aporta la arista (i, j) a la solución (con valor 0 si $(i, j) \notin E$), S el vector solución antes del movimiento y M la matriz de adyacencias.

Posteriormente se suman todas las aportaciones de un nodo, lo que refleja cómo afecta al valor de solución realizar un movimiento sobre ese nodo. Se elije el nodo que más aporte, y si el aporte es positivo, se realiza dicho movimiento. El algoritmo se repite hasta no encontrar ningún movimiento que mejore.

3. Resultados experimentales

Esta sección está destinada a comparar el rendimiento del algoritmo BVNS utilizando las dos búsquedas locales propuestas. El algoritmo ha sido implementado en C++ y NVIDIA CUDA y todos los experimentos se han llevado a cabo sobre un Intel Core 2 Duo E8400 CPU (3 GHz) con 4 GB de RAM. La GPU utilizada para la implementación paralela es el modelo NVIDIA GTX-680. El conjunto de instancias utilizado se compone de 88 instancias divididas en tres conjuntos diferentes:

1. **Conjunto 1**: Se compone de instancias generadas por [12] que utiliza un generador de grafos para crear 54 instancias cuyo número de nodos varía entre 800 y 3000. Se trata de grafos cuyos pesos pueden valer -1,0,1 utilizados en [6,2].
2. **Conjunto 2**: Contiene 30 instancias descritas en [6]. Las 10 primeras son de pequeño tamaño con un promedio de 128 vértices y 300 aristas, las 10 siguientes de medio tamaño (promedio de 1000 vértices y densidad 0.60 %) y las 10 últimas son instancias grandes, con un promedio de 2744 vértices y densidad igual a 0.22 %. Los pesos de las aristas varían entre -1,0 y 1.
3. **Conjunto 3**: Se compone de cuatro instancias obtenidas del 7th DIMACS Implementation Challenge. El número de vértices está entre 1536 y 10125. En [2] se utilizan estas cuatro instancias mientras que en [6] se utiliza solo una de ellas.

Las instancias están disponibles en `http://www.optsicom.es/maxcut/`. En el experimento se lleva a cabo una comparativa del algoritmo BVNS utilizando la búsqueda local secuencial en CPU (VNS_CPU) frente al BVNS utilizando la búsqueda local paralela en GPU (VNS_GPU). Ambos algoritmos utilizan un valor de $k_{max} = 0.20 \cdot n$, es decir, el valor máximo que tomará k será el 20 % del número de nodos de cada instancia. La Tabla 1 muestra los resultados obtenidos en la ejecución de los dos algoritmos propuestos sobre cada conjunto de instancias descritas anteriormente. De cada algoritmo se reporta el promedio de la función objetivo (F.O.), el tiempo de ejecución medido en segundos (Tiempo), el promedio del porcentaje de la desviación estándar respecto al mejor valor del experimento (Desv.) y el número de mejores soluciones encontradas (#Mejores).

En los resultados obtenidos para el Conjunto 1 se puede apreciar como, aunque VNS_GPU es notablemente más rápido que VNS_CPU, este último proporciona soluciones ligeramente mejores, con un 0.1 % menos de desviación respecto al mejor valor obtenido en el experimento y dos mejores soluciones encontradas.

Conjunto 1				
Algoritmo	F.O.	Tiempo (seg.)	Desv. (%)	#Mejores
VNS_CPU	5071.3	249.1	0.5	28
VNS_GPU	5076.9	71.9	0.4	26
Conjunto 2				
Algoritmo	F.O.	Tiempo (seg.)	Desv. (%)	#Mejores
VNS_CPU	1053.5	301.6	1.3	10
VNS_GPU	1049.7	56.6	1.3	13
Conjunto 3				
Algoritmo	F.O.	Tiempo (seg.)	Desv. (%)	#Mejores
VNS_CPU	78426958.8	548.6	1.1	2
VNS_GPU	78426964.3	286.2	0.0	4

Tabla 1. Resultados obtenidos por los dos algoritmos propuestos sobre cada conjunto de instancias

Sin embargo, analizando los resultados del experimento sobre los Conjuntos 2 y 3, se puede observar como VNS_GPU proporciona mejores resultados en un menor tiempo de ejecución.

Una de las medidas más utilizadas para comparar algoritmos secuenciales frente a algoritmos paralelos es el *speedup* [1]. Dicha medida establece una relación entre el tiempo de ejecución paralelo y el secuencial para indicar cuánto de rápido es el algoritmo paralelo frente al secuencial. Sin embargo, esta medida solo es significativa si el algoritmo paralelo es una traducción directa de la versión secuencial del algoritmo [3]. En este trabajo se da dicha situación, por lo que resulta de interés llevar a cabo un análisis del *speedup* obtenido sobre cada conjunto de instancias y sobre el total de las mismas. El cálculo del *speedup* se lleva a cabo de la siguiente manera: $speedup = T_s/T_p$ siendo T_s el tiempo utilizado por el algoritmo secuencial y T_p el tiempo requerido por el algoritmo paralelo.

Instancias	Tiempo VNS_CPU	Tiempo VNS_GPU	Speedup
Conjunto 1	249.1	71.9	3.5
Conjunto 2	301.6	56.6	5.3
Conjunto 3	548.6	286.2	1.9
Total	280.6	76.4	3.7

Tabla 2. Comparativa del *speedup* obtenido en cada conjunto de instancias frente al *speedup* global.

Como se puede apreciar en la Tabla 2 el *speedup* obtenido sobre el conjunto total de instancias es de 3.67, siendo 5.33 el máximo y 1.92 el mínimo. Esto significa que, en términos generales, el algoritmo paralelo es entre 2 y 5 veces más rápido que el secuencial. Ya que el tiempo de ejecución necesario para VNS_GPU es mucho menor que el de VNS_CPU, se decide llevar a cabo un últi-

mo experimento para comprobar si el algoritmo VNS_GPU es capaz de generar mejores soluciones si aumentamos su tiempo de ejecución. Para ello, se establece el parámetro $k_{max} = 0.50$, en lugar del 0.20 original. La Tabla 3 muestra una comparativa de VNS_CPU frente a VNS_GPU sobre el conjunto total de instancias, estableciendo el parámetro k_{max} de VNS_GPU en 0.5 y manteniendo el de VNS_CPU en 0.2.

Algoritmo	F.O.	Tiempo (seg.)	Desv. (%)	#Mejores
VNS_CPU	3568332.9	280.6	0.7	53
VNS_GPU	3594570.8	207.6	0.5	40

Tabla 3. Comparativa del algoritmo secuencial VNS_CPU con $k = 0.2$ frente al paralelo VNS_GPU con $k = 0.5$

Como se puede observar en la Tabla 3, con esta modificación de k_{max} el algoritmo VNS_GPU sigue necesitando menos tiempo de ejecución, siendo 1.35 veces más rápido que el algoritmo secuencial. Además, la desviación estándar de la versión paralela ha disminuido hasta casi la mitad de la versión secuencial, obteniendo también un mayor número de mejores soluciones.

4. Conclusiones

En este artículo se presenta una comparativa entre un algoritmo secuencial y uno paralelo basados en la metodología *Variable Neighborhood Search*. Se ha utilizado el problema de la maximización del corte en grafos para ilustrar el comportamiento de ambos algoritmos. En concreto se propone un método *shake* así como dos búsquedas locales (una secuencial y otra paralela). La comparativa se ha llevado a cabo mediante la ejecución de ambos algoritmos sobre un total de 88 instancias ampliamente utilizadas en el estado del arte, divididas en tres subconjuntos diferentes. Los resultados experimentales muestran como el algoritmo paralelo obtiene soluciones ligeramente mejores que las obtenidas por el algoritmo secuencial. Además, la comparativa del tiempo de ejecución ilustra como el algoritmo paralelo necesita hasta 5 veces menos tiempo que el algoritmo secuencial para obtener la solución.

Agradecimientos

Este trabajo ha sido parcialmente financiado por la Comunidad de Madrid a través del proyecto con referencia S2009/TIC-1542 y por el Ministerio de Educación y Ciencia a través de los proyectos con referencias TIN2012-35632 y TIN2011-28151.

Referencias

1. Barr R.S., Hickman B.L. Reporting Computational Experiments with Parallel Algorithms: Issues, Measures, and Experts Opinions. ORSA Journal on Computing, 5(1):2–18 (1993)
2. Burer S., Monteiro R.D.C. Rank-two relaxation heuristics for max-cut and other binary quadratic programs. SIAM J. Optim, 12:503–521 (2001)
3. Crainic, T.G., Gendreau M., Hansen P., Mladenović N. Cooperative Variable Neighborhood Search for the p-Median Journal of Heuristics, 10:293–314 (2010)
4. Chang K.C., Du D.-Z. Efficient algorithms for layer asignment problems. IEEE Trans. Comput.-Aided Design, 6:67–78 (1987)
5. Chen R., Kajitani Y., Chan S. A graph-theoretic via minimization algorithm for two layer printed circuit boards. IEEE Trans. Circuits System, 30:284–299 (1983)
6. Festa P., Pardalos P.M., Resende M.G.C., Ribeiro C.C. Randomized heuristics for the max-cut problem. Optim. Methods Software, 7:1033–1058 (2002)
7. García-López F., Melián-Batista B., Moreno-Pérez J.A., Moreno-Vega M. The Parallel Variable Neighborhood Search for the p-Median Problem. Journal of Heuristics, 8:375–388 (2002)
8. Goemans M.X., Williamson D.P. Improved approximation algorithms for the max-cut and satisfability problems using semidefinite programming J. ACM, 42:1115–1145 (1995)
9. Hansen P., Mladenović N., Moreno J.A. Variable Neighborhood Search: methods and applications. Annals of Operations Research, 175(1):367–407 (2010)
10. Karp R.M. Reducibility among combinatorial problems. R.Miller, J.Tatcher, eds. Complexity of Computer Computations. Plenum Press, New York. 85–103 (1972)
11. Haglin D.J., Venkatesen M. Approximation and intractability results for the maximum cut problem and its variants. IEE Trans. Comput., 40:110–113 (1991)
12. Helmberg C., Rendl F. A spectral bundle method for semidefinite programming. SIAM J. Optim. 10:673–696
13. Homer S., Peinado M. Design and performance of parallel and distributed approximation algorithms for max cut. J. Parallel Distributed Comput., 46:48–61
14. Martí R., Duarte A., Laguna M. Advanced Scatter Search for the Max-Cut Problem INFORMS Journal on Computing, 21(1):26–38
15. Mladenović N., Hansen P. Variable Neighborhood Search Computers & Operations Research, 24(11):1097–1100
16. Montemayor, A.S., Duarte., A., Pantrigo, J.J., Cabido R.: High-Performance VNS for the Max-Cut Problem using Commodity Graphics Hardware 18th Mini EURO Conference on VNS, Tenerife, Spain (2005)
17. Poljak S., Tuza Z. A polynomial algorithm for constructing a large bipartite subgraph, with an application to a satisfiability problem. Canadian J.Math, 34:519–524 (1982)
18. Sahni S., Gonzales T. P-complete approximation problem. J. ACM 46:48–61

Diversificación de la VNS para el problema del árbol generador de etiquetado mínimo *

Sergio Consoli[1], Nenad Mladenović[2], and José A. Moreno-Pérez[3]

[1] Joint Research Centre, European Commission,
Via Enrico Fermi 2749, 21027 Ispra (VA), Italy

[2] School of Information Systems, Computing and Mathematics, Brunel University,
Uxbridge, Middlesex, UB8 3PH, United Kingdom

[3] DEIOC, Instituto Universitario de Desarrollo Regional
Universidad de La Laguna, 38271, Tenerife, Spain

Resumen En este trabajo se trata el problema MLSTP (*Minimum Labelling Spanning Tree Problem*) consitente en, dado un grafo con sus aristas etiquetadas o coloreadas, determinar el árbol generador con el menor número de etiquetas distintas en sus aristas. Este problema se ha mostrado NP-duro, aplicaciones reales y para su solución se han propuesto diversas metaheurísticas, entre las que destaca VNS (*Variable Neighbourhood Search*). Sin embargo, VNS adolece de suficiente capacidad de exploración en instancias grandes. En este trabahjo se propone una componente reactiva para dotar a la VNS de mayor capacidad de exploración. La comparación experimental analiza la mejora en el comportamiento de la metaheurística propuesta.

Keywords: Arbol generador de etiquetado mínimo, Variable Neighbourhood Search, Metaheurísticas

1. Introducción

El problema de determinar el árbol generador con el mínimo número de etiquetas o colores se ha mostrado que es un problema NP-duro con importante aplicaciones reales. El problema se plantea sobre un grafo conexo no dirigido donde sus aristas han recibido unas etiquetas o colores para clasificarlas según algunas determinadas características. En principio se pretende determinar el menor número de categorías o clases de aristas (identificadas con las etiquetas o colores) que hacen conexo al subgrafo resultante al considerar sólo las aristas de tales categorías. Dado que el problema es equivalente a determinar el árbol generador cuyas aristas pertenecen al menor número de categorías posibles, y la relación con los problemas de árbol generador mínimo se suele utilizar esta relación para aprovechar la terminología, conceptos y procedimientos. El problema

* Este trabajo ha sido parcialmente financiado por el Ministerio de Economía y Competitividad (proyecto TIN2012-32608) y el Ministerio de Industria, Turismo y Comercio (proyecto TSI-020100-2011-298).

se conoce con las siglas MLSTP correspondiente a *Minimum Labelling Spanning Tree Problem*.

La prueba de que se trata de un problema NP-duro se puede encontrar en [3]. Aplicaciones en el diseño de redes de comunicaciones se pueden encontrar en [15], en la compresión de datos en [6] y en sistemas de transporte multimodal en [14]. Una red de transporte multimodal puede ser representada por un gráfo en el que se asigna a cada enlace de una etiqueta, denotando la compañía que gestiona ese enlace. En este contexto, se trata de establecer el menor numero de compañías que pueden proporcionar un servicio completo entre todos los nodos de la red [16]. Por lo tanto, el objetivo es encontrar un árbol de expansión del grafo que utilice el mínimo número de etiquetas posible. Esto permitirá simplificar las tareas de gestión de la red o de construcción en algunos casos. Un ejemplo práctico de esta situación de redes de transporte multimodal se presenta durante eventos de crisis humanitarias como, por ejemplo, erupciones volcánicas, amenazas terroristas, inundaciones, tsunamis, etc [12]. En la gestión de las situaciones delicadas que se presentan en las crisis en necesario contar con diferentes tipos de intervención humana y reorganizar dinámicamente toda la red de transporte de la zona dañada, teniendo en cuenta la existencia de zonas inaccesibles o prohibidas. El objetivo consiste en garantizar un servicio de transporte básico mínimo, entre las principales ciudades, los hospitales, los aeropuertos, u otros servicios, con el número mínimo de diferentes compañías de transporte y/o empresas.

El resto del trabajo se organiza de la siguiente forma. En la siguiente sección se describe formalmente el problema y las heurísticas constructivas básicas. En la sección 3 se describen brevemente las diferentes metaheurísticas propuestas. En la siguiente sección se describen las componentes de mejora de la diversificación de la VNS. En la sección 5 se muestran los resultados de los experimentos comparativos desarrollados. Finalmente, el trabajo concluye con la sección de conclusiones y las principales referencias.

2. Formulación del problema

El problema del árbol generador de coloración o etiquetado mínimo (MLSTP) se formula como un problema sobre un grafo no dirigido etiquetado. Sea $G = (V, A, L)$ un grafo no dirigido, conexo y etiquetado, donde V es un conjunto de n vértices, A es un conjunto de m aristas sobre V, y L un conjunto de ℓ etiquetas asignadas a las aristas de A de forma que cada arista $a \in A$ tiene una única etiqueta $l(a) \in L$. El propósito consiste en encontrar un árbol generador T de G tal que el conjunto de etiquetas de T, $L(T) = \{l \in L | \exists a \in A : L(a) = l\}$, es de tamaño mínimo. En la figura 1 se muestra a la izquierda un ejemplo de un grafo etiquetado y a la derecha el árbol generador de coloración mínima; la solución óptima del MLSTP.

Aunque una solución final del problema sea un árbol generador, es más adecuado trabajar en primer lugar con conjuntos de etiquetas. Una solución factible vendrá dada como un conjunto de etiquetas o colores $C \subseteq L$, tal que las aristas con etiquetas en C dan lugar a un subgrafo conexo de G dado por

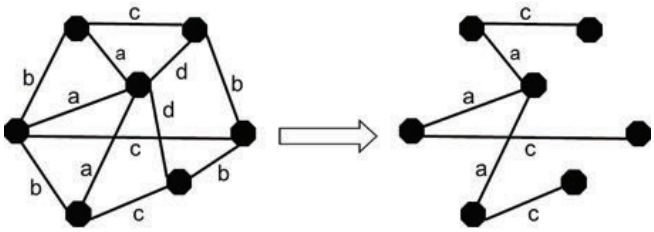


Figura 1. Se muestra un grafo y la solución del MLSTP.

$G(C) = (V, A(C))$ donde $A(C) = \{a \in A/l(a) \in C\}$. Dado un conjunto de colores C, cualquier árbol generador de $G(C)$ tiene como máximo $|C|$ etiquetas. Por tanto, si C es una solución factible de cardinal mínimo, entonces cualquier árbol generador de C es una solución óptima del MLSTP [19].

El MLSTP fue introducido en [3] quien demostró que es un problema NP-duro y proporcionó una heurística constructiva. Esta heurística conocida por las siglas MVCA (*Maximum Vertex Covering Algorithm*) tuvo que ser revisado en [13]. El procedimiento se inicia con un grafo vacío; sin aristas. Sucesivamente, se añade la etiqueta que más reduzca el número de componentes conexas, hasta que quede una única componente conexa. En [13] también se probó que este algoritmo produce una solución con cardinal no más de $(1 + 2\log n)$ veces el óptimo. Posteriormente, en [18] se mostró que el algoritmo es una $(1+\log(n-1))$-aproximación de la solución óptima, para $n > 1$.

En [1] se utiliza un enfoque diferente usando técnicas de búsqueda local basada en entornos de j-intercambios a una versión restringida del problema MLST. En este trabajo se demostraban un cierto número de resultados sobre complejidad y se demostró que si cada etiqueta aparece a lo sumo dos veces en el grafo, el MLSTP es resoluble en tiempo polinomial. En [20] se obtienen cotas más ajustadas que las de [18]. Para cualquier grafo con frecuencia de etiqueta acotada por b, se demostró que la cota del peor caso del MVCA es el b-ésimo número armónico H_b dado por

$$H_b = \sum_{i=1}^{b} \frac{1}{i} = 1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{b}.$$

Además, construyen una familia de grafos del peor caso tales que la solución del MVCA es H_b veces la solución óptima. Puesto que $H_b < (1 + \log(n-1))$ y $b \leq (n-1)$ (ya que de lo contrario, el subgrafo inducido por las etiquetas de máxima frecuencia contiene un ciclo y se pueden eliminar con seguridad las aristas del ciclo), se alcanza la cota ajustada H_b, por lo tanto, proporcionando una mejora de la cota anterior $(1 + \log(n-1))$ dada en [18].

La regla general propuesta [13] de seleccionar la etiqueta que minimiza el número total de componentes conexas en cada paso, da rápidamente soluciones de buena calidad. Sin embargo se presentan problemas cuando en un paso concreto

varias etiquetas dan lugar al mismo número mínimo de componentes conexas. Dado que frecuentemente hay varias etiquetas con este número mínimo de componentes conexas, los resultados dependen principalmente de la regla elegida para seleccionar un candidato de este conjunto. Si se elige la primera etiqueta de este conjunto, los resultados se ven claramente afectados por la ordenación de las etiquetas. Por lo tanto, diferentes ejecuciones del algoritmo pueden dar lugar a diferentes soluciones con un número diferente de las etiquetas.

En [21] se implementan diversas versiones para mejorar el rendimiento del MVCA centrándose en las etiquetas añadidas inicialmente. Las etiquetas son inicialmente ordenadas de acuerdo con sus frecuencias (número de aristas con cada etiqueta), desde la etiqueta de frecuencia más alta hasta la de frecuencia más baja. La idea es usar en la primera iteración un cierto porcentaje de las etiquetas más frecuentes. En una primera versión el porcentaje usado es del 10 %, por tanto se considera sólo cada una de las $\ell/10$ etiquetas más frecuentes como primera etiqueta del conjunto C. A continuación se ejecuta MVCA para determinar las restantes etiquetas y finalmente se selecciona la mejor de las $\ell/10$ soluciones resultantes. Otra versión modificada en [21] es similar a la anterior pero probando el 30 % de las etiquetas de mayor frecuencia en el paso inicial. En [21] también se propone otra manera de modificar MVCA. En ella se considera en cada paso los tres etiquetas que más reducen el número de componentes conexas, y le asigna una probabilidad de selección proporcional a sus frecuencias. En [8] proponemos una versión del MVCA en el que en cada paso se selecciona al azar una de las etiquetas que más reduce el número de componentes conexas. Esta versión del MVCA utilizada como fase constructiva de un GRASP (*Greedy Randomize Adaptive Search Procedure*) se combina con una postoptimización para dar mejores resultados que los aparecidos hasta entonces. El proceso de post-optimización consiste en probar si se puede eliminar alguna etiqueta manteniendo la conexión del grafo. En [10] se propone una versión probabilística del MVCA en el que la selección de las etiquetas se realiza con un modelo probabilístico exponencial derivado de la regla de selección de movimientos de un *Simulated Annealing*.

Como métodos exactos, además del proceso enumerativo mencionado en los primeros trabajos aparecidos en la literatura, se han realizado dos propuestas. En [8] propusimos un método exacto útil para obtener soluciones óptimas de instancias de pequeño tamaño y así obtener bancos de prueba importantes para estudios computacionales con los que analizar las metaheurísticas. Nuestro procedimiento exacto se basa en un algoritmo de tipo A^* o *backtracking* para probar subconjuntos de etiquetas. Se realiza un proceso de ramificación y poda en el espacio de soluciones parciales implementado con un procedimiento recursivo que trata de completar de forma óptima la solución parcial actual. En [5] se presentan varios métodos de programación entera mixta exactos, incluyendo *branch-and-cut* y *branch-and-cut-and-price*, que son capaces de encontrar soluciones exactas a las pequeñas instancias del MLSTP.

3. Soluciones metaheurísticas

Se ha propuesto un buen número de metaheurísticas para el MLSTP. Esta aplicación de las metaheurísticas se facilita por el hecho de que las soluciones son conjuntos de colores o etiquetas representadas por vectores booleanos.

En primer lugar, en [19] se presentó un Algoritmo Genético que mejoró los resultados del MVCA en la mayoría de las instancias. Posteriormente, en [2] se aplica una búsqueda Tabú, un Simulated Annealing y una VNS, y se propone un algoritmo basado en el Método Piloto ideado por [17] que da mejores resultados que las metaheurísticas anteriores en casi todos los casos estudiados pero con tiempos de ejecución bastante altos. En [21] se presenta un Algoritmo Genético Modificado que mostró tener un buen rendimiento en términos de calidad de la solución y el tiempo de ejecución.

En [8] propusimos una heurística GRASP que utiliza como proceso constructivo una versión del MVCA en la que, en cada paso, se selecciona al azar una de las etiquetas que más reduce el número de componentes conexas y como fase de postoptimización el proceso consistente en eliminar las etiquetas que permitan mantener la conexión del grafo. Una segunda versión del GRASP propuesta en [8] incluye un criterio de creación por nivel de la lista restringida de candidatos estableciendo un valor relativo α de diferencia con respecto a la selección óptima.

En [9] propusimos un método de arranque múltiple y un PSO. En [4] también se propone un ACO. Estos trabajos de aplicación de las estrategias bio-inspiradas tienen la relevancia de mostrar como las estrategias cooperativas centralizada y descentralizada pueden aportar soluciones con rendimiento razonable aunque no llegan a ser competitivos con los procedimientos más especializados que tratan de explotar las características del problema como la VNS. En [10] se propone un procedimiento híbrido introduciendo en la búsqueda local de la VNS una idea similar a la aquí propuesta en combinación con una versión probabilística del MVCA inspirado en Simulated Anealig.

4. VNS Complementaria

El algoritmo de tipo VNS propuesto en [8] es una implementación de la versión básica de la VNS, la BVNS (ver [11]). Se combina la anterior búsqueda local con la agitación de amplitud variable k. La agitación se realiza en el entorno k-ésimo de la solución actual C constituido por los conjuntos de etiquetas que están a una distancia k de C aplicando la distancia de Hammimg; es decir, $N_k(C) = \{C' \subseteq L / |C' \Delta C \ = k\}$. La solución agitada se obtiene repitiendo k veces el movimiento aleatorio siguiente. Sea ξ un número aleatorio uniforme en [0,1]; si $\xi < 1/2$ se elimina una etiqueta de C elegida al azar y an caso contrario se añade una etiqueta de $L - C$ también elegida al azar.

Esta heurística VNS desarrollada para el MLSTP en [8] se ha mostrado como la metaheurística con el mejor rendimiento de las aparecidas hasta el momento, en cuanto a la calidad de las soluciones aportadas. Sin embargo, el procedimiento muestra una falta de capacidad de diversificación del proceso de búsqueda ya

que suelen encontrar soluciones de buena calidad para instancias medianas y pequeñas del problema en un relativo corto tiempo de cálculo, pero para las grandes instancias del problema tiende a ser generalmente lenta y menos robusta. Además, muestra un rendimiento muy dependiente de la adecuada elección del valor del parámetro k_{max} que resulta diferente según las características de la instancia.

Por lo tanto, en esta sección consideramos un mecanismo de autoregulación del parámetro k_{max} y la integración de un mecanismo para dotar a la VNS de capacidad extra de exploración. Se trata de una componente que se introduce sobre la VNS para que, en caso de tener que volver a arrancar la VNS, lo haga desde una zona distante del espacio de búsqueda. Además, el límite del tamaño de los entornos en la agitación se hace oscilar entre la mitad y el doble del tamaño de la mejor solución actual. El método resultante se conoce como VNS Complementaria (Co-VNS).

La Co-VNS aplica reiteradamente una VNS hasta que se cumpla un criterio de parada. Antes de aplicar cada VNS, Co-VNS extrae una solución del *espacio complementario* de la solución actual C y sustituye a la solución actual con la solución extraída. El espacio complementario de una solución de C se define como el conjunto de todas las etiquetas que no estén incluidas en C, es decir $(L \Delta C)$. A dicha solución, Co-VNS le aplica una heurística constructiva, como el MVCA en el subgrafo de G con etiquetas en el espacio complementario de la solución actual. Téngase en cuenta que Co-VNS se detiene si se obtiene ya sea una solución factible directamente (es decir, con una sola componente conexa), o el conjunto de colores no utilizados en el espacio complementario está vacío (es decir, $(L \Delta C) = \emptyset$), produciendo una solución final no factible. A continuación, se aplica la VNS básica con el fin de mejorar la solución resultante.

El proceso iterativo de selección de una nueva solución del espacio complementario de la solución actual, si no se produce mejora, está destinado a aumentar la capacidad de diversificación de la VNS básica para el MLSTP. Cuando la búsqueda local está atrapada en el entorno de un mínimo local, Co-VNS extrae una solución complementaria factible que se encuentra en una zona muy diferente del espacio de búsqueda, y se fija como nueva solución actual para la búsqueda local. Esta nueva solución de partida permite que el algoritmo pueda escapar de la región donde está atascado, produciendo un aumento inmediato de la diversificación.

Con el fin de ilustrar el funcionamiento de la Co-VNS, considere el ejemplo mostrado en la figura 2. Dada una solución aleatoria inicial de X_0, la VNS busca de nuevas soluciones en entornos cada vez más alejados de X_0. En este ejemplo, no se detectan las mejores soluciones y la solución actual es todavía X_0. Ahora, COMPL extrae una solución factible del espacio complementario de X_0, que se define como $(L \Delta X_0)$. Supongamos que la nueva solución es X_0^{compl}. Entonces, el algoritmo busca nuevas soluciones en los entornos de X_0^{compl}. En este ejemplo, se encuentra una solución factible mejor X_1. El algoritmo continúa con este procedimiento hasta que el se cumplen las condiciones de parada. En el ejemplo, la última solución es la denotada por X_2.

Algoritmo 1: VNS complementaria

Input: Un grafo no dirigido conexo y etiquetado $G = (V, A, L)$, con n vertices, m aristas y ℓ etiquetas;
Output: Un árbol generador T;
Initialización:
begin
  $Best_C \leftarrow$ *Generar-Solución-Inicial-Aleatoria()*;
  Hacer: $k_{max} \leftarrow |Best_C|$;
  repeat
    Extraer una solución del espacio complementario de $Best_C$:
    $C \leftarrow$ *Complementary*$(Best_C)$;
    Hacer $k \leftarrow 1$;
    while $k < k_{max}$ **do**
      $C' \leftarrow$ *Agitación*$(N_k(C))$;
      Local search(C');
      if $|C'| < |C|$ **then**
        Mover $C \leftarrow C'$;
        Volver a empezar con el primer entorno: $k \leftarrow 1$;
      else
        Aumentar el tamaño de la estructura de entornos: $k \leftarrow k + 1$;
      end
    end
    if $|C| < |Best_C|$ **then**
      Mover $Best_C \leftarrow C$;
      Disminuir el mayor tamaño de la agitación:
      $k_{max} \leftarrow max(k_{max} - 1; |Best_C|/2)$;
    else
      Aumentar el mayor tamaño de la agitación:
      $k_{max} \leftarrow min(k_{max} + 1; 2 \cdot |Best_C|)$;
    end
  until *Condición de Parada*;
  Tomar un árbol generador arbitrario T del subgrafo $H = (V, A(Best_C))$.
end

5. Resultados Computacionales

Para probar el rendimiento del algoritmo propuesto en este trabajo, se realizaron experimentos con los mismos conjuntos de instancias utilizados en [8] que están accesibles en (ver [7]). Se trata de instancias generadas al azar a partir del número de nodos (n), la densidad del grafo (d), y el número de etiquetas (ℓ). La batería de instancias consta de 48 conjuntos de 10 instancias de características determinadas (lo que supone un total de 480 instancias), donde el número de vértices n y de etiquetas ℓ van desde 20 hasta 500. El número de aristas, m, se obtiene indirectamente de la densidad d, cuyos valores son 0.8, 0.5, y 0.2. La complejidad de las instancias aumenta con la dimensión del grafo (es decir, con el aumento de n y/o ℓ), y la reducción en la densidad del grafo.

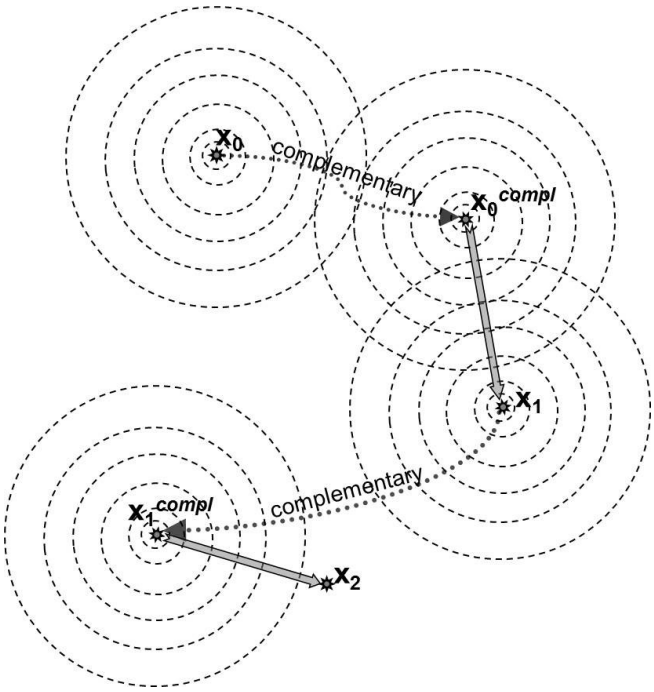


Figura 2. Ejemplo de ejecución del Co-VNS.

Para cada conjunto de datos, se evalúa el promedio de la calidad de la solución dada por el valor de la función objetivo (es decir, el número de colores de la solución) entre las 10 instancias del problema. Se eligió un tiempo de CPU permitido máximo, como condición de parada para las metaheurísticas, determinado en función de la dimensión de las instancias del problema. Por cuestiones de espacio mostramos sólo los resultados para las instancias de mayor tamaño; $n = 500$ (la condición de parada para estas instancias fue un tiempo máximo 5 minutos de CPU). Todos los experimentos se realizaron en un ordenador Intel Xeon(R) con microprocesador de 4.0 GHz y 6 GB de memoria RAM.

En los resultados mostrados en la tabla 1 se observa que cuando la densidad es baja ($d = 0.2$) para instancias grandes y el problema se hace más difícil de resolver es cuando el aumento de la capacidad de exploración de la VNS da lugar a mejores soluciones en promedio.

6. Conclusiones

En este trabajo se muestra un mecanismo para mejorar la capacidad de exploración de la VNS en la resolución del MLSTP. Además se introduce un mecanismo de auto regulación del parámetro más relevante de la VNS cuya

Tabla 1. Resultados Computacionalespara $n = 500$, $\ell = 0{,}25n$, $0{,}5n$, n, $1{,}25n$

Parámetros			*Objetivo promedio*	
n	ℓ	d	VNS	Co-VNS
500	125	0.8	2	2
		0.5	2.6	2.6
		0.2	6.2	6.2
	250	0.8	3	3
		0.5	4.1	4.2
		0.2	9.9	9.9
	500	0.8	4.7	4.9
		0.5	6.5	6.5
		0.2	15.8	15.8
	625	0.8	5.1	5.1
		0.5	7.9	7.9
		0.2	18.3	18
Parámetros			*tiempo (seg.)*	
n	ℓ	d	VNS	Co-VNS
500	125	0.8	0.012	0.425
		0.5	1.1	0.993
		0.2	3.8	3.8
	250	0.8	0.042	0.551
		0.5	83.7	19.2
		0.2	4.9	4.7
	500	0.8	22.5	21.2
		0.5	31.4	44.6
		0.2	138.1	121.6
	625	0.8	10.1	20.3
		0.5	32.2	35.2
		0.2	156.5	137.4

adecuada elección resultaba muy dependiente de las caraterísticas de la instancia. Los experimentos realizados muestran la efectividad de estas componentes.

Referencias

1. T. Brüggemann, J. Monnot, and G. J. Woeginger. Local search for the minimum label spanning tree problem with bounded colour classes. *Operations Research Letters*, 31:195–201, 2003.
2. R. Cerulli, A. Fink, M. Gentili, and S. Voß. Metaheuristics comparison for the minimum labelling spanning tree problem. In B. L. Golden, S. Raghavan, and E. A. Wasil, editors, *The Next Wave on Computing, Optimization, and Decision Technologies*, pages 93–106. Springer-Verlag, New York, 2005.
3. R. S. Chang and S. J. Leu. The minimum labelling spanning trees. *Information Processing Letters*, 63(5):277–282, 1997.
4. A. M. Chwatal and G. R. Raidl. Solving the minimum label spanning tree problem by ant colony optimization. *Proceedings of the 7th International Conference on*

Genetic and Evolutionary Methods (GEM 2010), pages 91–97, 2010. Las Vegas, Nevada.
5. A. M. Chwatal and G. R. Raidl. Solving the minimum label spanning tree problem by mathematical programming techniques. *Advances in Operations Research*, 2011:38, 2011. Article ID: 143732.
6. A. M. Chwatal, G. R. Raidl, and O. Dietzel. Compressing fingerprint templates by solving an extended minimum label spanning tree problem. *Proceedings of the 7th Metaheuristics International Conference (MIC 2007)*, 2007. Montreal, Canada.
7. S. Consoli. Test datasets for the minimum labelling spanning tree problem, [online]. 2008. Available at http://www.sergioconsoli.com/MLSTP.htm, last accessed: October 2012.
8. S. Consoli, K. Darby-Dowman, N. Mladenović, and J. A. Moreno-Pérez. Greedy randomized adaptive search and variable neighbourhood search for the minimum labelling spanning tree problem. *European Journal of Operational Research*, 196(2):440–449, 2008.
9. S. Consoli, J. A. Moreno-Pérez, K. Darby-Dowman, and N. Mladenović. Discrete Particle Swarm Optimization for the minimum labelling Steiner tree problem. *Natural Computing*, 9(1):29–46, 2010.
10. S. Consoli, and J. A. Moreno-Pérez Solving the minimum labelling spanning tree problem using hybrid local search. *Electronic Notes in Discrete Mathematics*, 39:75–82, 2012.
11. P. Hansen, N. Mladenović, and J. A. Moreno Pérez. Variable Neighbourhood Search: methods and applications. *Annals of Operations Research*, 175(1):367–407, 2010.
12. J. Koźlak. Learning in cooperating agents environment as a method of solving transport problems and limiting the effects of crisis situations. In Y. Shi, G. D. Van Albada, J. Dongarra, and P. M. A. Sloot, editors, *Proceedings of the 7th International Conference on Computational Science (ICCS 2007), LNCS*, volume 4488, pages 872–879. Springer-Verlag, Heidelberg, Germany, 2007.
13. S. O. Krumke and H. C. Wirth. On the minimum label spanning tree problem. *Information Processing Letters*, 66(2):81–85, 1998.
14. J. S. Miller. *Multimodal statewide transportation planning: A survey of state practices.* Transportation Research Board, National Research Council, Virginia, US, 2006.
15. A. S. Tanenbaum. *Computer networks.* Englewood Cliffs, New Jersey: Prentice-Hall. 1989.
16. R. Van-Nes. *Design of multimodal transport networks: A hierarchical approach.* Delft University Press, The Netherlands, 2002.
17. S. Voß, A. Fink, and C. Duin. Looking ahead with the pilot method. *Annals of Operations Research*, 136:285–302, 2004.
18. Y. Wan, G. Chen, and Y. Xu. A note on the minimum label spanning tree. *Information Processing Letters*, 84:99–101, 2002.
19. Y. Xiong, B. Golden, and E. Wasil. A one-parameter genetic algorithm for the minimum labelling spanning tree problem. *IEEE Transactions on Evolutionary Computation*, 9(1):55–60, 2005.
20. Y. Xiong, B. Golden, and E. Wasil. Worst case behavior of the mvca heuristic for the minimum labelling spanning tree problem. *Operations Research Letters*, 33(1):77–80, 2005.
21. Y. Xiong, B. Golden, and E. Wasil. Improved heuristics for the minimum labelling spanning tree problem. *IEEE Transactions on Evolutionary Computation*, 10(6):700–703, 2006.

Cálculo de separadores de grafos utilizando búsqueda de vecindad variable reducida

D. Fernández-López, J. Sánchez-Oro, y A. Duarte[1]

Dept. Ciencias de la Computación, Universidad Rey Juan Carlos (Spain)
{ david.fernandez, jesus.sanchezoro, abraham.duarte }@urjc.es

Resumen Dado un grafo conexo y no dirigido $G = (V, E)$ y un límite entero positivo b, el problema del *Vertex Separator* (VS) consiste en encontrar el conjunto de vértices C, que eliminado divide G en componentes aisladas A, B, donde $\max(|A|, |B|) \leq b$. Este es un problema de optimización NP-completo y puede ser utilizado en un amplio rango de aplicaciones. Por ejemplo, en redes de telecomunicaciones un separador determina la capacidad y fragilidad de la red. En el campo de los algoritmos para grafos, el cálculo de pequeños separadores balanceados es muy útil, especialmente en los algoritmos del tipo divide y vencerás. Este trabajo presenta un nuevo algoritmo heurístico basado en el método *Variable Neighborhood Search* (VNS) para el cálculo de separadores. El método es comparado con el estado del arte (dos procedimientos de ramificación y poda recientes). Los resultados muestran que el método obtiene la solución óptima en todas las instancias pequeñas y medianas, y obtiene buenos resultados en instancias grandes. Aunque los métodos anteriores aseguran encontrar la solución óptima, tienen un tiempo de ejecución mucho mayor que el método presentado (aunque éste no asegura que la solución obtenida sea la óptima).

Keywords: Optimización combinatoria, Metaheurísticas, VNS, Separadores de grafos

1. Introducción

Se define un grafo $G = (V, E)$ mediante su conjunto de vértices V y de aristas E. Sea c_j el coste asociado con cada vértice $j \in V$. Un separador en G se define como un subconjunto de vértices o aristas que al ser eliminados convierten el grafo en inconexo. Los separadores también son conocidos por otros nombres, incluyendo bisectores, bifurcadores, cortes balanceados y particiones.

Este trabajo se centra en el problema del *Vertex Separator* (VS), que consiste en encontrar una partición de V en tres conjuntos no vacíos A, B y C, tal que no existe ninguna arista entre A y B, el tamaño de ambos conjuntos esta limitado por un entero positivo b (normalmente representado como una función de $|V|$), y que minimiza la suma de los pesos de los vértices en C.

Dado un grafo $G = (V, E)$, una solución x para el problema del VS puede representarse como tres conjuntos A, B, y C tales que $A \cup B \cup C = V$ y $A \cap B =$

$A \cap C = B \cap C = \emptyset$. Por lo tanto, el valor de la función objetivo, Sep, de una solución, $x = \{A, B, C\}$, se define como $Sep(x, G) = \sum_{j \in C} c_j$. Así, el problema de optimización puede ser formulado del siguiente modo:

$$\begin{aligned} &\min \sum_{j \in C} c_j \\ &s.t. \\ &\qquad \max\{|A|, |B|\} \leq b \\ &\qquad \forall i \in A \wedge \forall j \in B, (i, j) \notin E \end{aligned}$$

Nótese que el problema del VS es completamente equivalente a maximizar la suma de los pesos de los vértices en A y B (ver por ejemplo [2]).

La Figura 1.(a) muestra un ejemplo de un grafo G con cinco vértices y seis aristas. La Figura 1.(b) representa una posible solución, donde los conjuntos A, B, y C están delimitados mediante una línea discontinua. Si se asume, por simplicidad, que cada vértice tiene coste 1, el valor de esta solución es 1 ya que sólo hay un vértice en el conjunto C.

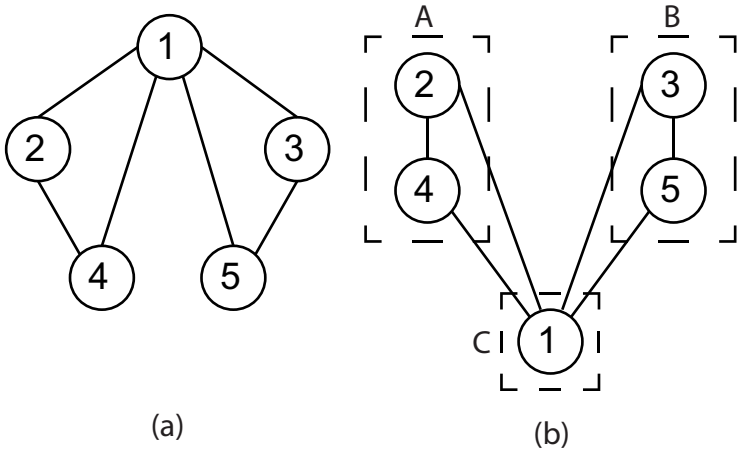


Figura 1. Ejemplo de un grafo (a) y una posible solución para el *Vertex Separator* (b)

Este problema de optimización fue originalmente presentado en [4] para el diseño de VLSI. Sin embargo, encontrar separadores balanceados y de pequeño tamaño es importante en muchos contextos. Por ejemplo, en redes de telecomunicaciones ([15], [13]). En el campo de la algoritmia para grafos, el calculo de separadores balanceados de pequeño tamaño es muy útil, especialmente en los algoritmos del tipo divide y vencerás ([17]). En bioinformática y biología computacional, se utilizan para simplificar la representación de las proteínas.

Encontrar el separador mínimo de un grafo general es un problema NP-completo [5]. Por lo tanto, los métodos exactos solo resuelven instancias de tamaño moderado. En concreto, Souza and Balas [2] propusieron una formulación de programación entera mixta, investigaron el politopo del VS y la envolvente

convexa de los vectores incidentes de los separadores. Este estudio teórico fue posteriormente incluido en un procedimiento de ramificación y poda. Los resultados computacionales mostraron que el método exacto encontraba el óptimo para instancias con tamaños comprendidos entre 50 y 150 en un tiempo de ejecución moderado. Biha and Meurs [7] introdujeron nuevas restricciones válidas y un límite inferior simple. Los experimentos mostraron que el nuevo método exacto resolvía de manera óptima todas las instancias propuestas por Balas and Souza [23] en menos tiempo. Por lo tanto, este método aparece como el estado del arte con respecto a los métodos exactos.

Muchos de los trabajos previos en el problema del VS están basados en el diseño de algoritmos de aproximación [16], mientras que otros aproximan la solución al VS mediante la solución del problema del *Edge Separator* [14,1,9,10]. En la literatura, hasta donde sabemos, solo existe un método heurístico [23], con el objetivo de encontrar soluciones factibles para el procedimiento de ramificación y poda, utilizando una relajación lineal del problema.

En este trabajo se propone un método basado en *Variable Neighborhood Search* (VNS) [11] para el problema del *Vertex Separator*. Esta metaheurística ha sido aplicada satisfactoriamente en un gran número de problemas de optimización. Por ejemplo en [21,8,18,22,20]. Concretamente se utiliza la variante *Reduced Variable Neighborhood Search* (RVNS) para diseñar un procedimiento de resolución eficiente y efectivo para el problema del VS. La Sección 2 presenta el procedimiento constructivo diseñado. La Sección 3 describe un nuevo procedimiento *shake*, diseñado para balancear la intensificación y la diversificación. La Sección 4 describe el procedimiento VNS en sí mismo, y cómo utiliza el constructivo y el método de *shake*. La Sección 5 muestra una extensa experimentación para validar el algoritmo propuesto comparando su rendimiento y tiempo de ejecución con los métodos del estado del arte. Por último, la Sección 6 resume las principales conclusiones de la investigación.

2. Procedimiento constructivo

Los árboles de nivel (ver [6]) son procedimientos constructivos basados en la partición de los vértices de un grafo en diferentes niveles, $L_1, \ldots, L_s$, de tal manera que los extremos de cada arista en el grafo estén en el mismo nivel L_i o en dos niveles consecutivos, L_i y L_{i+1}. Esta estructura de nivel garantiza que los vértices en niveles no consecutivos no son adyacentes. El procedimiento constructivo genera la estructura de nivel usando un método de búsqueda en anchura. Este algoritmo recibe como parámetro de entrada un vértice v y se construye un árbol de expansión con raíz en v. Por lo tanto, si un grafo tiene $|V|$ vértices, se construirán $|V|$ árboles de expansión diferentes.

El Algoritmo 1 muestra el pseudocódigo del procedimiento constructivo. En concreto, recibe como parámetro de entrada el árbol de expansión T y devuelve una solución factible para el problema del VS. El algoritmo comienza identificando el conjunto de niveles del árbol (paso 2). Después se inicializa un conjunto de variables auxiliares (pasos 3 al 7). El algoritmo intenta alternativamente asignar

un nivel del árbol a cada conjunto. Concretamente, el nivel L_1 es asignado a A y el nivel L_s es asignado a B. Si ningún conjunto está lleno, se realiza otra iteración asignando L_2 y L_{s-1} a A y B, respectivamente (ver pasos 9 al 24). Está lógica se mantiene hasta que se alcanza el nivel medio del árbol ($\lfloor s/2 \rfloor$) o uno de los subconjuntos está lleno (comprobar pasos 9 y 16). Si ni A ni B alcanza el máximo de su capacidad, el nivel crítico será $\lfloor s/2 \rfloor$. En otro caso, el nivel crítico será aquel en el que se llenó uno de los niveles (ver pasos 13 y 20). En el momento que un conjunto está lleno, los niveles restantes, menos el crítico, se introducen en el otro conjunto.

Algoritmo 1 Pseudocódigo del procedimiento constructivo

```
 1: procedure Constructivo(T)
 2: T = {L_1, L_2, ..., L_s}
 3: limite ← ⌊s/2⌋
 4: i ← 1
 5: j ← s
 6: llenoA ← false
 7: llenoB ← false
 8: while (i < limite) or (j > limite) do
 9:    if (|A| + |L_i| < b) and not llenoA then
10:       A ← A ∪ L_i
11:    else
12:       llenoA ← true
13:       critico ← i
14:       limite ← i + 1
15:    end if
16:    if (|B| + |L_j| < b) and not llenoB then
17:       B ← B ∪ L_j
18:    else
19:       llenoB ← true
20:       critico ← j
21:       limite ← j − 1
22:    end if
23:    i ← i + 1
24:    j ← j − 1
25: end while
26: for all v ∈ L_critico do
27:    N(v) → {u ∈ V : (u, v) ∈ E}
28:    N_A(v) → N(v) ∩ A
29:    N_B(v) → N(v) ∩ B
30:    if |N_A(v)| = ∅ then
31:       B → B ∪ {v}
32:    else if |N_B(v) = ∅ then
33:       A → A ∪ {v}
34:    else
35:       C → C ∪ {v}
36:    end if
37: end for
```

Los elementos pertenecientes al nivel crítico son tratados en los pasos del 26 al 37. En concreto, para cada vértice, se comprueba si tiene adyacentes en A o B (pasos 28 y 29). Si el vértice correspondiente no tiene adyacentes en A, será asignado a B o viceversa (pasos 30 al 33). Finalmente, si el vértice tiene adyacentes en A y B será asignado a C.

3. Método *Shake*

Hay varios problemas de optimización donde la estructura de la solución apenas permite el diseño de una búsqueda local eficiente. Este es el caso del problema del VS. En concreto, una solución x está definida por tres subconjuntos disjuntos A, B, y C. Los movimientos en esa solución pueden ser muy complejos. Por ejemplo, eliminando un vértice en A (simétricamente en B) e insertándolo en B (simétricamente en A) la función objetivo no varía. En la misma línea, eliminando vértices de A o B e insertándolos en C la función objetivo empeora. Finalmente, casi nunca es posible eliminar un vértice de C e insertarlo en A o B porque la solución obtenida es infactible. La mayoría de algoritmos VNS utilizan el método *shake* para salir de un óptimo local mediante perturbaciones aleatorias de la solución. Sin embargo, en este caso es necesario incluir una componente intensificadora para reducir la aleatoriedad del método.

La función *shake* propuesta en este trabajo, $Shake(x, k)$ se muestra en el Algoritmo 2. En primer lugar identifica los tres conjuntos disjuntos A, B, y C (paso 2). Después, se identifica el conjunto de k vértices (llamado $Cand$) el cual será perturbado. Para diversificar la búsqueda, estos vértices son seleccionados de $A \cup B$ de manera aleatoria (paso 3). A continuación, el método genera una solución parcial eliminando los vértices k seleccionados (pasos 4 y 5). En esa solución parcial, se intenta reducir la función objetivo eliminando elementos de C e insertándolos en A o B (fase de intensificación). Primero se prueba si eliminando un elemento de C e insertándolo en A (paso 7) o B (paso 8) la solución es factible. Se utiliza la función `Factible` $(A \cup \{c\})$,la cual devuelve `true` si al insertar el elemento c no se sobrepasa el límite b (es decir, $A \cup \{c\} \leq b$) y $N_B(c) = \emptyset$. La función `Factible` $(B \cup \{c\})$ es completamente simétrica.

La inserción de un elemento en A o B se puede separar en tres casos diferentes (pasos 9 al 19). Si es posible insertar el elemento en ambos conjuntos, se insertará en el conjunto con menor cardinalidad para asegurar la similitud de tamaños entre conjuntos (pasos 10 al 14). En otro caso, el elemento se insertará en el único conjunto que genera una solución factible (pasos 15 al 19).

Por último, el procedimiento *shake* termina insertando los k elementos eliminados, primero, en A o B (de nuevo, manteniendo el tamaño de los conjuntos lo más similar posible). Cuando esto no sea posible, el elemento finalmente será insertado en C (ver pasos 21 al 37).

Algoritmo 2 Pseudocodigo del método *shake*

```
1: procedure Shake(x, k)
2: x = {A, B, C}
3: Cand ← SeleccionAleatoria(A ∪ B, k)
4: A ← A \ Cand
5: B ← B \ Cand
6: for all c ∈ C do
7:    feasA ← Factible(A ∪ {c})
8:    feasB ← Factible(B ∪ {c})
9:    if (feasA and feasB) then
10:      if ((|A| < |B|)) then
11:         A ← A ∪ {c}
12:      else
13:         B ← B ∪ {c}
14:      end if
15:   else if feasA then
16:      A ← A ∪ {c}
17:   else if feasB then
18:      B ← B ∪ {c}
19:   end if
20: end for
21: for all c ∈ Cand do
22:   feasA ← Factible(A ∪ {c})
23:   feasB ← Factible(B ∪ {c})
24:   if (feasA and feasB) then
25:      if ((|A| < |B|)) then
26:         A ← A ∪ {c}
27:      else
28:         B ← B ∪ {c}
29:      end if
30:   else if feasA then
31:      A ← A ∪ {c}
32:   else if feasB then
33:      B ← B ∪ {c}
34:   else
35:      C ← C ∪ {c}
36:   end if
37: end for
```

4. Búsqueda de vecindad variable reducida

La búsqueda de vecindad variable reducida (*Variable Neighborhood Search*, (VNS)) es una metodología para resolver problemas de optimización basada en el cambio de estructuras de vecindad. En los últimos años, se han propuesto una gran cantidad de variantes para el método del VNS. En [11,12] se presentan dos completas revisiones de la metodología.

Sea N_k con $1 \leq k \leq k_{max}$ un conjunto finito de estructuras de vecindades preseleccionadas, donde $N_k(x)$ es el conjunto de soluciones vecinas de x en la

k-esima vecindad. Cuando se soluciona un problema de optimización mediante el uso de estructuras de vecindad, el método VNS propone tres métodos diferentes para su exploración: (i) aleatorio, (ii) determinista, o (iii) mixto, el cual mezcla (i) y (ii). Este trabajo se centra en la variante RVNS, que consiste en la exploración (generación) de soluciones aleatorias en cada vecindad N_k. Esta variante no considera la aplicación de un procedimiento de búsqueda local. De hecho, los valores de las soluciones generadas son directamente comparados con el valor de la solución correspondiente, actualizando la mejor solución en caso de mejora.

Algoritmo 3 Pseudocódigo del RVNS

```
1: procedure RVNS(k_max, t_max)
2: repeat
3:    x ← Constructivo()
4:    k ← 1
5:    repeat
6:       x' ← Shake(x, k)
7:       if Sep(x', G) < Sep(x, G) then
8:          x ← x'
9:          k ← 1
10:      else
11:         k ← k + 1
12:      end if
13:   until k = k_max
14:   t ← TiempoCPU()
15: until t > t_max
16: end RVNS
```

El pseudocódigo del RVNS se muestra en el Algoritmo 3. Tiene dos argumentos de entrada: la vecindad predefinida más grande (k_{max}) y el máximo tiempo de ejecución (t_{max}). El procedimiento comienza construyendo una solución factible (paso 3), utilizando el método descrito en la Sección 2. El RVNS comienza la búsqueda en la primera estructura de vecindad (paso 4). Después, la solución es perturbada utilizando la función `Shake` en el paso 6, obteniendo una nueva solución x' (ver Sección 3). En el paso 7, se decide si el RVNS realiza un movimiento (lo que implica que x' es mejor que x) o no. Si es así, la solución es actualizada (paso 8) y el método vuelve a la primera vecindad (paso 9). En otro caso, (es decir, x' es peor que x) el RVNS explora una vecindad mayor incrementando k (paso 11). Los pasos 6 y 12 son repetidos hasta que se alcanza k_{max}. Este parámetro determina el número máximo de vecindades diferentes a explorar en la iteración actual cuando no hay mejora en la solución. Por último, se repiten los pasos del 3 al 14 hasta que se alcanza t_{max}, comenzando cada iteración desde la construcción de una nueva solución.

Para diversificar la búsqueda el RVNS se ejecuta $|V|$ iteraciones independientes. Específicamente, se considera en cada iteración un árbol de expansión construido con un algoritmo de búsqueda en profundidad. El método completo devuelve la mejor solución encontrada.

5. Experimentos computacionales

Está sección muestra los experimentos realizados para evaluar la eficiencia del RVNS propuesto para solucionar el problema del VS. El algoritmo ha sido implementado en Java SE 6. Se han considerado cuatro conjuntos de instancias utilizadas previamente en este problema.

Las instancias DIMACS provienen del reto DIMACS en coloreado de grafos. Los grafos incluidos en este conjunto no tienen más de 150 vértices, con excepción de la instancia myciel7 que tienen 191 vértices. La segunda clase de instancias deriva de los grafos del *Matrix Market* (grafos MM para abreviar). Esta colección está compuesta por un conjunto de matrices de prueba estándar $M = M_{uv}$ derivadas de problemas en sistemas lineales, mínimos cuadrados y calculo de autovalores de una amplia variedad de disciplinas científicas e ingenieriles. Los grafos derivan de estas matrices considerando una arista (u, v) por cada elemento $M_{uv} = 0$. Se utilizan tres categorías de grafos MM. La primera, llamada MM-I, corresponde a todas las matrices con entre 20 y 100 columnas. La segunda, llamada MM-II, fue obtenida de las matrices cuyo número de columnas está entre 100 y 200. La tercera categoría de instancias, denotada MM-HD, solo contiene grafos con al menos una densidad del 35 %.

Los resultados del RVNS son comparados con los obtenidos por De Souza y Balas [23] (Pentium 4 a 2.5 GHz y 2 GB de RAM) y Biha y Meurs [7] (Pentium M740 a 1.73 Ghz y 1 GB de RAM). En nuestra opinión, el rendimiento de los ordenadores donde se ejecutaron los algoritmos no difiere demasiado. El algoritmo propuesto se ha ejecutado en un Intel Core 2 Duo T6400 a 2GHz y 3 GB de memoria RAM.

La Tabla 1 muestra los resultados experimentales. Cada línea se corresponde a uno de los conjuntos de instancias. La primera columna contiene el nombre del conjunto de instancias (Instancias) seguido, entre paréntesis, por el número de instancias que lo forman. La segunda columna contiene el valor promedio de la solución óptima (Opt.). La tercera y cuarta columna muestran, respectivamente, el promedio de tiempo de ejecución de los dos métodos exactos (Tiempo SB y Tiempo BM). La quinta y sexta columna contienen el promedio del valor de la función objetivo (RVNS) y su tiempo de ejecución en segundos (Tiempo). Finalmente, la séptima columna reporta el número de óptimos encontrados en cada conjunto de instancias.

Instancias	**Óptimo**	**Tiempo SB**	**Tiempo BM**	**RVNS**	**Tiempo**	**#Opt**
MM-I (24)	49.63	48.46	7.87	49.63	0.13	24
MM-II (20)	107.8	147.92	53.98	107.75	1.43	19
MM-HD (39)	72.49	13.74	13.67	72.21	0.46	33
DIMACS (21)	69.80	634.97	609.37	69.76	1.07	20

Tabla 1. Resultados obtenidos en los diferentes conjuntos de instancias.

En el conjunto más sencillo, MM-I, nuestro procedimiento obtiene todos los óptimos (24) en 0.13 segundos, frente a los 7.87 y 48.46 segundos que necesi-

tan BM y SB, respectivamente. En el segundo conjunto, MM-II, el algoritmo propuesto obtiene todos los óptimos menos uno (19 de 20) en 1.43 segundos, mientras que BM y SB necesitan 147.92 y 53.98 segundos. Además, en la instancia donde RVNS no encuentra el óptimo, se queda a una sola unidad. En el conjunto MM-HD el método RVNS no encuentra el óptimo en tan solo 6 instancias, quedándose como mucho a 2 unidades, y utilizando 0.46 segundos en promedio. En este caso, tanto SB como BM necesitan aproximadamente 13 segundos para finalizar. Finalmente, en el conjunto DIMACS se obtienen todos los óptimos salvo uno, en los que RVNS se queda a una unidad del óptimo. El tiempo promedio es de 1.07 segundos, frente a los 600 segundos que necesitan tanto SB como BM para encontrar el valor óptimo.

6. Conclusiones

En este trabajo se propone el uso del método *Reduced Variable Neighborhood Search* (RVNS) para la solución del problema del *Vertex Separator* (VS). Se presenta un procedimiento constructivo muy efectivo que permite encontrar regiones muy prometedoras en el espacio de búsqueda. Con el objetivo de diversificar la búsqueda, el RVNS comienza con diferentes soluciones iniciales. Además, también se propone un procedimiento *shake* que consigue un compromiso entre intensificación y diversificación. Se proporciona una extensiva comparación experimental con los mejores métodos anteriores en el estado del arte sobre un conjunto de 104 instancias. Los resultados experimentales muestran que el algoritmo propuesto encuentra la solución óptima en la mayoría de los casos con tiempos de ejecución realmente cortos.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el Ministerio de Educación y Ciencia a través de los proyectos con referencias TIN2009-07516, TIN2011-28151, y TIN2012-35632, y por la Comunidad de Madrid a través del proyecto con referencia S2009/TIC-1542.

Referencias

1. Arora, S., Rao, S. & Vazirani,U. (2004).Expander flows, geometric embeddings, and graph partitionings. 36th Annual Symposium on the Theory of Computing, 222—231.
2. Balas, E. & de Souza, C. (2005). The vertex separator problem: a polyhedral investigation. Mathematical Programming, 103:583–608.
3. Basu, A., Bonami, P., Cornuejols, G, & Margot, F. (2009). On the Relative Strength of Split, Triangle and Quadrilateral Cuts. Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2009), New York, USA, 1220–1229.
4. Bhatt,S.N. & Leighton,F.T.(1984). A framework for solving VLSI graph layout problems. J. Computer System Sci., 28:300—343.

5. Bui,T.N. & Jones,C.(1992). Finding good approximate vertex and edge partitions is NP-hard. Inf. Process. Lett., 42:153—159.
6. Díaz, J., Petit, J. & Serna,M. (2002) A survey of graph layout problems. Journal ACM Computing Surveys, 34(3):313–356.
7. Didi Biha, M. & Meurs, MJ. (2011). An exact algorithm for solving the vertex separator problem. Journal of Global Optimization, 49:425–434.
8. Duarte, A., Escudero L.F., Martí R., Mladenovic N., Pantrigo J.J., Sánchez-Oro, J. (2012). Variable Neighborhood Search for the Vertex Separation Problem. Computers and Operations Research, 39(12):3247—3255.
9. Feige, U. & Kogan,S. (2004) Hardness of approximation of the balanced complete bipartite subgraph problem. Technical report MCS04-04, Department of Computer Science and Applied Math., The Weizmann Institute of Science.
10. Feige, U., Hajiaghayi, M. & Lee, J.R. (2005). Improved approximation algorithms for minimum-weight vertex separators. Proceedings of the thirty-seventh annual ACM symposium on Theory of computing, 563–572.
11. Hansen, P. & Mladenović, N. (2003). Variable Neighborhood Search. International Series in Operations Research & Management Science, 57:145–184.
12. Hansen, P., Mladenovic, N., Perez, J.A.M. (2010). Variable neighbourhood search: methods and applications. Annals of Operations Research 175:367-–407.
13. Leighton F.T. (1983). Complexity Issues in VLSI: Optimal Layout for the Shuffle-Exchange Graph and Other Networks. MIT Press, Cambridge, MA.
14. Leighton, T. & Rao, S. (1999) Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. J. ACM. 46:787-–832.
15. Leiserson, C. (1980). Area-efficient graph layouts (for VLSI). 21th Annual Symposium on Foundations of Computer Science. IEEE Computer Soc., Los Alamitos, CA, 270—280.
16. Lipton, R.J. & Tarjan, R.J. (1979). A Separator Theorem for Planar Graphs. SIAM Journal on Applied Math, 36:177–189.
17. Lipton, R.J. & Tarjan,R.E. (1980). Applications of a planar separator theorem. SIAM J. Comput., 9:615-–627.
18. Lozano, M., Duarte, A., Gortázar, F. & Martí R. (2012). Variable Neighborhood Search with Ejection Chains for the Antibandwidth Problem. Journal of Heuristics, 18:919–938.
19. Mladenovic, N., Hansen P.,(1997). Variable neighborhood search. Computers and Operations Research 24 (11): 1097-–1100.
20. Montemayor. A.S., Duarte A., Pantrigo J.J., Cabido R. (2005). High-Performance VNS for the Max-Cut Problem Using Commodity Graphics Hardware. Mini-Euro Conference on VNS (MECVNS 05), Tenerife (Spain), 1–11.
21. Pardo, E.G., Mladenovic, N., Pantrigo, J.J. & Duarte, A. (2013). Variable Formulation Search for the Cutwidth Minimization Problem. Applied Soft Computing, 13:2242–2252.
22. Sánchez-Oro, J., Pantrigo J.J., Duarte A. (2013). Balancing intensification and diversification strategies in VNS. An application to the Vertex Separation Problem. Technical Report. Dept. Ciencias de la Computación. Universidad Rey Juan Carlos.
23. de Souza, C. & Balas, E. (2005) The vertex separator problem: algorithms and computations. Mathematical Programming, 103:609-–631.
24. Strunk, W. Jr., & White, E.B. (2000). The elements of style. (4th ed.). New York: Longman.
25. Van der Geer, J., Hanraads, J.A.J., & Lupton, R.A. (2010). The art of writing a scientific article. Journal of Scientific Communications, 163:51-–59.

Búsqueda de Formulación Variable aplicada al problema de la Minimización de la Anchura de Corte en ordenaciones lineales

Eduardo G. Pardo[1], Juan J. Pantrigo[1], Abraham Duarte[1], and Nenad Mladenovic[2]

[1] Dpto. Ciencias de la Computación. Universidad Rey Juan Carlos. 28933, Móstoles.
{eduardo.pardo, juanjose.pantrigo, abraham.duarte}@urjc.es
[2] School of Mathematics. Brunel University. UB8 3PH, Uxbridge, London.
nenad.mladenovic@brunel.ac.uk

Resumen Muchos problemas de optimización son formulados como problemas min-max donde la función objetivo consiste en minimizar un valor máximo. En este caso, es habitual que muchas soluciones al problema tengan asociadas el mismo valor de la función objetivo. Cuando esto ocurre es difícil determinar cuál de las soluciones comparadas es más prometedora para continuar la búsqueda. En este artículo se propone una nueva variante dentro de la metodología Búsqueda de Vecindad Variable para abordar este tipo de problemas. Esta nueva variante, denominada Búsqueda de Formulación Variable, está basada en la utilización de formulaciones alternativas al problema para determinar la solución más prometedora, dentro de las funciones de agitación, búsqueda local y cambio de vecindad, en el esquema Búsqueda de Vecindad Variable Básico. La metodología propuesta es aplicada al problema de la Minimización de la Anchura de Corte en ordenaciones lineales y los resultados obtenidos son comparados con algoritmos previos en el estado del arte.

1. Introducción

Entre los problemas de optimización, existen muchos que consisten en minimizar un valor máximo o, alternativamente, maximizar un valor mínimo. Este tipo de problemas presentan un espacio de búsqueda con un horizonte plano. Es decir, muchas soluciones al problema tienen asociado el mismo valor de la función objetivo, $\mathcal{F}(x)$, para la formulación original del problema [11,12,9]. Esto hace que este tipo de problemas sean aún más difíciles de lo habitual, dado que la manera típica de búsqueda, conducida por movimientos dentro del espacio de soluciones factibles, resulta inerte en la mayoría de los casos.

El cambio de vecindad es una estrategia utilizada por la metodología Búsqueda de Vecindad Variable (VNS, acrónimo de su nombre en inglés *Variable Neighbourhood Search* [8]) para evitar caer en óptimos locales. Esta estrategia se basa en la idea de que los mínimos locales con respecto a una estructura determinada no tienen porqué serlo con respecto a otra [8,5]. Pese a que el cambio de vecindad

puede ser de utilidad para superar, parcialmente, el problema relativo al horizonte plano en el espacio de soluciones, existen problemas donde incluso esto no es suficiente para producir mejoras en la función objetivo tras un movimiento.

En este trabajo se propone una nueva variante de la metodología VNS, denominada Búsqueda de Formulación Variable (VFS, acrónimo de su nombre en inglés *Variable Formulation Search*) la cual combina el cambio de vecindad del esquema VNS con la utilización de formulaciones alternativas al problema para tratar de superar la dificultad anteriormente mencionada. Esta variante toma en consideración diferentes formulaciones de un problema cuando para una vecindad determinada, múltiples soluciones toman el mismo valor de la función objetivo. El cambio de formulación se lleva a cabo durante las etapas de búsqueda local, agitación y cambio de vecindad, dentro del esquema VNS.

Para validar la eficacia de la propuesta se ha aplicado la nueva variante al problema de la Minimización de la Anchura de Corte en ordenaciones lineales y se han comparado los resultados obtenidos con los de los métodos presentes en el estado del arte. El resto del artículo está estructurado de la siguiente manera: en la Sección 2 se describe la metodología propuesta. En la Sección 3 se presenta el problema abordado y la adaptación de VFS. Los resultados de esta validación se presentan en la Sección 4, terminando con las conclusiones en la Sección 5.

2. Búsqueda de Formulación Variable

En esta sección se presenta la Búsqueda de Formulación Variable, una nueva variante de la metodología VNS para problemas que presentan un horizonte plano del espacio de búsqueda.

2.1. VNS

La Búsqueda de Vecindad Variable es una metaheurística propuesta en [8] como un marco general para la resolución de problemas difíciles de optimización. Está basada en una idea simple: cambios sistemáticos de estructuras de vecindad durante el proceso de búsqueda. La metaheurística original ha sido ampliamente evolucionada con muchas extensiones, destacando: *Variable Neighbourhood Descent* (VND), *Reduced VNS* (RVNS), *Basic VNS* (BVNS), *Skewed VNS* (SVNS), *General VNS* (GVNS), Variable *Neighbourhood Decomposition Search* (VNSD) y *Reactive VNS*. En [4] y [5] se pueden encontrar amplias revisiones sobre las principales variantes de la metodología. De entre las anteriormente mencionadas, este trabajo centra su atención en el esquema BVNS como punto de partida.

Un pseudocódigo de BVNS se presenta en Algoritmo 1. Este procedimiento recibe tres parámetros de entrada: una solucion inicial (denotada como x), la máxima vecindad predefinida (k_{max}) y el tiempo máximo de cómputo (t_{max}). El procedimiento comienza realizando una perturbación a la solución actual a través de la función `Shake`, en el paso 5, y obteniendo así una nueva solución x'. A continuación, en el paso 6, un óptimo local x'', es alcanzado a través de la utilización de una estrategia de mejora. En el paso 7, se decide si BVNS necesita

explorar una vecindad mayor incrementando k (x'' es una solución peor que x) o no, lo que implica restablecer el valor de $k = 1$ (x'' es mejor que x). Los pasos 5 a 7 son repetidos hasta que se alcanza k_{max}. Este parámetro determina la máxima diferencia de vecindades que se probarán en la actual iteración cuando no hay mejoras en la solución. Los pasos 3 a 9 son repetidos hasta que t_{max} es alcanzado, comenzando en cada iteración desde la mejor solución encontrada.

Algoritmo 1 Pseudocódigo de BVNS

```
1: Procedimiento BVNS (x, k_max, t_max)
2: repetir
3:    k ← 1
4:    repetir
5:       x' ← Shake(x, k)
6:       x'' ← LocalSearch(x')
7:       NeighbourhoodChange(x, x'', k)
8:    hasta que k = k_max
9:    t ← CPUTime()
10: hasta que t > t_max
11: fin BVNS
```

2.2. VFS

Asúmase que además de la formulación original para un problema dado y su correspondiente función objetivo $\mathcal{F}_0(x) = \mathcal{F}(x)$ existen otras (p) formulaciones diferentes para el problema, que pueden denotarse como $\mathcal{F}_1(x)$, $\mathcal{F}_2(x)$, $\ldots, \mathcal{F}_p(x)$, $\forall\, x \in X$, donde x es una solución y X es el conjunto de soluciones factibles. Nótese que dos formulaciones son equivalentes si la solución óptima en una de ellas es equivalente a la solución óptima en la otra. Por lo tanto, dos formulaciones diferentes pero equivalentes, podrían tener el mismo valor de la función objetivo. No obstante, por simplicidad, considérense diferentes formulaciones como diferentes funciones objetivo $\mathcal{F}_i(x), i = 1, \ldots, p$.

La idea principal de VFS consiste en utilizar múltiples formulaciones de un mismo problema para comparar soluciones con el mismo valor de la función objetivo, en la formulación original. Para ello, se introduce el procedimiento `Accept`(x, x', p), presentado en el Algoritmo 2, en las tres etapas clave del esquema BVNS donde se realizan comparaciones de soluciones: `Shaking`, `LocalSearch` y `NeighbourhoodChange`. Como es evidente, si tras realizar un movimiento una solución no mejora en ninguna de las p formulaciones del problema propuestas, el movimiento es rechazado. El procedimiento `Accept` (x, x', p) recibe tres parámetros, donde x y x' representan las soluciones comparadas y p es el número de formulaciones consideradas. Así, para cualquier problema particular, es necesario diseñar diferentes formulaciones y decidir el orden en que serán empleadas dentro del procedimiento `Accept`. Las respuestas a estas preguntas son específicas del problema y en ocasiones difíciles de resolver.

Algoritmo 2 Procedimiento `Accept`

```
1: Procedimiento Accept (x, x', p)
2: para i = 0 a p hacer
3:    condicion1 = F_i(x') < F_i(x)
4:    condicion2 = F_i(x') > F_i(x)
5:    si (condicion1) entonces
6:       devolver Cierto
7:    si no
8:       si (condicion2) entonces
9:          devolver Falso
10:      si no
11:         continuar /*con la siguiente formulación*/
12:      fin si
13:   fin si
14: fin para
15: devolver Falso
16: fin Accept
```

3. Problema de la Minimización de la Anchura de Corte

El problema de la Minimización de la Anchura de Corte en ordenaciones lineales (CMP, por su nombre en inglés *Cutwidth Minimization Problem*) es un problema NP-Difícil [3] de tipo min-max. El CMP consiste en encontrar una ordenación de los vértices de un grafo en una línea, tal que minimice el máximo número de aristas que cortan una línea entre vértices consecutivos. Este problema puede formularse tanto de manera combinatoria como matemática [6].

La formulación combinatoria para el CMP, en la que se centra este trabajo, se describe de la siguiente manera: dado un grafo $G = (V, E)$ con $|V| = n$, una ordenación o etiquetado $f : V \to \{1, 2, \ldots, n\}$ de G asigna los enteros $\{1, 2, \ldots, n\}$ a los vértices en V, de tal modo que cada vértice recibe una etiqueta diferente. Dada la ordenación, el *cutwidth* de un vértice v, denotado como $CW_f(v)$, es el número de aristas $(u, w) \in E$ que satisfacen $f(u) \leq f(v) < f(w)$. Por lo tanto, $CW_f(v) = |\{(u, w) \in E : f(u) \leq f(v) < f(w)\}|$. Como consecuencia de la definición previa es posible definir el *cutwidth* de G con respecto a f, denotado como $CW_f(G)$, como el máximo de entre todos los valores de *cutwidth* entre todos sus vértices. Formalmente $CW_f(G) = \text{máx}_{v \in V}\, CW_f(v)$, por lo que el *cutwidth* óptimo de G se define como el mínimo $CW_f(G)$ sobre Π_n, que representa todas las posibles ordenaciones de los vértices del grafo de tamaño n:

$$CW(G) = \min_{f \in \Pi_n} CW_f(G)$$

Ejemplo. En las Figuras 1.a y 1.b se muestran respectivamente un grafo G y una ordenación f de los vértices de G junto con el valor del *cutwidth* asociado a cada vértice. En este ejemplo, el *cutwidth* de G con respecto a f es $CW_f(G) = 6$, debido a que es el máximo *cutwidth* de cualquier vértice en la ordenación. Está representado por una línea discontinua resaltada.

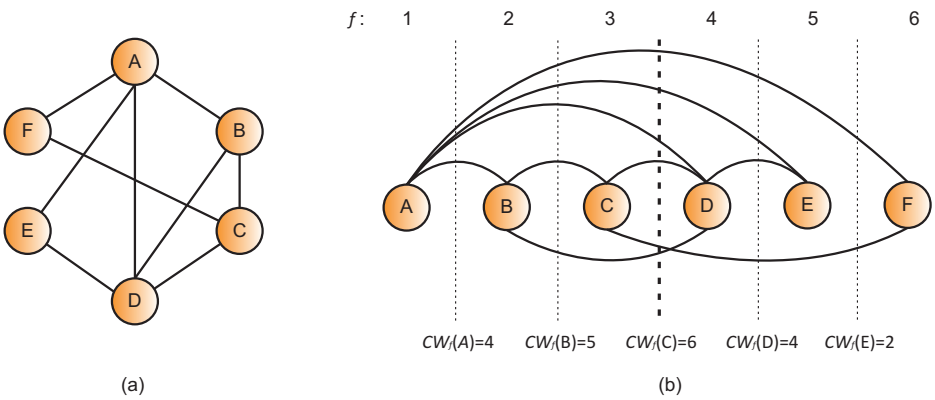


Figura 1. (a) Grafo G con 6 vértices y 9 aristas. (b) Ordenación f de los vértices del grafo en (a) con el correspondiente valor del *cutwidth* asociado.

Este problema ha sido abordado ampliamente desde diferentes perspectivas. Las primeras aproximaciones a la resolución heurística del problema se remontan a los años ochenta, cuando se propuso un algoritmo de *Simulated Annealing* (SA) para una versión generalizada del problema [2] . Tiempo después, en [1] se desarrolló un algoritmo tipo GRASP con *Path Relinking* (GPR). Las heurísticas más recientes para el problema pueden ser encontradas en [9] y [10] donde se propusieron, respectivamente, un *Scatter Search* (SS) y un VNS para el CMP. Existen también numerosos métodos exactos para tipos de grafos particulares, así como algún método genérico. Entre estos últimos, en [6] se propuso una formulación entera y en [7] un algoritmo de Ramificación y Acotación.

3.1. Formulación basada en subconjuntos

Esta formulación alternativa está basada en las ideas presentadas en [9]. Dada una ordenación f de los vértices de un grafo, es posible definir subconjuntos de vértices de acuerdo con el valor de su *cutwidth* asociado. Sea S_f^i el conjunto de vértices con valor de *cutwidth* i en la ordenación f. En términos matemáticos,

$$S_f^i = \{v \in V : CW_f(v) = i\}.$$

Considerando el ejemplo representado en la Figura 1.b, se tendrían los siguientes subconjuntos: $S_f^0 = \{\texttt{F}\}$, $S_f^1 = \emptyset$, $S_f^2 = \{\texttt{E}\}$, $S_f^3 = \emptyset$, $S_f^4 = \{\texttt{A,D}\}$, $S_f^5 = \{\texttt{B}\}$ y $S_f^6 = \{\texttt{C}\}$. Sea i_{max} el índice que identifica el conjunto que contiene los vértices con el mayor *cutwidth*. En este ejemplo, el valor de i_{max} es 6. Obviamente, $i_{max} = CW_f(G)$, por lo que el CMP es equivalente a reducir i_{max}. No obstante, con esta formulación alternativa se podrían comparar dos soluciones más allá del valor de i_{max}, tomando en cuenta la cardinalidad de los subconjuntos S_f^i.

En general, dadas dos soluciones, f y f', y un índice j $(1 \leq j \leq i_{max})$ se podría afirmar que f es mejor f' si y solo si $|S_f^j| < |S_{f'}^j|$ y $|S_f^i| = |S_{f'}^i|$ para

todo i, tal que $j < i \leq i_{max}$. Juntando las anteriores condiciones en una misma expresión matemática, se propone la definición de una nueva función objetivo para el CMP, $CW_f^2(G)$, de la siguiente manera:

$$CW_f^2(G) = \sum_{i=1}^{i_{max}} n^i |S_f^i|$$

Por lo tanto, el CMP puede ser reformulado así:

$$CW^2(G) = \min_{f \in \Pi_n} CW_f^2(G)$$

donde $CW^2(G)$ es el mínimo valor de CW_f^2 sobre todos las posibles ordenaciones Π_n. Notar que una solución óptima en $CW^2(G)$ es también óptima en $CW(G)$, pero lo contrario no es necesariamente cierto.

3.2. Formulación basada en la distancia a una solución ideal

Sean f_1 y f_2 dos soluciones al CMP. Se podría definir $d(f_1, f_2)$ como la función de distancia entre estas dos soluciones. Particularmente, si una de las dos soluciones (f^*) es óptima, entonces el CMP podría ser formulado como sigue:

$$CW^3(G) = \min_{f \in \Pi_n} d(f, f^*)$$

donde $CW^3(G)$ es el mínimo valor de $d(f, f^*)$ sobre todas las posibles ordenaciones en Π_n. Desafortunadamente, considerar la solución óptima como punto de referencia no es posible dado que, de hecho, es el objetivo del problema. No obstante, en lugar de utilizar una solución óptima, se podría considerar una solución «ideal» (quizá infactible) $f_$, como punto de referencia.

La solución «ideal» debería tener un *cutwidth* al menos tan bueno como el óptimo. Esto significa que $CW_{f_}(G)$ es una cota inferior para el CMP. A pesar de que minimizar $d(f, f_)$ no necesariamente implica aproximarse a la solución óptima para el CMP, puede ser de utilidad a la hora de encontrar buenas soluciones. Cualquier solución (probablemente infactible) cuya función objetivo sea una cota inferior para el CMP, puede ser utilizada como solución «ideal». Aquí se propone utilizar el procedimiento propuesto en [7] (ver Sección 3.4 en dicho artículo) para construir una solución ideal.

Se proponen además dos funciones de distancia distintas. La primera, inspirada en la formulación presentada en la Sección 3.1, se basa en la comparación de subconjuntos. Dada una solución f, definimos $d_1(f, f_)$ de la siguiente manera:

$$d_1(f, f_) = \sum_{i=1}^{i_{max}} \text{abs}\left(|S_f^i| - |S_{f_}^i|\right)$$

No obstante, la distancia d_1 presenta el inconveniente de que solo considera la diferencia absoluta entre los subconjuntos S_f^i y $S_{f_}^i$, pero no dónde aparecen estas diferencias. La segunda distancia propuesta, d_2, está basada en la definición de

etiquetado. Cada vértice v en el grafo tiene asignada una etiqueta $f(v)$ diferente que indica la posición i que el vértice ocupa en la ordenación (Ver Figura 1.b). Considérese $f^{-1}(i)$ como la función que, dada una posición i, determina el vértice correspondiente asociado a esa posición. Por definición, la relación entre f y f^{-1} es tal que si $f(v) = i$ entonces $f^{-1}(i) = v$. A través de la minimización de d_2 se intenta obtener soluciones próximas a la solución «ideal». Específicamente, se minimizan las diferencias entre el *cutwidth* de los vértices situados en la misma posición en $f_$ y f. Por lo tanto, la distancia se define de la siguiente manera:

$$d_2(f, f_) = \text{mín}\Big(\sum_{1 \le i < n} \text{abs}(CW_f(f^{-1}(i)) - CW_{f_}(f_^{-1}(i))), \sum_{1 \le i < n} \text{abs}(CW_f(f^{-1}(i)) - CW_{f_}(f_^{-1}(n-i)))\Big)$$

La distancia d_2 considera que el etiquetado inverso es equivalente para el CMP. Sean $f = (v_1, v_2, \ldots, v_n)$ y $f' = (v'_1, v'_2, \ldots, v'_n)$ dos etiquetados de un conjunto de n vértices donde $f^{-1}(i) = f'^{-1}(n-i+1)$, $1 \le i \le n$. Entonces, se puede decir que f y f' son etiquetados inversos y, por lo tanto, equivalentes para el CMP. En particular, $CW_f(f^{-1}(i))$ es igual a $CW_{f'}(f'^{-1}(n-i))$, $\forall\, 1 \le i < n$.

3.3. Búsqueda de Formulación Variable para el CMP

El objetivo principal de este trabajo no es describir con detenimiento las fases de construcción de la solución, estrategia de agitación, búsqueda local y cambio de vecindad (elementos estándar en VNS). Por ello, se han empleado las estrategias propuestas en [10] como punto de partida para aplicar las estrategias aquí propuestas. La modificación principal ha consistido en añadir el procedimiento `Accept` (ver Algoritmo 3) a las fases de agitación, búsqueda local y cambio de vecindad a la hora de comparar soluciones. En él se utilizan la formulación original ($CW(G)$) y las formulaciones alternativas ($CW^2(G)$ y $CW^3(G)$) para comparar dos soluciones f_1 y f_2. Si $CW_{f_1}(G) < CW_{f_2}(G)$, se puede concluir que f_1 es mejor que f_2 (paso 2). En cambio, si ambas soluciones tienen el mismo valor de la fución objetivo original, no es posible asegurar cuál es mejor sin la utilización de formulaciones alternativas. En particular, si $CW_{f_1}(G) = CW_{f_2}(G)$, se puede considerar que f_1 es más prometedora que f_2 cuando $CW^2_{f_1}(G) < CW^2_{f_2}(G)$ (paso 4). Similarmente, también se puede considerar f_1 más prometedora que f_2 cuando $CW_{f_1}(G) = CW_{f_2}(G)$ y $CW^3_{f_1}(G) < CW^3_{f_2}(G)$ (paso 6). Por último, si ninguna de las anteriores condiciones es satisfecha, el procedimiento devuelve Falso, lo que significa que f' no es más prometedora que f.

4. Resultados experimentales

En esta sección se presentan los resultados experimentales llevados a cabo para validar la eficiencia de la metodología propuesta. En primer lugar se presentan los conjuntos de instancias empleados para su evaluación, a continuación

Algoritmo 3 Procedimiento Accept para el CMP

1: **Procedimiento** `Accept` (f, f')
2: **si** $(CW_{f'}(G) < CW_f(G))$ **entonces**
3: **devolver** Cierto
4: **si no**, **si** $(CW_{f'}(G) = CW_f(G))$ **y** $(CW^2_{f'}(G) < CW^2_f(G))$ **entonces**
5: **devolver** Cierto
6: **si no**, **si** $(CW_{f'}(G) = CW_f(G))$ **y** $(CW^3_{f'}(G) < CW^3_f(G))$ **entonces**
7: **devolver** Cierto
8: **si no**
9: **devolver** Falso
10: **fin si**
11: **fin** `Accept`

se muestra el impacto de la introducción de las formulaciones alternativas al problema y, por último, se compara la mejor versión del algoritmo con algoritmos previos del estado del arte. Los algoritmos fueron implementados en Java 6 SE y los experimentos realizados en un Intel Core 2 Quad CPU, con 6 GB RAM.

Para evaluar los algoritmos propuestos se han utilizado dos conjuntos de instancias ("*Grid*" y "*Harwell-Boeing*"). Estas instancias fueron propuestas en [9], donde pueden encontrarse descripciones detalladas de las mismas. Además, todas las instancias se encuentran disponibles en `http://www.optsicom.es/cutwidth`.

El primer experimento está destinado a la comparación del uso de las diferentes formulaciones propuestas para el problema, en combinación, en lugar de utilizar solo una de ellas. Para realizarlo se ha utilizado un subconjunto representativo de instancias de los conjuntos "*Grid*" y "*Harwell-Boeing*". Concretamente se han tomado 16 instancias (el 10 % de los conjuntos).

	BVNS	**VFS$_1$**	**VFS$_2$**	**VFS$_3$**
Promedio F.O.	137.31	93.56	91.56	90.75
Desviación (%)	192.44	60.40	49.23	48.22
Tiempo CPU (s)	30.17	30.47	30.50	30.96

Tabla 1. Impacto de la utilización de formulaciones alternativas.

En la Tabla 1, se presentan los resultados obtenidos con cuatro algoritmos diferentes tras ser ejecutados durante 30 segundos sobre cada una de las 16 instancias mencionadas. La columna BVNS representa el algoritmo basado en la metodología BVNS que utiliza únicamente la formulación original para el CMP en el proceso de búsqueda. VFS$_1$ es la primera versión de la variante metodológica propuesta en este trabajo y está basada en la combinación del BVNS mencionado junto con la formulación CW^2 presentada en la Sección 3.1. VFS$_2$ es equivalente a VFS$_1$ pero utilizando esta vez CW^3 (Ver Sección 3.2) en lugar de CW^2. Finalmente, la cuarta columna (VFS$_3$) combina la formulación original con las dos anteriores empleando el procedimiento descrito en la Sección

3.3. Todos los algoritmos fueron configurados con $k_{max} = 0{,}1n$ (siendo n el número de vértices del grafo) y comenzaron desde la misma solución inicial.

Los resultados presentados en la Tabla 1 claramente confirman que la utilización de más de una formulación para determinar con qué solución continuar la búsqueda es más eficaz que emplear solo la formulación original. En este sentido, la combinación de varias formulaciones (columna VFS_3) fue ligeramente mejor que la utilización de cada formulación por separado, por lo que es la configuración elegida para la experimentación final.

Finalmente, se compara la mejor versión de VFS con los principales algoritmos presentes en el estado del arte sobre los conjuntos de instancias completos. Específicamente, los procedimientos comparados son: Simulated Annealing (SA), [2]; Greedy Randomized Adaptive Search Procedure con Path Relinking (GPR), [1]; Scatter Search (SS), [9] y VFS. El algoritmo de VFS fue configurado de la siguiente manera: se construyeron 1000 soluciones con el procedimiento constructivo descrito en [10], de entre las que se escogió la mejor como punto de partida. Se empleó la búsqueda local basada en inserciones propuesta en [10]. El parámetro k_{max} se estableció a $0{,}2n$ de manera experimental y, el máximo tiempo de cómputo para cada instancia se fijó en $0{,}4n$, siendo n el número de vértices de cada instancia. Finalmente, se utilizó la variante VFS_3 donde las dos formulaciones alternativas propuestas fueron utilizadas para determinar qué solución es la más prometedora para continuar la búsqueda.

En la Tabla 2 se presenta la comparación con los algoritmos anteriormente mencionados sobre el conjunto de instancias *Grid* y *HB*. En ambos casos es posible observar como la variante de VFS elegida mejora a los algoritmos previos del estado del arte. En particular, en el conjunto *Grid*, VFS obtuvo un 3.2 % de desviación, mientras que el siguiente mejor algoritmo (SS) obtuvo un 7.8 % de desviación, más del doble en más del doble del tiempo de cómputo. De manera similar, en el conjunto *HB*, VFS es nuevamente el mejor método, obteniendo un 1.8 % frente al método más cercano (SS) con un 3.4 %. Nuevamente, en más del doble del tiempo de cómputo. Para complementar esta información, se ha aplicado el test de Friedman a las mejores soluciones obtenidas por cada método. Este experimento fue llevado a cabo de manera independiente para los conjuntos *Grid* y *HB*. En ambos casos, el p-valor resultante de 0.000 indica que existen diferencias significativas entre los métodos comparados.

	Grid (81)				**HB (87)**			
	GPR [1]	**SA** [2]	**SS** [9]	**VFS**	**GPR** [1]	**SA** [2]	**SS** [9]	**VFS**
Promedio F.O.	38.4	16.1	13.0	12.2	364.8	346.2	315.2	314.4
Desviación (%)	201.8	25.4	7.8	3.2	95.1	53.3	3.4	1.8
#Mejores	2	37	44	59	2	8	47	61
Tiempo CPU (s)	235.2	216.1	210.1	90.3	557.5	435.4	430.6	128.1

Tabla 2. Comparación con los algoritmos del estado del arte (Instancias Grid y HB).

5. Conclusiones

En este trabajo se ha propuesto una nueva variante de la metodología VNS. La nueva variante, denominada Búsqueda de Formulación Variable, utiliza las características principales de VNS y añade nuevos criterios para la comparación de soluciones. El propósito de esta metodología es determinar qué solución es más prometedora para continuar la búsqueda, más allá del valor de la función objetivo original. Este hecho es de especial relevancia en problemas de tipo min-max donde existen muchas soluciones con el mismo valor de la función objetivo. Se ha ilustrado la efectividad de la propuesta mediante su aplicación al problema de la Minimización de la Anchura de Corte en ordenaciones lineales. Los resultados obtenidos han sido comparados favorablemente en términos de calidad y tiempo de cómputo con los métodos previos más detacados en el estado del arte.

Agradecimientos Esta investigación ha sido parcialmente financiada por los proyectos de investigación con referencias: TIN2009-07516, TIN2011-28151 y TIN2012-35632.

Bibliografía

1. D. V. Andrade and M. G. C. Resende. GRASP with Path-Relinking for Network Migration Scheduling. In *Proceedings of International Network Optimization Conference (INOC)*, 2007.
2. J. Cohoon and S. Sahni. Heuristics for backplane ordering. *Journal of VLSI and Computer Systems*, 2:37–61, 1987.
3. F. Gavril. Some NP-complete problems on graphs. In *Proceedings of the Eleventh Conference on Information Sciences and Systems*, pages 91–95, Baltimore, 1977.
4. P. Hansen, N. Mladenovic, J. Brimberg, and J. A. Moreno Pérez. *Handbook of metaheuristics*, chapter Variable Neighbourhood Search, pages 61–86. Number 146. Kluwer, 2nd edition, 2010.
5. P. Hansen, N. Mladenovic, and J. A. Moreno Pérez. Variable neighbourhood search: algorithms and applications. *Annals of Operations Research*, 175:367–407, 2008.
6. J. Luttamaguzi, M. Pelsmajer, Z. Shen, and B. Yang. Integer programming solutions for several optimization problems in graph theory. Technical report, Center for Discrete Mathematics and Theoretical Computer Science, DIMACS, 2005.
7. R Martí, J. J. Pantrigo, A. Duarte, and E. G. Pardo. Branch and bound for the Cutwidth Minimization Problem. *Computers & Op. Res.*, 40(1):137–149, 2013.
8. N. Mladenović and P. Hansen. Variable Neighborhood Search. *Computers & Operations Research*, 24(11):1097–1100, 1997.
9. J. J. Pantrigo, R. Martí, A. Duarte, and E. G. Pardo. Scatter search for the cutwidth minimization problem. *Annals of Operations Research*, 199:285–304, 2012. DOI:10.1007/s10479-011-0907-2.
10. E. G. Pardo, N. Mladenovic, J. J. Pantrigo, and A. Duarte. A variable neighbourhood search approach to the cutwidth minimization problem. *Electronic Notes in Discrete Mathematics*, 39(0):67 – 74, 2012.
11. E. Piñana, I. Plana, V. Campos, and R. Martí. GRASP and Path Relinking for the Matrix Bandwidth Minimization. *European Journal of Operational Research*, 153(1):200 – 210, 2004.
12. M. G. C. Resende, R. Martí, M. Gallego, and A. Duarte. Grasp and path relinking for the max-min diversity problem. *Computers & Op. Res.*, 37:498–508, 2010.

Algoritmo de Búsqueda Tabú para el HVRP con ventanas de tiempo y criterios de sostenibilidad

Ignacio Eguía (ies@us.es); José Carlos Molina (jmolina12@us.es); Jesús Racero (jrm@us.es); Fernando Guerrero (fergue@us.es)

Escuela Superior de Ingeniería. Universidad de Sevilla
Camino de los Descubrimientos s/n. 41092 Sevilla, Spain

Abstract.

En el problema de rutas de vehículos con flota heterogénea (HVRP) hay diferentes tipos de vehículos y un número limitado de cada tipo. La resolución de este problema consiste en asignar los clientes a los vehículos existentes, definiendo el orden de visita a cada cliente y cumpliendo las limitaciones de capacidad de cada vehículo. Cuando existen ventanas de tiempo además se deben cumplir dichas limitaciones de tiempo por cliente. La flota heterogénea y las ventanas de tiempo dotan de mayor realismo al problema VRP. El objetivo tradicional de los problemas HVRP se centran en minimizar los costes fijos (por uso del vehículo) y variables (por distancia recorrida), que cambian para cada tipo de vehículo.

Dado el interés cada vez mayor en la preservación del medio ambiente, los aspectos ambientales juegan un papel importante en las políticas estratégicas y operacionales, y más concretamente en el diseño de rutas. En este artículo, se presenta un algoritmo de Búsqueda Tabú para la resolución de problemas VRP con criterios económicos y medioambientales y donde se incorporan aspectos realistas como ventanas de tiempo y flota limitada de vehículos heterogéneos.

Keywords: Búsqueda Tabú, Vehicle Routing Problem, Flota Heterogénea, Green Logistics, Sostenibilidad.

1 Introducción

El problema de enrutamiento de vehículos (VRP) es un problema de optimización combinatoria de gran importancia en diferentes entornos logísticos. Consiste en encontrar el conjunto de rutas óptimas para una flota de vehículos homogénea cuya finalidad es la de servir una demanda conocida a una serie de clientes ubicados geográficamente de manera dispersa, saliendo y volviendo a un único depósito.

En la industria, una flota de vehículos es raramente homogénea, ya que la necesidad de estar presente en diversos segmentos del mercado, obliga a muchas empresas a disponer de vehículos que se adapten a la tipología de la mercancía transportada. De igual modo, el disponer de vehículos con diferentes unidades de carga, permite una mejor adaptación a la demanda. La variante VRP para flotas heterogéneas (HFVRP),

aparece cuando los diferentes vehículos que conforman la flota difieren en equipamiento, capacidad, antigüedad, estructura de costes o incluso nivel de emisiones. A los problemas con un número limitado de vehículos se les denomina "Heterogeneous VRP" (HVRP), mientras que la versión ilimitada se les denomina "Fleet Size and Mix" (FSM). Por otra parte, a las variantes que consideran restricciones de ventanas de tiempo, se les añade "TW" al acrónimo del problema específico.

El transporte tanto de mercancías como de pasajeros es una de las principales fuentes de contaminación del aire en Europa, dando lugar a otro tipo de externalidades como son la congestión, los ruidos y los accidentes. En este entorno, las empresas deben incluir todas estas externalidades en los criterios de diseño de las rutas, con el fin de reducir estos factores externos que representan un coste no percibido directamente por el usuario y que son fuente de problemas de salud graves.

El objetivo de este artículo es la descripción de un algoritmo basado en la búsqueda tabú para los problemas VRP, que incluya estas externalidades y con entornos realistas como son flotas heterogéneas y restricciones de ventanas de tiempo en los clientes servidos. Se dispone de una flota heterogénea de vehículos caracterizados por diferentes capacidades, costes internos (fijos, mantenimiento, conductor y consumos) y tecnologías del motor (Euro) que determinarán las emisiones de contaminantes a la atmósfera. La función objetivo considerada contempla tanto los aspectos relativos a los costes internos como los debidos a las externalidades. Los *costes internos* incluyen los siguientes componentes: (i) costes por horas de conducción, (ii) costes energéticos por consumo de combustible, (iii) costes fijos por uso del vehículo, (iv) costes de mantenimiento por distancia recorrida y (v) costes de peaje si los hubiera. Mientras que los *costes externos* son atribuidos a: (i) las emisiones de gases de efecto invernadero (GEI), (ii) las emisiones de contaminantes atmosféricos, (iii) las emisiones de ruido y (iv) los accidentes. Estos cuatro componentes reflejan el 88% de la carga de los costes externos media total en la Unión Europea, con exclusión de los costes de congestión (INFRAS / IWW, 2004).

Los principales factores de costes asociados al cambio climático se deben al contenido de carbono del combustible. El coste total se obtendrá multiplicando los litros consumidos por un vehículo, por el factor de emisión de CO_2 en su combustión, por el coste recomendado de las emisiones asociadas al cambio climático. Los costes de contaminación son causados por la emisión de otros contaminantes atmosféricos tales como partículas inferiores a 2,5 micras o inferiores a 10 micras, NOx y compuestos orgánicos volátiles diferentes al metano. La internalización de costes por contaminación de cada contaminante puede ser obtenida mediante el producto del número de kilómetros realizados por los gramos de contaminantes emitidos por kilómetro y el coste asociado a cada gramo. Los costes de ruido están basados en estudios de exposición de las personas al ruido y de los costes de molestia y salud. Los costes de accidentes se basan en estudios de costes de accidentalidad no cubiertos por el seguro. Ambos dependen de la distancia recorrida y de la carga transportada.

El resto del artículo se organiza de la siguiente manera. En la Sección 2 se describe el estado del arte de heurísticas y metaheurísticas para la resolución de problemas HVRP y de estudios sobre VRP con criterios de sostenibilidad. En la Sección 3 se describe el algoritmo basado en búsqueda tabú presentado en este artículo, mientras

que en la Sección 4 se detallan los resultados computacionales obtenidos en problemas de la literatura. Finalmente, la Sección 5 muestra las observaciones finales y las sugerencias para investigaciones futuras.

2 Estado del arte

2.1 Algoritmos heurísticos para el problema HVRP

Al ser el problema HFVRP un caso especial del VRP, pertenece igualmente a la categoría de problemas NP-Hard, en los que los tiempos computacionales requeridos para su resolución por medio de métodos exactos se incrementan exponencialmente con el tamaño del problema. Debido a la dificultad que representa este tipo de problemas, la mayor parte de las soluciones expuestas en la literatura han sido obtenidas mediante la utilización de heurísticas y metaheurísticas.

Los primeros estudios de problemas de enrutamiento de vehículos con flotas heterogéneas, en particular a problemas FSM, son debidos a Golden et al. (1984). En ellos, los autores describen heurísticas constructivas basadas en el cálculo de ahorros y en la ruta gigante (*Giant Tour)*. El enfoque del Giant Tour propuesto por Golden et al. (1984) consiste primeramente en la formación de una gran ruta en la que se visita a todos los clientes, resolviendo el problema del agente viajero (TSP). A continuación, en una segunda fase, se divide dicha ruta en otras más pequeñas haciéndolas partir desde el *depot* o almacén, mediante la resolución del problema del camino más corto. Gheysens et al (1984) presentan heurísticas para el problema FSM, incorporando las limitaciones de la capacidad del vehículo en la función objetivo junto con los costes fijos y variables.

Taillard (1999) fue el primer autor que aborda el problema HVRP, presentando una heurística conocida como el método de generación de columnas. Liu and Shen (1999) fueron los primeros en tratar el problema FSM con TW y describieron varias heurísticas de inserción basadas en ahorros para su resolución.

Todas las soluciones planteadas en la literatura son, por lo general, adaptaciones o ampliaciones de heurísticas aplicadas a las variantes clásicas VRP. De esta forma, desde finales de los años 90, se comenzaron a aplicar conjuntamente las metaheurísticas para la búsqueda de soluciones de alta calidad a los problemas HFVRP.

Una de las metaheurísticas más aplicadas a los problemas VRP es la búsqueda tabú. Osman and Salhi (1996), aplicaron por primera vez la búsqueda tabú al problema FSM. Gendreau et al. (1999) propusieron heurísticas basadas en la búsqueda tabú para los problemas FSM. Prins (2002), consideró el problema HVRP en el que la función objetivo dependía de la duración de las rutas. Para ello, formuló heurísticas constructivas y una búsqueda tabú que podía abordar la variante de multiviaje. Posteriormente, Lee et al. (2008) desarrollaron un algoritmo basado en la búsqueda tabú y partición de conjuntos y Brandao (2009) desarrolló una heurística tabú determinista que proporciona excelentes resultados. Las soluciones iniciales se obtienen a partir del algoritmo GENIUS y del Giant Tour. Ambos autores compararon los resultados obtenidos para los problemas FSM. Brandao (2011) adaptó su algoritmo anterior, haciendo algunos

cambios, para resolver el problema HVRP, comprobando los resultados en los problemas de Taillard.

Otras metaheurísticas utilizadas para problemas HFVRP son los algoritmos de aceptación de umbral (Tarantilis et al. (2003)), algoritmos genéticos (Liu et al. (2009)), algoritmos meméticos (Prins (2009)), algoritmos record-to-record (Li et al. (2007)) o algoritmos de búsqueda en la vecindad variable VNS (Imran et al. 2009).

2.2 Criterios de sostenibilidad en VRP

Al mismo tiempo que el problema VRP persigue la reducción de las distancias recorridas y de los vehículos totales asignados, se están satisfaciendo los requerimientos de la *Green Logistics*, al reducir consecuentemente las cantidades de combustible utilizadas. La reducción de las cantidades de fuel, implica la reducción de las emisiones de CO_2, aunque existen otros factores en la *Green Logistics* que podrían ser considerados en un modelo de VRP y que han sido estudiados durante los últimos años.

Kara et al. (2007) introdujo el problema VRP de Minimización de Energía (EMVRP). En él se define una nueva función de coste que tiene en cuenta tanto la carga del vehículo como las distancias. Esto conduce a minimizar el consumo de fuel total. Como resultado, se pudo comprobar que existe una diferencia considerable entre las soluciones obtenidas al minimizar distancias y al minimizar energías.

Bektas and Laporte (2011) definen el problema PRP (Pollution Routing Problem) como una variante del VRP donde se incluye como objetivo reducir al mínimo el coste de las emisiones de CO_2, junto con los costes operativos de los conductores y el consumo de combustible. Analizan y comparan diferentes medidas de rendimiento, tales como la distancia, la carga, las emisiones y los costes a través de una variedad de funciones objetivo. Asumen que la carga y la velocidad pueden cambiar de un arco a otro en el viaje de un vehículo, permaneciendo constante el resto de parámetros.

Úbeda et al. (2011) presentaron un caso de estudio para mostrar que la introducción de prácticas ecológicas en el proceso de toma de decisiones de todos los días en una empresa de transportes, puede cumplir simultáneamente la eficiencia y los objetivos medioambientales. Incorporaron las emisiones de CO_2 en la función objetivo del CVRP con viajes de regreso (*backhauls*) y como restricción un tiempo máximo permitido de conducción.

En el estudio desarrollado por Eguía et al. (2012), se propone un modelo matemático de programación lineal que internaliza los costes externos en problemas tipo HVRPTW-B apoyándose en las conversiones establecidas por el INFRAS/IWW (2004). Se evalúa los costes de la flota de vehículos y la cuantía de sus emisiones de contaminantes para añadirlos al modelo clásico. Finalmente, se corrobora que este nuevo concepto de diseño de rutas con criterios medioambientales obtiene diferentes resultados al problema que minimiza únicamente distancias o costes internos.

Los estudios anteriormente descritos se han publicado recientemente. Esto demuestra que el tema está en sus orígenes de desarrollo, siendo aún su temática atractiva y bastante demandada.

3 Algoritmo de Búsqueda Tabú

Se han visto diversos enfoques heurísticos del problema HFVRP, destacando el uso de metaheurísticas y más concretamente la Búsqueda Tabú (BT). Por ello, se ha decidido en este artículo desarrollar un algoritmo de BT para la resolución de problemas VRP con flotas heterogéneas, ventanas de tiempo y con criterios de sostenibilidad.

La búsqueda tabú es un algoritmo de optimización, que partiendo de una solución inicial, trata de mejorarla mediante la exploración de soluciones vecinas. En la búsqueda de una solución óptima, permite no quedarse atrapados en mínimos locales, mediante el uso de estructuras de memoria, que previenen volver a soluciones ya examinadas, o de una forma más genérica previene volver hacia soluciones que compartan unos ciertos atributos con soluciones ya examinadas.

El diseño del algoritmo de BT de este artículo se basa en los algoritmos de Osman and Salhi (1996) y de Gendreau et al. (1999), incluyendo una serie de adaptaciones al problema con ventanas de tiempo y criterios de sostenibilidad.

El primer paso del algoritmo de BT consiste en representar cada solución del problema. En dicha representación deben aparecer los clientes a visitar, los vehículos y en el orden de paso. Una solución queda unívocamente definida por tres vectores: *vec_clientes* contiene todos los clientes del grafo ordenados de izquierda a derecha por orden de visita; *vec_vehiculos* indica el número de clientes que atiende el vehículo *i*; y la posición *i* de *vec_punteros* señala la posición en *vec_clientes* donde se sitúa la ruta asignada al vehículo *i*.

3.1 Generación de la solución inicial

La generación de una solución inicial proporciona una asignación de vehículos a rutas admisibles tanto en capacidad como en ventanas de tiempo, de forma rápida. En este artículo se ha optado por una generación en dos fases: primero se adapta la heurística de inserción secuencial de M&J (Mole and Jameson, 1976), y a continuación se mejora la solución mediante un procedimiento de búsqueda local con el operador λ-opt.

La heurística de inserción de M&J utiliza dos medidas para decidir el próximo cliente a insertar en la ruta actual. En primer lugar, para cada cliente no visitado se calcula la mejor posición para ubicarlo en la ruta actual teniendo en cuenta solamente las distancias y sin reordenar los nodos que ya están en la ruta. Se tiene una ruta (v_0, v_1,…., v_t, v_{t+1}) donde $v_0 = v_{t+1} = 0$. Si w es un cliente no visitado, el costo de insertar w entre v_i y v_{i+1} ($0 \leq i \leq t$) se define como $c_1(v_i,w)=c_{vi,w}+c_{w,vi}-\lambda c_{vi,vi+1}$ si (v_0,…., v_i,w,v_{i+1},….,v_{t+1}) es factible. La mejor posición para insertar el cliente w en la ruta actual está dado por, $i(w) = arg\ min\{c_1(v_i,w): i=0,...t\}$.

Si se utilizara solamente la medida c_1 para decidir el próximo cliente a insertar, es probable que los clientes lejanos al depósito solo sean tenidos en cuenta en las iteraciones finales del algoritmo. Por lo tanto es necesario utilizar un incentivo adicional para la inserción de clientes lejanos al depósito. Se define $c_2(v_i,w)=\mu c_{0w}-c_1(v_i,w)$ para cada cliente w. En cada iteración se busca el cliente que maximiza la medida c_2 (llamada medida de urgencia) y se inserta en la posición con el mínimo valor de c_1. Tanto λ como μ son parámetros del algoritmo de M&J. Además de las medidas anteriores,

debe considerarse la factibilidad de las inserciones. Cuando ninguna inserción es factible y si aún quedan clientes sin visitar, se selecciona un cliente para comenzar una nueva ruta con un nuevo vehículo.

Esta heurística originalmente se desarrolla para flota homogénea y se ha adaptado al problema en estudio de la siguiente forma: (i) la flota se ordena según su capacidad (creciente o decreciente) y se eligen los vehículos en ese orden, (ii) los costes se adaptan a los criterios de sostenibilidad, y (iii) se consideran los tiempos de llegadas a cada cliente para comprobar la admisibilidad de la ventana de tiempo, incluida la llegada final al *depot.*

El operador λ-opt es un mecanismo de búsqueda local que elimina λ arcos (λ>1) y reconecta los λ segmentos restantes. Dependiendo del número de nodos, la cantidad de intercambios posibles varía, siendo lo habitual la selección de los operadores 2-opt y 3-opt. En nuestro caso se ha optado por aplicar estos dos operadores al final del algoritmo de M&J con la idea de mejorar el orden de paso por los clientes.

3.2 Movimientos y soluciones vecinas

Se van a definir dos grupos de movimientos para generar las soluciones vecinas de una solución, los movimientos de intercambio y los movimientos de inserción.

Cada solución viene definida por grupos de clientes ordenados o rutas, asignadas a distintos vehículos de la flota. Este conjunto de rutas se recogen en $R = \{r_1,\ldots,r_h\}$ y cada ruta, r_i, está compuesta por un conjunto finito de clientes $V = \{v_1,\ldots,v_n\}$.

En un *movimiento de intercambio* intervienen un par de clientes, v_i y v_j, ($i<>j$) que pertenecen, respectivamente, a un par de rutas, r_q y r_p, ($q<>p$). El intercambio se define como el movimiento de v_i a r_p y de v_j a r_q. Dentro de dicho intercambio se busca que las nuevas posiciones de v_i y v_j sean las que resultan de menor coste en r_p y r_q, respetando las ventanas de tiempo y el máximo tiempo de operación de cada vehículo.

En cada iteración se analizan todas las combinaciones de pares de clientes y de rutas, representado por el vector $\{v_i,r_q,v_j,r_p\}$. A su vez, para cada vector $\{v_i,r_q,v_j,r_p\}$, se analizan todas las posibles posiciones de los clientes v_i en r_p y de v_j en r_q cuyo coste sea menor. Las posibles posiciones que se analizan del v_i en r_p está limitado por el límite superior de la ventanas de tiempo, reduciéndose así el espacio de búsqueda.

De dicha iteración se obtiene el mejor movimiento que no pertenezca a la lista tabú y se convierte en una nueva solución. El concepto de lista tabú se explicará en el siguiente apartado. El mejor movimiento no tabú se convierte en la nueva solución. Se puede entender a este tipo de movimientos como *mecanismos de intensificación*.

Si el mejor movimiento es además el de menor coste encontrado hasta el momento, se almacena como mejor movimiento absoluto. Como *criterio de aspiración* del algoritmo de BT se aceptan mejores movimientos absolutos aunque sean tabú.

Un *movimiento de inserción* se define como el movimiento admisible en el que se extrae un cliente v_i de una ruta r_q y se inserta en otra ruta r_p, pero, al contrario que en un intercambio, ninguna ciudad de r_p se transfiere a r_q, quedando representado por los vectores $\{v_i,r_q,-1,r_p\}$ o $\{-1,r_q,v_j,r_p\}$. En cada movimiento de inserción también se busca la mejor posición de la ciudad v_i en la ruta r_p, manteniendo los criterios de admisi-

bilidad en tiempos. Se puede afirmar que los movimientos de inserción proporcionan a este algoritmo de BT de *mecanismos de diversificación*.

3.3 Listas tabú, inadmisibilidad y criterio de finalización

En este algoritmo se considera que un *movimiento tabú* cuando al menos una de los clientes implicados (v_i ó v_j) ya ha sido previamente intercambiado o insertado. De esta forma tal vez se limita en exceso la movilidad de los nodos, pero si se considerara la pareja de clientes y no los clientes en sí, el algoritmo puede caer en bucles indeseados y dejar de explorar el espacio de soluciones. Cada tipo de movimiento lleva asociado una lista tabú diferente.

Para este tipo de problemas con flotas heterogéneas de tamaño fijo HVRP, los autores suelen aceptar soluciones no admisibles. En nuestro algoritmo hemos incorporado *inadmisibilidad de soluciones* en los movimientos de intercambio respecto a la capacidad de los vehículos, pero siempre cumpliendo admisibilidad en tiempos. Por ello, la lista tabú de intercambio se divide en dos listas y se almacena la información sobre la admisibilidad de la solución actual.

3.4 Mecanismo principal del algoritmo de BT

El algoritmo se inicia con la generación de una solución inicial admisible mediante la heurística de M&J adaptada y la mejora mediante los operadores 2-opt y 3-opt. En cada iteración del algoritmo de BT se analizan todos los posibles movimientos de intercambio respecto de la solución actual, seleccionando el de menor coste no tabú (o tabú si es el mejor absoluto) admisible o no en capacidad. La lista tabú de intercambio correspondiente es actualizada con las ciudades de dicho movimiento. Se actualiza la nueva solución actual y la mejor absoluta si lo fuese. Este procedimiento de movimientos de intercambio se repite hasta que no haya mejora en la solución durante un cierto número de iteraciones.

Cuando se termina el bucle de iteraciones por intercambio (mecanismo de intensificación), se inicializan las listas tabú de intercambio y se pasa a realizar una iteración con todos los posibles movimientos de inserción sobre la solución actual (mecanismo de diversificación). Si no es posible una inserción admisible o se llega a un número determinado de iteraciones totales, se finaliza el algoritmo. En caso contrario se vuelve a iniciar los movimientos de intercambio con la solución actual de la inserción.

4 Experiencias computacionales

Para la realización de pruebas del algoritmo de BT sobre el problema planteado, se han adaptado 56 casos de problemas FSMTW de Liu and Shen (1999) con 100 clientes, añadiendo un número limitado de vehículos y especificando los parámetros de costes de sostenibilidad. Los problemas se clasifican en 3 grupos: “R”, “C” y “RC” (localización de clientes *Random*, *Clustered* o mixta) y 2 subgrupos: “1” y “2” (hori-

zonte de TW corto o largo), formándose las instancias LS_C1xxA, LS_C2xxA, LS_R1xxA, LS_R2xxA, LSRC1xxA y LS_RC2xxA.

Los 56 casos han sido resueltos con dos funciones objetivos diferentes: minimizar el tiempo total de las rutas o minimizando los costes de sostenibilidad (internos más externos). Se van a comparar los resultados de ambas funciones al aplicar la solución inicial y al aplicar el algoritmo de BT. Se han usado varios valores de los parámetros, mostrando los mejores resultados obtenidos.

Los resultados experimentales para algunas instancias se muestran en la tabla 1. En la misma se muestra para cada instancia de Liu and Shen: la solución inicial de M&J+3opt y la solución del algoritmo de BT, indicando el valor del Tiempo total de las rutas (TR) y del Coste total de las soluciones (CT) al ser resuelto por ambas funciones objetivos.

Tabla 1. Valores de las funciones objetivos en varias instancias (12 de 56)

	M&J+3opt				**BT**			
	FO Tiempo		**FO Coste**		**FO Tiempo**		**FO Coste**	
Archivo	**TR**	**CT**	**TR**	**CT**	**TR**	**CT**	**TR**	**CT**
LS_C101A	268,63	20449,03	248,78	22700,50	236,57	27739,63	253,91	18336,25
LS_C201A	258,44	30655,14	207,55	27255,53	227,64	41401,48	207,55	27255,53
LS_R101A	132,78	13712,53	135,24	13503,85	119,47	14722,95	131,78	12435,00
LS_R201A	77,43	41170,19	78,72	41928,98	62,77	64080,12	87,59	27526,80
LS_RC101A	124,12	17496,09	126,87	18061,52	111,52	19735,38	123,79	16054,06
LS_RC201A	210,67	22545,92	192,69	29216,45	147,47	39489,06	194,87	19926,07

En primer lugar se observa que en varios casos cuando se aplica la heurística de M&J+3opt al minimizar los Tiempos, la solución obtenida es menor en Costes que al minimizar los Costes y viceversa. Esto se justifica porque el método constructivo de inserción de M&J busca buenas rutas al principio pero empeora en las últimas rutas. Se van a visualizar los resultados gráficos de cada instancia agrupada por grupo comparando: cada función objetivo antes (M&H+3opt) y después de aplicar el algoritmo de BT, y cada objetico (TR frente a CT) cuando se aplica cada función objetivo.

En la figura 1, se observa que hay una mejora (16,51%) en los CT entre la solución inicial de M&J+3opt y la solución final dada por la BT cuando se aplica dicha función objetivo. Esta mejora es sustancial cuando el horizonte de ventanas de tiempo es largo (tipo 2 ó VL). El mismo análisis se ha realizado con el objetivo de minimizar TR, obteniendo una mejora del 18,64% en el TR entre la solución inicial y la final. Esta mejora es más notable cuando los clientes están organizados en modo Cluster.

En la figura 2 se comparan los resultados de la BT de cada instancia para el CT con ambas funciones objetivos. De media se obtiene una diferencia del 36,95%, es decir, si se minimiza el problema usando como objetivo los tiempos de las rutas, los costes totales obtenidos serán peores en esa proporción. En el estudio inverso esa cantidad es

de media un 15,73%, indicando que el objetivo de reducir tiempos empeora bastante los costes de sostenibilidad de las rutas obtenidas, pero no tanto al revés.

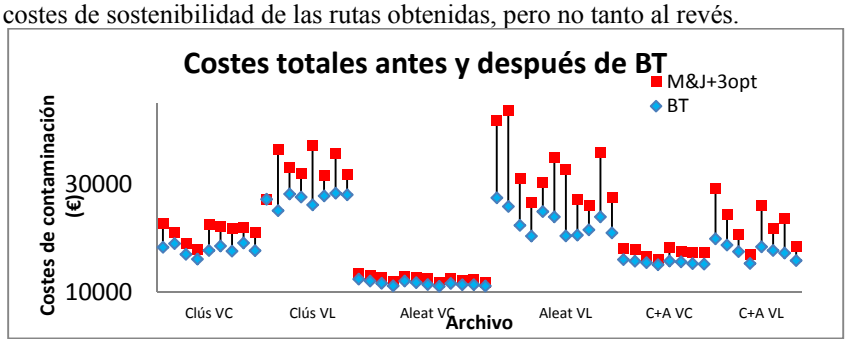


Fig. 1. Reducción del coste total (CT) entre la solución inicial y la final

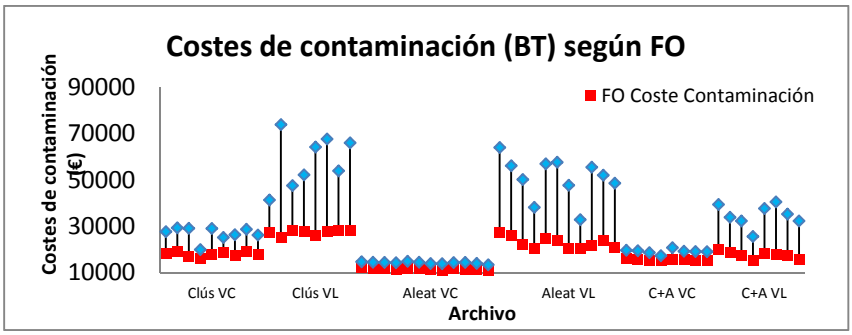


Fig. 2. Costes totales (CT) usando ambas funciones objetivos en BT

5 Conclusiones

En este artículo se ha presentado el problema VRP, incluyendo aspectos realistas de las empresas de distribución, como flotas heterogéneas y ventanas de tiempo, y comparando los criterios tradicionales asociados a distancias y tiempos con nuevos criterios de sostenibilidad energética y medioambiental. Para su resolución se ha adaptado un algoritmo de Búsqueda Tabú que usa como solución inicial una heurística clásica adaptada, Mole & Jameson, con una mejora con operadores 2-opt y 3-opt. Los resultados demuestran que el algoritmo lleva a soluciones mejores que la heurística y también que las soluciones obtenidas mediante los criterios basados en tiempos no consideran los criterios de sostenibilidad, pero que al revés existe una clara correlación.

Para el futuro se deja abierta la posibilidad de mejorar el algoritmo de BT mediante el uso de otra heurística inicial basada en *Giant Tour*, y con nuevos movimientos de inserción basados en *GENIUS*, que funcionan bien en HFVRP.

AGRADECIMIENTOS

Este trabajo ha sido financiado por la Consejería de Economía, Innovación y Ciencia de la Junta de Andalucía a través del proyecto de investigación P10-TEP-6332.

6 Bibliografía

1. Bektas, T. and Laporte, G. (2011). The Pollution-Routing Problem. Transportation Research Part B, doi:10.1016/j.trb.2011.02.004.
2. Brandão, J. (2009): A deterministic tabu search algorithm for the fleet size and mix vehicle routing problem. Eur.J. Oper. Res. 195, 716–728
3. Brandão, J. (2011): A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem. Computers & Operations Research 38, 140-151.
4. Eguia, I., Racero, J., Guerrero F. (2012): Evaluating Vehicle Routing Problems with Sustainability Costs. ICORES (2012).
5. Gendreau M., Laporte G., Musaraganyi C. and Taillard E.D. (1999) A tabu search heuristic for the heterogeneous fleet vehicle routing problem. Computers & Operations Research, 26(12):1153-1173, 1999
6. Gheysens F.G., Golden B.L. and Assad A.A. (1984) A comparison of techniques for solving the fleet size and mix vehicle routing problem. *OR Spectrum*, 6(4):207-216.
7. Golden B., Assad A., Levy L. and Gheysens F.G. (1984), The Fleet Size and Mix Vehicle Routing Problem, Computers and Operation Research, Vol 11, pp. 49-66.
8. Imran, A., Salhi, S., Wassan, N.A. (2009): A variable neighborhood-based heuristic for the heterogeneous fleet vehicle routing problem. Eur. J. Oper. Res. 197, 509–518
9. INFRAS/IWW. (2004). External costs of transport: update study, INFRAS, Zurich.
10. Kara, I., Kara, B.Y., Yetis, M.K., (2007). Energy minimizing vehicle routing problem. In: Dress, A., Xu, Y., Zhu, B. (Eds.), Combinatorial Optimization and Applications, Lecture Notes in Computer Science, vol. 4616. Springer, Berlin/Heidelberg, pp. 62–71.
11. Lee, Y., Kim, J., Kang, K., Kim, K. (2008): A heuristic for vehicle fleet mix problem using tabu search and set partitioning. J. Oper. Res. Soc.. 59, 833–841
12. Li F., Golden B.L. and Wasil E.A. (2007). A record-to-record travel algorithm for solving the heterogeneous fleet vehicle routing problem. Computers & Operations Research, 34(9): 2734-2742.
13. Liu F.H. and Shen S.Y. (1999). The fleet size and mix vehicle routing problem with time windows. Journal of the Operational Research Society, 50(7):721-732, 1999
14. Liu, S., Huang, W., Ma, H. (2009): An effective genetic algorithm for the fleet size and mix vehicle routing problems. Transp. Res., Part B, Methodol. 45, 434–445
15. Mole, R.H., Jameson, S.R. (1976): A sequential route-building algorithm employing a generalised saving criterion. Operational Research Quarterly, 27: 503-511
16. Osman I.H. and Salhi S. (1996) Local search strategies for the vehicle fleet mix problem. In V.J. Rayward-Smith, I.H. Osman, C.R. Reeves, and G.D. Smith, editors, Modern Heuristic Search Methods, pages 131-153. Wiley: Chichester.
17. Prins, C. (2002): Efficient heuristics for the heterogeneous fleet multitrip vrp with application to a large-scale real case. J. Math. Model. Algorithms **1**, 135–150
18. Prins, C. (2009): A GRASP evolutionary local search hybrid for the Vehicle Routing Problem. In: Bio-inspired Algorithms for the Vehicle Routing Problem, Studies in Computational Intelligence, vol. 161, pp. 35–53. Springer, Berlin (2009a)
19. Taillard E.D. (1999) A heuristic column generation method for the heterogeneous fleet vrp. *RAIRO* Recherche Opérationnelle, 33(1):1-14, 1999.
20. Tarantilis C.D., Kiranoudis C.T. and Vassiliadis V.S. (2003) A list based threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. Journal of the Operational Research Society, 54(1):65-71.
21. Ubeda, S., F. J. Arcelus, and J. Faulin. (2011) "Green logistics at Eroski: A case study." Int. J. Production Economics 131, 2011: 44 –51.

Local search method for a real case assembly problem with single machines on production stage

Sara Hatami[a], Carlos Andrés Romano[b]

[a] Departamento de Estadística e Investigación Operativa, Facultad de Ciencias Matemáticas, Universitat de València, Spain
[b] Departamento de Organización de Empresas, Universidad Politécnica de València,Camino de Vera s/n, 46021, València, Spain

Abstract. This paper presents initial results of a research study in a real assembly problem with single machines in the first (production) stage with the objective to minimize the makespan. The problem arises in a manufacturing facility devoted with the production of merchandising and house ware items where parts must be manufactured at the first stage and then assembled at the second stage. The paper describes some heuristics and a local search based heuristic developed to solve the real problem.

1 Introduction

Generally assembly problems are consisted of production and assembly stages. Depend on production strategy, different production systems are considered for the first stage. In this paper, *m* different single machines are considered as production stage and also a single assembly machine assembled jobs that are produced in the first stage. There are *n* jobs to be processed. In the first stage, there are *m* different and independent single machines that each of them is considered to process some define jobs. In assembly stage, a single assembly machine produces *t* different products by assembling jobs according to an assembly program. Assembly of product *h* at the second stage can start only after all jobs of it are completed at the first stage. Besides, each job needs a processing time only on one single machine, where these processing times are known in advance, non-negative and deterministic. Each machine can perform only one operation at a time. No breakdowns exist on machines and they are always available. All jobs are ready to start processing at time 0. No preemption is allowed in processing of jobs.

The consider problem is almost a case study of a mold/tooling producer that produces some products like, belt ice bucket, light signs, cocktail shakers, etc.

In this paper, in spite of processing time for jobs on single production and assembly machines, setup times are also considered, where is very important in practice. Setup times are considered as preparation time for machine, putting in position work-in-process material time, cleaning up time. Setup times are divided into two groups, sequence-independent setup times (SIST), where the setup time value only depends on the new job that is to be processed and sequence-dependent setup times (SDST),

where the amount of the setup time depends on both the new job and on the job previously processed on the machine [1].

There are various studies on assembly problems in literature, most of them are considered flowshop systems in production stage ([2], [3], etc.), Javadian et al., (2009) considered parallel machines in production stage.

Single machine scheduling has attained most attention in scheduling literature, where only a single machine occupies to processing jobs ([5], [6], etc.).

The rest of the paper is organized as follow: section 2 describes the considered problem, in section 3, we present a simple constructive heuristic for the problem and in section 4 a local search is suggested for considered problem and finally in section 5 presented conclusion and future work suggestion.

2 Problem description

The considered production model in this paper, is consisted of two stages: production and assembly. In the first stage that is a production stage, *n* independent jobs are fabricated on *m* independent feeding machines where each produces its own component associated with a job. There are *t* products that is consisted of a set of identical jobs. Each product at least is consisted of two jobs and each job only belongs to only one product. Job J_j consists of only one task, for $j = 1, \dots, n$, therefore each job can process only on one machine. The components of the products are assembled on single assembly machine at the second stage according to an assembly program. In the first stage, there are *m* machines assigned to produce maximum m different types of component, which is required for each product. Each product has a defined assembly program; in other words, each product consists of some defined jobs.

N_h and J_j are used, respectively, to represent product h assembly program and the jobs that belong to the product h assembly program, N_h: $\{J_j\}$, $j \in N_h$. Each product h has $|N_h|$ jobs and job j is needed for the assembly of one product, where $|N_h| \leq m$ for all h. Therefore, $\sum_{h=1}^{t} |N_h| = n$. Any assembly operation cannot start until fabrication of all the related component jobs has been finished. In fact the production model is consisted of m production centers where each consisted of only one single machine. Processing times of job J_j, $j = 1, .., n$, constants as p_{ij}, $i = 1, .., m$, on machine *i* of the first stage and product s assembly time as p_s, $s = 1, .., t$ are given. All processing times are known, finite and fixed positive number.

The performance criterion is minimizing the makespan (C_{max}). Note that the completion time at the second stage is influenced by the completion time at the first stage due to the *m* single machines.

3 Heuristic

An example is presented to explain better the proposed simple constructive heuristic. The example consisted of seven jobs seven jobs, three machines and three products. Assembly programs for products are: $N_1 = \{1, 3, 6\}$, $N_2 = \{2, 5\}$, $N_3 = \{4, 7\}$.

Table 1 presents job processing times on production machines and product assembly time on assembly machine. Table 2 presents setup matrixes for production and assembly machines.

To start processing the first job on production machines and assembling the first product on assembly machine, an initial setup is necessary, hence a dummy job and product are considered as J_0 and T_0 to comply this consumption.

Table 1. Job processing and product assembly times for the example

	Job						
	J_1	J_2	J_3	J_4	J_5	J_6	J_7
M_1	4	3	-	-	-	-	-
M_2	-	-	-	5	-	7	-
M_3	-	-	3	-	4	-	5
	Product 1			Product 2		Product 3	
M_A	5			4		6	

Table 2. Production and assembly machines setup time matrix

	Machine 1								Machine 2							
Jobs	J_1	J_2	J_3	J_4	J_5	J_6	J_7	Jobs	J_1	J_2	J_3	J_4	J_5	J_6	J_7
J_0	3	3	5	6	7	1	3	J_0	4	8	1	6	4	5	3
J_1	0	3	1	9	8	8	4	J_4	4	4	2	0	1	3	6
J_2	2	0	5	1	3	1	5	J_6	3	2	5	1	4	0	7
	Machine 3							Products	Assembly machine						
	J_1	J_2	J_3	J_4	J_5	J_6	J_7		T_1		T_2		T_3		
J_0	4	3	7	2	1	5	3	T_0	4		5		1		
J_3	3	8	0	2	4	8	4	T_1	0		3		6		
J_5	2	3	7	3	0	4	6	T_2	1		0		5		
J_7	1	4	5	6	1	1	0	T_3	2		5		0		

In heuristic explanation, sequence of products is presented by π, where in it, the product with minimum summation of processing and setup times on assembly machine occupies the first position. The next product that results minimum makespan by lying after the first job, will placed in second position of π. Rest of the unscheduled products follow up the same procedure until all products enter to π. Jobs of the product are started to assign to production machines according to product order of π.

Each product has a start time for assembling on assembly stage and this time is named earliest start assemble time of product s, E_s. Where E_s is equal to maximum completion time of jobs that belongs to product s on all production machines, $ES_s = Max_{i=1}^{m} C_{ij}$. After assigning jobs of products following π, second product sequence π^*, is made by sorting E_s for all products s.

Here, we considered mentioned example to explain better the heuristic. The makespans of the products 1 to 3, by considering them, separately on assembly machine are 9, 9 and 7 respectively by considering setup and processing times.

Hence product 3 is the first product that enters to set SP, which is used to hold scheduled products for π, $SP = \{3\}$. Makespan of the rest of the products are exam-

ined to determine the second product in π, hence the makespans of sequences $C_{max}\{3,1\} = 14$, $C_{max}\{3,2\} = 16$. Then SP is updated to $\{3,1\}$. The procedure is repeated similarly for all rest of the unscheduled products until all products entered to π. Finally $\pi = \{3,1,2\}$. Jobs that belongs to defined products are assigned to production machines following to products order of π. Completion time of job j on machine i, C_{ij}, for product 1 are: $C_{11} = 7$, $C_{33} = 16$, $C_{26} = 21$, for product 2: $C_{12} = 13$, $C_{35} = 24$, and for product 3: $C_{24} = 11$, $C_{37} = 8$.

Earliest start assembling times for products 1 to 3 are 21, 24 and 11, respectively. Therefore, second product sequence, π^*, that makes from earliest start assembling times on assembly machine is equal to $\{3,1,2\}$, with makespan of 33. The Gantt chart of the heuristic of the example is presented in figure 1.

Fig. 3. Gantt chart of the example of proposed simple heuristic

4 Local Search

Local search is a meta-heuristic method for solving optimization problems, where it moves from solution to solution in the space of candidate solutions (the search space) by applying neighborhood structures, until a solution deemed optimal is found or a time bound is elapsed. There are enormous studies about scheduling that uses local search as a tool to solve problem ([7], [8], etc.).

Two local searches are proposed for the considered problem. The first one (*LS1*) tries to improve the objective function by inter-exchange of jobs for each machine, where the second one (*LS2*) tries to find a better product sequence by remove one product from π and reinsert it in all possible position of it to improve objective function. Both of the local search are explained in figures 2 and 3.

- Proposed heuristic result as initial solution
- $h = 1$

while ($h <= t$)
- Remove product h and reinsert it in all possible positions of π

if (*objective function improve*)
- Update π

else
- $h = h + 1$

end if

end while

Fig. 2. Job local search (*LS1*) for proposed problem

```
- Proposed heuristic result as initial solution
- i = 1
  while (i <= m)
        - Inter-exchange of jobs on machine i
        if (objective function improve)
- Update job sequence of machine i
  else
  - i = i + 1
    end if
 end while
```

Fig. 3. Product local search (LS2) for proposed problem

5 Conclusion

In this paper we present a real case study assembly problem that consists of two stages. The first stage includes *m* single machines that are responsible to produce *n* defined jobs and also a single assembly machine constructs the second stage.

Jobs are assembled through an assembly program to constitute *t* products by single assembly machine. A simple heuristic method is presented to solve the problem and also two local search as a meta-heuristic are proposed to find better solutions and try to improve the proposed simple heuristic. Consider parallel machine in place of single machines for first stage and consideration of a transport stage between the two existed stages can be a suggestion for future work.

References

1. Yang, W.H., Liao, C.J., Survey of scheduling research involving setup times. International Journal of Systems Science, 30 (2), (1999) 143–155.
2. Koulamas, C., Kyparisis, G.J., The three stage assembly flowshop scheduling problem. Computers & Operations Research 28 (7), (2001) 689-704.
3. Al-Anzi, F., Allahverdi, A., A hybrid tabu search heuristic for the two-stage assembly scheduling problem. The International Journal of Operational Research, 3 (2), (2006) 109-119.
4. Javadian, N., Mozdgir, A., Ghazani Kouhi, E., Davalloo Qajar, M.R., Shiraqai, M.E., Solving Assembly Flowshop Scheduling Problem with Parallel Machines Using Variable Neighborhood Search. 2009. CIE 2009. International Conference on Computers & Industrial Engineering, (2009) 102–107.

5. Phojanamongkolkij, N., Grabenstetter, D., Ghrayeb, O., Scheduling jobs to improve weighted on-time performance of a single machine. Journal of Manufacturing Systems, 22 (2), (2003) 148-156.
6. Kethley, R.B, Alidaee, B., Single machine scheduling to minimize total weighted late work: a comparison of scheduling rules and search algorithms. Computers & Industrial Engineering 43(3), (2002) 509-528.
7. Glass, C.A., Potts, C.N., Shade, P., Unrelated parallel machine scheduling using local search. Mathematical and Computer Modelling, 20 (2), (1994) 41-52.
8. Pan, Q.K., Ruiz, R., Local search methods for the flowshop scheduling problem with flowtime minimization. European Journal of Operational Research, 222 (1), (2012) 31-43.

Variable Neighborhood Search: Application to a Routing Problem with Flexibility in Delivery Dates

Joaquín Pacheco (Universidad de Burgos), Irma García (Universidad Autónoma de Coahuila, Mexico), Ada Álvarez (Universidad Autónoma de Nuevo León, Mexico)

jpacheco@ubu.es, idgarcia@cima.uadec.mx, ada.alvarezs@uanl.mx

Abstract. In this work a real problem of designing the routes over a planning period with flexibility in the dates of delivery is addressed. Specifically a method based on the strategy variable neighborhood search (VNS) is designed for this problem. The VNS-based method takes some ideas from *routing-first cluster-second* strategies for routing problems. In addition, it has two important features: it uses a memory-based shaking procedure and it allows, in some conditions, move from the current solution to a worse solution (Skewed VNS). This procedure performs better than its basic version (not memory, not skewed). Computational experiments in real-data-based instances show that our VNS obtains better results than previous methods for this problem. The method is simple, with easy implementation and can be adapted to other routing problems.
Keywords: Logistics, Routing, Metaheuristics, Variable neighborhood search, Memory

1 Introduction

The vehicle routing problem (VRP) is undoubtedly one of the most studied combinatorial optimization problems in the literature. It can be defined as the problem of designing routes for delivering vehicles of given capacities, to supply a set of customers with known locations and demands from a single depot. Routes for the vehicles are designed to minimize some objective such as the total distance traveled.

The problem addressed in this paper is motivated by a real problem in a bakery company in Northern Spain and it was recently introduced in [1]. The company has to meet required orders, known in advance, for a set of distribution centers over a week using a fleet of homogeneous vehicles. Each order includes a number of required pallets and a specif-

ic delivery day (deadline). Each trip finishes on the same day it started and the vehicles return to the depot at the end of the route.

The total travel costs over the week might be reduced by combining orders, since it is possible to take advantage of certain flexibility in the delivery date. Nevertheless, as the company is dealing with perishable products, every order should not be delivered more than one day earlier than the original deadline. This policy will result in stock at the distributors, but it is controlled and acceptable for the company. Although initially this problem has been proposed by a bakery company it is clear that the same problem or similar can be found in other areas.

In this work a method based on the metaheuristic VNS for this problem is proposed. VNS is a local search based metaheuristic first proposed in [2].

The work is organized as follows. In the next section the notation and the description of the problem are presented. In section 3 we describe our VNS method for this problem. In section 4 we present the results of different computational experiments. Finally, section 5 concludes with our main findings.

2 Problem description and notation

In this paper we address a problem that arises in a bakery company, which has to deliver orders placed by geographically dispersed distribution centers in northern Spain. Orders for the week are known in advance, commonly on Friday of the previous week. The problem has the following characteristics.

Regarding orders:

- Each order includes a number of required pallets which need to be delivery by a specific day (deadline)
- Each order must be served by a single vehicle (due to labor constraints at distribution centers).

- Each order should be delivered within the original deadline and no more than a fixed number of days before the deadline, but always within the same week. This means that if the deadline of an order is Monday, it cannot be delivered before.

Regarding routes:

- To meet the orders a fleet of homogeneous vehicles with known capacity is available.
- Each trip finishes on the same day it started and the vehicles return to the depot at the end of the route
- Transportation costs are proportional to the distance traveled.

The problem consists of designing routes to deliver the orders placed for the week with the objective of minimizing the total transportation cost, that is, the total traveled distance. Flexibility in the day of delivery plays an important role in the solution to the problem. Managing this flexibility it is intended to create savings in transportation costs, and is achieved as is shown in [1].

To formalize the statement of the problem let us introduce the following notation:

H = Number of days of the planning horizon;
n = Number of orders;
V = $\{0, 1, \ldots, n\}$ set of locations; where 0 corresponds to the depot and i corresponds to the location of i-th order, $\forall\ i = 1,...,n$. (We prefer to identify every location by its corresponding order to simplify further descriptions and notations)
d_{ij} = Distance between location i and j, $\forall\ i, j \in V$;
q_i = Number of pallets of order i, $\forall\ i = 1,...,n$;
e_i = Deadline of order i, $\forall\ i = 1,...,n$; $e_i \in \{1,..,H\}$
Q = Capacity of vehicles (in pallets)
K = Size of the fleet
g = Maximum number of days that an order could be served before its deadline.

As it has been have mentioned before, in real instances the parameter H is taken equal to 5, where 1 corresponds to Monday and 5 to Friday.

3 The VNS method

In this section, the VNS method designed for this problem is described. In all methods described in this work all orders and all routes over the planning horizon are considered jointly, that is, not day by day. Thus, our algorithm treats the problem as a CVRP with two additional constraints: 1) the orders belonging to the same route should have at least a common feasible date; 2) the number of routes assigned to each day must not exceed the fleet size. Initially, every route is assigned to the minimum deadline of the orders supplied by this route. Obviously, this constraint replaces the constraint regarding to the maximum number of routes over the planning horizon.

Therefore, in all the steps of the different procedures (insertions, movements, exchanges, etc.) that compose the VNS method these two constraints are taken into account, in addition to the capacity of the vehicles. Let us denote by $f(S)$ the value of the objective function of a solution S, and by S_0 the initial solution. Our VNS method is described in Figure 1.

```
Method VNS (max_iter, rmax, S0; var S)
1. Set S = S0 and niter = 0
Repeat
  2. niter = niter+1
  3. r = 0
  Repeat
  4. r = r+1
  5. Execute Shaking(r, S, S1)
  6. Execute Local_Search (S1)
  until (r = rmax) or (f(S1) < f(S))
  If f(S1) < f(S) then S = S1
until   does not improve for max_iter iterations
```

Fig. 1. VNS method

The initial solution S_0 is obtained by using the well-known method of Fisher and Jaikumar [3], which is properly adapted to this problem, i.e., bearing in mind the two additional constraints mentioned above. Next the Shaking and Local_Search procedures are described.

3.1 Shaking procedure

This procedure generates a solution belonging to a neighborhood of the current solution *S*. It depends on the parameter *r*, so that a low value of *r* corresponds to a small neighborhood, that is, it generates solutions close to or similar to *S*, while a larger value of r corresponds to a big neighborhood. Figure 2 shows pseudocode for this procedure.

```
Procedure Shaking(r, S, var S1)

- From a solution S, form a "big chain" BC,
writing in a sequence all the routes that are in
S,  removing the depot 0.
- Randomly choose r points in the big chain BC
and divide it in r+1 parts or sub-chains. These
sub-chains are randomly permuted. Denote BC' the
new obtained big chain.
- From BC' construct a feasible solution S1 by
splitting the big chain into sub-chains corre-
sponding to  feasible
```

Fig. 2. Shaking procedure

First, a big chain is built from *S*. For example, if *S* is composed of routes 0 – 1 – 4 – 2 – 3 – 0 and 0 – 5 – 6 – 8 – 7 – 0, then, the corresponding big chain *BC* is 1 – 4 – 2 – 3 –5 – 6 – 8 – 7. Secondly, BC is divided randomly in *r*+1 pieces or sub-chains; for example, if *r* = 2, let us suppose the following 3 sub-chains: 1 – 4 – 2, 3 – 5 and 6 – 8 – 7. Third, the sub-chains are randomly permuted and re-linked, for exam-

ple: 3 – 5, 6 – 8 – 7 and 1 – 4 – 2. The new big chain BC' is: 3 – 5 – 6 – 8 – 7 – 1 – 4 – 2.

Finally, the big chain *BC'* is divided into sub-chains corresponding to feasible routes. To do that, let us define bi the location in position i in *BC'* (that is, $b_1 = 3$, $b_2 = 5$ and so on). For simplicity, define $b_0 = 0$. An auxiliary directed graph $G = (N, A)$ is created where $N = \{0, 1, \ldots, n\}$ is the set of nodes, and $A = \{(i, j) \mid i < j;\ i, j \in N\}$ is the set of arcs. In this case $n = 8$. The costs associated to the arcs in this graph are defined in the following way $\forall\ (i, j) \in A$:

u_{ij} = cost of the route that starts in 0, visits locations $b_{i+1}, b_{i+2}, \ldots, b_j$, and returns to 0, if this route is feasible; and ∞ otherwise.

Thus, in order to find an optimal partition of the (directed) big chain into feasible routes, (that is, routes with minimum total distance), a shortest path problema is solved from node 0 to n in graph G with arc costs u_{ij}. For example, if the solution of the SPP corresponding to previous *BC'* is 0 – 4 – 6 – 8, then the obtained routes are the following:

$$0 – 3 – 5 – 6 – 8 – 0,\ \ 0 – 7 – 1 – 0 \ \text{ and } \ 0 – 4 – 2 – 0$$

3.2 Procedure Local_Search

Solutions obtained by shaking procedure are improved using the procedure Local_Search. This is an iterative method that in every iteration moves from the current solution to a better one from its neighborhood. The procedure ends when there are no better solutions in the neighborhood. The neighborhood of a solution S, i.e., the set $N(S)$, is obtained by simple changes in S. Four types of changes or movements have been considered in our implementation:

- Or exchanges [4]
- Generalized Or exchanges
- CROSS [5]
- Insertion

The first two mechanisms (Or exchanges) are performed within a single route while CROSS and Insertion are performed on two different routes. These moves are illustrated in Figure 3, where the original routes are assumed to visit the locations in alphabetical order (i.e., A to H and U to Z).

As depicted in Figure 3, the Or move consists of exchanging two consecutive segments in a single route. In particular, segment B→C is exchanged with segment D→E→F, transforming the original route to the new route ADEFBCGH. The generalized version of this move allows the exchange of nonconsecutive segments. Figure 3 shows the exchange of the B→C and F→G segments which results in the new route AFGDEBCH. In the CROSS exchange, a segment in one route is exchanged with a segment in another route. The segments do not have to cover the same number of locations. However, we do enforce feasibility throughout the search and therefore these exchanges must be such that the capacity restrictions are satisfied by the resulting routes. The CROSS exchange illustrated in Figure 3 transforms the original routes into two new routes AVWXEF and UBCDYZ. We also execute insertions, where a segment is moved from one route to the other. The exchanges are also limited to those that will result in feasible routes. In the insertion depicted in Figure 3, the V→W→X segment is inserted between A and B to create two new routes AVWXBC and UYZ.

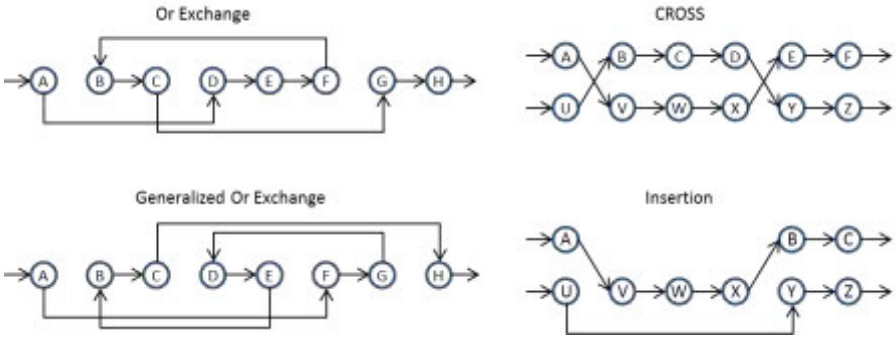


Fig. 3. Illustration of move mechanisms used within **Local_Search** procedure

The pseudocode for this procedure is shown in Figure 4.

```
Procedure Local_Search (var S)

Repeat
1. Set previous_value = f(S)
2. Search f(S*) = min { f(S') | S' ∈ N(S) }
3. If f(S*) < f(S) then S = S*
until f(S*) ≥ previous_value
```

Fig. 4. Local_Search Procedure

4 Computational Results

In order to check the quality of our proposed SVNSM method a set of computational tests were performed. In this section these computational experiences are described and their corresponding results are shown. For these experiments real-data-based instances used in [1].

In this sub-section, our VNS method is compared with the following methods for PVRP: The Tabu Search method proposed in [6] (CGL), Scatter Search method proposed in [7] (ALP) and the VNS method proposed in [8] (HDH). Also our VNS method is compared with the GRASP-Path Relinking (GRASP-PR) based method proposed in [1] for our problem. The CGL algorithm was run in a Dell Precision T7400 with four processors at 3.0 Ghz CPU (a single processor was used to run the tests); the ALP and GRASP-PR algorithms was run on a Corel 8400 PC at 2.66 GHz CPU; the HDH algorithm was run on a 64 bits dual core PC, with 3.2 GHz CPU. Table 2 shows the results obtained for every method, specifically the total distance (T.Dist) and the total computational time in seconds (Time).

Instance		CGL		ALP		HDH		GRASP+PR		VNS	
n	#	*T.Dist*	*Time*	*T.Dist*	*Time*	*T.Dist*	*Time*	*T.Dist*	*Time*	*T.Dist*	*Time*
41	1	**6425**	*36*	**6425**	*33*	**6425**	*80*	**6425**	*29*	**6425**	9
41	2	**6173**	*37*	6315	*31*	6177	*81*	**6173**	*28*	**6173**	8
41	3	**6356**	*41*	**6356**	*33*	**6356**	*82*	**6356**	*29*	**6356**	8
41	4	**6903**	*35*	**6903**	*35*	6922	*84*	**6903**	*26*	**6903**	7
41	5	**6833**	*30*	**6833**	*31*	**6833**	*87*	**6833**	*23*	**6833**	6
62	1	9298	*58*	9318	*98*	9336	*166*	**9297**	*81*	9309	52
62	2	**8765**	*62*	8824	*92*	8785	*171*	**8765**	*81*	8809	58
62	3	8833	*58*	8886	*95*	8849	*177*	**8814**	*71*	8844	50

Instance		CGL		ALP		HDH		GRASP+PR		VNS	
n	#	*T.Dist*	*Time*	*T.Dist*	*Time*	*T.Dist*	*Time*	*T.Dist*	*Time*	*T.Dist*	*Time*
62	4	**9089**	*41*	**9089**	*77*	**9089**	*174*	**9089**	*72*	**9089**	33
62	5	**8838**	*68*	8852	*117*	8878	*163*	**8838**	*96*	8915	76
82	1	**10779**	*89*	10913	*206*	10827	*276*	10789	*163*	10780	215
82	2	10460	*93*	10481	*217*	10501	*283*	10431	*203*	**10409**	193
82	3	10148	*98*	**10076**	*210*	10139	*287*	**10076**	*199*	10101	188
82	4	10555	*72*	**10550**	*234*	10620	*289*	**10550**	*177*	**10550**	135
82	5	10611	*91*	10593	*217*	10618	*307*	10664	*209*	**10604**	161
102	1	13313	*112*	13285	*377*	13192	*434*	13204	*349*	**13124**	360
102	2	13888	*92*	13835	*389*	13894	*465*	13828	*325*	**13801**	325
102	3	11921	*117*	11957	*468*	11981	*455*	**11899**	*389*	11914	448
102	4	12515	*126*	12424	*454*	12483	*438*	12355	*402*	**12324**	315
102	5	12591	*127*	12574	*519*	12640	*439*	12519	*416*	**12490**	544

Table 1. Comparison between our SVNSM method and previous methods

As can be observed in Table 2, VNS obtains the best results in 14 of 20 instances; CGL obtains 9 best results, ALP 7, HDH 4 and GRASP+PR 13. On the other hand, our VNS method, GRASP+PR, ALP and HDH employ similar amount of computational time while CGL spends less computational time in larger instances.

5 Conclusions

In this work a VNS-based method with memory is proposed for a routing problem with real applications which has been recently introduced. The problem consists of designing the daily routes over the week satisfying a set of orders. The objective is to minimize the transportation cost (total distance traveled). In order to save costs, flexibility is allowed in the delivery date. The problem can be considered as a generalization of the classic CVRP as well as a particular case of the PVRP. The proposed method outperforms other existing methods in the literature.

The proposed VNS method employs ideas based on strategies “Route-First Cluster-Second” that have been successfully adapted in the context of genetic algorithms. From our point of view the proposed method is simple and can be easily programmed. It uses known procedures for routing and networks problems (intra-routes and between routes movements, shortest path methods, etc.).

References

1. Pacheco J., A. Alvarez, I. García and F. Angel-Bello (2012) “Optimizing vehicle routes in a bakery company allowing flexibility in delivery dates“. *Journal of the Operational Research Society* 63, 5, pp. 569–581.
2. Mladenović N. (1995) A variable neighborhood algorithm - a new metaheuristic for combinatorial optimization. In *Abstracts of Papers Presented at Optimization Days,* pp. 112.
3. Fisher M. L. and R. Jaikumar (1981) "A Generalized Assignment Heuristics for Vehicle Routing Problem". *Networks* 11, 2, pp. 109–124.
4. Or I. (1976). Traveling Salesman Type Combinatorial Problems and Their Relations to the Logistics of Blood Banking. PhD Thesis, *Northwestern University*, 1976.
5. Taillard E., P. Badeau, M. Gendreau, F. Guertain and J. Y. Potvin (1997) "A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows" *Transportation Science* 31, 2, pp. 170-186.
6. Cordeau J-F.., M. Gendreau, and G. Laporte (1997) “A Tabu Search Heuristic for Periodic and Multidepot Vehicle Routing Problems”. *Networks*, 30, 2, pp. 105-119.
7. Alegre J., M. Laguna and J. Pacheco (2007). “Optimizing the periodic pick-up of raw materials for a manufacturer of auto parts”. *European Journal of Operational Research* 179, 3, pp. 736–746.
8. Hemmelmayr V. C., K. F. Doerner and R. F. Hartl (2009) “A variable neighbourhood search heuristic for periodic routing problems”. *European Journal of Operational Research* 195, 3, pp. 791–802.

A Multi Ant Colony Optimization algorithm for a Mixed Car Assembly Line

Raúl Pulido, Álvaro García-Sánchez, Francisco Javier Diego, Carlos Andrés-Romano

Technical University of Madrid. Research Group in Industrial Engineering and Logistics, Madrid, Spain, Polytechnic University of Valencia, Valencia, Spain

{raul.pulido,alvaro.garcia,javier.diego}@upm.es
candres@omp.upv.es

Abstract. This paper presents an ant colony optimization algorithm to sequence the mixed assembly lines considering the inventory and the replenishment of components. This is a NP-problem that cannot be solved to optimality by exact methods when the size of the problem growth. Groups of specialized ants are implemented to solve the different parts of the problem. This is intended to differentiate each part of the problem. Different types of pheromone structures are created to identify good car sequences, and good routes for the replenishment of components vehicle. The contribution of this paper is the collaborative approach of the ACO for the mixed assembly line and the replenishment of components and the jointly solution of the problem.

Keywords: Ant Colony Optimization, Vehicle Routing Problem, Car Sequencing

1 Introduction

There is a growing interest on solving multi-objective problems, this has led the researcher to combine algorithm and create an extension of the classical algorithms to achieve their objective [1]. This paper presents an ACO approach to deal with the sequencing of mid-size problems in a mixed car assembly line considering the inventory and the replenishment. Unfortunately, only small in-stances can be solved up to optimality; thus, the computational paradigm, Ant Colony Optimization, introduced by Dorigo et al. [2] was used.

Car sequencing: The mathematical model of the operation of the car assembly line was based upon the approach of sequence rules, where the assembly line can handle a pre-established production ratio for each option. In case that the task cannot be finished a utility worked is called to allow the task to be finished. “Instead of a detailed scheduling of work content, car sequencing considers and controls the succession of work intensive product options (e.g. sunroof, air conditioning) in order to avoid work overload’’ [3]. Giard [4] presents a model which offers the option of hiring utility workers to allow the violation of spacing constraints. This problem has

been tackled with greedy, local search, Ant Colony Optimization, Genetic Algorithm, and so on. Gottlieb [5] constructs the solution using the difference of two cars to decide how to construct the solution.

Solnon [6] uses two pheromone structure, the first to identifying good car sequences, and the second one to identify critical cars. This paper is the continuation of our previous work [7]. Where small instance of this problem is solve using a MIP. In that paper a compound approach is used to minimize the cost of replenishment of components keeping and a proper inventory. The formulation of the MIP is the same, and it will not be detailed here as the main objective of the problem is the heuristic formulation.

Inventory Routing Problem The replenishment and inventory control is a classical problem defined as the Inventory Routing Problem (IRP) is defined as "a starting point for studying the integration of different components of the logistics value chain, i.e. inventory management and transportation'' [8]. Traditionally; production and transportation have been dealt with separately. It is expected that savings could be obtained by coordinating production and transportation. Gendreau [9] categorizes the different heuristic used to solve this problem, which is ACO, genetic algorithms, greedy, Simulated Annealing, Tabu Search, and Variable Neighborhood. Bell [10] uses multiple ant colonies to identify typically used for each vehicle.

Ant colony optimization has been applied to solve large and difficult problems. Recently, ACO has been used for the solution of constraints satisfaction problems like the Car Sequencing,

We introduce different pheromone structures for the different parts of the problem to help the ants to construct the solution.

The main motivation behind this paper is to answer the following question, would it be possible to use the ACO characteristic to solve car assembly lines problem with a compound approach, without integrating problem-dependent heuristics?

2 Problem Statement

The process to assemble a car requires to pass through several workstations. Some configurations of cars require more than a takt time at the workstation to assemble the component. Other configurations require less time. Then it is necessary to find sequences that keep the ratio in each workstation. Each time that the production ratio is above this number is considered as a rule violation. Additionally, all the components required in the workstation have to be replenished from the warehouse.

Each model has a set of characteristics, such as types of wheels and tires, radio, sunroof, car seat, etc. In every workstation, a kit of components is installed; these components can have different trim levels. The combination of components and trims gives us the characteristics. To make it clearer; in the case of radio, High trim could mean radio/MP3, and Low trim could mean radio/CD (see table 1).

We define that the car is assembled when passed through all workstations which install the different components. The present problem is an extension to the problem

proposed for the ROADEF challenge in 2005, where we also consider the inventory level, and the replenishment of these components.

Table 1. Example of the classes with their components

	Radio	GPS	Sunroof	…	Comp n
Class A	Low	Low	Low	…	Low
Class B	Low	High	High	…	High
Class N	High	High	High	…	High

2.1 Definition

The problem is defined by a 8-tuple (C, O, S, h, n, r, V, T DIS) such that.

- C = $\{c_1, .., c_n\}$ is the set of cars to be produced.
- Classes the set of all different cars sharing the trim level for all components.
- O = $\{o_1, ..., o_m\}$ is the set of different components.
- A: trim levels
- J : characteristic J $\subseteq$O×A
- S = $\{s_1, ..., s_m\}$ is the set of stations to install the different components.
- Hj : Nj at most H_j of N_j successively cars may have characteristic j.
- r_{ij} : C × O →{0, 1}, r_{ij} = 1 if in station s_i the component with the characteristic H_j is installed, r_{ij} = 0 otherwise.
- V : it defines a maximum number of transportation vehicles.
- TDIS($s_{i,}s_{j)}$: defines displacement times from station S_i to station S_j.

2.2 Solution

The problem will be solved when finding the sequence that violates the minimum number of sequence rules and the routes for replenish all the components, such that the capacity and constraints are met. Each route will be attended by only one vehicle. The following notation is used to denote the change of sequences:

- a sequence is noted $\pi = (c_{i1}, c_{i2}, ..., c_{ik})$
- subset of options required by a car is defined as a class classOf (c_i)= $\{h_j \in H | r_{ij} = 1\}$
- a route is defined as a nonempty subset of stations attended by each vehicle, Rv_i = $\{s_0, s_1, ..., s_{m+1}\}$ where s0 = s_{m+1} denotes the depot
- the set of all sequences that may be built is π_c
- the concatenation $\oplus$ of two sequences is the first followed by the second
- a sequence π_1 is a subsequence of another π_2 , $\pi_1 \subseteq \pi_2$, if there exist two sequences that can be concatenated to π_1 to create π_2
- τ cycle (takt) time

– the cost of the sequence π and the route R depend upon the number of violated constraints, the vehicles used and the distance traveled by each vehicle, and the amount of stock in the assembly line (see equation 1).

$$\text{cost}(\pi, R)=\sum_{o_i \in o} \sum_{\pi_k \ \pi} \text{violation}(\pi_k, O_i) \times \text{violCost} + \sum_n (\text{travelCost}(V_n) + \text{vehiclesUsed}(V_n) \times \text{Cost}) + \sum_m \text{holdingCost}(s_m) \quad (1)$$

where

$$\text{violation}(\pi_k,O_i);\ \text{violation}(\pi_k,O_i)=0 \text{ if } \sum_{cl \ \ \pi_k} r_{lj} \leq H_j;\ 0 \text{ otherwise} \quad (2)$$

$$\text{travelCost}(V_n)=\sum_{l \in R} \text{TDIS}_{ll'} \times \text{unitCostKm} \quad (3)$$

$$\text{vehicleUsed}(V_n)=0 \text{ if distanceT raveled}(V_N)=0;\ 1 \text{ otherwise} \quad (4)$$

$$\text{holdingCost}(s_m)=\sum_{j\tau} \text{stock}_{j\tau} \times \text{unitCostStock} \quad (5)$$

A solution will be defined as the set of production sequence and the routes for the vehicles that permit replenishment of the components for the given production requirements.

2.3 Construction of Solutions

We give a detailed formulation of the construction algorithm in algorithm 1. The algorithm constructs the solution from an empty sequence and empty replenishment route. Cars are iteratively added until the sequence is complete. At every step, candidates are restricted to the ones that generated the minimum cost, it means the election is restricted only to cars create the minimum extra cost. With this set of candidates (cand) the next car is chosen using transition probability (eq. 6 or 7). Then the demand over the time is calculated (operation 16) and the replenishment route is built. We start from an empty route. We duplicate the depot the number of transportation vehicles. We keep doing this, until all the cars are sequenced, we repeat this until stop criteria, and we keep the best solution and calculate the demand over the time.

We start the solution from empty routes; we duplicate the depot a number of times equal to the maximum number of vehicle. We start to add stations from the non-attended locations (candS) between the ones that generate the minimum cost, we choose for each vehicle the one that adds the minimum cost; the probability (eq. 8) will depend on T3 and η values. Once that all the station has been attended, we decrease the number of vehicles and repeat the creation of routes, unless that the number of vehicles cannot attend all the stations on time. We should keep the best solution to lay pheromones. Finally, we repeat the entire process.

```
Algorithm 1
Initialize pheromones
Repeat
  π←empty {Start to sequence the cars}
  while |π|≤|C| do let C−π denote the set of cars of C
that are not sequenced
  cand←the minimum cost generated by {ck∈C−π}|∀cj∈C−π,
cost(π<ck>)≤cost(π<cj>)
  if ∀ci∈cand, cost(π<ck>)≤cost(π<cj>) then
    for every car class cc ∈{classOf(ci)}∈ C − π do
      T2(cc) ←T2(cc)+ cost(π<cj >) − cost(π)
    end for
  end if
  Choose ci ∈ cand with the probability p(ci, candCar, π)
  π ←<ci >
  end while
  keep the best sequence
  calculates the instant demand for the best sequence
  repeat
    R← empty {Routes for transportation vehicles}
    for vn = maxVehicles to vn =1 do
      while |R|≤|S| do
        let (R−S) denote the set of non-attended stations
        duplicate the depot a number=Vn
        the ants select a candS←the min cost generated by
{sk ∈ R − S} with p(si, candS, R)
      end while
      if stock at station ≤ safety stock then
        break
      end if
      decrease one replenishment vehicle vn--
      keep the best route for the transportation vehicles
    end for
  until stop criteria
  calculate the cost
  keep the best solution an update the pheromones
until stop criteria
```

The algorithm use two types of ants, the sequencing ants and the routing ants. The sequencing ant will be complete when the ant contains a full production sequence of cars. A routing ant will be complete when all the stations are visited.

The probability to build the sequence is the one described by Solnon for combining two pheromones [6] in section 6. The following part is inspired in the approach of Baran [11]. The first colony minimizes the number of vehicles, while the second colony minimizes the inventory cost. Both colonies use independent pheromone and

collaborate sharing a global best solution. This solution is used to update the pheromones.

$$p(c_i, \text{candCar}, \pi) = \frac{[\tau_1(c_j,c_i)]^{\alpha 1}[\tau_2 classOf(c_i)]^{\alpha 2}}{\sum_{c_k \in cand}[\tau_1(c_j,c_i)]^{\alpha 1}[\tau_2 classOf(c_i)]^{\alpha 2}} \text{ if the last car of } \pi \text{ is } c_j \quad (6)$$

$$p(c_i, \text{candCar}, \pi) = \frac{[\tau_2 classOf(c_i)]^{\alpha 2}}{\sum_{c_k \in cand}[\tau_2 classOf(c_i)]^{\alpha 2}} \text{ if } \pi \text{ is empty} \quad (7)$$

$$p(s_i, \text{candS}, R_{v_i}) = \frac{[\tau_3(s_i,s_j)]^{\alpha 3}[\ (s_i,s_j)]^{\beta}}{\sum_{s_k \in candS}\left[[\tau_3(s_i,s_j)]^{\alpha 3}[\ (s_i,s_j)]^{\beta}\right]} \text{ if } s_j \quad \text{candS, 0 otherwise} \quad (8)$$

Where $\alpha 1, \alpha 2, \alpha 3, \beta$ are relative weights for the pheromone and heuristic values respectively.

2.4 Pheromones

The three proposed pheromone structures achieve complementary goals; the first try to identify good sequence, the second try to identify critical cars, the third try to identify vehicle routes that could deliver on time the components.

- *pheromone* τ_1. Ants lay pheromone on a couple of cars (c_i,c_j) ∈ C×C associated with the amount of pheromone τ_1(c_i,c_j). It represents the past experience of sequence car c_j after c_i. This pheromone is bounded with [Tmin, Tmax] and it is initialized in the maximum.
- *pheromone* τ_2. Ants lay pheromone on car classes cc ∈ Classes(C) and the amount of pheromone τ_2(cc) represents the past experience with the car sequence of this class without violating constraints. This pheromone is bounded with [Tmin, Tmax] and it is initialized in the minimum.
- *pheromone* τ_3. Ants lay pheromone on the path between the current location and the possible location (s_i,s_j) ∈ Station(S) and the amount of pheromone levels of τ_3T3(s_i,s_j), indicating how proficient it has been visit station j after i. This pheromone is bounded with [Tmin, Tmax] and it is initialized in the minimum.
- heuristic η(s_i s_j). the dynamic attractiveness of the arc (i,j) will be: η(s_i s_j) = 1/$stock_j$, it will be computed dynamically depending of the inverse of the stock level in each station at each time.

2.5 Pheromone Update

Each pheromone will be laid and update according to with their characteristic.

Updating Pheromone τ_1 Once every ant has construed a sequence, pheromone trials are updated decreasing in order to simulate evaporation multiplying every arch by (1−ρ1). Then the best ant deposit along their paths a trail of pheromone inversely proportional with the total cost generated by the violated constrains. If the value is lower or higher than the range, it will be adjusted to the closest bound.

Updating Pheromone τ_2 Ants lay pheromone on car classes during the construction, when no more cars can be scheduled without new constraints, some pheromone is laid in the classes of the cars that have not been scheduled. The pheromone update occurs during the construction step. Every ant adds pheromone not only the best ant. In order to simulate evaporation each class is multiplying by $(1 - \rho 2)$.

Updating Pheromone τ_3 First, local updating is conducted by reducing the amount of pheromone on all visited arcs by multiplying by $(1 - \rho 3)$. Global trail updating is performed to all the arcs included in the best route founded by one of the m ants.

3 The algorithm

The algorithm follows the ACO scheme. Where each part of the problem is modeled as the search for a best Hamiltonian path in the graph. Solution are constructed using a pheromone model, then the solutions are used to modify the pheromone values. As we use utility workers for the sequence part, all the sequences are feasible. A big enough set of transportation vehicles is defined to ensure that all the routes are feasible and capable of deliver the components when are needed.

A full solution is defined as the sequence and the route of vehicles to replenish the components. After each iteration, we obtain full feasible solution to the problem which is improved after each iteration. The decrease of the use of the vehicle is given after several iterations where vehicles select the nil route.

Specialized ants are created to different part to the problems; this is intended to differentiate paths typically used by each vehicle. First, pheromone trails are initialized, then at each cycle sequence ants construct a sequence, and each route ant creates a route; the global cost function evaluates the results, and pheromone trails are updated. The algorithm stops iterating when a maximum number of cycles were performed.

1. Pheromones trials are initialized.
2. For the maximum iteration allowed to do:
   (a) The sequence ants construct a sequence.
   (b) Calculate the demand over the time
   (c) Each route ant constructs a full route.
   (d) Each route ant constructs a full route using one less vehicle.
   (e) Evaluate the solution using the global function.
   (f) Update the pheromone trails with their respective pheromone and rules.

The vehicles departure from the depot with a load of components equal to the number of vehicles v divided by the number of stations s times the number of car produced n, always respecting the vehicle capacity (see eq. 9)

$$\frac{v \times n}{s} = \leq \text{capacityOfVehicle} \tag{9}$$

Each vehicle from the replenishment route departs from the warehouse ($s_i = 0$), visit the stations and finish in the warehouse. The transportation time includes, traveling from station s_i to station s_j and unload the components.

In order to promote the exploration of different solutions each routing ant start to explore from a different part, we multiply the probability matrix for the equation 10 as a factor of the selection of the candS in the algorithm 1 in the operation 23.

$$\frac{(\text{ord}(\text{ant}) + 1)}{\text{totalNnumberOfAnts}} \tag{10}$$

The solution is driven by three main costs, the cost to the use utility workers for overloading of the station (violation of the sequence rule), the use of the transportation vehicles, and the inventory cost of the components.

The ACO was tuned using the instance small instance where an exact solution can be found by the MIP algorithm. The ACO algorithm finds solutions with the gap of 5% in less than 1 minute, when the MIP algorithm takes more than one hour. As bigger instance was not possible to solve by exact algorithm and no public instance for the entire problem was founded to compare with the optimal. We use the car sequencing instances reported by [12], and then we create the replenishment data, such as capacity, vehicle number, and so on.

4 Conclusion

The advantage of joint decision takes more importance when the cost of the space is higher than in a low-cost facility. The production space is a limited resource; the space has to be used in activity that adds value to the product and decreases the holding space. Becoming a key factor in factories where the inventory is limited, and there is no possibility to store more than one or two hours of inventory.

This paper uses the natural cooperative behavior of the ants to obtain a solution of joint problems. The contribution of this work consists in the development of a collaborative ant colony optimization system to obtain a good-quality solution for problems that cannot be solved to optimality, and the jointly solution to the problem using ACO which as far as the knowledge of the authors has not been described throughout the literature.

Future research should focus for larger problems, combining different techniques like additional types of pheromones, ranking methods, different construction strategies for the route, such as the local exchange, or candidate list.

References

1. M. López-Ibáñez, T. Stüutzle, Automatic configuration of multi-objective aco algorithms, in: Swarm Intelligence, Springer, 2010, pp. 95–106.

2. M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents, Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 26 (1) (1996) 29–41.
3. N. Boysen, M. Fliedner, A. Scholl, Sequencing mixed model assembly lines survey classification and model critique, European Journal of Operational Research 192 (2) (2009) 349–373.
4. V. Giard, J. Jeunet, Optimal sequencing of mixed models with sequence-dependent setups and utility workers on an assembly line, International Journal of Production Economics 123 (2) (2010) 290–300.
5. J. Gottlieb, M. Puchta, C. Solnon, A study of greedy, local search, and ant colony optimization approaches for car sequencing problems, in: Applications of evolutionary computing, Springer, 2003, pp. 246–257.
6. C. Solnon, Combining two pheromone structures for solving the car sequencing problem with ant colony optimization, European Journal of Operational Research 191 (3) (2008) 1043–1055.
7. R. Pulido, A. García-Sánchez, M. Ortega-Mier, The inventory routing problem for the mixed car model assembly line, in: International Conference on Industrial Engineering and Industrial Management., 2013.
8. A. Campbell, L. Clarke, A. Kleywegt, M. Savelsbergh, The inventory routing problem, Fleet management and logistics 1 (1998) 95–113.
9. M. Gendreau, J.-Y. Potvin, O. Braumlaysy, G. Hasle, A. Lokketangen, Metaheuristics for the vehicle routing problem and its extensions: A categorized bibliography, Springer, 2008.
10. J. E. Bell, P. R. McMullen, Ant colony optimization techniques for the vehicle routing problem, Advanced Engineering Informatics 18 (1) (2004) 41–48.
11. B. Barán, M. Schaerer, A multiobjective ant colony system for vehicle routing problem with time windows, in: IASTED International Multi-Conference on Applied Informatics, Vol. 21, 2003, pp. 97–102.
12. J.-C. Régin, J.-F. Puget, A filtering algorithm for global sequencing constraints, in: Principles and Practice of Constraint Programming-CP97, Springer, 1997, pp. 32–46.

Pallet building and truck loading strategies for an inter-depot transportation problem

M.T.Alonso*,R.Álvarez-Valdés**,J. Gromicho***,F.Parreño*,G. Post****

*University of Castilla-La Mancha, Math department, Albacete, Spain,mariateresa.alonso@uclm.es,Francisco.Parreno@uclm.es
**University of Valencia, Department of statistics and operational research, Burjassot, Valencia, Spain,ramon.alvarez@uv.es
***Vrije University, Econometric department, Amsterdam, The netherlands,Joaquim.Gromicho@ortec.com
****Twente University,Applied mathematics department, Enschede, The netherlands,Gerhard.Post@ortec.com

Abstract. The problem we deal is a inter-depot transportation problem.The company supplies a group of depots spread around the country. At each depot a daily decision has to be taken about which trucks will be sent to any of the other depots to distribute the goods. Each depot sends a weekly list of products they want to receive and these products are classified by days.The main depot has to put the products onto pallets and load the pallets onto the trucks for distribution, respecting as far as possible the priority and trying to operate with as few trucks as possible each day. Therefore, the objective here is to minimize the number of trucks needed to transport all the products while fulfilling a set of constraints. Our approach is to solve the problem in one phase building the products and loading them onto trucks in a GRASP algorithm.

Keywords: Loading trucks, GRASP, Loading pallets.

1 Introduction

The problem we deal is a real problem about inter-depot transportation in which a company has a main depot and it have to supplies a group of small depots spread around the country. Each small depot sends a weekly list of products they want to receive classified by day. The main depot has to put the products onto pallets and load the pallets onto the trucks for distribution, respecting as far as possible the priority and trying to operate with as few trucks as possible each day. Therefore, the objective here is to minimize the number of trucks needed to transport all the products while fulfilling a set of constraints.

When building the pallets, the units of the same product are arranged into layers. A layer is a predefined arrangement of units of the same product. Therefore, each layer has given dimensions and contains a given number of items. These layers have to be used whenever possible. Only if the number of units of

a product is not a multiple of the number of units per layer are the remaining units considered individually for packing.

All the products are placed on pallets of dimensions 1219.2 x 1016 millimetres, namely an ISO pallet, and there are three different types of pallets:

- Stock pallet, where all the layers are of the same product. In this case the number of layers composing the pallet is fixed and therefore the pallet height is also fixed.
- Case pallet, which combines layers of different products.
- Rest pallet, which contains the remaining units of each product that cannot form a layer.

The product layers can be *oversize* if their dimensions, length, width or both, are greater than the pallet surface. That is allowed up to a given percentage of the pallet dimensions. Otherwise, they are *non-oversize*.

If all the pallets are non oversize, 15 pallets can be positioned in the truck lengthwise, 2 widthwise and 2 heightwise, giving a total number of 60 pallets per truck. Because of the oversize, sometimes it is not possible to put 2 pallets widthwise and only 14 pallets can be placed lengthwise.

The problem has a list of constraints:

- **Priority constraint**: Each order has a priority or reception day that has to be respected, meaning that if truck x contains orders for day d, then the trucks $x+1$, $x+2$, etc. cannot contain orders for days before day d. In case items of day d do not fill the truck completely, the items of the day $d+1$ can be used to fill the truck.
- **Stackability constraint**: The pallet height is limited to half the truck height. Therefore, at most two pallets can be placed in each stack. However, not any 2 pallets can be placed in the same stack, as the selection of 2 pallets depends on certain properties such as weight, with lighter pallets being placed on top of heavy pallets. Each pallet has a stackability identification; denser pallets have a lower identification and less dense ones a higher identification, so that one pallet can be placed on the top of another if its identification is greater or equal to that of the pallet below.
- **75% support constraint**: At least, 75% of the pallet surface or the layer surface must be placed on another layer or pallet or on the floor of the truck.
- **Weight constraint**: The truck has a maximum weight that it can bear. The items can be light (cereals, tea, ...), or quite heavy (soups, cans of drinks,...). In this problem the weight constraint tends to be much more restrictive than the constraint of not exceeding the volume of the truck.
- **Stability constraint**: For load stability reasons, it is preferred that the height of the stacks decreases towards the end of the truck, placing pallets with the same height together.
- **Axle weight constraint**: The truck has two axles, each of which can bear a maximum weight, so the load has to be balanced throughout the truck so as not to exceed the weight limit. Let *cog* be the centre of gravity in longitudinal direction, x_1 the position of the front axle and x_2 the position

of the rear axle and g the total weight packed. Then, the weights supported by the front axle g_1 and the rear axle g_2 are

$$g1 = \frac{g * (cog - x_1)}{(x_2 - x_1)} \tag{1}$$

$$g2 = \frac{g * (x_2 - cog)}{(x_2 - x_1)} \tag{2}$$

– **Centre of gravity constraint**: The axle weight constraint means that the load should be well-spread lengthwise. Also, the centre of gravity of the load should not deviate too much from its ideal position, i.e. the centre of gravity should preferably be within a centre pallet.

2 Problem formulation

The problem can be defined as follows. A manufacturer has to send a set of different products, $i \in 1, \ldots I$, to meet daily orders, the days are represented by $k \in 1 \ldots K$, and each order line is composed of a product i and the amount of it for that day q_{ik}. The product is defined by its dimensions (x_i, y_i, z_i) in millimetres, its weight w_i in kilograms, and its possible rotations (rx_i, ry_i, rz_i), with $rx_i = 1$ if the dimension x of the product can be upward. For instance, a cereal box is defined by its dimensiones $(406.4, 338.1, 295.1)$ in millimeters,its weight 2.767 Kg and its allow orientation $(0, 0, 1)$, it means the box can be placed only in one orientation with the z dimension in vertical.

All the products have a pre-defined placement in layers. A layer is an arrangement of products of the same type put in a certain position, defined by its dimensions (lx_i, ly_i, lz_i) in millimetres, its weight lw_i in kilograms, the number of units in the layer lu_i, the number of layers of the same product that can be stacked ln_i and the stackability identification $ls_i \in 1, \ldots, 4$, which represents the density of the product, the denser products having a small identification of 1 and the less dense ones receiving a 4. One layer can be placed on top of another if the layer below has a stackability identification smaller than or equal than the lower layer. In Figure 1 a layer of cereal boxes is built with three boxes in the x and y dimensions. Each layer is composed of 9 items with those dimensions, it can be rotated in x and y, and only four layers of cereals can be stacked.

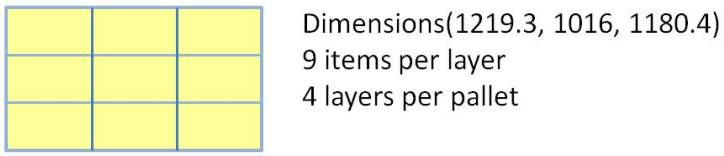


Fig. 1. layer of cereal boxes

The layers are stacked to build pallets. Each pallet has a support whose dimensions are (p_x, p_y, p_z) millimetres, a weight, w_p, and it can be rotated. The units of a product are first assigned to layers, and then the layers are placed onto the pallets. If there are enough layers of the same product, stock pallets are built first and then the remaining layers are combined into case pallets. Finally, if there are units of some products not assigned to layers, they are used to build rest pallets.

The pallets are loaded onto trucks, and a given set of trucks ($j \in 1 \dots J$) is available. For each truck j, we know its dimensions (tx_j, ty_j, tz_j), the weight it can bear on the front axle, W_{1j}, and on the rear axle, W_{2j}. The distances of the axles from the front part of the truck are d_{1j}, d_{2j}. In Figure 2 a truck is shown with its dimensions and axle distances.

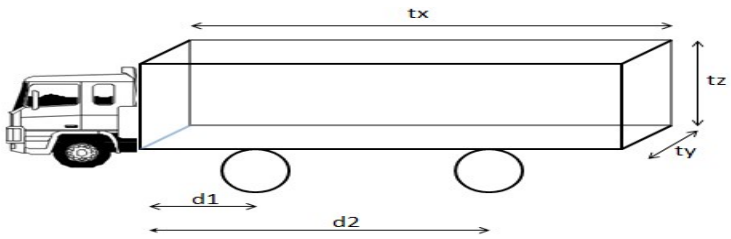


Fig. 2. A truck

3 Data analysis

A preliminary study of the data instances was carried out to obtain information about the characteristics of the problem, and then we considered what the most adequate approach might be. The company provided us with 11 instances. In those instances, we had the information about pallet support, the features of the trucks, the features of the products, and for each product the features of its layer and a list of orders ordered by day.

All the available trucks are of the same type with dimensions (14630.4, 2590.8, 2768.6)mm and the maximum weight that they can bear is 20048 Kg. With the characteristics of the trucks and the data of the products per day we can obtain lower bounds on the number of trucks needed for each day.

- Lower bound per volume, that is the number of trucks needed for loading the volume of all products. It is calculated dividing the product volume of a day by the volume of the truck.
- Lower bound per weight, that is the number of trucks needed for loading the weight of all products.
- Lower bound per position, that is the number of trucks that one needs to load all the pallets that can be formed with those products. For this lower bound, we first have to calculate an estimation of the number of pallets.

Analysing the lower bounds for the 11 instances,at table 1, the highest lower bound is the lower bound per weight. In this case the volume is not significant when determining the number of trucks. The weight of the products is the parameter with the greatest influence on the possible solutions, so if we are able to efficiently distribute this weight, we will probably obtain better solutions. As a result, we focused on developing strategies that help us to optimize the load so as to put more pallets onto the truck.

4 The problem approach

The problem can be divided into two problems. One for building the pallets where the input will be the list of orders and the output will be a pallet list with all the products to send. The second problem will be how to load the pallets onto the trucks, where the input will be the list of pallets provided by the first problem and the output will be the number of trucks that are necessary to load the whole list of pallets from the first problem.

With this approach we have two independent problems where the solution of the first problem is the input data of the second problem. The second one depends on the first solution, and if we want another solution to the second problem, it is necessary to rebuild the list of pallets from the first one, creating a new solution. The disadvantage of this approach is that the two problems are independent and one does not take into account the features and constraints of the other.

Our proposal is to keep a constant flow of information between the two problems, takening into account the constraints and the features of both. In the loading phase, we gather information about the features of the pallet, whereas in the building pallet phase, we collect the information about the placement of the pallet. With both pieces of information, we build the best pallet for that position using the remaining products.

To solve this problem, we propose a GRASP algorithm with a constructive phase in which a solution is built adding a pallet at each step, and then an improvement phase, where some movements are performed to improve the initial solution. In the next sections, this algorithm is described.

5 Constructive phase

The first phase of a GRASP algorithm is a randomized constructive heuristic algorithm. The constructive algorithm builds a solution by means of an iterative process in two steps, the first one consists in finding a space for placing the pallet and the second one in building the pallet and placing it onto the truck. Both steps are described in the next section.

5.1 Selecting a place

The first step is to select a position on the truck to place the next pallet. For that step we have to take account the constraints centre of gravity and axle weight,

because the position of the pallet affects the weight that each axle can bear, and this can displace the centre of gravity away from the middle. Since the weight has to be balanced between the two axles, we have decided to divide the truck into two parts, front and rear, starting from the middle. By placing a pallet on a different side each time, we can control the balance of the weight and keep the centre of gravity approximately in the middle of the truck.

The centre of gravity and the weight that each axle can bear each axle is calculated with Equations (1) and (2). If the weight that axle 1 can bear is greater than that of axle 2, the next space is selected at the front; otherwise it is chosen at the back. Starting at the middle of the truck and going in the chosen direction, front or back, we look for a big enough space to place a pallet. We search the floor of the truck first and, if there is no space, we look at the top of the pallets already loaded.

When the space is found their features are extracted: its dimensions (S_x, S_y, S_z), the maximum weight that can be put into this position W_s, and the stackability identification of the upper pallet, SI_s, if the space is above another pallet.

5.2 Building a pallet for that place

In this step, a pallet will be built that is tailored for the space selected, takening into account its characteristics. The layers of products are ordered by day, first days first, and by density, more dense first. We take the first layer of the list and check if there are enough layers of this product to build a complete pallet, in other words a stock pallet. If this is the case, we build the stock pallet. Otherwise, these layers are added to the pallet and we continue searching for layers that cannot form a stock pallet by themselves and add them to the pallet, building a case pallet, with layers of different products.

One layer is selected if it satisfies several conditions. First, it fits into the selected space. Second, the weight of the pallet, including this layer, does not exceed the maximum allowed weight for this space. Third, the height of the pallet does not exceed half the height of the truck.

The pallet is placed into the space touching the nearest wall of the truck, in such a way that the potential gap is left in the middle of the truck, where it can be used by pallets on the other side and also by pallets on top, always satisfying a 75% of support, as can be seen in Figures 3.

When the pallet is inserted into the solution a new step begins, calculating the new centre of gravity, the load bearing of the axles, the direction to be chosen, and updating the list of layers.

When the first level on the floor of the truck is completed, all the pallets are pushed to the front of the truck. The way in which we place the pallets, starting from the middle, leaves two gaps, one at the front and another at the back of the truck. By pushing the pallets to the front of the truck we join the gaps at the end, with the possibility of placing another pallet in that gap. This process is performed before starting the placement at the second level.

The constructive phase continues until all the layers have been placed or until no more pallets can be placed on the truck because one of the constraints,

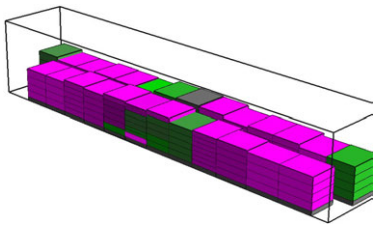


Fig. 3. First floor arrangement

whether it be limiting the volume, the weight or the number of pallets, has been reached. If there are more pallets to be sent, a new truck is opened and the process goes on.

Deterministic constructive results We have run the constructive algorithm in a deterministic way in order to know how far the solution from the lower bound and to study how they can be improved. Table 1 shows the information about the solutions.There are three possible reasons for closing the truck: one of the axles has reached the maximum weight, the truck volume is fully occupied, or no available positions are left for new pallets because there is no space or the stackability constraints do not allow to placing of another pallet on top of the existing pallets.Weight is the most influential factor, as well as the stackability conditions. Note that the stackability constraints are severely limiting the number of pallets per truck.

In Table 1 we can see the comparison between the lower bound and the number of trucks for the deterministic constructive algorithm. In some instances we have achieved the lower bound, but in the others there are differences of up to two trucks.

Instance	Lower bound	Constructive
Instance 1	2	2
Instance 2	3	3
Instance 3	3	3
Instance 4	3	3
Instance 5	6	8
Instance 6	5	6
Instance 7	3	4
Instance 8	3	5
Instance 9	4	5
Instance 10	7	7
Instance 11	4	4

Table 1. Comparing the number of trucks in the deterministic constructive

6 GRASP algorithm

Our proposal is a GRASP algorithm for randomized the election of the layer that componed a pallet. We have studied one type of randomization procedure for selecting the layer to be packed, and we have used the alternative introduced by Resende and Werneck [3], in which a restricted candidate list, *RCL*, is built by randomly selecting a fraction γ ($0 \leq \gamma \leq 1$) of the elements to be placed. Then, the first element of *RCL* in the ordered list is selected. To determine the value of γ we folloew the propouse of Prais and Ribeiro (2000)[1].

6.1 Constructive randomized results

Each instance was run 5000 times with the constructive randomized algorithm. The number of trucks in the best solution for each instance is shown in the third column of Table 2, together with the lower bound and the value of the deterministic constructive algorithm. In this process the number of trucks has been reduced for instances 8 and 9, but for instances 5, 6, and 7 the trucks are the same, indicating that some improving movements are needed.

Instance	Lower bound	Deterministic	Randomized
Instance 1	2	2	2
Instance 2	3	3	3
Instance 3	3	3	3
Instance 4	3	3	3
Instance 5	6	8	8
Instance 6	5	6	6
Instance 7	3	4	4
Instance 8	3	5	4
Instance 9	4	5	4
Instance 10	7	7	7
Instance 11	4	4	4

Table 2. Comparing the number of trucks in the randomized constructive

6.2 Postprocess

This process is performed after the constructive phase and is executed for all the solutions that are going to pass to the improvement phase. The idea behind this decision is to reorganize the pallets on the trucks smoothing out the randomization effects. In this process, the denser pallets are placed at the bottom, whereas the less dense pallets are placed on top whenever possible. In this way, it is possible to bring up pallets that, due to the randomization effects, are placed at the bottom preventing other pallets from being put on top of them due to the stackability constraint. The postprocess is applied to each truck in the solution. This postprocess can generate new positions to allocate pallets.

6.3 Improvement phase

One solution the constructive phase consists of a list of pallets packed on each truck, with their contents and positions, and some characteristics of each truck, in particular the weight on each axle. In the second phase, the solution is improved by a local search which uses several moves.

We will apply each movement according to the features of the solution. These moves are especially careful with the priority constraints. If we have a truck with products up to day h we only consider loading it with products of up to this day h. In the next subsections the movements are described.

Improvement 1: Filling a percentage of the container From the post-processed solution, a percentage of the pallets on each truck is extracted. The extraction method can follow different strategies. For instance, we can extract pallets randomly, only the upper pallets, only the pallets on one side, and so on. Then, the resulting gaps are filled with new pallets composed of the layers of the extracted pallets and those loaded on other posterior trucks. Note that we respect the priority order related to the days, so the layers corresponding to first day will be placed first, and so on.

Improvement 2: Insertion For those trucks that have enough space and can bear more weight on either of their axles, it is possible to insert pallets from the next truck. Then, if the free space is situated at the top of the truck, the algorithm looks for candidates among the pallets of the next truck that can fit into this gap in such a way that when the weight of the candidate is added to the weight supported by that axle, it does not exceed the total weight supported by that axle. Once the previous restrictions are satisfied, the candidate can be placed above the pallet which is below the free space or below it, depending on the stackability identifiers of both pallets. For instance, if the candidate has a stackability identifier lower than the existing pallet, the algorithm will insert the candidate below the former one.

Improvement 3: Swapping + insertion This improvement is for those trucks that have been closed because one axle has reached the maximum weight and we try to obtain better results by swapping pallets between consecutive trucks. The idea is to move heavy pallets from the first truck (named A) to the next truck (named B) so that the axles of truck A support lower weights, allowing us to load more pallets. In that way, we try to put as many pallets as possible in the first trucks with respect to the following ones. Consequently, if we are able to empty the final truck with this method, we save trucks in the complete solution.

This swapping must be followed by an insertion in order to increase the number of pallets on truck A. First, each pallet on truck A and all the pallets on truck B are examined in order to find candidates for swapping. Two pallets are swapped if the pallet on truck B can fit into the place of the pallet on truck A and it is lighter. Then we try to take advantage of the difference in weight by

loading more pallets. In the case of finding candidates for swapping, we evaluate the remaining pallets on truck B to check if one of them can be inserted into a gap on truck A (improvement 2: insertion). We only consider this improvement if we find successful candidates for both movements, swapping and insertion.

7 Conclusions and future works

For the problem of inter-depot transportation where products have to be put in pallets and load onto trucks, our propose is a feedback information between the two phases palletization and loading. All of this into a metaheuristic GRASP, in two phases one for building solutions selecting randomly the layers of the pallets and the second for improving the solutions.

The future works is to develop the second phase, check the results and continue improving the solutions the algorithm get.

References

1. M. Prais and C.C. Ribeiro. Reactive GRASP: An application to a matrix decomposition problem in TDMA traffic assignment. *INFORMS Journal on Computing*, 12:164-176, 2000.
2. M.G.C. Resende and C.C. Ribeiro. Greedy Randomized Adaptive Search Procedures. In F. Glover and G. Kochenberger, editors, *Handbook of Metaheuristics*, pages 219-249, Kluwer, Boston, 2003.
3. M.G.C. Resende and R.F. Werneck. A hybrid heuristic for the p-median problem. *Journal of Heuristics*, 10:59-88, 2004.

Algoritmos *GRASP* para solucionar el problema *Blocking Flow Shop*

Joaquín Bautista, Alberto Cano, Rocío Alfaro y Cristina Batalla

Universitat Politècnica de Catalunya.
Avenida Diagonal 647, 7 planta. 08028 Barcelona, Spain
{joaquin.bautista,alberto.cano-perez,rocio.alfaro,
cristina.batalla}@upc.edu
http://www.prothius.com

Resumen Se resuelve el *Blocking Flow Shop Problem* (*BFSP*), consiste en determinar una secuencia de piezas procesadas por una línea de máquinas sin espacios de almacenamiento entre ellas. Para ello, se implementan 14 algoritmos heurísticos *Greedy Randomized Adaptive Search Procedure* (*GRASP*) aplicados a las 60 primeras instancias de Taillard. Los resultados obtenidos con los 7 procedimientos que incorporan la heurística de Nawaz son los más competitivos cuando se comparan con otros los de la literatura.

Keywords: GRASP; Algoritmos de secuenciación; secuencias; Blocking Flow Shop.

1. Introducción

El *Flow Shop Scheduling Problem* (*FSP*) es uno de los problemas que ha recibido más atención en los últimos 50 años por parte de profesionales e investigadores debido a la gran variedad de contextos productivos que permite modelar. En el *FSP*, un conjunto I de n de trabajos o piezas debe ser procesado en un conjunto K de m de máquinas. Todos los trabajos deben pasar por todas las máquinas en el mismo orden, empezando en la máquina 1 y finalizando en la máquina m. Cada trabajo, $i \in I$, requiere un tiempo de proceso, $p_{i,k} > 0$, en cada una de las máquinas, $k \in K$. El objetivo es encontrar una secuencia de proceso de los trabajos que optimice un criterio de eficiencia.

En la versión más popular del problema, conocida como *Permutation Flow Shop Problem* (*PFSP*), el espacio de almacenamiento entre dos fases consecutivas del proceso, donde las piezas pueden esperar hasta que puedan ser procesados por la siguiente máquina, es ilimitado. Sin embargo, existen sistemas productivos, en los sectores químico, farmacéutico, modelado de plástico, electrónica, metalurgía y alimentación, en los que existen líneas de producción en las que el espacio de almacenamiento está limitado. Si se asume que no existe espacios de almacenamiento entre dos fases consecutivas del proceso, entonces se produce un gran cambio estructural en el comportamiento del sistema, debido a que una pieza no puede abandonar la máquina que lo está procesando hasta que la siguiente

máquina esté libre. En este caso, el trabajo debe permanecer en la máquina previa, bloqueando la máquina e impidiendo a ésta realizar otros trabajos.

Esta variante se conoce como *Blocking Flow Shop Problem* (*BFSP*) y es la que consideramos en este documento, teniendo como objetivo minimizar el instante de finalización de todos trabajos en el taller (*makespan*, C_{max}). Haciendo uso de la notación propuesta por Graham [1], el problema considerado se conoce como $Fm|block|C_{max}$ (y el *PFSP* como $Fm|prmu|C_{max}$).

Dada la naturaleza *NP-difícil* del problema, su resolución emplea heurísticas, debido a su capacidad de encontrar soluciones de calidad en un tiempo reducido. En la literatura se encuentra el procedimiento de Nawaz, Enscore y Ham (*NEH*) [2], y más recientemente, se encuentra Ronconi [3], que propone dos variantes de éste, y Bautista [4] donde se propone un algoritmo basado en *programación dinámica acotada* (*BDP*) y una lista actualizada de las mejores soluciones para las instancias de Taillard.

Para este trabajo se ha diseñado un procedimiento *GRASP extendido* (*GRASP-x*) que admite diversas variantes en función de los valores asignados a tres parámetros. Entre dichas variantes se encuentran las heurísticas *Greedy* constructivas con posterior optimización local, los procedimientos *Multistart*, los algoritmos *GRASP clásicos* [5,6] y los algoritmos *GRASP* en los que la probabilidad de selección de los candidatos se hace depender de la aptitud de éstos, la cual puede medirse a través de una función de calidad dependiente de cada candidato. Una extensa recopilación de trabajos sobre aplicaciones se puede encontrar en [7,8].

Nuestra propuesta contiene: (1) el diseño e implementación de cotas parciales para el problema; (2) siete algoritmos basados en el procedimiento *GRASP* que actúan como maestros para dirigir la exploración en el espacio de búsqueda; y (3) una experiencia computacional, con ejemplares de la literatura, para comparar los resultados obtenidos mediante los procedimientos implementados con los mejores resultados encontrados en la literatura.

Este trabajo se organiza de la siguiente forma: la sección 2 presenta la descripción del problema; la sección 3 contiene el diseño de cotas parciales y globales para el problema; en la sección 4 se describe el procedimiento *GRASP-x* adaptado para resolver el *BFSP*; la sección 5 se centra en la descripción y comparación de resultados de la experiencia computacional realizada, que explota siete algoritmos (*derivados del procedimiento GRASP-x tras fijar siete conjuntos de valores a los tres parámetros*) sobre ejemplares de la literatura; finalmente, la sección 6 muestra algunas conclusiones sobre el presente trabajo.

2. Descripción del problema

A partir del instante 0 se deben procesar n trabajos, en el mismo orden, en m máquinas. Los tiempos de proceso para cada operación se denominan $p_{i,k}$, donde k denota una máquina y i un trabajo. Los tiempos de proceso están fijados y son positivos. La función objetivo considerada es la minimización del *makespan* (C_{max}), que es el instante en que el taller finaliza todas sus operaciones.

En un programa factible asociado a la permutación π, definimos $s_{k,t}$ como el instante de inicio del proceso destinado en la máquina k al trabajo que ocupa la posición t y $e_{k,t}$ como el instante en que el trabajo que ocupa la posición t abandona la máquina k. El problema $Fm|prmu|C_{max}$ puede ser formalizado como sigue:

$$s_{k,t} + p_{[t],k} \leqslant e_{k,t} \qquad k = 1, ..., m; t = 1, ..., n \tag{1}$$

$$s_{k,t} \geqslant e_{k,t-1} \qquad k = 1, ..., m; t = 1, ..., n \tag{2}$$

$$s_{k,t} \geqslant e_{k-1,t} \qquad k = 1, ..., m; t = 1, ..., n \tag{3}$$

$$C_{max} = e_{m,n} \tag{4}$$

Siendo $p_{[t],k}$ el tiempo de proceso en la máquina k de la pieza que ocupa la *t-ésima* posición en la secuencia π. Además $e_{k,0} = 0 \quad \forall k$ y $e_{0,t} = 0 \quad \forall t$ son las condiciones iniciales.

El programa obtenido es semi-activo si la restricción (1) se escribe como $s_{k,t} + p_{[t],k} = e_{k,t}$ y las restricciones (2) y (3) se resumen como $s_{k,t} = max\{e_{k,t-1}, e_{k-1,t}\}$.

Cuando no existen espacios de almacenamiento entre etapas, caso del problema $Fm|block|C_{max}$, si el trabajo i finaliza su operación en la máquina k y la próxima máquina, $k+1$, se encuentra todavía ocupada con el trabajo anterior, el trabajo completado i ha de permanecer en la máquina k, bloqueándola. Esta condición requiere una restricción adicional (5) en la formulación del problema.

$$e_{k,t} \geqslant e_{k+1,t-1} \qquad k = 1, ..., m; t = 1, ..., n \tag{5}$$

siendo necesario añadir la condición inicial $e_{m+1,t} = 0$, $t = 1, ..., n$.

En el caso $Fm|block|C_{max}$ el programa obtenido es semi-activo si la restricción (1) y (5) se resumen como en (6):

$$e_{k,t} = max\{s_{k,t} + p_{[t],k}, e_{k+1,t-1}\} \qquad k = 1, ..., m; t = 1, ..., n \tag{6}$$

Consecuentemente, el problema $Fm|prmu|C_{max}$ puede ser visto como una relajación del problema $Fm|block|C_{max}$.

3. Acotando los valores de las secuencias

Asumamos que se ha construido una subsecuencia $\pi(t) = \{\pi_1, \pi_2, ..., \pi_t\}$ de t trabajos.

Supongamos ademas que disponemos de la información $x_i(t)$ y $e_k(t)$; donde $x_i(t)$ adopta el valor 1 si el trabajo i está presente en $\pi(t)$ y $e_k(t)$ representa el instante en que queda libre la máquina k tras procesar todos los trabajos contenidos en $\pi(t)$. El esquema de acotación empleado se muestra en la figura 1.

Para completar todos los trabajos, nos faltará añadir a la secuencia $\pi(t)$ un segmento conteniendo los $n - t$ trabajos pendientes no contenidos en $\pi(t)$. Si

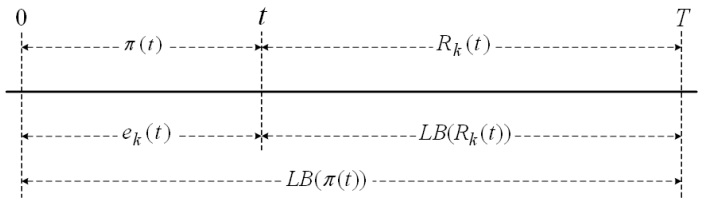


Figura 1. Esquema de acotación.

$R_k(t)$ es una secuencia de los trabajos pendientes en la máquina $k \in K$, es fácil obtener una cota inferior, $LB(R_k(t))$, para dicha secuencia. En efecto:

$$LB(R_k(t)) = \sum_{i \notin \pi(t)} p_{i,k} + \underset{i \in \pi(t)}{min} \left\{ \sum_{\kappa=k+1}^{m} p_{i,\kappa} \right\} \tag{7}$$

Y, por consiguiente, una cota global de C_{max} asociada a la secuencia $\pi(t)$ será:

$$LB(\pi(t)) = \underset{k \in K}{max} \{e_k(t) + LB(R_k(t))\} \tag{8}$$

4. Algoritmos *GRASP* para el *BFSP*

4.1. Preliminares

La complejidad del *BFSP* y el interés de obtener soluciones para ejemplares del problema con dimensiones industriales, hace recomendable el uso de procedimientos heurísticos capaces de ofrecer soluciones aceptables con bajo esfuerzo de computación.

La metaheurística *GRASP* [5,6] es de tipo multi-arranque y está provista de 2 fases en cada iteración: (1) un procedimiento *Greedy* que sirve para construir una solución aceptable sin que sea preciso alcanzar el óptimo global; y (2) una segunda fase para obtener un óptimo local dentro de un vecindario y teniendo como punto de partida la solución que resulta al aplicar el procedimiento *Greedy* de la fase_1. Obviamente, la solución ofrecida por el *GRASP* es la mejor entre las obtenidas en el conjunto de iteraciones.

Para la primera fase *Greedy* es importante definir un buen procedimiento que pueda ofrecer soluciones aceptables y una diversidad suficiente de soluciones que permitan explorar diferentes regiones en el espacio de soluciones. Para garantizar dicha diversidad se emplea el azar, de manera que el siguiente elemento a añadir a una solución parcial se sortea entre los elementos de una lista restringida de candidatos (*RCL*); dicha lista contiene los candidatos que presentan los mejores valores en relación a una función (*una cota para* C_{max}*, por ejemplo*) diseñada para la selección.

Para resolver un problema de optimización mediante un procedimiento *GRASP* es preciso definir los siguientes elementos: (1) el proceso aleatorio empleado en

la selección entre candidatos y el procedimiento *Greedy*; (2) el vecindario de una solución y, lógicamente, el procedimiento para explorar dicho vecindario; y (3) el criterio de finalización del algoritmo, normalmente vinculado a un máximo número de iteraciones o al tiempo de ejecución.

En la figura 2 se presenta un esquema general del algoritmo *GRASP*.

1. Inicialización
2. Mientras tiempo $CPU \leqslant CPU_{max}$
   2.1. Solución ⟵ Fase_constructiva (*Semilla*)
   2.2. Solución ⟵ Mejora_local (*Solución*)
   2.3. Actualizar_solución (*Solución, Mejor_solución*)
3. Salida: *Mejor_solución*

Figura 2. Esquema general de la metaheurística *GRASP*.

En algunos casos, al procedimiento *GRASP*, se le puede añadir un post-proceso que permita combinar las soluciones generadas [9].

4.2. Fase *Greedy* de construcción de una solución

El procedimiento implementado para esta fase del algoritmo (*ver figura 3*) construye progresivamente una secuencia seleccionando, en cada etapa asociada con el instante, $t = 1, ..., T = n$, un elemento candidato a partir de una lista restringida de éstos (*sea RCL*). En efecto, llegados a la etapa t, en la que se dispone de una secuencia (*solución*) parcial $\pi(t)$, para cada trabajo i no procesado aún, se determina el índice f_i ($\forall i : x_i = 0$) a partir del valor de la cota $LB(\pi(t) \cup \{i\})$; tras ello, la lista de candidatos se construye en 2 pasos:

1. En el primero, a partir del parámetro $Z \in [0, 1]$ denominado impedancia, se seleccionan todos los trabajos con un valor de cota no superior a $1/Z$ veces el valor de la menor de ellas (*cota correspondiente al mejor candidato*).
2. En el segundo paso, se seleccionan como máximo los L mejores candidatos (*ordenados por cota, de menor a mayor*) incluyendo en la lista, claro está, los trabajos empatados en cota con el *L-ésimo* candidato.

Nótese que gracias a la impedancia se pueden descartar soluciones peores a las de un valor de referencia, dejando en *RCL* menos soluciones que L. Además, la formulación propuesta en la figura 3 comprende como casos particulares los siguientes procedimientos: (1) los algoritmos *GRASP* tradicionales con tratamiento de empates incorporado, pues basta hacer $Z \to 0$, $f_0 \to \infty$, $\eta = 1$ y fijar un valor de L menor que el número de candidatos; (2) *Multistart*, haciendo $Z \to 0$, $f_0 \to \infty$, $\eta = 1$ y fijar un valor de L igual al número de candidatos; y (3) las heurísticas *Greedy* con tratamiento de empates con $Z = 1$; si $Z \to 0$ y L es suficientemente grande ($L = |I| = n$) todos los elementos compatibles

son candidatos y toda solución del *BFSP* tiene una probabilidad no nula de ser generada por el procedimiento.

---

0. Inicialización

Leer: $n = T$, I, K, $p_{i,k}(\forall i, \forall k)$, Z, L, f_0, η

Hacer:

$t = 0$

$\pi(t) = \{\varnothing\}$ siendo $\pi(t) = \{\pi_1, \pi_2, ..., \pi_t\}$ la secuencia parcial de los t trabajos procesados.

$x_i = 0$ $\forall i \in I$, siendo $x_i = 1$ si el trabajo i está contenido en $\pi(t)$ y 0 en caso contrario.

1. Cálculo del índice f

$\forall i \in I$: $x_i = 0$, determinar:

$f_i = LB(\pi(t) \cup \{i\})$
siendo $LB(\pi(t) \cup \{i\})$ el valor de la cota de C_{max} asociado a la secuencia parcial $\pi(t) \cup \{i\}$.

2. Creación de la lista de candidatos *RCL*

Sea $f^* = \text{mín}_{x_i=0}\{f_i\}$;

$RCL(f) = \{i \in I : (x_i = 0) \wedge (f_i \leq \text{mín}\{f_0, f^*/Z\}\}$

donde $Z \in [0, 1]$ es la impedancia sobre el conjunto de elementos compatibles y f_0 es la elasticidad aditiva que que se puede corresponder con el valor de una solución de referencia afectada por la impedancia:
- Si $|RCL(f)| \leq L \Rightarrow RCL = RCL(f)$
- Si $|RCL(f)| > L$:

Sea $i_L \in RCL(f)$ el trabajo que ocupa la *L-ésima* posición en la lista $RCL(f)$ ordenada no-decrecientemente respecto al índice f. Hacer:

$$RCL = \{i \in RCL(f) : f_i \leq f_{i_L}\}$$

3. Selección del trabajo a secuenciar

$\forall i \in RCL$, determinar:

$g_i = \frac{(f_0 - f_i)^\eta}{\sum_{j \in RCL}(f_0 - f_j)^\eta}$, donde η es la elasticidad potencial.

Seleccionar, por sorteo, con probabilidades g_i $\forall i \in RCL$ un trabajo; sea i^* el resultado de esta selección.

4. Actualización

$x_{i^*} \leftarrow x_{i^*} + 1$; $t \leftarrow t + 1$; $\pi_t = i^*$

5. Finalización

Si $t < T = n$, ir a paso-1.

Si no, finalizar.

Figura 3. Fase constructiva *GRASP* para el *BFSP*.

4.3. Fase de mejora local

Se propone una mejora local exhaustiva tipo 2-intercambio entre dos elementos de la secuencia en curso de mejora. La exploración del vecindario es determinista y se realiza de izquierda a derecha.

A partir de una secuencia en curso, $\pi_c(T)$, con valor $C_{max}(\pi_c(T))$, se genera una secuencia vecina, $\pi_v(T)$, mediante un 2-intercambio tentativo entre los elementos que ocupan las posiciones t y t' de la secuencia en curso. Si $C_{max}(\pi_c(T)) > C_{max}(\pi_v(T))$, el intercambio se consolida, $\pi_v(T)$ se convierte en la nueva secuencia en curso y se reinicia el proceso de intercambios. En caso contrario, se prosigue con la generación de una nueva secuencia tentativa, una secuencia vecina, mediante otro 2-intercambio tentativo. El procedimiento finaliza cuando ninguna solución vecina tiene un valor de C_{max} mejor que el de la solución en curso.

4.4. Fase de mejora a través de *NEH*

Además, a la secuencia obtenida en la fase de mejora anterior, cuando se ha alcanzado un óptimo local, se puede aplicar el procedimiento *NEH* consistente en seleccionar al azar un número fijo de elementos de la secuencia y extraerlos de la misma; posteriormente, los elementos extraídos se van insertando de uno en uno en la posición de la secuencia que genera menor C_{max}.

5. Experiencia computacional

Se ha realizado una experiencia computacional con las 6 primeros sets de ejemplares de Taillard [10]. Cada set contiene 10 ejemplares con el mismo número de trabajos (n) y mismo número de máquinas (m). En los 6 sets, el número de trabajos varía entre 20 (set 1) y 50 (set 6) y el número de máquinas entre 5 (set 1) y 20 (set 6). Un valor actualizado de las mejores soluciones para estos ejemplares se puede encontrar en [4].

Para obtener las soluciones de la fase constructiva se emplean 7 algoritmos derivados del procedimiento general *GRASP-x* para el que se han fijado los valores de los parámetros Z, L, f_0 y η (*ver tabla* 1) siendo f_G el valor de referencia, para cada ejemplar, ofrecido por el procedimiento *greedy (G)* con tratamiento de empates. Los algoritmos resultantes tras asignar valores a los 4 parámetros son: (G) una heurística *Greedy* constructiva; (M) un procedimiento *Multistart* tradicional; (*GR-01/*$0{,}5 * |I|$) un algoritmo *GRASP* tradicional con una lista *RCL* limitada a $|I|/2$ candidatos (50 % del número de trabajos); (*GR-5/*$0{,}5 * |I|$) un algoritmo *GRASP* con lista limitada a $|I|/2$ candidatos y con probabilidades de selección de éstos levemente dependientes de su aptitud; (*GR-8/*$0{,}5 * |I|$) un algoritmo *GRASP* con alta dependencia entre las probabilidades de selección de los candidatos y su aptitud y con lista restringida a $|I|/2$ candidatos; y dos

algoritmos *Multistart*, (*GR-5/*$|I|$) y (*GR-8/*$|I|$) con probabilidades de selección de los candidatos idénticas a (*GR-5/*$0{,}5*|I|$) y (*GR-8/*$0{,}5*|I|$), respectivamente.

Tabla 1. Algoritmos derivados de *GRASP-x*. Características.

Alg.	Z	L	f_0	η
G	1	1	∞	1
M	0.01	$\|I\|$	f_G/Z	1
GR-01/$0{,}5*\|I\|$	0.01	$0{,}5*\|I\|$	f_G/Z	1
GR-5/$0{,}5*\|I\|$	0.5	$0{,}5*\|I\|$	f_G/Z	1
GR-8/$0{,}5*\|I\|$	0.8	$0{,}5*\|I\|$	f_G/Z	1
GR-5/$\|I\|$	0.5	$\|I\|$	f_G/Z	1
GR-8/$\|I\|$	0.8	$\|I\|$	f_G/Z	1

Cada uno de los 60 ejemplares se ha resuelto, primero, empleando las 7 variantes de la fase constructiva del algoritmo *GRASP* y, posteriormente, se han aplicado, a cada solución obtenida en la primera fase, dos procedimientos de mejora local. El primer procedimiento de mejora local (*LS* en el texto) es un 2-intercambio exhaustivo, consolidando el intercambio cuando se produce mejora. El segundo procedimiento (*LS+NEH* en el texto) consiste en un bucle que aplica consecutivamente el primer procedimiento de mejora local mencionado, hasta alcanzar un óptimo local, y seguidamente la heurística *NEH* extrayendo e insertando 5 trabajos de la secuencia. El tiempo máximo de *CPU* concedido a la mejora de un ejemplar es de 10 *s*.

Los procedimientos *GRASP* han sido programados en *gcc v. 4.2.1*, en un ordenador Macintosh iMac con un procesador Intel Core i7, 2.93 Ghz. y 8 Gb de memoria RAM, usando MAC OS X 10.6.8 como sistema operativo. Ni la implementación ni el compilador hacen uso de *threads* ni de otra forma de código paralelo, y, por tanto, el ordenador actúa como un único procesador a 2.93 GHz.

En las tablas 2 y 3 se muestran los resultados más significativos del experimento, empleando los procedimientos de mejora *LS* y *LS+NEH*, respectivamente, tras la fase constructiva del *GRASP*. En la tabla 2 se recoge: (1) el número de óptimos alcanzado (*#opt*) por cada uno de los 7 algoritmos sobre los 60 ejemplares del *BFSP*; (2) el promedio de la *desviación porcentual relativa* (RPD=((*solución* - *óptimo*)/*óptimo*)×100), para los ejemplares y cada algoritmo, tanto para la fase_1 *Greedy* constructiva del *GRASP* ($\overline{RPD_1}$) como para el proceso completo ($\overline{RPD_2}$) que incluye el procedimiento de mejora local (*LS* o *LS+NEH*); (3) el tiempo medio de *CPU*, por ejemplar, requerido por una iteración de cada uno de los 7 algoritmos *GRASP* ($\overline{CPU}$); y (4) el promedio para RPD para cada uno de los sets.

A la vista de las tablas 2 y 3 observamos (en promedio para los 60 ejemplares) los siguientes hechos: (1) las soluciones obtenidas en la fase constructiva del *GRASP*, se alejan de las mejores soluciones conocidas aproximadamente entre el 13 % con el algoritmo *Greedy* determinista (G) y casi el 23 % con los algo-

Tabla 2. *#opt*, $\overline{RPD_1}$, $\overline{RPD_2}$, $\overline{CPU}$ y $\overline{RPD_2}$ para cada uno de los sets en el caso de *LS* para cada uno de los 7 algoritmos

	#opt	$\overline{RPD_1}$	$\overline{RPD_2}$	$\overline{CPU}$	S1	S2	S3	S4	S5	S6
G	0	13.01	4.01	0.28	3.67	3.85	3.84	3.31	4.07	5.31
M	0	22.75	3.22	0.38	2.79	2.37	2.41	3.89	4.04	3.82
GR-01/$0{,}5 * \|I\|$	0	20.64	3.06	0.40	2.42	2.56	1.80	3.55	4.09	3.94
GR-5/$0{,}5 * \|I\|$	0	20.46	2.92	0.40	2.13	2.40	1.81	3.62	3.76	3.81
GR-8/$0{,}5 * \|I\|$	0	20.76	3.01	0.41	2.45	2.11	2.25	3.62	3.87	3.76
GR-5/$\|I\|$	0	22.43	3.26	0.37	2.58	2.43	2.00	4.17	4.26	4.10
GR-8/$\|I\|$	0	22.38	3.16	0.36	2.50	2.62	1.85	3.80	4.14	4.03

Tabla 3. *#opt*, $\overline{RPD_1}$, $\overline{RPD_2}$, $\overline{CPU}$ y $\overline{RPD_2}$ para cada uno de los sets en el caso de *LS+NEH* para cada uno de los 7 algoritmos

	#opt	$\overline{RPD_1}$	$\overline{RPD_2}$	$\overline{CPU}$	S1	S2	S3	S4	S5	S6
G	19	13.01	0.69	10.00	0.07	0.12	0.16	0.95	1.27	1.56
M	25	22.75	0.70	10.00	0.13	0.00	0.03	1.10	1.37	1.57
GR-01/$0{,}5 * \|I\|$	20	20.64	0.64	10.00	0.14	0.13	0.07	0.85	1.25	1.43
GR-5/$0{,}5 * \|I\|$	24	20.46	0.68	10.00	0.10	0.11	0.00	1.04	1.29	1.53
GR-8/$0{,}5 * \|I\|$	28	20.76	0.66	10.00	0.00	0.00	0.02	1.12	1.32	1.53
GR-5/$\|I\|$	25	22.43	0.67	10.00	0.02	0.03	0.02	1.01	1.31	1.64
GR-8/$\|I\|$	25	22.38	0.71	10.00	0.04	0.12	0.02	1.20	1.35	1.52

ritmos *Multistart* (M) y *GRASP* con $L = |I|$ ($GR - 5/|I|$ y $GR - 8/|I|$); (2) *LS* desciende, desde la solución ofrecida por la fase constructiva del algoritmo *Greedy* determinista (G), a óptimos locales con peor valor que los alcanzados por el resto de procedimientos que incorporan azar; (3) atendiendo a todos los procedimientos, *LS* mejora del 20.35 % al 3.23 % la distancia a las mejores soluciones conocidas; (4) *LS+NEH* reduce dichas distancias al 0.68 %; y (5) en todos los procedimientos con mejora *LS*, se alcanza el primer óptimo local en 0.37 s.

Además, en los procedimientos con mejora *LS*, no se alcanza ninguna mejor solución conocida para los 60 ejemplares. En cambio, usando todos los procedimientos con mejora local *LS+NEH* se alcanzan 29 sobre 60. Finalmente, el procedimiento $GR - 8/0{,}5 * |I|$ alcanza 28 óptimos, incluyendo los del set 1 y 2.

6. Conclusiones

Los procedimientos *GRASP* implementados se muestran competitivos respecto a los existentes en la literatura. Concretamente el procedimiento $GR - 8/0{,}5 * |I|$ alcanza todas las mejores soluciones conocidas para los sets 1 y 2 de Taillard. El procedimiento *NEH* reduce las distancias al 0.68 % respecto a

las mejores soluciones conocidas, mientras que su exclusión de los procedimientos producen una reducción al 3.23 %; sin optimización local los procedimientos constructivos dejan esta distancia al 20.35 %.

Agradecimientos. Los autores agradecen la colaboración prestada por *Nissan Spanish Industrial Operations* (NSIO), la Cátedra Nissan UPC y al Gobierno Español por la financiación parcial de este trabajo a través del proyecto PROTHIUS-III: DPI2010-16759, incluyendo fondos FEDER.

Referencias

1. Graham, R.L., Lawler, E.L., Lenstra, J.K., Rinnooy Kan,A.H.G.: Optimization and approximation in deterministic sequencing and scheduling: a survey. Annals of Discrete Mathematics, vol. .5, 287-326 (1979).
2. Nawaz, M., Enscore, Jr E.E., Ham, I.: A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. Omega, vol. 11 (1), 91-95 (1983).
3. Ronconi, D. P.: A note on constructive heuristics for the flowshop problem with blocking. International Journal of Production Economics, vol. 87 (1), 39-48 (2004).
4. Bautista, J., Cano, A., Companys, R., Ribas, I.: Solving the $Fm|block|C_{max}$ problem using Bounded Dynamic Programming. Engineering Applications of Artificial Intelligence, vol. 25 (6), 12351245 (2012).
5. Feo, T.A., Resende, M.G.C.: Greedy randomized adaptive search procedures. J. of Global Optimization, vol. 6, 109-133 (1995).
6. Resende, M.G.C., Ribeiro, C.C.: Greedy randomized adaptive search procedures. Handbook of Metaheuristics, F. Glover and G. Kochenberger, eds., Kluwer Academic Publishers, 219-249 (2003).
7. Festa, P., Resende, M.G.C.: An annotated bibliography of GRASP-Part I: Algorithms. International Transactions in Operational Research, vol. 16, 1-24 (2009).
8. Festa, P., Resende, M.G.C.: An annotated bibliography of GRASP-Part II: Applications. International Transactions in Operational Research, vol. 16, 131-172 (2009).
9. Laguna, M., Martí, R.: GRASP and Path Relinking for 2-Layer Straight Line Crossing Minimization. INFORMS Journal on Computing, vol. 11(1), 44-52 (1999).
10. Taillard, E.: Benchmarks for basic scheduling problems. European Journal of Operational Research, vol. 64(2), 278-285 (1993) .

A multiobjective genetic algorithm and robustness information function for time and space assembly line balancing

M. Chica[1], O. Cordón[1,2], S. Damas[1], and J. Bautista[3]

[1] European Centre for Soft Computing, Mieres (Asturias), Spain
{manuel.chica, oscar.cordon, sergio.damas}@softcomputing.es
[2] DECSAI and CITIC-UGR, University of Granada, 18071 Granada, Spain
[3] Universitat Politècnica de Catalunya, Barcelona, Spain
{joaquin.bautista}@upc.edu **

Abstract. The time and space assembly line balancing problem (TSALBP) is a realistic multiobjective version of assembly line balancing industrial problems involving the joint optimization of conflicting criteria such as the cycle time, the number of stations, and the area of these stations. However, the existing problem formulation does not consider the industrial scenario where the demand of a set of mixed products is variable and uncertain. In this contribution we introduce a novel robustness function to measure the robustness of the line configuration when the production plans demand changes. The function is used as additional *a posteriori* information for the non-dominated solutions found by an advanced multiobjective genetic algorithm. Because of their independence, these robustness functions can be used in conjunction with any other multiobjective metaheuristic. Results show how the use of the robustness function can help the decision maker to select robust non-dominated solutions when future demand conditions vary in the assembly line configurations.

1 Introduction

An assembly line is made up of a number of workstations, arranged either in series or in parallel. Since the manufacturing of a production item is divided into a set of tasks which require an operation time for their execution, a usual and difficult problem, called assembly line balancing (ALB), is to determine how these tasks can be assigned to the stations fulfilling certain restrictions such as precedence relations. The final aim of ALB is to get an optimal assignment of subsets of tasks to the stations of the plant (1, 2). An excellent review on ALB and the existing solving methods for the different problems is given in (3). Within ALB, a well-known family of problems is the simple assembly line balancing problem (SALBP) (4, 5, 6). The SALBP only considers the assignment of each task to a single station in such a way that all the precedence constraints are satisfied and no station workload time is greater than the line cycle time.

** This work has been supported by Ministerio de Economía y Competitividad under project SOCOVIFI2 (TIN2012-38525-C02-01 and TIN2012-38525-C02-02), and under PROTHIUS-III: DPI2010-16759, both including EDRF funding.

As a result of the observation of the ALB operation in an automotive Nissan plant from Barcelona (Spain), (7) recently proposed a SALBP extension aiming to design a more realistic ALB model. They considered an additional space constraint to get a simplified but closer version to real-world situations, defining the time and space assembly line balancing problem (TSALBP). The TSALBP presents eight variants depending on three optimization criteria: m (the number of stations), c (the cycle time), and A (the area of the stations). In this paper we tackle the TSALBP-m/A variant[4] which tries to jointly minimize the number of stations and their area for a given product cycle time, a complex and realistic multicriteria problem in the automotive industry.

The multicriteria nature of the TSALBP-m/A favoured the application of multiobjective meta-heuristics (MOMHs) such as multiobjective ant colony optimization (MOACO) (8) or evolutionary multiobjective optimization (EMO) (9). Assembly lines are generally balanced for producing mixed products and their demand is not usually fixed and certain. When the assembly line is devoted to produce mixed products in a given sequence, the operation times of the required tasks are obtained from the average value of the different products and their demand. This is a problematic rough estimate of the actual operation times. If the demand changes, the operation times also change and a re-balancing could be necessary for the configuration.

Robustness can be applied to many components in an optimization process: noise in constraints, objective function, or uncertainties in data variables (10, 11). Real-world applications, as ALB, normally involve uncertainties because of operating conditions or manufacturing process (12). In our case, the interest lies on measuring the robustness of a specific operating condition, i.e. the operation times originated by the mixed products demand. In this work, a robustness function is defined based on the overloading production plans which occur when the demand changes and the line configuration is set. The latter robustness measure is used as additional *a posteriori* information associated to each non-dominated solution returned by the MOO method. The *advanced TSALBP-NSGA-II* is selected as the MOO method for the robustness experiments of this study because of the best performance of the metaheuristics with respect to others (9). However, the robustness functions can be applied to any other MOO method.

The rest of the paper is structured as follows. In Section 2, the TSALBP-m/A formulation and the uncertain demand scenario modelled by production plans are explained. Then, the metaheuristic applied for solving the TSALBP-m/A, the *advanced TSALBP-NSGA-II*, is described in Section 3. The numerical robustness function for assembly line balancing is given in Section 4. The experimentation results are discussed in Section 5. Finally, we present some concluding remarks in Section 6.

2 Demand variation in the TSALBP-m/A for mixed product products

We first introduce the TSALBP-m/A (Section 2.1) and then the real scenario of having a mixed products with changing demand (Section 2.2).

[4] Originally, this TSALBP variant is referred as TSALBP-1/3 (7). This new notation is introduced in this work for a better understanding.

2.1 Time and space assembly line balancing problem

The manufacturing of a production item is divided into a set J of n tasks. Each task j requires an operation time for its execution $t_j > 0$ that is determined as a function of the manufacturing technologies and the employed resources. Each station k ($k = 1, 2, ..., m$) is assigned to a subset of tasks S_k ($S_k \subseteq J$), called workload. Each task j can only be assigned to a single station k.

Each task j has a set of direct "preceding tasks" P_j which must be accomplished before starting it. These constraints are normally represented by means of an acyclic precedence graph, whose vertices stand for the tasks and where a directed arc (i, j) indicates that task i must be finished before starting task j on the production line. Thus, task j cannot be assigned to a station that is ordered before the one where task i was assigned. Each station k also presents a station workload time $t(S_k)$ that is equal to the sum of the tasks' processing time assigned to the station k. SALBP focuses on grouping tasks in workstations by an efficient and coherent way.

In this simplistic model there is a need of introducing space constraints in assembly lines' design based on two main reasons: (a) the length of the workstation is limited in the majority of the situations, and (b) the required tools and components to be assembled should be distributed along the sides of the line. Hence, an area constraint may be considered by associating a required area a_j to each task j and an available area A_k to each station k that, for the sake of simplicity, we shall assume it to be identical for every station and equal to $A = max_{k=1,2,...,m} A_k$. Thus, each station k requires a station area $a(S_k)$ that is equal to the sum of areas required by the tasks assigned to station k.

This leads us to a new family of problems called TSALBP (7). It may be stated as: given a set of n tasks with their temporal t_j and spatial a_j attributes ($1 \leq j \leq n$) and a precedence graph, each task must be assigned to a single station such that: (i) every precedence constraint is satisfied, (ii) no station workload time ($t(S_k)$) is greater than the cycle time (c), and (iii) no area required by any station ($a(S_k)$) is greater than the available area per station (A).

TSALBP presents eight variants depending on three optimization criteria: m (the number of stations), c (the cycle time) and A (the area of the stations). Within these variants there are four multiobjective problems and we will tackle one of them, the TSALBP-m/A. It consists of minimising the number of stations m and the station area A, given a fixed value of the cycle time c, mathematically formulated as follows:

$$f^0(x) = m = \sum_{k=1}^{UB_m} \max_{j=1,2,...,n} x_{jk}, \tag{1}$$

$$f^1(x) = A = \max_{k=1,2,...,UB_m} \sum_{j=1}^{n} a_j x_{jk}, \tag{2}$$

where UB_m is the upper bound for the number of stations m, a_j is the area information for task j, x_{jk} is a decision variable taking value 1 if task j is assigned to station k, and n is the number of tasks.

We chose this variant because it is realistic in the automotive industry since the annual production of an industrial plant (and therefore, the cycle time c) is usually

set by some market objectives. For more information about the problem we refer the interested reader to (8; 13).

2.2 Production plans for modelling changing demand

The latter TSALBP-m/A formulation assumes both a constant demand and fixed operation times t_j. However, real assembly lines are normally employed to assemble more than one single product, and when the demand of each product changes, the operation times of the tasks change in consequence. The demand of a set of mixed products is defined by means of production plans. In this work, the engine assembly line of the Nissan Spanish Industrial Operations (NSIO) plant is the chosen uncertain environment to define the different production plans.

Nine different engines are assembled in the main line of the NSIO plant, $m_1, \ldots, m_9$, having different destinations and assembly characteristics. The first three engine products are built for 4×4 vehicles; products m_4 and m_5 are for VANs; and the remaining four products are used by medium tonnage trucks. When demand is balanced (identical for the nine products) and the cycle time is 3 minutes, the assembly line is divided into 21 workstations having an average length A_k of 4 meters each.

In (7), authors grouped the primary operations of this assembly line in the so called Nissan TSALBP instance having 140 tasks. For each type of engine, operation times change. In Table 1 the operation times of five tasks are listed for illustration. The average operation time when having a balanced demand for the nine products is also shown in the *t-average* column.

Table 1. Operation times and average time for five tasks belonging to the NSIO engine assembly line.

task	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	m_9	t-average
1	64.8	61.2	60	54	58.8	55.2	63	66	57	60
3	18.4	18	20	19.6	19	21.6	21	20.4	22	20
5	19	19.6	18.4	20	21	20.4	18	21.6	22	20
8	9.8	9	10.5	10.8	9.5	11	9.2	10	10.2	10
9	20	19.6	19	18	20.4	18.4	21.6	21	22	20

Of course, it is difficult to always have the same uniform demand for all the engines within a global demand. Although the line is supposed to have a fixed daily production of, for instance, 270 products, the line should be capable of producing the required products for the specific product demand of a given production plan. In other words, the production plan of the 270 engines is not constant. Then, the goal is to have an assembly line configuration that is robust enough for different production plans.

There are currently 23 production plans for the nine engines and one working day at the NSIO. Each program corresponds to a set of operation times biased by the demand of each of the nine products. We have grouped them into seven categories according to the type of engine demand. The seven representative production plans, one per category, are shown in Table 2. Definitely, the demand variation of the production plan for mixed products conditions the average operation times of the 140 assembly line tasks. In that

case, a re-balancing of the assembly line could be necessary. For example, task 1 has operation times of 64.8, 61.2, 60, 54, 58.8, 55.2, 63, 66, and 57 seconds for products m_1 to m_9, respectively. On the other hand, production plan #12 has a demand of 24, 23, 23, 45, 45, 28, 28, 27, and 27 products for each of the engine products. Consequently, the average time for task 1 in the latter plan is 59.44 seconds ($= (64.8 \times 24 + 61.2 \times 23 + \cdots + 66 \times 27 + 57 \times 27)/270$) in comparison with the 60 seconds needed by production plan #1.

Table 2. Production units of the engine models for each production plan.

		Production plans						
Family	Product	# 1	#2	# 3	# 6	# 9	# 12	# 18
4 x 4	m_1	30	30	10	50	70	24	60
	m_2	30	30	10	50	70	23	60
	m_3	30	30	10	50	70	23	60
VAN	m_4	30	45	60	30	15	45	30
	m_5	30	45	60	30	15	45	30
Trucks	m_6	30	23	30	15	8	28	8
	m_7	30	23	30	15	8	28	8
	m_8	30	22	30	15	7	27	7
	m_9	30	22	30	15	7	27	7

The selected representative production plans are used in this work to present additional information to the decision maker (DM) about how robust a new assembly line configuration is under demand changes, i.e. how good it is with respect to those changes.

3 Advanced NSGA-II-based approach for the TSALBP-m/A

This design is based on the original NSGA-II search scheme (14) with an appropriate representation and more effective operators are used to solve the TSALBP-m/A. The algorithm will be referred as *advanced TSALBP-NSGA-II* because of its problem-specific design and potential application to other TSALBP variants (9). The main features and operators of the *advanced TSALBP-NSGA-II* are described in the next subsections.

3.1 Representation scheme

The allocation of tasks among stations is made by employing separators. Separators are thus dummy genes which do not represent any specific task and they are inserted into the list of genes representing tasks. In this way, they define groups of tasks being assigned to a specific station. The maximum possible number of separators is $n-1$ (with n being the number of tasks), as it would correspond to an assembly line configuration with n stations, each one composed of a single task. Tasks are encoded using numbers in $\{1, \ldots, n\}$, as in the previous representation, while separators take values in $\{n + 1, \ldots, 2 \cdot n - 1\}$. Hence, the genotype is again an order-based representation.

The number of separators included in the genotype is variable and it depends on the number of existing stations in the current solution. Therefore, the algorithm works with

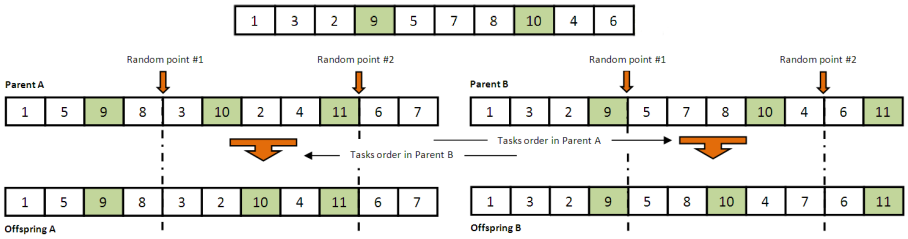


Fig. 1. Coding scheme and crossover example. Separators are those genes coloured.

a variable-length coding scheme, although its order-based representation nature avoids the need of any additional mechanism to deal with this issue. The maximum size of the chromosome is $2 \cdot n - 1$ to allow the presence of separators for the maximum number of possible stations. On the other hand, the representation scheme ensures the encoded solutions are feasible with respect to the precedence relations constraints. However, the cycle time limitation could be violated and it will be a task of the genetic operators to ensure feasibility with respect to that constraint. Figure 1 shows an example of the new coding scheme.

3.2 The crossover operator

As our representation is order-based, the crossover operator can be designed from a classical order-based one. Crossover operators of the latter kind which have been suggested in the literature include partially mapped crossover (PMX), order crossover, order crossover # 2, position based crossover, and cycle crossover, among others (15). We have selected one of the most extended ones, PMX, which has been already used in other genetic algorithm implementations for the SALBP (for example in (16)).

PMX generates two offspring from two parents by means of the following procedure: a) two random cut points are selected, b) for the first offspring, the genes outside the random points are copied directly from the first parent, and c) the genes inside the two cut points are copied but in the order they appear in the second parent. The same mechanism is followed up with the second offspring but with the opposite parents. Thanks to our advanced coding scheme and to the use of a permutation-based crossover, the feasibility of the offspring with respect to precedence relations is assured. See Figure 1 where an example of the operator is shown. In addition, a repair operator must be applied for each offspring after crossover.

3.3 Mutation operator

A mutation operator has been specifically designed and applied uniformly to the selected individuals of the population. It is based on reordering a part of the sequence of tasks and reassigning them to stations. It will be called scramble mutation operator.

The scramble mutation operator works as follows: after choosing two points randomly, the tasks between those points are scrambled forming a new sequence of tasks

in such a way the mutated solution keeps on being feasible with respect to the precedence relations. The existing separators among the two mutation points are ignored and a new reallocation of those tasks is considered by randomly generating new separator locations within the task sequence.

4 Evaluating robust solutions for assembly line balancing when demand is uncertain

Solving the TSALBP when the mixed products demand is uncertain belongs to the robust optimization case where the operating conditions change after the optimal solution is found (17). In our case, the operating conditions are the operation times originated by the different mixed products demands represented by the production plans of Section 2.2. The overall goal is to find a set of non-dominated solutions for the TSALBP-m/A and calculate their robustness for all the possible production plans. In the next paragraphs we will present our proposal for evaluating this robustness.

Let E be the set of possible production plans based on the demand variation and ε^0 a reference production plan, our evaluation proposal is based on determining the workload of the set K of stations of an assembly line configuration in the plans of E.

First, being S_k^0 the tasks assignment to the station k in ε^0 (normally, the balanced plan), the workload of this station k is obtained for all the production plans $\varepsilon \in E$: $t(S_k^0, \varepsilon)$.

Then, the relative station overloads with respect to the available cycle time c are calculated for all the existing production plans ε by applying Equation 3.

$$\omega(S_k^0, \varepsilon) = \frac{\max\{0, t(S_k^0, \varepsilon) - c\}}{c} \qquad \forall k \in K, \forall \varepsilon \in E. \tag{3}$$

From these overload values, the average and maximum station overload values are also calculated through Equations 4 and 5:

$$\overline{\omega}(S_k^0) = \overline{\omega}_k = \frac{1}{|E|} \sum_{\varepsilon=1}^{|E|} \omega(S_k^0, \varepsilon) \qquad \forall k \in K, \tag{4}$$

$$\omega_{max}(S_k^0) = \max_{\varepsilon \in E} \{\omega(S_k^0, \varepsilon)\} \qquad \forall k \in K. \tag{5}$$

Analogously, the average and maximum overloading values for each production plan are obtained by applying Equations 6 and 7:

$$\overline{\omega}(\varepsilon) = \frac{1}{|K|} \sum_{k=1}^{|K|} \omega(S_k^0, \varepsilon) \qquad \forall \varepsilon \in E, \tag{6}$$

$$\omega_{max}(\varepsilon) = \max_{k \in K} \{\omega(S_k^0, \varepsilon)\} \qquad \forall \varepsilon \in E. \tag{7}$$

The latter values allow us to define and calculate our proposed robustness function. This is based on counting the number of overloaded stations and/or the number

of overloading production plans. Given a station $k \in K$, a production plan $\varepsilon \in E$, a configuration line $(S_1^0, S_2^0, ..., S_m^0)$ for a reference production plan ε^0, and a cycle time c, we can state that there is an overload in $(k, S_k^0, \varepsilon, c)$ iff:

$$t(S_k^0, \varepsilon) > c \Leftrightarrow \omega(S_k^0, \varepsilon) > 0 \qquad \forall k \in K, \forall \varepsilon \in E. \tag{8}$$

Associated to the concept of overload, the set of total overloads can be defined as follows:

$$D = \{\forall \varepsilon \in E \wedge \forall k \in K \mid \omega(S_k^0, \varepsilon) > 0\}. \tag{9}$$

Finally and taking into account the set D, a robustness function R_3 can be defined in Equation 10. This will be the robustness function to be used in this study and represents the total number of overloads rate.

$$R_3 = \frac{|D|}{|E||K|} \in [0, 1] \tag{10}$$

5 Experiments

In this section we present the results of the experimentation and the analysis of them. The parameters of the *advanced TSALBP-NSGA-II* are presented in Table 3. The algorithm and the robustness function was applied to three TSALBP real-like instances (**P1, P2 and P4**) which have been created by the NTIGen software, publicly available at `http://www.prothius.com/TSALBP`.

Table 3. Used parameter values for the *advanced TSALBP-NSGA-II*.

Parameter	**Value**	**Parameter**	**Value**
Random seed	1212	Stopping criteria	300 s
Population size	100	Ishibuchi's similarity based mating γ, δ values	10
Crossover probability	0.8	Mutation probability	0.1
α values for scramble mutation	$\{0, 0.8\}$		

The obtained results of the robustness function for the non-dominated solutions are presented in Table 2. The first instance tackled by the algorithm is P1 (first table of Figure 2) where only two non-dominated solutions are found. However, even when having small non-dominated solution sets, the robustness information is important for the DM. Solution #2 reports robustness value of $R_3 = 0.12$. This value means that, when demand varies, the assembly line should support that a 12% of the stations are overloaded (R_3). Solution #2 is thus less robust than solution #1.

P2 is the original Nissan instance having 140 tasks. The number of non-dominated solutions obtained by the algorithm is five (see second table of Figure 2). The numerical value of R_3 allows us to conclude that solutions #1 and #2, those with objective values $(19, 5)$ and $(18, 6.09)$, respectively, are less robust than the remainder when demand changes. Then, if the number of stations (and then, workers) is not restricted, the best approach in terms of robustness is always choosing a solution with more than 19 stations.

There are seven non-dominated solutions for instance P4 (third table of Figure 2). Among them, there are two solutions which are less robust than the remainder. These are solutions #1 and #2 with 15 and 16 stations, respectively. In the latter pair of solutions, the 11% and 5.2% of the stations are overloaded by different production plans (R_3 function).

P1

Sols.	R_3
S1	0.01
S2	0.12

P2

Sols.	R_3
S1	0.139
S2	0.149
S3	0.092
S4	0.083
S5	0.029

P4

Sols.	R_3
S1	0.111
S2	0.052
S3	0.03
S4	0.046
S5	0.017
S6	0.033
S7	0.032

Fig. 2. Robustness values for the non-dominated solutions when solving the NTIGen instances of 100 tasks (P1), 140 tasks (P2), and 220 tasks (P4).

6 Concluding remarks

The existing TSALBP formulation and previous ALB works do not cover an important real scenario where the same assembly line is devoted to produce mixed products and their demand is not fixed. In this paper we have presented a new robustness function measure R_3. It is defined based on the number of overloaded stations. The proposed model was used to analyse the non-dominated solutions provided by the state-of-the-art MOO method for the TSALBP-m/A, the *advanced TSALBP-NSGA-II*, although the nature of the robustness model allows the use of any other MOO method instead. The results of the application of the robustness model are clear. There are some solutions which are less robust than others when demand changes and the DM can take advantage of this information before making her/his decision.

Some future works arise from this contribution: (i) propose and evaluate other robustness functions, (ii) include a graphical visualization model within the robustness representation of the non-dominated solutions, and (iii) integrate the robustness functions within the search of the MOO method (as a restriction or as another objective):

Bibliography

[1] Boysen, N., Fliedner, M., Scholl, A.: A classification of assembly line balancing problems. European Journal of Operational Research **183**(2) (2007) 674–693

[2] Boysen, N., Fliedner, M., Scholl, A.: Assembly line balancing: Which model to use when? International Journal of Production Economics **111**(2) (2008) 509–528

[3] Battaïa, O., Dolgui, A.: A taxonomy of line balancing problems and their solution approaches. International Journal of Production Economics **142**(2) (2013) 259–277

[4] Baybars, I.: A survey of exact algorithms for the simple assembly line balancing problem. Management Science **32**(8) (1986) 909–932

[5] Scholl, A.: Balancing and Sequencing of Assembly Lines (2nd. Edition). Physica-Verlag, Heidelberg (1999)

[6] Scholl, A., Becker, C.: State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. European Journal of Operational Research **168**(3) (2006) 666–693

[7] Bautista, J., Pereira, J.: Ant algorithms for a time and space constrained assembly line balancing problem. European Journal of Operational Research **177**(3) (2007) 2016–2032

[8] Chica, M., Cordón, O., Damas, S., Bautista, J.: Multiobjective, constructive heuristics for the 1/3 variant of the time and space assembly line balancing problem: ACO and random greedy search. Information Sciences **180** (2010) 3465–3487

[9] Chica, M., Cordón, O., Damas, S.: An advanced multi-objective genetic algorithm design for the time and space assembly line balancing problem. Computers and Industrial Engineering **61**(1) (2011) 103–117

[10] Roy, B.: Robustness in operational research and decision aiding: A multi-faceted issue. European Journal of Operational Research **200**(3) (2010) 629–638

[11] Beyer, H., Sendhoff, B.: Robust optimization - a comprehensive survey. Computer Methods in Applied Mechanics and Engineering **196**(33–34) (2007) 3190–3218

[12] Miettinen, K., Deb, K., Jahn, J., Ogryczak, W., Shimoyama, K., Vetschera, K.: Future challenges. In: Multiobjective Optimization. Interactive and Evolutionary Approaches. Volume 5252 of Lecture Notes in Computer Science. Springer (2008) 435–461

[13] Chica, M., Cordón, O., Damas, S., Bautista, J.: Multiobjective memetic algorithms for time and space assembly line balancing. Engineering Applications of Artificial Intelligence **25**(2) (2012) 254–273

[14] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation **6**(2) (2002) 182–197

[15] Poon, P.W., Carter, J.N.: Genetic algorithm crossover operators. Computers & Operations Research **22**(1) (1995) 135–147

[16] Sabuncuoglu, I., Erel, E., Tayner, M.: Assembly line balancing using genetic algorithms. Journal of Intelligent Manufacturing **11**(3) (2000) 295–310

[17] Ferreira, J., Fonseca, C.M., Covas, J.A., Gaspar-Cunha, A.: Evolutionary multi-objective robust optimization. In: Advances in Evolutionary Algorithms (www.i-techonline.com). InTech, Vienna, Austria (2008) 261–278

MOEAs para la Secuenciación de Operaciones con Restricciones de Disponibilidad de Máquinas y Capacidad del Buffer

Frutos Mariano[1], Méndez Babey Máximo[2], Tohmé Fernando[1], Broz Diego[1]

[1] Instituto de Investigaciones Económicas y Sociales del Sur (IIESS-CONICET), Universidad Nacional del Sur, Bahía Blanca, 8000, Argentina.
[2] Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería (SIANI), Universidad de Las Palmas de Gran Canaria, Las Palmas de Gran Canaria, 35017, España.
mfrutos@uns.edu.ar, maximo@dis.ulpgc.es, ftohme@criba.edu.ar, diego.broz@uns.edu.ar

Abstract. Existen en el modelado de sistemas productivos una variedad de problemas que pueden ser abordados con técnicas de optimización multi-objetivo. El problema de secuenciación de operaciones es uno de ellos aplicando además restricciones de disponibilidad de máquinas y capacidad del buffer. Nuestro estudio comprendió el análisis de diferentes algoritmos evolutivos multi-objetivo (MOEAs) para resolver este tipo de problemas, tomando como marco experimental la programación de la producción de tres casos extraídos del mundo real en un ámbito productivo tipo Job-Shop. Se realizaron comparaciones de tres algoritmos de referencia, a saber, NSGAII, SPEA2 e IBEA, a través de los indicadores de rendimiento Hipervolumen y R2. Se estableció que los algoritmos SPEA2 e IBEA resultaron ser los más eficientes para resolver este problema.

Keywords: Job-Shop Scheduling Problem, Algoritmos Evolutivos, Optimización Multi-objetivo, Simulación.

1 Introducción

En la mayoría de los procesos de planificación y programación de la producción se debe decidir sobre la asignación de recursos limitados a cada uno de los trabajos a realizar con el fin de optimizar uno o más objetivos en el corto plazo relacionados con la eficiencia del sistema productivo [1]. Para apoyar estas decisiones en la mayoría de los casos se utiliza el modelo de Job-Shop, el cual considera un conjunto de máquinas y un conjunto de trabajos compuestos por una secuencia ordenada de operaciones que se deben procesar en las máquinas con el objetivo de minimizar, entre otros, el tiempo de procesamiento de todos los trabajos, más conocido como makespan [2]. Este problema pertenece a la clase denominada NP-Hard [3], para los cuales aún no se disponen de algoritmos de resolución eficientes que los ejecuten en tiempos computacionales razonables. La mayoría de los trabajos encontrados en la literatura enfocan su desarrollo en problemas con un solo objetivo. Varios autores afirman que un problema genuino de secuenciación de operaciones involucra la optimización de más de un objetivo para lograr la eficiencia de un sistema productivo [4]. Luego de analizar

varios enfoques se optó por la utilización de algoritmos evolutivos [5,6]. Las principales ventajas de estos algoritmos son su fácil adaptación y su eficiencia al resolver problemas con más de un objetivo. Estos algoritmos requieren conocer solamente el valor de la función objetivo o fitness en la población discreta y no sus derivadas. Esto último permite abordar una gran variedad de problemas de optimización no tratables mediante métodos basados en gradientes. Se evaluará, en este artículo, el rendimiento de tres algoritmos evolutivos aplicados en la secuenciación de operaciones en un sistema productivo tipo Job-Shop [7,8]. Se incorporará en el modelado restricciones de disponibilidad de máquinas y capacidad del Buffer. Esto ocurre cuando la producción no estandarizada, como es el caso del Job-Shop, comparte las mismas máquinas con la producción estandarizada o seriada, la cual tiene prioridad sobre la primera.

2 Job-Shop Scheduling problem

El problema de secuenciación de operaciones en un ámbito productivo tipo Job-Shop, también conocido como Job-Shop Scheduling Problem, es particularmente complicado de resolver. El mismo se ha resuelto para la existencia de 1, 2 y 3 máquinas y para algunos valores arbitrarios del número de trabajos a realizar. Son relativamente pocos los desarrollos de algoritmos eficientes para asignar 4 o más máquinas y programar 3 o más trabajos ya que existen muchas secuencias posibles. En los siguientes puntos se hará un estado del arte sobre algunos de estos algoritmos, se definirá formalmente este problema y se presentarán los objetivos a optimizar objeto de éste estudio.

2.1 Estado del arte

Haciendo un breve recuento de los enfoques que intentan resolver el problema, tenemos que en [9], se presenta un método de búsqueda tabú para minimizar la tardanza total. El método utiliza reglas de secuenciación para obtener una solución inicial y búsqueda tabú para generar nuevas soluciones. En [10] se presenta un algoritmo GRASP (Greedy Randomized Adaptive Search Procedures) que trata de minimizar el makespan. En [11] se propone un algoritmo híbrido también para minimizar el makespan. Este incorpora un proceso de inicialización y una búsqueda local, sobre la base de un algoritmo genético. En [7], se desarrolla un algoritmo genético cuya representación está hecha para ser siempre factible. En [12], también se plantea un algoritmo genético y el trabajo se basa en el análisis de los parámetros de control del algoritmo optimizando el makespan. En [13] se desarrolla un algoritmo HACO (Hybrid Ant Colony Optimization), para optimizar el makespan. El algoritmo ACO trata de preservar y mejorar las soluciones existentes, y un algoritmo de post-procesamiento se aplica para explorar las soluciones tras su finalización. En [14], se propone un enfoque por localización, que permite encontrar buenas soluciones, minimizando el makespan y la carga de las máquinas. En [15] estudian el problema proponiendo reglas de despacho y en cada generación se guía el espacio de búsqueda mediante esquemas. Se estudian las representaciones de las soluciones, para encontrar la más eficaz en términos de tiempo y de alcanzar el mejor makespan. Este enfoque es generalizado en [16] donde se propone una arquitectura basada en algoritmos evolutivos, un aprendizaje con teoría de esquemas y un generador de poblaciones mediante reglas de despacho compuestas. En [17] se propone un modelo matemático para el Job-Shop y para probarlo generan diez instancias pequeñas y diez instancias medianas/grandes, aunque sólo logran resolver las instancias pequeñas. En [18] se propone un algoritmo genético

basado en escenarios múltiples, donde cada escenario corresponde a una operación y cada máquina factible a un estado. En [19] también se propone un algoritmo genético, pero utilizan el enfoque de localización propuesto en [14]. Con ello establecen una mutación que consiste en seleccionar una operación entre aquellas máquinas con más carga de trabajo y reasignar dicha carga a la máquina que tenga menos. En [2] se propone un algoritmo genético híbrido para resolver el problema con la restricción de que no se permite ningún tiempo de espera entre operaciones dentro de cualquier trabajo, minimizando el makespan. Por último, en [20], se propone un algoritmo evolutivo para minimizar el makespan, la carga de trabajo total y la máxima carga de trabajo.

2.2 Definición formal del problema y sus objetivos

Como fue mencionado en la sección anterior, el problema consiste en organizar la ejecución de diferentes trabajos. Es decir, se tiene un conjunto finito J de n trabajos $\{J_j\}_{j=1}^{n}$. Estos trabajos deben ser procesados por un conjunto finito M de m máquinas. $\{M_k\}_{k=1}^{m}$ Al procesamiento de una operación de un trabajo J_j en una máquina M_k se la denomina O_{ij}^{k}, donde i indica el orden en que debe ser realizado el conjunto de operaciones $\{S_j\}_{j=1}^{n}$ de un mismo trabajo J_j. La operación O_{ij}^{k} requiere el uso de una máquina M_k durante un tiempo ininterrumpido τ_{ij}^{k}, conocido como tiempo de procesamiento. La solución a este problema es determinar las fechas de inicio t_{ij}^{k} de cada operación O_{ij}^{k} orientadas por los objetivos a optimizar [8]. En este caso, se definieron tres objetivos. En primer lugar, se tomó como objetivo la minimización del makespan (1). En segundo lugar, se tomó como objetivo la minimización del tiempo medio de flujo (2). Plantear este objetivo permite disminuir el número de trabajos en proceso. Por último, se plantea minimizar los efectos que presentan las variaciones en los tiempos de procesamiento de cada operación O_{ij}^{k} en el makespan. Para ello se desarrollan micro-simulaciones sobre cada solución para generar la varianza que presenta el primer objetivo [21]. Minimizar esta varianza ($\sigma^2(C_{máx}^{j})$) garantiza una secuenciación de operaciones más estable (3).

$$f_1 : \text{mín}\, C_{máx}^{j} = \sum_{i \in S_j} \underset{k \in M}{\text{máx}} (t_{ij}^{k} + \tau_{ij}^{k}) \tag{1}$$

$$f_2 : \text{mín}\, \overline{F} = \frac{1}{n} \sum_{j \in J} \sum_{i \in S_j} (t_{mj}^{h} + \tau_{mj}^{h} - t_{1j}^{k}) \tag{2}$$

$$f_3 : \text{mín}\, \sigma^2(C_{máx}^{j}) \tag{3}$$

Para éste modelo se plantea inicialmente la restricción de no negatividad para las fechas de inicio t_{ij}^{k} de cada operación O_{ij}^{k} : $t_{ij}^{k} \geq 0$. Seguido a esto, se plantean las restricciones de precedencia (conjuntivas) para operaciones del mismo trabajo: $t_{ij}^{k} - t_{sj}^{h} \geq \tau_{sj}^{h}$. Esta se cumple para toda operación O_{sj}^{h} que precede a O_{ij}^{k}. Por último, se plantean las restricciones de no traslape (disyuntivas) para operaciones que se procesan en la misma máquina: $t_{ij}^{k} - t_{sp}^{k} \geq \tau_{sp}^{k}$. Esta se cumple para toda operación O_{sp}^{k} que precede a O_{ij}^{k}. El propósito de este último grupo de restricciones es garantizar que una máquina no procese dos operaciones a la vez. Se incorporan al modelo original dos nuevos grupos de restricciones: disponibilidad de máquinas y capacidad del Buffer. El primer grupo de restricciones limita el intervalo operativo de cada M_k, es decir el intervalo $(t_{ij}^{k}, t_{ij}^{k} + \tau_{ij}^{k}) \in$ al intervalo operativo de producción no estandarizada para M_k. El

segundo grupo de restricciones limita la cantidad de operaciones O_{ij}^{k} en la máquina M_k que aún no se han completado y están en espera. El límite que se ha tomado para el análisis es: Buffer 0 (no-wait) y n-1 (not-restricted).

3 Algoritmos Evolutivos Multi-objetivo

Los algoritmos evolutivos imitan los procesos genéticos mejorando las soluciones mediante apareamiento de soluciones existentes como si fueran cadenas de ADN que van generando otras cadenas. Un cromosoma contiene estas cadenas de ADN, que se dividen en pequeñas unidades llamadas genes. Para los problemas de secuenciación de operaciones, se generan estos cromosomas con una secuencia especial que identifica cada programación. Se trabaja sobre la idea de que, tantos cromosomas se generan, que alguno debe contener la secuencia óptima buscada. Para explicar con mayor detalle la fase evolutiva, se utiliza un ejemplo que corresponde a una instancia que contiene tres trabajos y tres máquinas (3×3). El problema total consta de nueve operaciones cuyos tiempos de procesamiento (τ_{ij}^{k}) y su varianza (σ^2), y la máquina en donde se debe procesar (M_k) se muestran en la Tabla 1.

Tabla 1. Problema de Secuenciación de Operaciones

Problema 3 × 3 con 9 operaciones									
J_j		J_1			J_2			J_3	
$O_i^k{}_j$	$O_1{}^1{}_1$	$O_2{}^2{}_1$	$O_3{}^3{}_1$	$O_1{}^2{}_2$	$O_2{}^3{}_2$	$O_3{}^1{}_2$	$O_1{}^3{}_3$	$O_2{}^1{}_3$	$O_3{}^2{}_3$
$\tau_i^k{}_j$	10	15	8	12	9	10	11	5	16
σ^2	4	4	2	4	3	5	2	1	4
M_k	1	2	3	2	3	1	3	1	2

3.1 La fase evolutiva

Para representar al individuo, se propone una variante del cromosoma de secuencia (variables enteras) desarrollado en [8]. El cromosoma propuesto contiene variables binarias y la cadena de ADN tendrá tantos genes como máquinas el problema. Cada gen se dividirá en una cantidad de elementos llamados alelos y dependerá del número de trabajos del problema. El tamaño del gen será igual a $\lceil \log(n!) / \log(2) \rceil$. El tamaño total del cromosoma tendrá una longitud igual a $m * \lceil \log(n!) / \log(2) \rceil$. Para cada gen la sucesión de números binarios representa una secuencia pre-establecida de trabajos en cada máquina. Para el ejemplo, 000→1 | 2 | 3, 001→1 | 3 | 2, 010→2 | 1 | 3, 011→2 | 3 | 1, 100→3 | 1 | 2 y 101→3 | 2 | 1. En el primer cromosoma de la figura 1, First Parent, se ve que para la máquina M_1 el gen es 000. Esto significa que la secuencia de trabajos para esta máquina será 1, 2 y 3. Para la máquina M_2 el gen es 010, por lo que la secuencia de trabajos será 2, 1 y 3. Por último, para la máquina M_3, el gen es 011, por lo que la secuencia de trabajos será 2, 3 y 1.

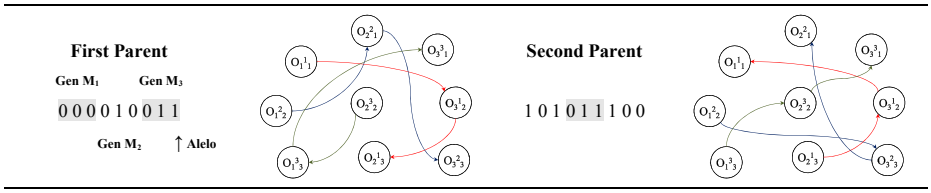


Fig. 1. Cromosoma de Secuencia

El operador de cruce opera entre parejas de cromosomas. Estos cromosomas se alinean, luego se fraccionan en ciertas partes y posteriormente intercambian fragmentos entre sí. A modo de ilustrar el procedimiento de cruce sobre el cromosoma de secuencia, se puede observar, en la figura 1, los padres a cruzar: First Parent y Second Parent. Se seleccionó el cruce uniforme ya que dio mejor resultado en la exploración de los puntos que se encuentran sobre la frontera de Pareto. Con respecto a éste operador, la descendencia se forma incorporando elementos de ambos padres uno a uno alternando elementos de manera aleatoria (figura 2, Offspring). El segundo descendiente se forma con los elementos opuestos que aporta cada padre. La mutación provoca la variación de los valores binarios de uno o más alelos del gen. Esta variación aplicada de manera aleatoria en diferentes puntos de la cadena de ADN produce individuos con pequeñas diferencias con respecto al individuo original. Para nuestro caso se produce la variación de los valores de los alelos para el 10% del cromosoma. En la figura 2 se puede observar el cromosoma mutado (Offspring*).

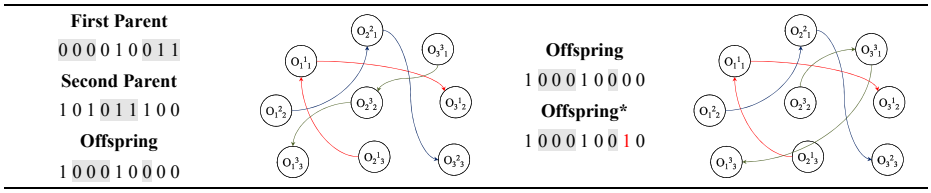


Fig. 2. Cruce y Mutación / Cromosoma de Secuencia

3.2 Selección de los MOEAs

Los algoritmos evolutivos multi-objetivo utilizados fueron: Nondominated Sorting Genetic Algorithm II (NSGAII) [22], Strength Pareto Evolutionary Algorithm 2 (SPEA2) [23] e Indicator-Based Evolutionary Algorithm (IBEA) [24]. Estos algoritmos ya han sido utilizados como referencia en algunas aplicaciones en ingeniería. NSGAII es un algoritmo que incorpora elitismo y reduce la complejidad del procedimiento de ordenamiento rápido por no dominancia de su antecesor NSGA. Realiza una clasificación de la población por frentes. A cada individuo se le asigna un rango equivalente a su nivel de no dominancia. Los mejores individuos son aquellos que tienen rangos menores. Este método de ordenación asegura que las mejores soluciones formarán parte de la población en la siguiente iteración por lo que este algoritmo no necesita de la incorporación de un mecanismo de elitismo externo. Además permite la preservación de la diversidad mediante una técnica de crowding. El algoritmo SPEA2 es una versión revisada de SPEA. Este algoritmo incorpora una estrategia fina de asignación del fitness que considera, para cada individuo, el número de los individuos que lo dominan y el número de los individuos por los cuales es dominado. Además, utiliza la técnica del "vecino más cercano" para la valoración de la densidad, dirigiendo la búsqueda en forma más eficiente. Por último, el algoritmo IBEA, incorpora indicadores de calidad multi-objetivo para guiar la búsqueda, y por lo tanto no se basa directamente en la noción de Pareto dominancia, aunque puede hacerlo de manera indirecta.

4 Implementación y diseño de los experimentos

Para la experimentación se utilizaron tres instancias complejas extraídas de casos reales: C_1 (Problema 15 × 20 con 157 operaciones), C_2 (Problema 20 × 20 con 242

operaciones), C_3 (Problema 20 × 25 con 412 operaciones). Cada una de ellas se experimentó teniendo en cuenta las características del Buffer: No-wait y Not-restricted. Habiendo definido una cantidad adecuada de generaciones para la fase evolutiva y la configuración productiva para las micro-simulaciones, se trabajó con la herramienta PISA (A Platform and Programming Language Independent Interface for Search Algorithms) [25]. Los parámetros y características de los experimento son los que se muestran en la Tabla 2. Para el caso de IBEA se eligió el indicador épsilon aditivo. El resto de los valores de los parámetros se mantuvieron con su valor predefinido en PISA. Cada algoritmo fue corrido 30 veces y se tomaron todas las soluciones eliminándose de este conjunto las dominadas. Cabe mencionar, que el tiempo computacional para todos los casos se mantuvo por debajo de los 30 minutos (Processor: 2.1 GHz AMD Turion X2 Ultra Dual-Core, Memory: 4 GB 800 MHz DDR2).

Tabla 2. Parámetros y características de los experimentos

	C_1 15 × 20 con 157 Op.	**C_2 20 × 20 con 242 Op.**	**C_3 20 × 25 con 412 Op.**
Inicialización	Aleatoria	Aleatoria	Aleatoria
Representación	Binaria	Binaria	Binaria
Número de Genes	20	20	25
Tamaño del Gen	22	62	62
Tamaño del Cromosoma	820	1240	1550
Tamaño de la Población	50	50	100
Generaciones	1000	1000	1000
Tipo de Cruce	Uniforme	Uniforme	Uniforme
Probabilidad de Cruce	0.85	0.85	0.85
Tipo de Mutación	Alterar	Alterar	Alterar
Alelos Mutados	82	124	155
Probabilidad de Mutación	0.05	0.05	0.05
Funciones objetivo	3	3	3
Número de Ejecuciones	30	30	30
Límite del Buffer	0 y 14	0 y 19	0 y 19

5 Resultados

En los siguientes puntos se realizó un análisis de las fronteras de Pareto generadas y una comparación de los tres algoritmos a través de indicadores de calidad.

5.1 Análisis de las Fronteras de Pareto

La primera parte de la experimentación consistió en el análisis de las fronteras de Pareto generadas. Los resultados pueden verse en figuras 3, 4 y 5. Observando f_1 vs. f_2, los frentes generados por los tres algoritmos (Referencia: NSGAII (verde), SPEA2 (violeta) e IBEA (azul)) parecen comportarse de manera similar, aunque IBEA generó en todos los casos el frente completo. La frontera más incompleta la generó NSGAII. Con respecto a la varianza, es decir f_1 vs. f_3, se observó que para el primer problema C_1 (Figura 3, Buffer: 0), los algoritmos SPEA2 e IBEA obtuvieron mejores valores que NSGAII. Para el segundo problema C_1 (Figura 3, Buffer: 14), el algoritmo IBEA obtuvo los mejores valores para la varianza. Para el primer problema C_2 (Figura 4, Buffer: 0), no hay diferencia para los valores obtenidos. Para el segundo problema C_2 (Figura 4, Buffer: 19), el algoritmo IBEA obtuvo los mejores valores para la varianza. Para el primer problema C_3 (Figura 5, Buffer: 0), los algoritmos NSGAII e IBEA obtuvieron mejores valores que SPEA2. Por último, para el segundo problema C_3 (Figura 5, Buffer: 19), los algoritmos NSGAII y SPEA2 obtuvieron mejores valores que el algoritmo IBEA.

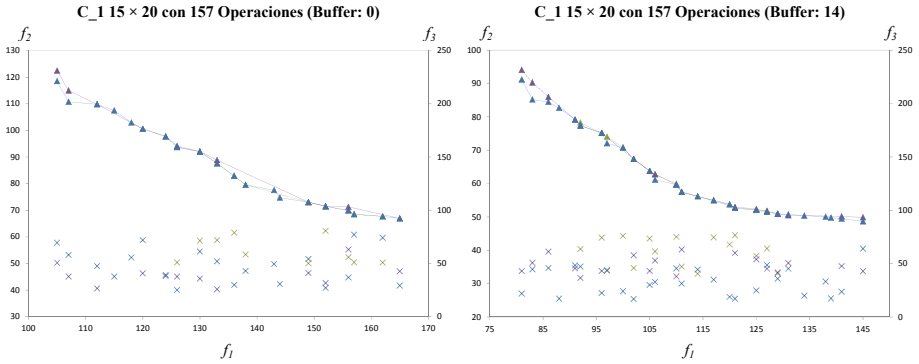


Fig. 3. Frontera de Pareto, f_1 vs. f_2 y f_1 vs. f_3, C_1

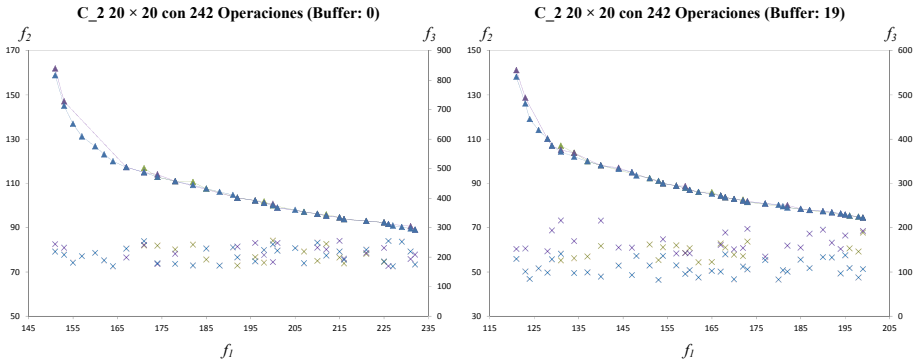


Fig. 4. Frontera de Pareto, f_1 vs. f_2 y f_1 vs. f_3, C_2

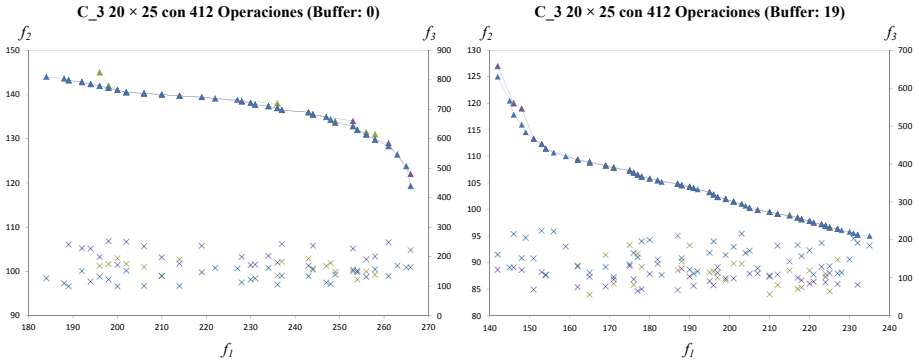


Fig. 5. Frontera de Pareto, f_1 vs. f_2 y f_1 vs. f_3, C_3

5.2 Comparación a través de Indicadores de Calidad

Para analizar con más detenimiento hemos aplicado dos indicadores de calidad bien conocidos que establecen el rendimiento de cada algoritmo: El indicador hipervolumen [26] y el indicador R2 [27]. El primer indicador necesita de un punto de referencia para establecer un área que es dominada por un punto específico de cada vector solución que es gene-

rado por los optimizadores objetos de estudio (cuanto más grande es este indicador mejor es el algoritmo). El segundo indicador estima que tan cerca está la solución generada por un algoritmo sobre el frente real o alguna referencia (cuanto más pequeño es este indicador mejor es el algoritmo). Para las figuras 6 y 7, una observación empírica nos permite inferir que SPEA2 e IBEA parecen comportarse mejor. Ambos algoritmos tienen un desempeño muy parejo. Para la figura 8, la observación nos indica que IBEA tiene mejor rendimiento. Uno de los motivos de que NSGAII no sea eficiente, fue que al tener el problema más de 2 objetivos el operador de crowding no funcionó adecuadamente. Además, como ya es sabido, NSGAII no es eficiente con representaciones binarias.

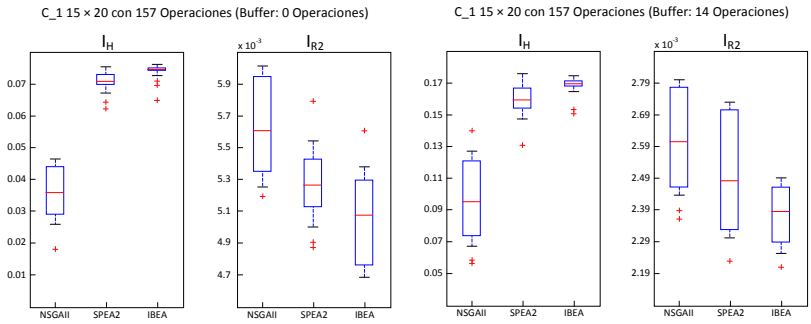


Fig. 6. Boxplot / Valores para NSGAII, SPEA2 e IBEA, C_1

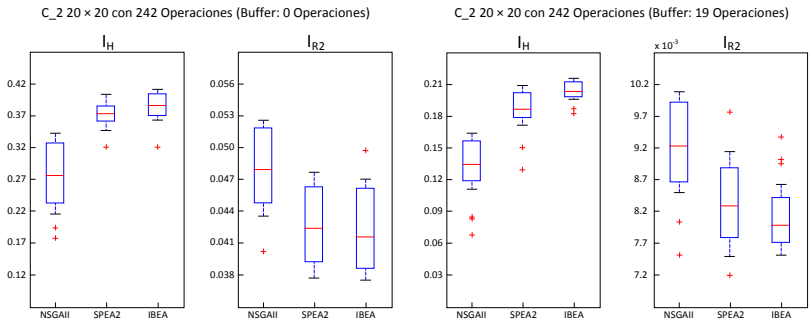


Fig. 7. Boxplot / Valores para NSGAII, SPEA2 e IBEA, C_2

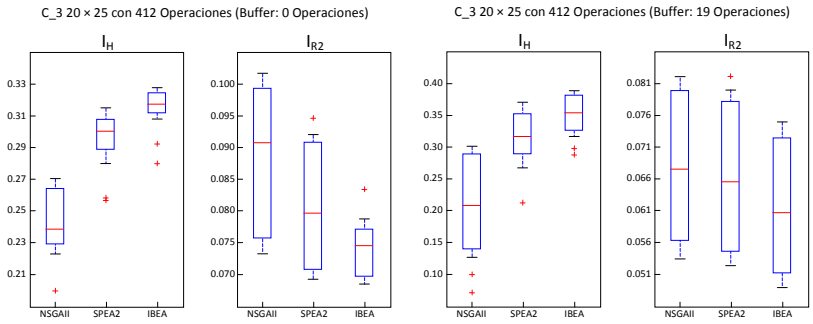


Fig. 8. Boxplot / Valores para NSGAII, SPEA2 e IBEA, C_3

A esta simple observación se le añadió un análisis estadístico paramétrico como es el Test de Fisher. Esta prueba se realizó con un nivel de significancia de 0.05. Para todos los problemas se establecieron diferencias significativas a favor de SPEA2 e IBEA frente a NSGAII. Pese a que en la observación para los problemas C_3 había un mejor comportamiento de IBEA frente a SPEA2, el análisis estadístico no acuso diferencias significativas entre estos dos algoritmos para ninguno de los problemas.

6 Conclusiones

En este artículo se presentó un análisis de diferentes algoritmos evolutivos que tuvieron como objetivo de diseño su uso en el problema de secuenciación de operaciones. La aplicación de los algoritmos requirió fijar parámetros, por lo que los valores son válidos para el problema analizado y constituyen una referencia para problemas similares. Una parte importante del tiempo computacional correspondió a las micro-simulaciones realizadas para estimar la varianza del makespan de cada solución generada no repetida. La comparación realizada entre los algoritmos evolutivos permitió seleccionar a SPEA2 y a IBEA como buenos candidatos. Como parte de trabajos futuros, estamos interesados en evaluar lo propuesto en otros tipos de problemas en el ámbito productivo. Además, se cree trascendente analizar con mayor detalle la varianza ya que se relaciona directamente con la robustez de los otros dos objetivos. Esto es un tema de investigación importante en la Ingeniería Industrial.

Agradecimientos

Los autores agradecen al Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), al Departamento de Ingeniería de la Universidad Nacional del Sur (Proyecto PGI 24/J056), a la Agencia Nacional de Promoción Científica y Tecnológica (Proyecto PICT-2011-0396) y al Departamento de Informática y Sistemas de la Universidad de Las Palmas de Gran Canaria, por el apoyo económico.

Referencias

1. Bihlmaier R., Koberstein A. and Obst R. (2009). Modeling And Optimizing Of Strategic And Tactical Production Planning In The Automotive Industry Under Uncertainty. OR Spectrum. Vol. 31 (2). pp. 311-336.
2. Chao-Hsien J. and Han-Chiang H. (2009). A Hybrid Genetic Algorithm For No-Wait Job Shop Scheduling Problems. Expert Systems with Applications. Vol. 36 (3). pp. 5800-5806.
3. Ullman J. D. (1975). NP-Complete Scheduling Problems. Journal of Computer System sciences. Vol. 10. Pp. 384-393.
4. Chinyao L. and Yuling Y. (2009). Genetic Algorithm-Based Heuristics For An Open Shop Scheduling Problem With Setup, Processing, And Removal Times Separated. Robotics and Computer-Integrated Manufacturing. Vol. 25 (2). pp. 314-322.
5. Goldberg D. (1989). Genetic Algorithms In Search. Optimization, and Machine Learning, Addison Wesley. 1989.
6. Coello Coello C. A., Van Veldhuizen D. A. and Lamont G. B. (2002). Evolutionary Algorithms For Solving Multi-Objective Problems. Kluwer Academic Publishers, New York.
7. Park B. J., Choi H. R. and Kim H. S. A. (2003). Hybrid Genetic Algorithm For The Job-Shop Scheduling Problems. Computers and Industrial Engineering. Elsevier. Vol. 45 (4). pp. 597-613.

8. Frutos M. and Tohmé F. (2012). A Multi-Objective Memetic Algorithm For The Job-Shop Scheduling Problem. Operational Research: An International Journal. Published Online: 25 April 2012.
9. Armentano V. A. and Scrich C. R. (2000). Tabú Search For Minimizing Total Tardiness In A Job Shop. Int. J. Production Economics. Vol. 63. pp. 131-140.
10. Binato S., Hery W. J., Loewenstern D. M. and Resende M. G. C. (2000). A GRASP For Job Shop Scheduling, Essays And Surveys In Meta-Heuristics. Kluwer Academic Publishers, Boston. pp. 59-80.
11. Tsai C. F. and Lin F. C. (2003). A New Hybrid Heuristic Technique For Solving Job-Shop Scheduling Problem. Intelligent Data Acquisition and Advanced Computing Systems. Second IEEE International Workshop.
12. Wu C. G., Xing X. L., Lee H. P., Zhou C. G. and Liang Y. C. (2004). Genetic Algorithm Application On The Job-Shop Scheduling Problem. Machine Learning and Cybernetics, International Conference. Vol. 4. pp. 2102- 2106.
13. Heinonen J. and Pettersson F. (2007). Hybrid Ant Colony Optimization And Visibility Studies Applied To A Job-Shop Scheduling Problem. Applied Mathematics and Computation. Vol. 187 (2). pp. 989-998.
14. Kacem I., Hammadi S. and Borne P. (2002). Approach By Localization And Multi-Objective Evolutionary Optimization For Flexible Job-Shop Scheduling Problems. IEEE Trans. Syst. Man Cybernetics. Vol. 32 (1). pp. 1-13.
15. Ho N. B. and Tay J. C. (2005). Evolving Dispatching Rules For Solving The Flexible Job-Shop Problem. Proc. IEEE Congress on Evolutionary Computation. Vol. 3. pp. 2848-2855.
16. Ho N. B., Tay J. C. and Lai E. M. (2007). An Effective Architecture For Learning And Evolving Flexible Job-Shop Schedules. European Journal of Operational Research. Vol. 179 (2). pp. 316-333.
17. Fattahi P., Saidi M. and Jolai F. (2007). Mathematical Modeling And Heuristic Approaches To Flexible Job-Shop Scheduling Problems. Jounal of Intelligent Manufacturing. Vol. 8 (3). pp. 331-342.
18. Zhang G. and Gen M. (2005). Multistaged-based Genetic Algorithm For Flexible Job-Shop Scheduling Problem. Complexity International. Vol. 11. pp. 223-232.
19. Pezzella F., Morganti G. and Ciaschetti G. (2008). A Genetic Algorithm For The Flexible Job-Shop Scheduling Problem. Journal Computers and Operations Research. Vol. 35 (10). pp. 3202-3212.
20. Chiang T. C. and Lin H. J. (2013). A Simple And Effective Evolutionary Algorithm For Multiobjective Flexible Job Shop Scheduling. International Journal of Production Economics. Vol. 141 (1). pp. 87-98.
21. Evans J. R. and Olson D. L. (1998). Introduction To Simulation And Risk Analysis. Editorial Prentice Hall. New Jersey.
22. Deb K., Pratap A., Agarwal S. and Meyarivan T. (2002). A Fast And Elitist Multi-Objective Genetic Algorithm: NSGAII. IEEE Transactions on Evolutionary Computation. Vol. 6 (2). pp. 182-197.
23. Zitzler E., Laumanns M. and Thiele L. (2002). SPEA2: Improving The Strength Pareto Evolutionary Algorithm For Multi-Objective Optimization, In Giannakoglou, Tsahalis, Periaux, Papailiou, and Fogarty (eds), Evolutionary Methods for Design, Optimisations and Control. pp. 19-26.
24. Zitzler E. and Künzli S. (2004). Indicator-Based Selection In Multi-objective Search. In: X. Yao, E. K. Burke, J. Lozano, J. Smith, J. Merelo Guervós, J. Bullinaria, J. Rowe, P. Tiño, A. Kabán, H-P. Schwefel (Eds.): PPSN VIII. LNCS (Springer). Vol. 3242. pp. 832-842.
25. Bleuler S., Laumanns M., Thiele L. and Zitzler E. (2003). PISA: A Platform And Programming Language Independent Interface For Search Algorithms. Evolutionary Multi-Criterion Optimization. pp. 494-508.
26. Zitzler E. and Thiele L. (1998). Multi-Objective Optimization Using Evolutionary Algorithms, A Comparative Case Study. In Parallel Problem Solving from Nature V., A. E. Eiben, Ed., Amsterdam. Springer-Verlag. pp. 292-301.
27. Hansen M. P. and Jaszkiewicz A. (1998). Evaluating The Quality Of Approximations To The Non-dominated Set. Tech. Rep. IMM-REP-1998-7. Institute of Mathematical Modelling Technical University of Denmark.

Híbrido basado en VNS para la Planificación Real de Rutas de Vehículos con Ventanas de Tiempo

Jesica de Armas, Belén Melián-Batista, Julio Brito y José A. Moreno-Pérez*

Dpto. de Estadística, I.O. y Computación
Escuela Técnica Superior de Ingeniería Informática
Universidad de La Laguna
38271 La Laguna, Spain
{jdearmas,mbmelian,jbrito,jamoreno}@ull.es

Resumen Este trabajo aborda el problema de rutas de vehículos con ventanas de tiempo, considerando un conjunto de restricciones adicionales que se presentan en una situación real. Se ha propuesto un procedimiento metaheurístico para resolver este problema particular, y se ha integrado en el sistema de gestión de flotas de la empresa. Los resultados computaciones indican que el método propuesto tiene un comportamiento competitivo incluso con las instancias de la literatura que no presentan estas restricciones. Para problemas reales con todas las restricciones, se ha comprobado la influencia que estas ejercen sobre la planificación final de rutas.

Keywords: Problema de Rutas de Vehículos, Ventanas de Tiempo, Restricciones Reales, Metaheurísticas, VNS.

1. Introducción

El problema de rutas de vehículos con ventanas de tiempo (VRPTW - Vehicle Routing Problem with Time Windows) es una parte crucial en los sistemas de transporte y logística. Este problema se puede definir como el diseño de rutas para una flota de vehículos que ha de servir a un conjunto de clientes. Cada ruta comienza y termina en el depósito y cada cliente se visita sólo una vez por un único vehículo. Las rutas deben respetar las capacidades de los vehículos y las restricciones de tiempo que vienen dadas en forma de ventanas de tiempo. Habitualmente, los objetivos primarios de este tipo de problemas son minimizar el número de vehículos para realizar las rutas y minimizar la distancia total recorrida. Sin embargo, en aplicaciones reales como la tratada en este trabajo incorpora nuevos objetivos y restricciones adicionales que se han de combinar para proporcionar una solución adecuada.

Los métodos exactos propuestos en la literatura para el VRPTW tienen importantes limitaciones en cuanto al tamaño de las instancias que pueden afrontar.

* Este trabajo ha sido parcialmente financiado por el Ministerio de Economía y Competitividad (proyecto TIN2012-32608) y el Ministerio de Industria, Turismo y Comercio (proyecto TSI-020100-2011-298).

Por ello, las investigaciones más recientes se centran en proponer heurísticas y metaheurísticas. Las heurísticas propuestas se pueden clasificar en tres tipos. Por un lado se encuentran las heurísticas constructivas, que a su vez se dividen en secuenciales [13,19], y paralelas [17], según el modo en el que se inicie la creación de las rutas. Por otro lado, las heurísticas de mejora empiezan con una solución factible y luego se inicia una búsqueda local mediante modificaciones de la solución actual [3,7]. Finalmente, las heurísticas mixtas incluyen simultáneamente procedimientos de construcción y mejora de rutas [14]. Mediante el uso de metaheurísticas para la resolución de VRPTW se han conseguido incluso mejores resultados. Así se han propuesto búsquedas locales [1,11], búsquedas tabú [3,6], algoritmos evolutivos [2], algoritmos meméticos [15] y multitud de algoritmos híbridos [21,22]. Algunos trabajos incluyen algunas adicionales y realizan un enfoque multi-objetivo del problema [5,16,20].

El presente trabajo se incoropran al VRPTW nuevos objetivos y restricciones adicionales atendiendo a necesidades planteadas por empresas de distribución y se aborda el problema utilizando una heurística constructiva y una búsqueda por entorno variable. En este trabajo se tratan por primera vez de forma conjunta estas restricciones reales propuestas por las empresas. El algoritmo metaheurístico propuesto se ha integrado en un sistema de gestión de flotas de vehículos que utilizan dichas empresas. Este sistema de gestión de flotas es proporcionado por un proveedor tecnológico que es el nexo entre las empresas y esta investigación. Después de comprobar que se obtienen buenos resultados incluso para las intancias presentes en la literatura que no incluyen ninguna de las restricciones adicionales, se han realizado los correspondientes experimentos computacionales sobre una instancia real, mostrando los efectos que las restricciones tienen sobre el plan de rutas obtenido.

El resto del artículo se organiza de la siguiente manera. La sección 2 describe el problema real propuesto por las empresas. La sección 3 resume el método metaheurístico que se ha utilizado para resolver el problema. La sección 4 proporciona los resultados de los experimentos que se han realizado en este trabajo. Finalmente, las conclusiones se presentan en la sección 5.

2. Descripción del problema

El problema de ruta de vehículos real tratado en este trabajo se formula sobre una red que contiene un depósito y un conjunto de n clientes. Cada cliente i requiere de un servicio, y por lo tanto tiene asociada su demanda, su localización, el tiempo de duración del servicio, y una ventana de tiempo que puede estar dividida en intervalos.

En el depósito, que también tiene asociada una ventana de tiempo, se dispone de un conjunto de m vehículos con diferentes capacidades por lo que se trata de un VRPTW con flota heterogénea (HFVRPTW, Heterogeneous Fleet Vehicle Routing Problem with Time Windows). Además, cada vehículo j también tiene su ventanas de tiempo que puede estar divida en intervalos que representan los turnos de trabajo y pueden ser diferentes para cada vehículo. Cada vehículo parte

del depósito y debe regresar a él dentro de la ventana de tiempo del depósito y dentro de uno de los intervalos de su propia ventana de tiempo.

A cada cliente se le asigna una prioridad lo que permitirá posponer el servicio de los clientes de menor prioridad para otro día en caso de no ser viable atenderlos a todos. Cada día se fija un nivel de prioridad de manera que todos los clientes que tengan una prioridad inferior podrán ser pospuestos en caso de necesidad. Algunos vehículos especificados pueden realizar horas extras por lo que sus ventanas de tiempo podrán ser ampliadas para que éstos empiecen antes o terminen después de los límites de los intervalos. De este modo, cada vehículo tendrá también un coste asociado a las horas normales de conducción y un coste distinto asociado a las horas extras de conducción. Finalmente, los vehículos consumen una cantidad de combustible por kilómetro recorrido y esa cantidad puede ser diferente para cada uno.

Debido a la incompatibilidad entre algunos clientes y vehículos, se han agregado tres tipos de restricciones, de manera que si un vehículo y un cliente tienen activa la misma restricción, ese vehículo no podrá servir a ese cliente. Por ejemplo, si algún tipo de vehículo no pasar por las carreteras necesarias para llegar hasta el cliente.

Con toda esta información y la que proporciona la empresa con sus sistemas de gestión de flotas acerca de los tiempos y distancias que existen entre clientes, se ha de obtener una planificación óptima de rutas. Esta planificación final estará formada por una serie de k rutas $P = \{r_1, ..., r_k\}$ y tendrá asociado un número de clientes pospuestos, un número de horas extra empleadas, una distancia total recorrida, un número de rutas empleadas, un balanceo de distancia, un balanceo de tiempo, un consumo total y un coste total. El número de clientes pospuestos NPC corresponde con el número total de clientes que no son visitados. El número de horas extra NEH empleadas es la suma de las horas extra empleadas en cada ruta. La distancia total TD es la suma de las distancias recorridas por cada vehículo de cada ruta. El balanceo de distancia DB se calcula como la diferencia de distancias recorridas por el vehículo que realiza la mayor distancia y por el que realiza la menor. El balanceo de tiempo TB es la diferencia de tiempo empleado entre el vehículo que utiliza más tiempo y el que menos. Por último, el coste total tiene dos componentes; la suma de los consumos de cada ruta en horas normales CN y la suma de los costes asociados las horas extra CS.

Para las empresas interesadas en esta optimización, existen tres tipos de criterios, en función del objetivo principal que se tenga en cuenta a la hora de optimizar: 1) la distancia total recorrida TD, 2) el balanceo de tiempo entre rutas utilizando toda la flota de vehículos TB y 3) el balanceo de distancia entre rutas utilizando toda la flota de vehículos DB. Adicionalmente, se requiere que se tengan en cuenta otros criterios como: el número de clientes pospuestos NPC, el número de horas extra NEH, el número de rutas K, el consumo CN, y el coste CS. Para cada uno de los tres criterios anteriores se fija una jerarquía de objetivos que son tratados siguiendo un orden lexicográfico dinámico [4]. Si el objetivo principal es 1) la distancia total TD, los otros objetivos a minimizar y su orden son NPC, NEH, TD, K, TB, CN y CS. Si el objetivo principal

Algoritmo 1: Pseudocódigo del algoritmo general

```
1 Inicializar la solution S*;
2 while (no se alcance un número máximo de iteraciones) do
3 |   S ← Ejecutar Heurística de Solomon adaptada;
4 |   S' ← Aplicar la VNS propuesta por De Armas et al. [8] a S;
5 |   if (S' es mejor que S*) then
6 |   |   S* ← S';
```

es 2) el balanceo de tiempos TB, los otros objetivos a minimizar y su orden son: NPC, NEH, TB, CN y CS. Si el objetivo principal es 3) el balanceo de distancias DB, los objetivos a minimizar y su orden son: NPC, NEH, DB, CN y CS. Sin embargo, en todos los casos, el número de clientes pospuestos y el número de horas extra se evaluan antes que los criterios principales, ya que se le da prioridad a obtener una planificación de rutas que incluya todos los clientes y sin necesidad de horas extras, es decir, que cumpla con los requisitos especificados inicialmente.

3. Resolución basada en metaheurísticas

El método propuesto en este trabajo para resolver el HFVRPTW con las restricciones agregadas por las empresas se resume en el Algoritmo 1, que actúa de la siguiente manera. En primer lugar, se obtiene una solución inicial mediante el uso de la heurística de Solomon [19] adaptada para tener en cuenta las particularidades del problema (Algoritmo 2). A continuación, se mejora la solución ejecutando el algoritmo de búsqueda por entornos variables (VNS - Variable Neighborhood Search) propuesto en [8]. Este proceso (líneas $3-6$) se repite un cierto número de iteraciones y se propone a la empresa la mejor solución alcanzada.

La heurística de Solomon puede utilizar 'vehículos ficticios' si faltan clientes por agregar a las rutas. Luego, la metaheurística VNS podrá reducir el número de rutas de la solución aportada por la heurística de Solomon.

Los vehículos ficticios tienen las características menos restrictivas posibles en cuanto a ventana de tiempo y capacidad con el objetivo de poder servir con mayor facilidad un mayor número de clientes. Posteriormente, en la VNS se tendrá como prioridad vaciar las rutas de los vehículos ficticios, moviendo los clientes a otras rutas con vehículos reales. Si se consigue se habrá obtenido un plan de rutas válido, pero en caso contrario el plan se descartará.

La VNS utilizada, cuyos detalles se tratan en profundidad en [8], sigue el esquema general de tres fases: agitación (*shaking*), búsqueda local (*local search*) y movimiento (*move*) [10]. En la fase de agitación se selecciona aleatoriamente una solución s' del k-ésimo entorno de la solución actual. Luego se aplica la búsqueda local para encontrar el mejor vecino $s*$ de s'. Si $f(s*) < f(s)$, s se reemplaza por $s*$ en la fase de movimiento y entonces se repite todo el esquema.

Algoritmo 2: Heurística de Solomon modificada

```
1  S ← ∅;
2  while (queden clientes por enrutar) do
3      if (no quedan vehículos disponibles) then
4          Agregar un vehículo ficticio;
5      Escoger el vehículo disponible de mayor capacidad;
6      Crear una ruta vacía r;
7      Escoger un cliente c entre los dos más lejanos del depósito;
8      if (es viable insertar c en r) then
9          Insertar el cliente c en r;
10     if (no se ha insertado el cliente c en r y se permiten las horas extra) then
11         Expandir las ventanas del vehículo con horas extra;
12         Escoger un cliente c entre los dos más lejanos del depósito;
13         if (es viable insertar c en r) then
14             Insertar el cliente c en r;
15     if (no se ha insertado el cliente c y el vehículo no es ficticio) then
16         Agregar un vehículo ficticio;
17         Asignar el vehículo ficticio a r;
18         Escoger un cliente c entre los dos más lejanos del depósito;
19         if (es viable insertar c en r) then
20             Insertar el cliente c en r;
21     if (no se ha insertado el cliente c) then
22         if (prioridad dos clientes más lejanos del depósito < prioridad mínima)
           then
23             Posponer esos clientes;
24             continue;
25         else
26             ERROR: Ningún vehiculo puede llegar a un servicio restante
               cumpliendo sus horarios y los del depósito;
27     while (se puedan insertar servicios en r) do
28         for (c ∈ clientes sin enrutar) do
29             Calcular la mejor posición donde se podría insertar c dentro de r;
30         Escoger el mejor cliente a insertar e insertarlo en su mejor posición en r;
31     S ← S ∪ r;
32 return S;
```

En otro caso, k se incrementa y se empieza una nueva fase de agitación utilizando otra estructura de entorno, hasta un cierto número de veces. Para establecer los entornos se han utilizado los movimientos y el orden de los mismos fijados en [18] como la mejor opción: GENI, Or-Opt, Cross, 2-Opt, 0-1 Relocate y 1-1 Interchange. Cuando se consigue alguna mejora con algún entorno se vuelve a comenzar desde el primero. En la fase de búsqueda local se han utilizado los

movimientos: 0-1 Relocate, 1-1 Interchange y Swap. La solución escogida será la mejor obtenida por estos tres movimientos.

Mediante unos previos se fijaron algunos parámetros de la heurística constructiva de Solomon y la VNS de mejora. Para que el proceso constructivo proporcione una solución diferente en cada ejecución, se ha utilizado un parámetro x para la selección del primer cliente que se inserta en una ruta. Se selecciona el cliente semilla entre los x más lejanos del depósito que aún no se hayan incluido en ninguna ruta; en la propuesta de Solomon se escoge siempre el más lejano al depósito. A partir de los experimentos previos, el valor de este parámetro ha sido fijado en 2, dados los buenos resultados. La VNS usa otros parámetros similares que también han sido ajustado en base a experimentos previos. De esta forma, para realizar movimientos de clientes entre rutas, la ruta de destino se selecciona entre las 3 rutas más cercanas a la ruta desde la que se mueve el cliente. El cliente a eliminar de una ruta se selecciona entre los 3 que se puedan eliminar factiblemente más cercanos a la ruta en la que se va a insertar. El tercer parámetro se utiliza para seleccionar El cliente de una ruta después del cual se puede insertar un cliente previamente escogido se selecciona entre los 3 factibles más cercanos al cliente previamente escogido. Estas medidas de cercanías se han realizado tomando como referencia los centroides de las rutas. También se fijó experimentalmente en 20 el número de soluciones de cada entorno que se usan en la agitación antes de dar por terminada su ejecución si no hay mejora. De esta forma el número de búsqueda locales que se ejecutan es de al menos 120. Finalmente todo el proceso consiste en la aplicación del algoritmo de Solomon seguido de la VNS 10 veces, dada la buena relación entre la calidad de la solución y el tiempo necesario para obtenerla. Para este ajuste de parámetros preliminar se ha realizado mediante el uso del test de Friedman, que proporciona la mejor configuración de parámetros y detecta diferencias significativas entre configuraciones.

4. Experiencia computacional

Para analizar el rendimiento de la metaheurística propuesta se han realizado dos experimentos computacionales. El primer experimento estudiamos los resultados obtenidos por nuestra propuesta en el VRPTW estándar con las instancias clásicas de Solomon [19]. Estos resultados se han comparado con los mejores resultados conocidos en la literatura [9,12]. Es necesario destacar que, el algoritmo propuesto en este trabajo está diseñado para resolver el problema VRPTW incluyendo todas las restricciones reales que se han explicado anteriormente. Por esa razón, al método no se le debería exigir ser tan competitivo para el problema y las instancias estándares de la literatura, particularmente debido al hecho de que las instancias reales tienes diferentes características. La Tabla 1 resume la comparativa, en la que se muestra la media de los valores del número de vehículos (NV) y la distancia total (TD) en cada categoría de instancias. La primera columna de la tabla indica la categoría de las instancias de Solomon. En el peor de

los casos, la desviación mostrada en la última columna es de aproximadamente un 7,6 %, lo cual es reseñable.

Tabla 1. Resultados computacionales para las instancias de Solomon

	Algoritmo		Mejor		
	NV	TD	NV	TD	Desv (%)
$C1$	10.00	835.88	10.00	828.38	0.94
$C2$	3.12	621.01	3.00	589.90	4.28
$R1$	14.08	1252.41	11.91	1203.16	4.28
$R2$	5.00	988.26	3.00	941.87	6.00
$RC1$	13.62	1402.50	12.00	1345.56	7.59
$RC2$	6.00	1127.62	3.62	1111.99	3.86

En segundo lugar, ya que el algoritmo considera muchas restricciones diferentes y las instancias de la literatura no las tienen en cuenta, se han realizado pruebas con una instancia real proporcionada por una empresa de las Islas Canarias. Dicha instancia consta de 100 clientes y se encuentra disponible en *https://sites.google.com/site/gciports/vrptw/HFVRPTW*.

Se ha podido comprobar que dependiendo de cómo se modifique el conjunto de restricciones, la planificación obtenida para el mismo conjunto de clientes se ve claramente afectada. Primero se han alterado las características de la flota de vehículos. Si se modifica la capacidad de los mismos, pasando de tener vehículos de bastante capacidad a tener vehículos con una capacidad más pequeña, se puede observar que el número de vehículos necesario para realizar las rutas se incrementa tal y como cabría esperar. Asimismo, la distancia total se ve incrementada como consecuencia de ello (Tabla 2). Si se utiliza una flota de vehículos con ambos tipos de capacidades, se obtiene un resultado intermedio.

Si el aspecto que se modifica de los vehículos es la amplitud de sus ventanas de tiempo, haciendolas más restrictivas, se obtienen los resultados de la Tabla 3. Como se puede observar en la primera columna, si se permiten las horas extra, éstas aparecerán como resultado de los períodos tan cortos de trabajo. Si en lugar de permitir las horas extra, se establecen prioridades que permitan posponer clientes para otro día, surgirán clientes que se han de posponer, como muestra la segunda columna de la tabla, al no ser posible servirlos a todos con ventanas de trabajo tan cortas.

Tabla 2. Vehículos de distinta capacidad

	Mucha capacidad	*Poca capacidad*	*Mezcla capacidades*
Distancia (m)	989247	1214530	1098950
Rutas	7	11	10

Tabla 3. Vehículos con ventanas restrictivas

	Horas extra permitidas	*Horas extras no permitidas*
Distancia (m)	1111560	1027710
Rutas	10	8
Tiempo extra (s)	2039	0
Clientes pospuestos	0	3

Tabla 4. Clientes con ventanas amplias o estrechas

	Ventanas no restrictivas	*Ventanas restrictivas*
Distancia (m)	1032300	1664520
Rutas	7	10
Clientes pospuestos	0	5

Tabla 5. Restricción en clientes 25 y 73, y en vehículos

	Vehículos con restricción	*Vehículos con restricción, excepto uno*
Distancia (m)	1064450	1060340
Rutas	7	7
Clientes pospuestos	2	0

Posteriormente se han alterado las características de las ventanas de tiempo de los clientes. Cuando dichas ventanas son muy restrictivas, ocurre lo que muestra la Tabla 4 se necesitan más rutas para poder servir a todos los clientes a tiempo, aumenta consecuentemente la distancia total recorrida e incluso se han de posponer algunos clientes que no se pueden atender.

Por último, para analizar el efecto de las restricciones entre vehículos y clientes, se han estudiado dos casos. Uno de ellos es que todos los vehículos tengan una restricción que algunos clientes también tengan. En ese caso, tal y como se puede observar en la Tabla 5, dichos clientes tendrán que ser pospuestos. Pero si al menos un vehículo no tiene esa restricción, los clientes serán servidos por él.

Para finalizar la experimentación, se ha integrado el algoritmo en el sistema utilizado por la empresa, pudiendo realizar pruebas de visualización de rutas sobre el mapa, como se puede apreciar en la Figura 1. En dicha imagen se puede ver claramente cómo se han separado los clientes en dos rutas para cubrir dos áreas completamente diferentes.

5. Conclusiones

Este trabajo trata una variante del VRPTW, que contempla las restricciones reales requeridas por empresas. Desde nuestro conocimiento, este conjunto de restricciones consideradas conjuntamente en este problema no se han tratado antes en trabajos de la literatura. Como método de resolución se ha propuesto

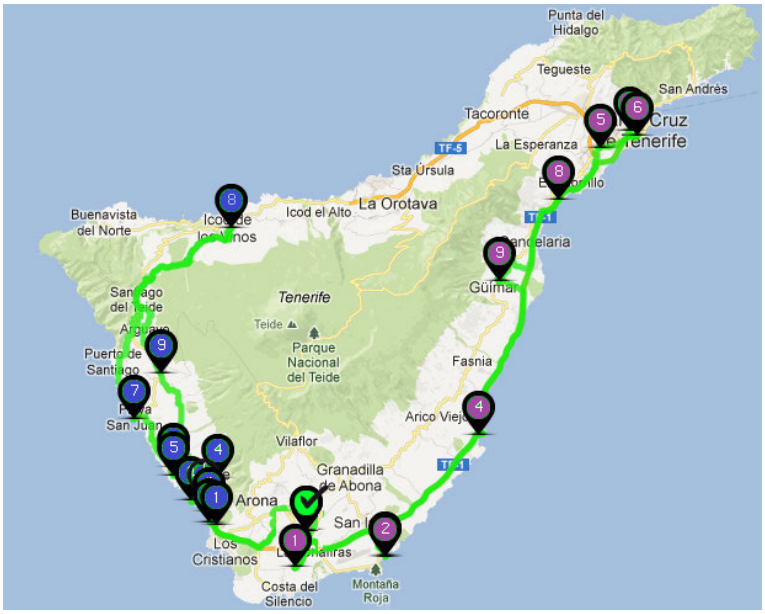


Figura 1. Rutas sobre el mapa

una metaheurística. Su implementación permite la activación y desactivación de las distintas restricciones, obteniendo así un método de resolución flexible. Además, el método se ha integrado en la herramienta de optimización de un sistema de gestión de flotas. La experiencia computacional llevada a cabo en este trabajo ha permitido corroborar el buen comportamiento de la propuesta de resolución cuando se aplica a las instancias más conocidas en la literatura para el problema estándar. Además, se ha probado su funcionamiento sobre un conjunto de instancias generadas a partir de datos reales, permitiendo así activar y desactivar ciertas restricciones y ver el efecto de éstas sobre la solución obtenida. Como trabajo futuro se abordará el problema dinámico VRPTW en el que es posible intercambiar clientes entre rutas.

Referencias

1. O Bräysy. A reactive variable neighborhood search for the vehicle routing problem with time windows. *INFORMS Journal on Computing*, 15:347–368, 2003.
2. O. Bräysy, W. Dullaert, and M. Gendreau. Evolutionary algorithms for the vehicle routing problem with time windows. *Journal of Heuristics*, 10(6):587–611, 2004.
3. O. Bräysy and M. Gendreau. Tabu Search heuristics for the Vehicle Routing Problem with Time Windows. *TOP: An Official Journal of the Spanish Society of Statistics and Operations Research*, 10(2):211–237, 2002.
4. J. Castro-Gutierrez, D. Landa-Silva, and J.A. Moreno-Pérez. Improved Dynamic Lexicographic Ordering for Multi-objective Optimisation. In *Parallel Problem Sol-*

ving from Nature - PPSN XI, Lecture Notes in Computer Science, volume 6239, pages 31–40, 2010.
5. J. Castro-Gutierrez, D. Landa-Silva, and J.A. Moreno-Pérez. Nature of real-world multi-objective vehicle routing with evolutionary algorithms. In Anchorage Alaska IEEE Press, editor, *Proceedings of the 2011 IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC 2011)*, pages 257–264, 2011.
6. J.F. Cordeau, G. Laporte, E. Hautes, Études Commerciales, and Les Cahiers Du Gerad. A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operationa Research Society*, 52:928–936, 2001.
7. R. Cordone and R.W. Calvo. A Heuristic for the Vehicle Routing Problem with Time Windows. *Journal of Heuristics*, 7(2):107–129, Marh 2001.
8. J. De Armas, B. Melián-Batista, J.A. Moreno-Pérez, and J. Brito. Real-World Heterogeneous Fleet Vehicle Routing Problem with Soft Time Windows. *Under review*, 2013.
9. Y-J. Gong, J. Zhang, O. Liu, R-Z. Huang, H.S.-H. Chung, and Y. Shi. Optimizing the Vehicle Routing Problem With Time Windows: A Discrete Particle Swarm Optimization Approach. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 42(2):254–267, 2012.
10. P. Hansen, N. Mladenović, and J.A. Moreno Pérez. Variable Neighbourhood Search: methods and applications. *Annals of Operations Research*, 175(1):367–407, 2010.
11. H. Hashimoto, M. Yagiura, and T. Ibaraki. An iterated local search algorithm for the time-dependent vehicle routing problem with time windows. *Discrete Optimization*, pages 434–456, 2008.
12. L. Hong. An improved LNS algorithm for real-time vehicle routing problem with time windows. *Computers & Operations Research*, 39(2):151–163, Feb 2012.
13. G. Ioannou, M. Kritikos, and G. Prastacos. A greedy look-ahead heuristic for the vehicle routing problem with time windows. *Journal of The Operational Research Society*, 52:523–537, 2001.
14. G. Kontoravdis and J.F. Bard. A grasp for the vehicle routing problem with time windows. *INFORMS Journal on Computing*, 7(1):10–23, 1995.
15. N. Labadi, C. Prins, and M. Reghioui. A memetic algorithm for the vehicle routing problem with time windows. *RAIRO - Operations Research*, 42:415–431, 6 2008.
16. B. Ombuki, B.J. Ross, and F. Hanshar. Multi-objective Genetic Algorithms for Vehicle Routing Problem with Time Windows. *Applied Intelligence*, 24:2006, 2006.
17. K-W. Pang. An adaptive parallel route construction heuristic for the vehicle routing problem with time windows constraints. *Expert Syst. Appl.*, 38(9):11939–11946, Sep 2011.
18. P.P. Repoussis, D. C. Paraskevopoulos, C.D. Tarantilis, and G. Ioannou. A Reactive Greedy Randomized Variable Neighborhood Tabu Search for the Vehicle Routing Problem with Time Windows. 4030:124–138, 2006.
19. M.M. Solomon. Algorithms for the Vehicle-Routing and Scheduling Problems with Time Window Constraints. *Operations Research*, 35(2):254–265, Mar-Apr 1987.
20. K.C. Tan, Y.H. Chew, and L.H. Lee. A hybrid multi-objective evolutionary algorithm for solving truck and trailer vehicle routing problems. *European Journal of Operational Research*, 172(3):855–885, 2006.
21. T. Vidal, T.G. Crainic, M. Gendreau, and C. Prins. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers & Operations Research*, 40(1):475–489, 2013.
22. B. Yu, Z.Z. Yang, and B.Z. Yao. A hybrid algorithm for vehicle routing problem with time windows. *Expert Systems with Applications*, 38(1):435–441, 2011.

PSO Discreto para el VRPTW orientado a la calidad del servicio

A. Expósito, J. Brito y J.A. Moreno*

Instituto Universitario de Desarrollo Regional
Universidad de La Laguna
38271 La Laguna, Spain
{airamexpmar}@gmail.es,{jbrito,jamoreno}@ull.es

Resumen En este artículo proponemos una nueva variante del Problema de Rutas de Vehículos con Ventanas de Tiempo, orientado a la calidad del servicio a los clientes. En este modelo el objetivo incorpora una medida de calidad asociada a que los vehículos lleguen a servir a los clientes tan pronto como sea posible dentro de la ventana de tiempo. Para la obtención de soluciones implementamos una metaheurística bioinspirada basada en una modificación de la Optimización por Enjambre de Partículas para abordar problemas discretos como el propuesto. Se evalúa el comportamiento de este procedimiento ajustando los parámetros y se llevan a cabo algunas comparaciones con otro modelo similar.

Keywords: Problemas de Rutas de Vehículos, Ventanas de Tiempo, Calidad del Servicio, Optimización Discreta por Enjambre de Partículas

1. Introduction

La planificación de rutas de distribución juega un importante papel en la gestión de la cadena de suministro, entre otras razones porque mejora la calidad del servicio, la satisfacción del cliente y reduce los costes. El Problema de Rutas de Vehículos (VRP-Vehicle Routing Problems) es un problema bien conocido cuya finalidad es encontrar el mejor conjunto de rutas para una flota de vehículos que sirva la demanda de bienes de un conjunto de clientes geográficamente dispersos [16], [7]. Un objetivo estándar de este problema es determinar el conjunto de rutas factibles para los vehículos disponibles que minimice el coste total de operaciones de la flota, en general expresados en distancias o tiempos. El Problema de Rutas de Vehículos con Ventanas de Tiempo (VRPTW-Vehicle Routing Problem with Time Windows) es una variante del VRP donde cada cliente tiene que ser servido por un vehículo dentro de unos límites de tiempo establecidos [10].

El objetivo más común en las empresas es minimizar los costes, sin embargo con la alta competitividad de los mercados actuales, estratégicamente están

* Este trabajo ha sido parcialmente financiado por el Ministerio de Economía y Competitividad (proyecto TIN2012-32608) y el Ministerio de Industria, Turismo y Comercio (proyecto TSI-020100-2011-298).

poniendo el foco de atención en la calidad y satisfacción de la prestación de servicios al cliente. La reducción de los tiempos en los que se satisface la demanda es uno de los factores que repercute en el incremento de la calidad del servicio y por tanto una cuestión importante a tener en cuenta en la gestión de la cadena de suministro. En este sentido y como resultado, la planificación de rutas de distribución debe considerar entres sus objetivos este criterio, la mejora de los tiempos de servicio a cada cliente. En el caso de que el cliente haya establecido una ventana de tiempo como intervalo aceptable para ser servido, una referencia lógica podría ser que el tiempo en el que vehículo llega al cliente para servirle sea lo antes posible dentro de estos límites.

En este trabajo consideramos un VRPTW bi-objetivo, cuyos objetivos están asociados a los costes operacionales y a la calidad del servicio al cliente. Los costes operacionales se miden utilizando la distancia total recorrida y/o el total del tiempo que la flota de vehículos tarda en cubrir todas rutas. La calidad del servicio es evaluada con el total de las diferencias entre el tiempo en el que el vehículo llega a servir al cliente y el inicio o cierre de su ventana de tiempo. Esta variante la vamos a denominar como VRPTW orientado a la calidad del servicio (QSVRPTW-Quality Service-oriented VRPTW). En la literatura podemos encontrar modelos del VRP cuya finalidad es responder con prioridad a la satisfacción de las necesidades de los clientes. Estos atienden a problemas como, el suministro urgente de bienes, la logística en desastres naturales, la recogida de niños para ir al colegio o los servicios de mantenimiento, los cuales pueden formalizarse con objetivos y restricciones similares al modelo propuesto. Entre otros modelos podemos encontrar, el problema del viajante de reparaciones con multiples rutas (m-TVRP, m-Traveling Repairman Problem) cuyo objetivo es minimizar el tiempo total de latencia, retardo o llegada a un cliente [9]; el problema de rutas de autobuses escolares [6] entre cuyos objetivos está el minimizar las medias de distancia de viaje, o la media de tiempo utilizado por los escolares en llegar al centro educativo; y el problema de rutas de vehículos capacitado y acumulativo (CCVRP-Cumulative Capacitated Vehicle Routing Problem), cuyo objetivo es minimizar la suma de los tiempos en los que el vehículo llega a servir a cada cliente, [13], [12], [4]. Dada la orientación dada a estos modelos la incorporación de restricciones de ventana de tiempo no está considerada y así no aparece ninguno en la literatura con tales restricciones. En estos modelos, de forma similar al nuestro, los vehículos deben llegar a todos los clientes tan pronto como sea posible, en nuestro modelo específicamente, se toma como referencia lo antes que llega dentro de la ventana de tiempo.

Para encontrar soluciones a este problema utilizando el modelo propuesto, esta investigación considera la aplicación de una versión discreta de la Optimización por Enjambre de Partículas (PSO- Particle Swarm Optimization) denominada Optimización por el Salto de la Rana (JFO, Jumping Frog Optimization)[11],[5].

El resto de este artículo está organizado a continuación como sigue. La siguiente sección describe el modelo propuesto, el QSVRPTW. En las sección 3 presentamos la metaheurística PSO discreta implementada, la JFO utilizada para resolver el modelo propuesto. En la sección 4 describimos los resultados de

la experimentación llevada a cabo. Finalmente, la sección 5 propone algunas conclusiones y líneas de futuros trabajos.

2. Formulación del modelo VRPTW orientado a la calidad del servicio

El problema de rutas de vehículos,VRP o VRP con capacidad limitada (CVRP, Capacitated VRP) consiste en, teniendo una flota de vehículos con capacidad limitada que tiene que servir la demanda a un conjunto geográficamente disperso de clientes, determinar el conjunto de m rutas cuyo coste total sea mínimo tal que: (a) cada cliente es visitado exactamente una vez por un solo vehículo asignado a la ruta, (b) cada ruta comienza y finaliza en un único depósito o almacén, y (c) el total de la demanda de los clientes servida por una ruta no excede de la capacidad del vehículo Q_k.

Una de las variantes más extendidas del VRP se origina al incluir periodos de tiempo limitados durante los cuales realizar el servicio a los clientes. En concreto el VRPTW es un VRP con la restricción adicional que asocia una ventana de tiempo a cada cliente en la que solo está dispuesto a recibir los bienes o servicios. Si un vehículo llega al cliente antes de tiempo el vehículo debe esperar, si llega en el intervalo de la ventana de tiempo, el vehículo suministra la demanda en el momento de la llegada y si llega con posterioridad el cliente queda sin atender. El VRPTW es un modelo clásico que, dada la aplicación real que tiene en la planificación de rutas de distribución son muchas las investigaciones desarrolladas. A partir del artículo seminal de Solomon en el año 1987 [14], han aparecido en la literatura muchas referencias, entre otras las que incluyen revisiones amplias de los métodos de resolución de estos problemas [1], [15], [8], [2], [3].

El objetivo general de este problema es minimizar el total de los costes asociados a los vehículos al recorrer las rutas, generalmente distancias o tiempos. Consideramos los siguientes índices y parámetros del modelo:

- Los índices de los vehículos son: $k \in [1..m]$;
- Los índices de los clientes son: $i \in [1..n]$;
- El índice del depósito es: $i = 0$;
- La cantidad de demanda del cliente i es q_i, $i \in [1..n]$;
- La capacidad del vehículo k es Q_k , $k \in [1..m]$;
- Los costes de ir del cliente i (o depósito 0) a j es c_{ij}, $i, j \in [0..n]$.
- El tiempo de viaje desde el cliente i a j es t_{ij}, $i, j \in [0..n]$.
- El tiempo de descarga al cliente i es u_i , $i \in [1..n]$;
- La ventana de tiempo del cliente i es $[e_i, l_i]$; $i \in [1..n]$;

Las variables de decisión son:

- Las variables binarias x_{ij}^k, $i, j \in [1..n]$, $k \in [1..m]$, donde $x_{ij}^k = 1$ si el vehículo k va de i a j y $x_{ij}^k = 0$ en otro caso.

- Las variables continuas r_i^k representando la carga del vehículo k cuando alcanza el cliente i. Si el vehículo k va del cliente i al cliente j ($x_{ij}^k = 1$) entonces $r_j^k = r_i^k - q_i$.
- Las variables continuas s_i^k que representan el tiempo en el que el vehículo k inicia el servicio a cada cliente i. Análogamente, si $x_{ij}^k = 1$ entonces $s_j^k = \text{máx}\{e_j, s_i^k + u_i + t_{ij}\}$.

La formulación del VRPTW como problema lineal de optimización denominada 'de tres índices' es la siguiente:

$$\text{Minimizar} \sum_{k=1}^{m}\sum_{i=0}^{n}\sum_{j=0}^{n} c_{ij}x_{ij}^k \tag{1}$$

Sujeto a:

$$\sum_{k=1}^{m}\sum_{i=0}^{n} x_{ij}^k = 1, \quad j \in [1..n] \tag{2}$$

$$\sum_{k=1}^{m}\sum_{i=0}^{n} x_{ji}^k = 1, \quad j \in [1..n] \tag{3}$$

$$\sum_{i=0}^{n} x_{ij}^k = \sum_{i=0}^{n} x_{ji}^k = 1, \quad j \in [1..n], k \in [1..m] \tag{4}$$

$$\sum_{i=1}^{n} x_{0i}^k = 1, \quad k \in [1..m] \tag{5}$$

$$\sum_{i=1}^{n} x_{i0}^k = 1, \quad k \in [1..m] \tag{6}$$

$$r_j^k - q_i - r_i^k \leq M(1 - x_{ij}^k), \quad i \in [0..n], j \in [1..n], k \in [1..m] \tag{7}$$

$$r_0^k \leq Q_k, \quad k \in [1..m] \tag{8}$$

$$s_i^k + t_{ij} + u_i - s_j^k \leq M'(1 - x_{ij}^k), \quad i \in [0..n], j \in [1..n], k \in [1..m] \tag{9}$$

$$\begin{array}{c} e_i \leq s_i^k, \quad j \in [1..n], k \in [1..m]. \\ s_i^k + u_i \leq l_i, \quad j \in [1..n], k \in [1..m]. \end{array} \tag{10}$$

La función objetivo (1) refleja el criterio de minimizar los costes totales de recorrer todas las rutas, los cuales pueden ser expresados en términos de distancia o tiempos. Las restricciones (2) y (3) aseguran que cada cliente es servido exactamente una vez. Las restricciones (4) establecen las condiciones de conservación del flujo y la continuidad de la ruta. Las restricciones (5) y (6) establecen que cada vehículo parte del depósito y acaba en él, garantizando que ningún vehículo es usado más de una vez. El conjunto de restricciones (7), donde M es una constante suficientemente grande, establecen las relaciones entre las variables r_i^k para clientes consecutivos de cada ruta y las restricciones (8) garantizan que la

carga de los vehículos no superan su capacidad. Análogamente las restricciones (9) reflejan las relaciones entre las variables s_i^k para clientes consecutivos de cada ruta. Por último las restricciones (10) establecen que cada cliente tiene que ser servido respetando su ventana de tiempo.

Proponemos extender este modelo introduciendo un objetivo que evalúa la calidad del servicio, utilizando para ello el criterio, que el cliente desea que sea atendido lo antes posible dentro de su ventana de tiempo. Si tomamos esta medida de la calidad del servicio como el tiempo total que llegan los vehículos antes de que cierre la ventana de tiempo de los clientes, el nuevo objetivo a maximizar es:

$$\sum_{k=1}^{m}\sum_{i=0}^{n}(l_i - s_{ik}) \tag{11}$$

Con un criterio similar el objetivo del modelo Cumulative CVRP propuesto en la literatura, conincide con la salvedad de un sumando, siendo este:

$$\text{Minimizar } \sum_{k=1}^{m}\sum_{i=0}^{n} s_{ik} \tag{12}$$

3. La Optimización Discreta por Enjambre de Partículas propuesta

En este artículo utilizamos una versión discreta de la PSO, denominada JFO. El algoritmo utilizado que representa esta versión es el que se describe en la Figura 1 (más detalles en [5]).

La posición de cada partícula del enjambre es una solución del problema. La solución inicial ha sido generada con un algoritmo constructivo aleatorio, que inserta los clientes al final de una ruta hasta que se completa la capacidad del vehículo o no se cumple las restricciones de ventana. También hemos utilizado un constructivo con inserción en la mejor posición, que encuentra una mejor solución inicial.

Las partículas del enjambre se mueven por medio de cuatro tipo de movimientos, dependiendo de qué partícula actúa como atractora en cada caso. Para seleccionar cual de los cuatro movimientos aplicar a cada partícula, se divide el intervalo [0,1] en cuatro sub-intervalos c_i, $i = 1, 2, 3, 4$, de tamaño variable igual a la probabilidad asignada por el usuario a cada movimiento. Se genera un número aleatorio ξ con distribución $U[0, 1]$ y la partícula se moverá a una nueva posición dependiendo del valor de ξ, como sigue:

- Si $\xi \in c_1$ entonces no existe partícula atractora y la partícula realiza un movimiento aleatorio con respecto a su posición actual. El propósito de este movimiento aleatorio es explorar el área alrededor de la posición de la partícula. Para ello se realiza un movimiento haciendo uso del $\lambda - intercambio$ (con $\lambda = 1$) que sencillamente intercambia la posición de dos clientes en diferentes sub-rutas.

- Si $\xi \in c_2$ entonces la partícula se mueve hacia la mejor posición alcanzada por un partícula de su vecindario, denotada g_i. Este movimiento es realizado aplicando un procedimiento de intercambio de sub-rutas que consiste en copiar de manera aleatoria una sub-ruta de la partícula atractora a la seguidora, eliminando los clientes de la partícula seguidora que se encuentran en la sub-ruta copiada. Este representa el movimiento social.
- Si $\xi \in c_3$ entonces el movimiento se realiza basándose en la mejor posición encontrada hasta el momento para la partícula actual, denotada b_i. La partícula actual en la posición adquiere una sub-ruta de la mejor posición encontrada hasta el momento para la partícula mediante un procedimiento de intercambio de sub-rutas. Este movimiento cognitivo ayuda a la partícula a explorar estructuras similares a ella misma en su mejor posición en beneficio de mejorar su posición actual.
- Si $\xi \in c_4$ entonces la partícula atractora es la mejor posición que globalmente ha tenido el enjambre hasta el momento, denotada g^*. Se realiza así mediante un procedimiento de intercambio de sub-rutas, un intercambio entre la mejor partícula y la actual. Este movimiento global permite utilizar parte de la mejor partícula encontrada hasta el momento para diseñar mejores soluciones.

Además en el procedimiento propuesto, después de encontrar una nueva solución, se aplica una búsqueda local para mejorar la calidad de la solución encontrada.

JFO (PSO discreto):

1. Inicalizar S:
   - Obtener posición aleatoria x_i,$\forall i \in S$.
   - Obtener los valores iniciales de b_i, g_i y g^*.
2. Repetir hasta $CriterioParada()$

   a) Para cada $i \in S$:
   $\xi \in rand[0,1]$
   $\xi \in c_1$ entonces $x_i \leftarrow RandomMove(x_i)$
   $\xi \in c_2$ entonces $x_i \leftarrow SocialMove(x_i, g_i)$
   $\xi \in c_3$ entonces $x_i \leftarrow PropioMove(x_i, b_i)$
   $\xi \in c_4$ entonces $x_i \leftarrow GlobalMove(x_i, g^*)$

   b) Para cada $i \in S$: Actualizar
   Si $F(x_i) < F(b_i)$ entonces $b_i \leftarrow x_i$ // Mejor Propio
   $\forall i \in N_j$: Si $F(x_i) < F(g_j)$ entonces $g_j \leftarrow x_i$ // Mejor entorno
   Si $F(x_i) < F(g^*)$ entonces $g^* \leftarrow x_i$// Mejor Global

---

Figura 1. Jumping Discrete PSO

4. Experimentación

Esta sección describe los resultados obtenidos de la experimentación computacional realizada. La finalidad de la experimentación es evaluar el rendimiento del procedimiento metaheurístico JFO propuesto, para resolver el VRPTW orientado a la calidad del servicio

Para la experimentación hemos utilizado las clásicas instancias de Solomon [14]. En concreto hemos seleccionado al azar una instancia de las RCxxx, la RC207. Estas instancias RCxxx combinan la distribución geográfica de los clientes agrupados en zonas con clientes ubicados al azar con distribución uniforme. Además las RC2xx tienen ventanas de tiempo amplias. Para este tipo de instancias hemos realizado la experimentación con diferentes tamaños, en particular para 25, 50 y 75 clientes.

Se han utilizado en la JFO un tamaño de enjambre de 50 partículas y 10 para el vecindario. Para dada una de los resultados obtenidos se han llevado a cabo 500 generaciones.

En primer lugar evaluamos la JFO para diferentes valores de los parámetros. Utilizamos diferentes combinaciones de los valores de c_i que representan el peso de los cuatro tipos de movimientos (aleatorio, cognitivo, social y global) dentro del procedimiento y así probar la eficiencia de los mismos para realizar el mejor ajuste. En la Cuadro 1 podemos observar el comportamiento del procedimiento JFO para estas combinaciones de parámetros y para cada una de las instancias de 25, 50 y 75 clientes. La tabla contiene los mejores resultados obtenidos para los valores del objetivo de calidad del servicio y los valores medios de este objetivo para el total de iteraciones.

Cuadro 1. *Resultados computacionales del ajuste de parámetros*

(c1,c2,c3,c4)	25 clientes		50 clientes		75 clientes	
	Mejor	Media	Mejor	Media	Mejor	Media
(25, 25, 25, 25)	8145	6353,9	17204,8	13162,8	27705,4	21080,2
(20, 20, 30, 30)	8596	6359,8	17545	13313,6	27073	20762,3
(05, 05, 45, 45)	8873,9	6343,6	17063,4	13250,9	26701	20908,1
(15, 15, 15, 55)	8676,7	6326,8	18186,2	13196,6	27254	21150,7
(55, 15, 15, 15)	8410	6311,5	17825,4	13147,6	27513,5	21001,5
(15, 55, 15, 15)	8023	6307,0	17173,8	13179,5	26675	20859,7
(15, 15, 55, 15)	7892	6314,4	18147,1	13260,1	26642	20800,2

También hemos realizado una comparativa de la ejecución de las instancias haciendo uso de entornos fijos y entornos variables. Es decir, que en una ejecución calculamos una sola vez el entorno de las partículas y no se actualizan a lo largo de las generaciones o por el contrario se recalcula cada vez que la mejor partícula de la generación anterior tiene una mejor posición que la mejor de la

generación actual. En el Cuadro 2 que presentamos a continuación, se muestra una comparativa del uso de ambos entornos para instancias con diferentes tamaño (número de clientes) y los valores de los parámetros que presentaban los mejores resultados. Como podemos observar que haciendo uso de entornos variables podemos encontrar mejores resultados que con entornos fijos.

Cuadro 2. *Resultados computacionales con diferentes entornos*

Inst/Param.	Mejores resultados			Medias	
	$QS.$	TD	TT	NV	$QS.$
Fijos					
$25C, (20, 20, 30, 30)$	7648,3	1098,4	2255,2	4	6265,6
$50C, (20, 20, 30, 30)$	16050,7	2332,3	4008,2	8	13146,8
$75C, (15, 15, 15, 55)$	26501,7	2703,6	4462,6	9	21009,3
Variables					
$25C, (20, 20, 30, 30)$	8142,4	907,6	2295,3	4	6275,4
$50C, (20, 20, 30, 30)$	16669,4	2004,9	4135,0	8	13264,8
$75C, (15, 15, 15, 55)$	25501,7	2759,3	5278,8	10	21111,2

Por otra parte, evaluamos el comportamiento del modelo CCVRPTW y el modelo QSVRPTW comparando los resultados utilizando una solución inicial obtenida por el constructivo de inserción final y el constructivo de mejor inserción. En este caso podemos ver la diferencia de comportamiento de ambos procedimientos y modelos en la Tabla 3.

5. Conclusiones

En esta investigación, abordamos un nuevo modelo del VRPTW como extensión del mismo en los objetivos, tratando de lograr la máxima satisfacción de los clientes asociada al criterio de que los vehículos alcancen y sirvan al cliente lo antes posible dentro de su ventana de tiempo. Se proporciona la formulación de este modelo denotado QSVRPTW y del CCVRPTW, una variante tratada en la literatura similar pero sin ventanas de tiempo. Para resolver estos problemas hemos utilizado una metaheurística para problemas discretos, basada en el PSO, el JFO. La experimentación computacional incluye una evaluación del comportamiento de los parámetros de la metaheurística y de la obtención de los entornos de las partículas utilizadas en el mismo. Esto nos ha ayudado a determinar la mejor combinación de parámetros en el procedimiento, para resolver este problema. Los resultados experimentales revelan que el procedimiento propuesto ayuda a resolver estos modelos con ventanas de tiempo y objetivos centrados en la calidad del servicio. Las soluciones obtenidas en el CCVRPTW son diferentes a las obtenidas en el QSVRPTW, las primeras son más eficientes

Cuadro 3. *Resultados computacionales comparando CCVRPTW y QSVRPTW*

Inst/Param	Mejores resul			
	QS.	*TD*	*TT*	*NV*
CCVRPTW	Inserción Final			
$25C, (20, 20, 30, 30)$	16085,6	598,5	1049,5	2
$50C, (20, 20, 30, 30)$	33723,1	1623,7	2794,1	4
$75C, (15, 15, 15, 55)$	46726,7	1687,5	2811,5	5
QSVRPTW	Inserción Final			
$25C, (20, 20, 30, 30)$	7535,3	807,7	2781,5	5
$50C, (20, 20, 30, 30)$	15634,9	2133,9	4009,3	7
$75C, (15, 15, 15, 55)$	27705,4	2456,3	4935,2	10
CCVRPTW	Mejor Inserción			
$25C, (20, 20, 30, 30)$	15931,2	604,7	1059,3	2
$50C, (20, 20, 30, 30)$	31419,6	1098,0	1945,4	3
$75C, (15, 15, 15, 55)$	45423,8	1573, 3	2677,9	4
QSVRPTW	Mejor Inserción			
$25C, (20, 20, 30, 30)$	6567,6	612,8	1145,1	2
$50C, (20, 20, 30, 30)$	14733,0	1112,1	1908,9	3
$75C, (15, 15, 15, 55)$	23073,3	1632,0	2591,6	4

en cuanto al uso y disponibilidad de recursos, en cambio atender a los clientes dentro de las ventanas de tiempo lo antes posible, hace que el número de rutas crezca y en consecuencia, los tiempos y distancias totales en recorrer las rutas. Futuros trabajos, nos conducirán a estudiar algunas mejoras en la metaheurística implementada, la experimentación con instancias reales y el tratamiento de la incertidumbre en algunas componentes del modelo.

Referencias

1. J. Brandao. *Meta-heuristics Advances and Trends in Local Search Paradigms for Optimization*, chapter Metaheuristic for the vehicle routing problem with time windows, pages 19–36. Kluwer Academic Publishers, 1999.
2. Olli Bräysy and Michel Gendreau. Vehicle routing problem with time windows, part i: Route construction and local search algorithms. *Transportation Science*, 39:104–118, 2005.
3. Olli Bräysy and Michel Gendreau. Vehicle routing problem with time windows, part ii: Metaheuristics. *Transportation Science*, 39:119–139, 2005.
4. Ping Chen, Xingye Dong, and Yanchao Niu. An iterated local search algorithm for the cumulative capacitated vehicle routing problem. In Honghua Tan, editor, *Technology for Education and Learning*, volume 136 of *Advances in Intelligent Systems and Computing*, pages 575–581. Springer Berlin Heidelberg, 2012.
5. S. Consoli, J. A. Moreno-Perez, K Darby-Dowman, and N. Mladenovic. Discrete particle swarm optimization for the minimum labelling steiner tree problem. *Natural Computing*, 9(1):29–46, 2010.

6. A. Corberan, E. Fernández, and M. LagM. Laguna. Marti. Heuristic solutions to the problem of routing school buses with multiple objectives. *J. Operational Research Society*, 53(4):427–435, 2002.
7. Jean-Francois Cordeau, Gilbert Laporte, Martin W.P. Savelsbergh, and Daniele Vigo. Vehicle routing. In Cynthia Barnhart and Gilbert Laporte, editors, *Transportation*, volume 14 of *Handbooks in Operations Research and Management Science*, pages 367 – 428. Elsevier, 2007.
8. J.F. Cordeau, G. Desaulniers, J. Desrosiers, M. M. Solomon, and Saumis F. *The vehicle routing problems*, chapter VRP with Time Windows, pages 157–186. SIAM, 2002.
9. H. Jin-Kao, P. Galinier, and M. Habib. Métaheuristiques pour l'optimisation combinatoire et l'affectation sous contraintes. *Revue d'Intelligence Artificielle*, 13(2):283–324, 2000.
10. Brian Kallehauge. Formulations and exact algorithms for the vehicle routing problem with time windows. *Computers & Operations Research*, 35(7):2307 – 2330, 2008. ¡ce:title¿Part Special Issue: Includes selected papers presented at the EC-CO'04 European Conference on combinatorial Optimization¡/ce:title¿.
11. J.A. Moreno-Pérez, J. P. Castro-Gutiérrez, F. J. Martínez-García, B. Melián, J. M. Moreno-Vega, and J. Ramos. Discrete particle swarm optimization for the p-median problem. In *7th Metaheuristics International Conference*, Montreal, Canada, 2007.
12. Sandra Ulrich Ngueveu, Christian Prins, and Roberto Wolfler Calvo. An effective memetic algorithm for the cumulative capacitated vehicle routing problem. *Computers & Operations Research*, 37(11):1877 – 1885, 2010. Metaheuristics for Logistics and Vehicle Routing.
13. Sandra Ulrich Ngueveu, Christian Prins, and Roberto Wolfler-Calvo. An effective evolutionary algorithm for the cumulative capacitated vehicle routing problem. In Mario Giacobini, Anthony Brabazon, Stefano Cagnoni, GianniA. Caro, Anika Ekart, AnnaIsabel Esparcia-Alcazar, Muddassar Farooq, Andreas Fink, and Penousal Machado, editors, *Applications of Evolutionary Computing*, volume 5484 of *Lecture Notes in Computer Science*, pages 778–787. Springer Berlin Heidelberg, 2009.
14. Marius M. Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2):254–265, 1987.
15. K.C. Tan, L. H. Lee, Q.L. Zhu, and K. Ou. Heuristic method for vehicle routing problem with time windows. *Artificial Intelligence in Engineering*, 15:281–295, 2001.
16. P. Toth and D. Vigo. *The vehicle routing problem*, volume 9 of *Monographs on Discrete Mathematics and Applications*. Society for Industrial and Applied Mathematics Publishing, 2002.

Un Algoritmo de Estimación de Distribución para el Problema de Planificación de Grúas en una Terminal Marítima de Contenedores

Christopher Expósito-Izquierdo, Belén Melián-Batista y
J. Marcos Moreno-Vega

Departamento de Estadística, Investigación Operativa y Computación
Universidad de La Laguna
38271 La Laguna, Spain
{cexposit, mbmelian, jmmoreno}@ull.es

Resumen Este trabajo presenta un algoritmo de estimación de distribución para el Problema de Planificación de Grúas en el contexto de una terminal marítima de contenedores. El objetivo de este problema es determinar las secuencias de trabajo de las grúas asignadas a un barco portacontenedores para la realización de sus tareas de carga y descarga. Los resultados computacionales muestran que el algoritmo propuesto presenta un rendimiento superior al de las técnicas de optimización de la literatura, alcanzando soluciones de alta calidad mediante tiempos de computación reducidos en un amplio conjunto de escenarios reales.

Palabras Clave: Problema de Planificación de Grúas, Algoritmo de Estimación de Distribución, Terminal Marítima de Contenedores

1 Introducción

Las terminales marítimas de contenedores son infraestructuras que permiten el intercambio de contenedores entre diferentes medios de transporte dentro de una red global de transporte intermodal. Los medios de transporte más habituales suelen ser barcos portacontenedores, camiones y trenes. Con el objetivo de dar servicio a estos medios de transporte las terminales marítimas típicamente se dividen en tres grandes áreas funcionales: línea de atraque, patio de contenedores e interfaz continental. La línea de atraque permite dar servicio a los barcos portacontenedores llegados hasta el puerto. El patio de contenedores es un almacén temporal donde se ubican los contenedores hasta su posterior retirada mediante algún medio de transporte para su reparto fuera del puerto. Por último, la interfaz continental conecta la terminal de contenedores con los medios de transporte terrestres. Una descripción exhaustiva de la estructura de las terminales y los principales procesos que tienen lugar en ellas puede encontrarse en los trabajos de Vis y de Koster [12], Steenken et al. [11] y Stahlbock y Voβ [10].

El indicador de rendimiento más representativo de una terminal es el tiempo de estadía de los barcos portacontenedores llegados a puerto (Wiegmans et al.

[13]). En este sentido, el servicio de los barcos sigue una secuencia de pasos bien definida. En primer lugar, se ha de proporcionar una ubicación de atraque para cada uno de los barcos acorde a sus características (tamaño, carga, etc.). Este problema es habitualmente conocido como Problema de Asignación de Atraques (Lalla-Ruiz et al. [5]). Una vez establecida la posición de atraque de los barcos, un subconjunto de las grúas disponibles en el muelle es asignado a cada barco. Este problema es denominado como Problema de Asignación de Grúas (Raa et al. [8]). Por último, el Problema de Planificación de Grúas planifica la realización de las operaciones de carga y descarga asociadas a cada uno de los barcos.

La disposición de los contenedores dentro de un barco portacontenedores viene determinada por su plan de estiba (Ambrosino et al. [1]). Éste define la posición de cada uno de los contenedores a cargar/descargar durante la travesía del barco. Los contenedores se ubican en el barco en secciones transversales denominadas bahías. Dentro de una bahía un contenedor puede ubicarse en la bodega o en la cubierta. Para facilitar la estiba de los contenedores éstos se suelen ubicar conjuntamente formando grupos dentro de la misma bahía de acuerdo con su peso, puerto de destino, etc. Por tanto, el objetivo principal de una terminal es la minimización del tiempo empleado para la realización de las tareas asociadas a cada uno de los barcos.

La principal contribución de este trabajo es el diseño de un algoritmo de estimación de distribución para la resolución del Problema de Planificación de Grúas. Éste es un algoritmo basado en poblaciones con una gran capacidad para detectar regiones prometedoras del espacio de búsqueda, pero adolece de suficiente explotación de las soluciones encontradas. Con el objetivo de superar esta limitación, este algoritmo es combinado con una búsqueda local. Los resultados computacionales indican que el algoritmo propuesto presenta una alta eficacia y eficiencia en un amplio conjunto de escenarios reales.

El resto de este trabajo se estructura como sigue. La Sección 2 describe el Problema de Planificación de Grúas. La Sección 3 resume las contribuciones de los principales trabajos de la literatura. La Sección 4 propone un algoritmo de estimación de distribución para la resolución del problema planteado. La Sección 5 analiza el comportamiento del algoritmo propuesto a través de un conjunto de experimentos computacionales. Finalmente, la Sección 6 resume las principales conclusiones extraídas del trabajo y establece algunas líneas de trabajo futuras.

2 Problema de Planificación de Grúas

El Problema de Planificación de Grúas (PPG) es un problema $\mathcal{NP}$-duro (Sammarra et al. [9]) cuyo objetivo es determinar las tareas de transbordo (carga y descarga) realizadas por las grúas de muelle asignadas a un barco.

Los datos de entrada del PPG son un conjunto de tareas $\Omega = \{1, ..., n\}$ y un conjunto de grúas $Q = \{1, ..., m\}$ con similares características técnicas (dimensiones, velocidad, etc.). Una tarea representa las operaciones de transbordo de los contenedores pertenecientes al mismo grupo y ubicados de forma adyacente en una bahía. Cada tarea $t \in \Omega$ requiere un tiempo de procesamiento,

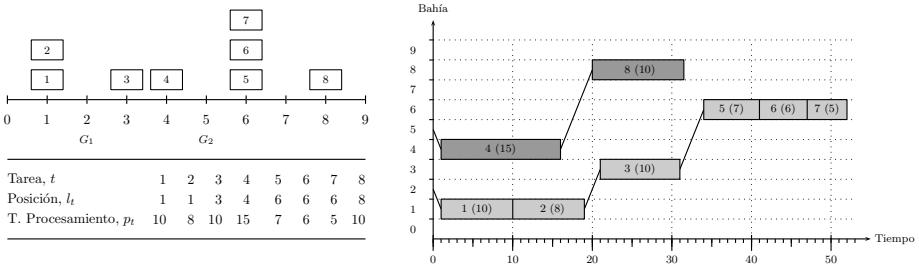


Figura 1. Ejemplo con 8 tareas y 2 grúas para el PPG

p_t, y está ubicada en una bahía del barco, l_t. Por motivos de conveniencia, se incluyen dos tareas ficticias 0 y $T = n + 1$, donde $p_0 = p_T = 0$, para representar respectivamente el comienzo y finalización del servicio del barco. Las bahías l_{min} y l_{max} son aquellas con al menos una tarea situadas más a la izquierda y derecha respectivamente. Tal como indican Kim y Park [4], existen relaciones de precedencia entre pares de tareas ubicadas en la misma bahía.

Estas precedencias están definidas mediante el conjunto Φ, donde

$$\Phi = \{(i,j) \mid i,j \in \Omega, i \text{ tiene que ser terminada antes del comienzo de } j\}. \quad (1)$$

Por otro lado, la grúa $q \in Q$ está disponible a partir del instante $r^q \geq 0$ y se encuentra inicialmente ubicada en la bahía l_0^q. El tiempo que requieren las grúas para moverse entre bahías es $\hat{t} > 0$. Por tanto, el tiempo requerido por la grúa q para moverse desde la ubicación de la tarea i hasta la ubicación de la tarea j es $t_{ij}^q = \hat{t}|l_i - l_j|$. Las grúas se encuentran montadas sobre un sistema de raíles que permite que se desplacen a lo largo del muelle. Consecuentemente, éstas no pueden cruzarse a la vez que, por motivos de seguridad, deben mantener una distancia de seguridad $\delta > 0$ entre ellas. Esto da lugar a que ciertos pares de tareas no puedan realizarse a la vez. Estas restricciones están definidas mediante el conjunto $\Psi \supseteq \Phi$, donde

$$\Psi = \{(i,j) \mid i,j \in \Omega, i \text{ y } j \text{ no pueden ser realizadas a la vez}\}. \quad (2)$$

Una solución para el PPG es una planificación para la cual el instante de comienzo y finalización de cada tarea está completamente determinado. El criterio de optimización del PPG es la minimización del *makespan* (instante de finalización de la tarea T) e instante de finalización de las grúas, $y^q, \forall q \in Q$ (Kim y Park [4]). Esto es,

$$min\ \alpha_1\omega + \alpha_2 \sum_{q \in Q} y^q. \quad (3)$$

La Figura 1 muestra un ejemplo para el PPG con 8 tareas realizadas por 2 grúas con $\delta = 1$. Las tareas se encuentran ordenadas en cada bahía de acuerdo a sus relaciones de precedencia (Ecuación 1). En este caso, $\omega = 52$, mientras que $y^1 = 52$ e $y^2 = 32$. Si $\alpha_1 = \alpha_2 = 1$, de acuerdo con la Ecuación 3 el valor objetivo de la planificación propuesta es 136.

3 Revisión Bibliográfica

La importancia del PPG para el incremento de la competitividad de las terminales marítimas de contenedores ha hecho que en los últimos años haya sido objeto de estudio en diversos trabajos.

Kim y Park [4] presentan un Branch & Bound para la resolución exacta del PPG. La carga computacional de esta técnica es alta para instancias medianas por lo que los autores proponen también un GRASP. La calidad de las soluciones alcanzadas por el GRASP difieren en menos de un 10% de la optimalidad.

Sammarra et al. [9] descomponen el PPG en un problema de ruteo, que determina las secuencias de trabajo para las grúas, y un problema de planificación que define el instante de comienzo de las tareas. Los autores presentan una Búsqueda Tabú con alta efectividad y tiempos de computación razonables.

Bierwirth y Meisel [2] proponen un modelo matemático para el PPG. Además los autores desarrollan un método de optimización denominado *Unidirectional Scheduling* (UDS) basado en el esquema general del Branch & Cut para encontrar la planificación unidireccional óptima. El UDS es capaz de encontrar soluciones de alta calidad con un tiempo máximo de computación de una hora.

Legato et al. [6] presentan un método de optimización denominado *LTM* que es una extensión del *UDS* propuesto por Bierwirth y Meisel [2] en el que se incluyen nuevas cotas inferiores que permiten reducir el tamaño del espacio de búsqueda. Los experimentos computacionales demuestran su superioridad sobre las propuestas anteriormente publicadas.

La motivación en este trabajo para el desarrollo de una nueva propuesta de resolución del PPG surge de que, a pesar de la existencia de las citadas técnicas de optimización, éstas no presentan un comportamiento homogéneo en escenarios diferentes o bien requieren tiempos de computación elevados.

4 Algoritmo de Estimación de Distribución

Los algoritmos de estimación de distribución (AED) son una familia de técnicas de optimización basados en poblaciones que fundamentan su comportamiento en la teoría de probabilidad (Lozano et al. [7]).

El AED propuesto en este trabajo sigue el diagrama de flujo mostrado en la Figura 2. A lo largo del proceso de búsqueda mantiene un modelo de probabilidades $M(g)$ con m filas y n columnas para cada generación g que registra información estadística sobre el espacio de búsqueda explorado. Esto es,

$$M(g) = \begin{pmatrix} m_{11}(g) & m_{12}(g) & \cdots & m_{1n}(g) \\ m_{21}(g) & m_{22}(g) & \cdots & m_{2n}(g) \\ \vdots & \vdots & \ddots & \vdots \\ m_{m1}(g) & m_{m2}(g) & \cdots & m_{mn}(g) \end{pmatrix}$$

Cada elemento m_{qt} representa la probabilidad de que la tarea $t \in \Omega$ sea asignada a la grúa $q \in Q$ en la próxima solución generada. La asignación de tareas a

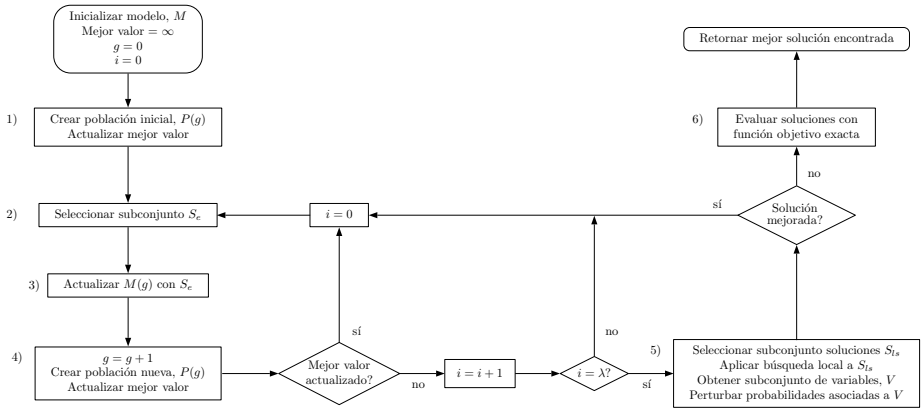


Figura 2. Diagrama de flujo del AED propuesto para el PPG

grúas es exclusiva, de forma que la siguiente restricción debe ser satisfecha por el modelo de probabilidades:

$$\sum_{q \in Q} m_{qt}(g) = 1, \forall t \in \Omega, \forall g \tag{4}$$

La inicialización del modelo se realiza fijando cada asignación de grúa a tarea. Las características del problema sugieren que, con el objetivo de minimizar la distancia recorrida, las planificaciones más interesantes son aquellas en las que las tareas son realizadas por una grúa que se encuentra situada inicialmente cerca. En este caso, el modelo de probabilidades se inicializa en base a la distribución gaussiana discreta (Johnson [3]). Esto es:

$$m_{qt}(0) = \frac{h(q,t)}{\sum_{q \in Q} h(q,t)}, \forall t \in \Omega, \forall q \in Q \tag{5}$$

donde $h(q,t)$ se define como

$$h(q,t) = \frac{1}{\sqrt{l_{max} - l_{min}}\sqrt{2\pi}} \exp\left\{ -\frac{|l_0^q - l_t|^2}{2(l_{max} - l_{min})} \right\} \tag{6}$$

En cada generación de la búsqueda se genera una población de N soluciones, $P(g)$, a partir del modelo de probabilidades. $P(g)$ está compuesta por el subconjunto de mejores soluciones de la población anterior, S_e, y las nuevas soluciones generadas. El tamaño de S_e es definido mediante el porcentaje α. Las nuevas soluciones generadas se obtienen asignando secuencialmente las tareas a una de las grúas en base a las probabilidades establecidas en el modelo.

El modelo de probabilidades es actualizado con información estadística de la población actual. En primer lugar, sea x_{qt}^{σ} igual a 1 si la tarea $t \in \Omega$ es realizada por la grúa $q \in Q$ en la planificación σ. $M(g)$ es actualizado como

$$m_{qt}(g) = \frac{k(q,t)}{\sum_{q \in Q} k(q,t)}, \forall t \in \Omega, \forall q \in Q \tag{7}$$

donde $k_{qt}(g)$ es definido como sigue:

$$k_{qt}(g) = m_{qt}(g-1) + \frac{1}{|S_e|} \sum_{\sigma \in S_e} x_{qt}^{\sigma}, \forall q \in Q, \forall t \in \Omega \tag{8}$$

Tal como se indicó brevemente en la introducción de este trabajo, los AEDs son técnicas con una alta capacidad de exploración del espacio de búsqueda, sin embargo, muestran una escasa capacidad para explotar dichas regiones. En este trabajo, se propone la combinación del AED con una búsqueda local basada en la reasignación de tareas y el intercambio de pares de tareas entre grúas adyacentes. Así bien, la búsqueda local es aplicada al subconjunto de mejores soluciones de la población actual, S_{ls}, tras λ generaciones sin mejora en el valor objetivo de la mejor solución alcanzada por la búsqueda. El tamaño de S_{ls} es definido mediante el porcentaje β, cuyo valor es establecido por el usuario. Con el objetivo de alcanzar una mayor diversificación de la búsqueda, una vez aplicada la búsqueda local, las probabilidades del modelo asociadas a aquellas tareas realizadas por distintas grúas en las planificaciones de S_{ls} con respecto a la mejor planificación de la población son uniformemente reiniciadas.

Finalmente, cabe señalar que la evaluación de las planificaciones generadas se realiza sin tener en cuenta las interferencias entre grúas. Esto proporciona una reducción superior al 95% en el tiempo empleado en la evaluación de las planificaciones a la vez que un error reducido. Sin embargo, las planificaciones incluidas en la población durante la última generación son evaluadas de forma exacta mediante el esquema propuesto por Sammarra et al. [9].

5 Experiencia Computacional

El objetivo de esta experimentación es comparar el rendimiento del AED con el del método exacto LTM (Legato et al. [6]). Para la experimentación se han empleado las instancias propuestas por Bierwirth y Meisel [2]. Son 90 instancias agrupadas en 9 categorías de diferentes tamaños. Tal como se indica en el artículo original, el método LTM ha sido ejecutado en un PC 3.07 GHz Intel T9900 y un criterio de parada basado en un tiempo máximo por instancia de una hora o bien 100 millones de nodos del árbol de búsqueda explorados. Por su parte, el AED ha sido ejecutado en un PC Intel Core 2 Duo E8500 3.16 GHz. Los valores de parámetros empleados se han establecido mediante un estudio preliminar en el que se han fijado como $N = 2000$, $\alpha = \beta = 20\%$, $\lambda = 20$ y $p_{min} = \frac{1}{mn}$.

Las tablas 1 y 2 muestran respectivamente los resultados obtenidos en instancias pequeñas y grandes. En cada caso se muestra el número de tareas y grúas de la instancia, el valor objetivo (Ecuación 3 con $\alpha_1 = 3$ y $\alpha_2 = 0$) y el tiempo de computación empleado por ambas técnicas de optimización. En la Tabla 2 se incluye una cota inferior propuesta por Legato et al. [6]. En el caso del método LTM, únicamente se proporcionan tiempos de computación medios para instancias pequeñas en el artículo original.

Los resultados presentados en la Tabla 1 muestran que, para instancias pequeñas, el AED propuesto proporciona una planificación óptima en todos los

casos en menos de 9 segundos. El espacio de búsqueda en este tipo de instancias es reducido por lo que técnicas de optimización exhaustivas como el método LTM son altamente aconsejables. Por otro lado, el AED es capaz de alcanzar una planificación óptima en la mayoría de las instancias grandes, a la vez que, en aquellas instancias donde no la ha alcanzado, el gap es de 1.03% en el peor caso (instancia *k100*). Si se realiza un análisis por grupos se comprueba que el mayor gap se presenta en las instancias del grupo I, donde se alcanza un gap de 0.07%. Es importante destacar que, el AED propuesto ha permitido mejorar la calidad de la mejor planificación conocida para la instancia *k101*, alcanzándose una mejora del 0.37%. Cabe recordar que, a pesar de tratarse de un método exacto, el método LTM se ha ejecutado con un criterio de parada, por lo que la optimalidad de las planificaciones proporcionadas no está garantizada en aquellos casos donde la ejecución no haya finalizado. La calidad media de las planificaciones proporcionadas por el AED es solo un 0.018% inferior a la de las proporcionadas por el método LTM. En este sentido, el test de Wilcoxon con un nivel de confianza del 95% indica que no existen diferencias significativas entre la calidad de las planificaciones proporcionadas por ambas técnicas. Finalmente, se comprueba que, a medida que el tamaño de las instancias crece, los tiempos computacionales requeridos por el método LTM se ven gravemente afectados. Este hecho se manifiesta en múltiples instancias para las que el tiempo de computación es superior a 10 minutos. En los mismos casos, el tiempo empleado por el AED es notablemente inferior. Así bien, el tiempo medio mayor es de unos 77 segundos (grupo I). El test de Wilcoxon con un 95% de confianza indica que existen diferencias significativas entre los tiempos de computación.

6 Conclusiones y Líneas Futuras

El tiempo de estadía de los barcos portacontenedores es el principal indicador de rendimiento de las terminales marítimas de contenedores. Éste se deriva del tiempo empleado por las grúas en realizar las operaciones de transbordo de contenedores. En este trabajo se propone un algoritmo de estimación de distribución para la resolución del Problema de Planificación de Grúas, el que persigue minimizar el tiempo de estadía de los barcos así como el empleado por las grúas de muelle. Este algoritmo presenta una alta capacidad para la exploración del espacio de búsqueda y encontrar regiones prometedoras en el mismo, sin embargo, no es capaz de explotar suficientemente dichas regiones. Con el objetivo de superar esta debilidad se integra con una búsqueda local. La búsqueda local se emplea en una estrategia de reinicio del algoritmo. Al mismo tiempo, la evaluación de las planificaciones generadas se relaja de forma que no se consideran las interferencias entre grúas. Esto permite una reducción superior al 95% en el tiempo de computación empleado durante la evaluación de las planificaciones.

A pesar de la naturaleza cambiante de las terminales de contenedores, el Problema de Planificación de Grúas no ha sido estudiado hasta la fecha en entornos dinámicos e inciertos. Por otro lado, la integración de este problema con

				LTM		AED		
Grupo	n	m	Instancia	f_{LTM}	$t_{LTM}(m)$	f_{AED}	$t_{AED}(m)$	Gap (%)
A	10	2	k13	453	—	453	1.29E-3	0.00
			k14	546	—	546	1.46E-3	0.00
			k15	513	—	513	1.28E-3	0.00
			k16	312	—	312	9.38E-4	0.00
			k17	453	—	453	1.41E-3	0.00
			k18	375	—	375	1.19E-3	0.00
			k19	543	—	543	1.38E-3	0.00
			k20	399	—	399	1.50E-3	0.00
			k21	465	—	465	1.20E-3	0.00
			k22	540	—	540	2.12E-3	0.00
			Avg.	**459.9**	**1.12E-5**	**459.9**	**1.38E-3**	**0.00**
B	15	2	k23	576	—	576	4.53E-3	0.00
			k24	666	—	666	6.48E-3	0.00
			k25	738	—	738	4.18E-3	0.00
			k26	639	—	639	5.47E-3	0.00
			k27	657	—	657	4.89E-3	0.00
			k28	531	—	531	4.52E-3	0.00
			k29	807	—	807	5.22E-3	0.00
			k30	891	—	891	3.74E-3	0.00
			k31	570	—	570	4.32E-3	0.00
			k32	591	—	591	6.17E-3	0.00
			Avg.	**666.6**	**3.68E-5**	**666.6**	**4.95E-3**	**0.00**
C	20	3	k33	603	—	603	1.52E-1	0.00
			k34	717	—	717	1.54E-1	0.00
			k35	684	—	684	7.46E-2	0.00
			k36	678	—	678	1.27E-1	0.00
			k37	510	—	510	6.82E-2	0.00
			k38	618	—	618	1.11E-1	0.00
			k39	513	—	513	1.09E-1	0.00
			k40	564	—	564	1.19E-1	0.00
			k41	588	—	588	1.11E-1	0.00
			k42	573	—	573	1.05E-1	0.00
			Avg.	**604.8**	**6.26E-4**	**604.8**	**1.13E-1**	**0.00**

Tabla 1. Comparación entre el método LTM (Legato et al. [6]) y el AED propuesto en este trabajo para instancias pequeñas

					LTM		AED		
Grupo	n	m	Instancia	LB	f_{LTM}	$t_{LTM}(m)$	f_{AED}	$t_{AED}(m)$	Gap (%)
D	25	3	k43	860	876	0.21	876	0.21	0.00
			k44	821	822	0.20	822	0.19	0.00
			k45	825	834	0.18	834	0.14	0.00
			k46	690	690	0.19	690	0.16	0.00
			k47	792	792	0.17	792	0.17	0.00
			k48	629	639	0.19	639	0.14	0.00
			k49	880	894	0.18	894	0.22	0.00
			k50	727	741	0.17	741	0.28	0.00
			k51	778	798	0.17	798	0.20	0.00
			k52	942	960	0.17	960	0.22	0.00
			Avg.	**794.4**	**804.6**	**0,18**	**804.6**	**0.19**	**0,00**
E	30	4	k53	660	717	18.02[a]	717	0.29	0.00
			k54	756	774	0.30	774	0.33	0.00
			k55	667	684	0.30	684	0.32	0.00
			k56	670	690	0.38	690	0.31	0.00
			k57	684	705	0.31	705	0.18	0.00
			k58	769	786	0.32	786	0.33	0.00
			k59	669	687	0.29	687	0.27	0.00
			k60	766	783	0.32	783	0.23	0.00
			k61	620	639	0.30	639	0.23	0.00
			k62	831	837	0.30	837	0.39	0.00
			Avg.	**709.2**	**730.2**	**2.08**	**730.2**	**0.29**	**0.00**
F	35	4	k63	930	948	0.56	948	0.28	0.00
			k64	718	741	0.51	741	0.45	0.00
			k65	821	837	0.70	837	0.55	0.00
			k66	910	924	0.38	924	0.44	0.00
			k67	861	882	0.39	882	0.35	0.00
			k68	951	963	0.35	963	0.42	0.00
			k69	787	807	0.56	807	0.24	0.00
			k70	939	957	0.44	957	0.21	0.00
			k71	814	834	0.99	834	0.74	0.00
			k72	727	744	0.38	744	0.41	0.00
			Avg.	**845.8**	**863.7**	**0.53**	**863.7**	**0.41**	**0.00**
G	40	5	k73	840	870	6.82	870	0.30	0.00
			k74	825	843	1.56	843	0.36	0.00
			k75	660	675	0.90	675	0.64	0.00
			k76	830	852	0.81	852	0.50	0.00
			k77	679	699	0.86	702	0.53	0.43
			k78	623	642	1.45	642	0.44	0.00
			k79	722	744	0.91	744	1.05	0.00
			k80	724	750	0.88	750	0.38	0.00
			k81	711	738	0.93	738	0.26	0.00
			k82	701	717	0.91	717	0.80	0.00
			Avg.	**731.5**	**753.0**	**1.60**	**753.3**	**0.53**	**0.04**
H	45	5	k83	928	948	1.02	948	0.68	0.00
			k84	882	897	0.98	897	0.68	0.00
			k85	952	972	1.08	972	0.46	0.00
			k86	791	816	5.88	819	0.53	0.37
			k87	846	867	16.29	867	1.21	0.00
			k88	747	768	1.89	768	0.34	0.00
			k89	825	843	1.09	843	0.64	0.00
			k90	1029	1053	2.65	1053	1.42	0.00
			k91	814	837	1.83	837	0.76	0.00
			k92	876	897	2.78	897	0.79	0.00
			Avg.	**869**	**889.8**	**3.55**	**890.1**	**0.75**	**0.03**
I	50	6	k93	792	810	8.43	810	1.35	0.00
			k94	768	786	19.84[a]	786	1.66	0.00
			k95	806	834	18.87[a]	834	1.68	0.00
			k96	783	801	3.66	801	1.14	0.00
			k97	693	717	33.70[a]	717	1.32	0.00
			k98	717	735	1.52	735	1.08	0.00
			k99	823	852	20.69[a]	852	0.72	0.00
			k100	858	876	2.84	885	1.68	1.03
			k101	771	813	39.48[a]	**810**	1.11	**-0.37**
			k102	874	897	5.62	897	1.13	0.00
			Avg.	**788.5**	**812.1**	**15.45**	**812.7**	**1.29**	**0.07**

[a] El método LTM ha sido detenido al haber alcanzado el número máximo de nodos explorados

Tabla 2. Comparación entre el método LTM (Legato et al. [6]) y el AED propuesto en este trabajo para instancias grandes

otros problemas como el Berth Allocation Problem o el Quay Crane Allocation Problem es un campo que hasta ahora ha sido insuficientemente estudiado.

Agradecimientos

El presente trabajo ha sido parcialmente financiado por el Ministerio de Economía y Competitividad (proyecto TIN2012-32608). Christopher Expósito Izquierdo agradece a CajaCanarias la ayuda económica que le presta a través de la beca para posgraduados que disfruta.

Referencias

1. Daniela Ambrosino, Anna Sciomachen, and Elena Tanfani. Stowing a containership: the master bay plan problem. *Transportation Research Part A: Policy and Practice*, 38(2):81 – 99, 2004.
2. Christian Bierwirth and Frank Meisel. A fast heuristic for quay crane scheduling with interference constraints. *Journal of Scheduling*, 12:345–360, 2009.
3. Norman L. Johnson, Samuel Kotz, and Adrienne W. Kemp. *Univariate Discrete Distributions (Wiley Series in Probability and Statistics)*. Wiley-Interscience, 2 edition, February 1993.
4. Kap Hwan Kim and Young-Man Park. A crane scheduling method for port container terminals. *European Journal of Operational Research*, 156(3):752–768, 2004.
5. Eduardo Lalla-Ruiz, Belén Melián-Batista, and J. Marcos Moreno-Vega. Artificial intelligence hybrid heuristic based on tabu search for the dynamic berth allocation problem. *Engineering Applications of Artificial Intelligence*, 25(6):1132 – 1141, 2012.
6. Pasquale Legato, Roberto Trunfio, and Frank Meisel. Modeling and solving rich quay crane scheduling problems. *Computers & Operations Research*, 39(9):2063 – 2078, 2012.
7. J.A. Lozano, P. Larrañaga, P. Inza, and E. Bengoetxea. *Towards a New Evolutionary Computation. Advances in Estimation of Distribution Algorithms*. Springer Verlag, 2006.
8. Birger Raa, Wout Dullaert, and Rowan Van Schaeren. An enriched model for the integrated berth allocation and quay crane assignment problem. *Expert Systems with Applications*, 38(11):14136 – 14147, 2011.
9. Marcello Sammarra, Jean-François Cordeau, Gilbert Laporte, and M. Monaco. A tabu search heuristic for the quay crane scheduling problem. *Journal of Scheduling*, 10:327–336, 2007.
10. Robert Stahlbock and Stefan Voβ. Operations research at container terminals: a literature update. *OR Spectrum*, 30:1–52, 2008.
11. Dirk Steenken, Stefan Voβ, and Robert Stahlbock. Container terminal operation and operations research - a classification and literature review. *OR Spectrum*, 26:3–49, 2004.
12. Iris F.A. Vis and René de Koster. Transshipment of containers at a container terminal: An overview. *European Journal of Operational Research*, 147(1):1–16, 2003.
13. Bart W. Wiegmans, Piet Rietveld, and Peter Nijkamp. Container terminal services and quality. Serie Research Memoranda 0040, VU University Amsterdam, Faculty of Economics, Business Administration and Econometrics, 2001.

Estudio de la Planificación de Atraque en una Terminal de Contenedores

Eduardo Lalla Ruiz, Belén Melián Batista y J. Marcos Moreno Vega

Departamento de Estadística, Investigación Operativa y Computación
Universidad de La Laguna
38271 La Laguna, Spain
{elalla, mbmelian, jmmoreno}@ull.es

Resumen La creciente demanda de transporte marítimo y el incremento de la competitividad entre las terminales portuarias fuerza a sus gestores a reducir los costes de forma eficiente a través del uso apropiado de los recursos de la misma. En este sentido, los atraques son un factor clave dado que un uso ineficiente de estos se traduce en una baja productividad global de la terminal. En este trabajo se estudia y propone la aplicación de técnicas de optimización en las tareas de planificación de atraques de una terminal de contenedores.

Palabras Clave: Metaheurísticas, Terminal de Contenedores, Planificación de Atraques

1 Introducción

Las terminales de contenedores son los elementos principales dentro del intercambio marítimo de mercancías (Vis y Koster [11]). Estas instalaciones son sistemas abiertos dedicados al intercambio de contenedores en las redes de transporte multimodal. En términos generales, de acuerdo con Petering et al. [10] una terminal de contenedores se puede dividir en tres grandes partes en función de las operaciones que se llevan acabo en las mismas. En primer lugar, la zona del lado del mar es la parte del puerto en donde los buques son atracados con el objetivo de ser cargados y descargados de contenedores para continuar su ruta hacia otro puerto. En segundo lugar, la zona del patio es el punto del puerto donde los contenedores son almacenados. Por último, la zona del lado continental es la sección que conecta la terminal con el transporte de carretera o ferroviario.

A la hora de medir la competitividad de las operaciones que tienen lugar en la zona del lado del mar de una terminal portuaria, uno de los índices más utilizados es el tiempo requerido para que un barco sea servido desde su llegada al puerto. En este sentido, las operaciones del lado del mar, en especial aquellas que relacionan el muelle, están directamente relacionadas con el servicio que se le ofrece a los barcos. Asimismo, los atraques, además de su papel relevante a la hora de dar servicio a los barcos, adquieren un papel fundamental como punto de conexión entre distintos modos de transporte que tienen lugar en una terminal de contenedores. Es por tanto prioritario para los gestores del puerto hacer un uso

apropiado de estos recursos tan limitados e influyentes en el desempeño global de la terminal.

La discusión previa lleva a la definición del problema logístico portuario conocido como el Problema de Asignación de Atraques (BAP). Este problema surge en las terminales de contenedores con el objetivo de asignar posiciones de atraque a lo largo del muelle a barcos entrantes de forma eficiente. En este proceso, los responsables de la terminal de contenedores tienen que considerar factores tales como las ventanas de tiempo de los barcos y los atraques, el número de contenedores que han de cargarse o descargarse, la profundidad de los atraques, el calado de los barcos, las operaciones de trasbordo de contenedores, el resultado de las negociaciones con las navieras, etc.

En este trabajo se estudia el Problema de Asignación de Atraques en los distintos niveles de planificación y se analiza la aplicación de técnicas de optimización avanzadas para su resolución. Para ello, se presentan dos algoritmos, uno basado en la Búsqueda Tabú y otro en Búsqueda por Entorno Variable. Para medir la efectividad de los métodos propuestos se utilizan un conjunto de instancias basadas en mediciones reales de la terminal de Gioia Tauro (Italia).

El resto de este trabajo está organizado de la siguiente forma. En la Sección 2 describe el Problema de Asignación de Atraques y los niveles de planificación en donde ha sido tratado. En la Sección 3 se muestra la experiencia computacional derivada de aplicar técnicas de optimización para la resolución de estos problemas. La experiencia computacional realizada en este trabajo se presenta en la Sección 4. Finalmente, en la Sección 5 se enumeran las conclusiones y líneas de trabajo futuras.

2 El Problema de Asignación de Atraques

Debido a la variedad de las operaciones que tienen lugar en una terminal de contenedores, se puede establecer una clasificación de las mismas atendiendo al horizonte de planificación y control en el que se enmarcan. De acuerdo con Vis y Koster [11], se pueden definir tres niveles de planificación. Estos son los niveles estratégico, táctico y operacional. En el nivel estratégico, el horizonte de planificación va de uno a varios años y las tareas que se llevan a cabo son relativas a la estructura y material de la terminal portuaria. En el nivel táctico, el horizonte de planificación varía de una semana a varios meses y se decide qué utilización y planificación de la terminal se puede hacer. Dentro del nivel operacional, el horizonte de planificación va de un día a varias semanas y se toman decisiones para los problemas diarios de la terminal.

La relevancia del Problema de Asignación de Atraques (BAP) en la productividad global de una terminal, y su importante relación con otros problemas afines que surgen en la misma, impulsaron el estudio de este problema en los niveles operacional y táctico. En este sentido, a pesar de que el objetivo del BAP sigue siendo el mismo, esto es, ubicar barcos entrantes en posiciones de atraque, la finalidad de definirlo en niveles de planificación diferentes persigue contemplar e incorporar los distintos aspectos prácticos que surgen en el con-

texto de una terminal portuaria. Mientras que en el nivel operacional se tiene una consideración más técnica (estado de las mareas, profundidad de los barcos, etc.), en el nivel táctico, gracias a su horizonte de planificación mayor, se pueden incluir operaciones que responden más a las necesidades de gestión comercial o contractual. Como se describe en la sección 2.2, en este nivel se puede incorporar el proceso de negociación entre la terminal portuaria y las navieras teniendo en cuenta los recursos de la terminal (atraques y grúas de muelle).

2.1 Nivel Operacional

En el presente trabajo se tratan variantes de los problemas de asignación de atraques dinámicos discretos a nivel operacional y a nivel táctico introducidos por Imai et al. [10] y Giallombardo et al. [7], respectivamente.

En estos problemas se asume que el muelle se encuentra dividido en secciones, llamadas atraques, y en su resolución se tienen en cuenta a los barcos que arribarán al muelle en algún instante dentro del horizonte de planificación.

El Problema de Asignación de Atraques a nivel operacional (OBAP) ha sido ampliamente estudiado por la comunidad científica llegando a denominarse frecuentemente como BAP.

El problema de asignación de atraques dinámico discreto a nivel operacional (OBAP) fue introducido por Imai et al. [10], quienes lo formularon como un problema de programación entera mixta. Cordeau et al. [3] presentan una nueva formulación que incluye ventanas de tiempo tanto para barcos como atraques. Además, desarrollan una Búsqueda Tabú con una estrategia de recomienzo para su resolución. Para evaluar la efectividad del modelo y de la heurística se presenta un conjunto de instancias. En el trabajo presentado por Buhrkal et al. [1], se describen y comparan este y otros modelos para el OBAP. La principal conclusión es que el modelo presentado por Christensen y Holst [2] (GSPP) es superior a los otros modelos, obteniendo el valor óptimo en un tiempo computacional notablemente inferior para las instancias presentadas por Cordeau et al. [3]. Sin embargo, como muestran Lalla et al. [11], el modelo GSPP no es útil para resolver instancias del problema de dimensión realista debido a los requerimientos de memoria y tiempo de CPU necesarios. Otro de los aspectos señalados en ese trabajo es la necesidad de desarrollar métodos que ofrezcan soluciones de alta calidad en tiempos muy pequeños. Esta necesidad radica en la ejecución de los mismos en marcos integrados con otros problemas afines, donde los datos del OBAP varían provocando que su resolución se repita múltiples veces para un mismo problema. Este es el caso de la integración con el Problema de Planificación de Grúas de Muelle (QCSP), ya que, como señalan, por ejemplo, Expósito et al. [7] el tiempo de servicio de los barcos depende del número y la calidad de servicio dado por las grúas del muelle asignadas a cada barco.

En la variante del OBAP considerada por Cordeau et al. [3] se pretende determinar la posición y el tiempo de atraque de cada barco que arriba al puerto a lo largo de un horizonte temporal bien definido. Se dispone un conjunto de barcos V, un conjunto de atraques B. Cada barco $i \in V$ debe ser asignado a un atraque vacío $j \in B$ dentro de su ventana de tiempo $[tv_i, tv_i']$ y teniendo en cuenta la

```
procedure T2S* + PR;
begin
     s_{R-G} = R-G();
     s_{FCFS-G} = FCFS-G();
     ES = T2S*_{min}(s_{R-G});
     ES = ES∪ T2S*_{min}(s_{FCFS-G});
     repeat
          s_{initial} = R-G();
          s_{elite} = Select(ES);
          s = PR(s_{initial}, s_{elite});
          ES = ES∪ T2S*_{min}(s);
     until (stopping condition);
end.
```

Figura 1: Búsqueda Tabú con recomienzo basado en Reencadenamiento con Trayectorias

ventana de tiempo del atraque asignado $[tb_j, tb'_j]$. El objetivo principal del DBAP es minimizar el tiempo de espera y servicio de cada barco. Una explicación más detallada puede consultarse en el trabajo de Cordeau et al. [3].

Para mejorar la eficacia y eficiencia de la Búsqueda Tabú ($\mathtt{T^2S}$) presentada por Cordeau et al. [3] se propone en este trabajo la incorporación de un Reencadenamiento de Trayectorias como estrategia de recomienzo. Para ello se crea un conjunto élite (CE) con los óptimos locales obtenidos durante la ejecución de la Búsqueda Tabú $\mathtt{T^2S}$ que es utilizado para intensificar la búsqueda en regiones prometedoras de la región factible. Los elementos de este conjunto se usarán para crear puntos de inicio de nuevas búsquedas. Tomando como guía una solución élite, s_{elite}, y como solución inicial, $s_{inicial}$, suministrada por el procedimiento `R-G` que se describirá más adelante, se aplica el procedimiento basado en Reencadenamiento de Trayectorias que, iterativamente, acerca $s_{inicial}$ a s_{elite}. Desde el punto medio de este camino se aplica una nueva búsqueda $\mathtt{T^2S}$ con cuyos óptimos locales se actualiza el conjunto élite. Estos pasos se repiten hasta que se cumpla el criterio de parada. Llamaremos $\mathtt{T^2S}$ + `PR` a nuestra propuesta.

Para generar la solución inicial de $\mathtt{T^2S}$, Cordeau et al. [3] proponen dos procedimientos: `R-G` (*Random Greeedy*) y `FCFS-G` (*First Come, First Served - Greedy*). En `R-G`, dada una permutación aleatoria de los barcos, éstos se asignan uno a uno siguiendo este orden al mejor atraque posible; en `FCFS-G`, los barcos se ordenan según su tiempo de llegada y se asignan como en `R-G`.

Sean s_{R-G} y s_{FCFS-G} las soluciones obtenidas al usar, respectivamente, los métodos `R-G` y `FCFS-G`. Sean, asimismo, $\mathtt{T^2S}_{min}(s_{R-G})$ y $\mathtt{T^2S}_{min}(s_{FCFS-G})$ los conjuntos de óptimos locales obtenidos al ejecutar $\mathtt{T^2S}$ desde s_{R-G} y s_{FCFS-G}, respectivamente. En Figura 1 se muestra el pseudocódigo de $\mathtt{T^2S}$ + PR.

El Reencadenamiento de Trayectorias procede como sigue. Sean $s(i)$ la solución actual del camino y $B(s(i))$ el conjunto de barcos con asignaciones distintas en $s(i)$ y en s_{elite}. Para obtener la siguiente solución del camino, $s(i+1)$, se toma

un barco $j \in B(s(i))$ y se le asigna el atraque y la posición en el atraque que tiene en s_{elite}. Estos pasos se iteran hasta alcanzar la solución guía.

2.2 Nivel Táctico

El Problema de Asignación de Atraques en el nivel de planificación táctico, a diferencia del OBAP, no cuenta con muchos trabajos en la literatura relacionada. Formalmente fue presentado por Giallombardo et al. [5] como el Problema de Asignación de Atraques Táctico (TBAP). Sin embargo, se pueden encontrar trabajos previos tales como los presentados por Moorthy et al. [9] y Cordeau et al. [4] que incluyen aspectos propios del nivel táctico.

En este trabajo se aborda el Problema Táctico de Asignación de Atraques presentado por Giallombardo et al. [5]. Su objetivo es determinar la posición y el tiempo de atraque, y el número de grúas que atenderán cada barco que llegue al puerto a lo largo de un horizonte temporal bien definido. En el TBAP, tenemos un conjunto de barcos V, un conjunto de atraques B y un número máximo de grúas Q. Cada barco $v \in V$ debe ser asignado a un atraque vacío $b \in B$ dentro de su ventana de tiempo $[t_v, t'_v]$. Las posición de atraque de cada barco tiene que ser cercana a la posición de salida de sus contenedores. En este sentido, los costes de housekeeping, $h_{ss'}$, representan el coste asociado de mover los contenedores entre las posiciones s y s' del muelle (Giallombardo et al. [5]). Para cada barco $v \in V$, el tiempo de servicio, s_v, es definido de acuerdo a los contenedores incluídos en su plan de estiba y las negociaciones entre el barco y la terminal. Para ello, se introducen los perfiles de grúa, que se utilizan para determinar la distribución del número de grúas a lo largo del tiempo hasta que el barco sea servido. El conjunto de posibles perfiles de grúas es denotado por P. El coste de usar cada perfil $p \in P$ es c_p. El objetivo principal del TBAP es maximizar el valor de los perfiles de grúas asignados a los barcos y minimizar el coste de housekeeping derivado del trasbordo de contenedores entre barcos. Una explicación más detallada puede consultarse en el trabajo de Giallombardo et al. [5].

En la Figura 2 se muestra un ejemplo de una planificación para el TBAP con $|B| = 3$ atraques y $|V| = 3$ barcos y un número máximo de $Q = 8$ grúas. Para cada barco $i \in V$ la carga de trabajo necesaria viene representada por m_i, definida como el número de horas de grúa solicitadas para el mismo. Los bloques debajo de los rectángulos representan la distribución de grúas que servirán al barco, esto viene establecido por el perfil de grúa. En este caso, para el barco 1 se solicitan 10 horas de grúas y el perfil asociado para dar este servicio establece que será servido por 2, 3, 3 y 2 grúas respectivamente.

Para resolver el TBAP se presenta una Búsqueda por Entorno Variable (VNS). Esta metaheurística ha sido utilizada de forma satisfactoria en un gran número de problemas optimización. Su utilización se basa en el cambio sistemático de varias estructuras de entornos (Hansen y Mladenović [6]). En el Algoritmo 1 se muestra el pseudocódigo de la VNS propuesta para resolver el TBAP. La VNS utiliza dos estructuras de entornos: (a) basada en el movimiento de reinserción $N_a(S, \lambda)$, esto es, λ barcos son removidos del atraque $b \in B$ y

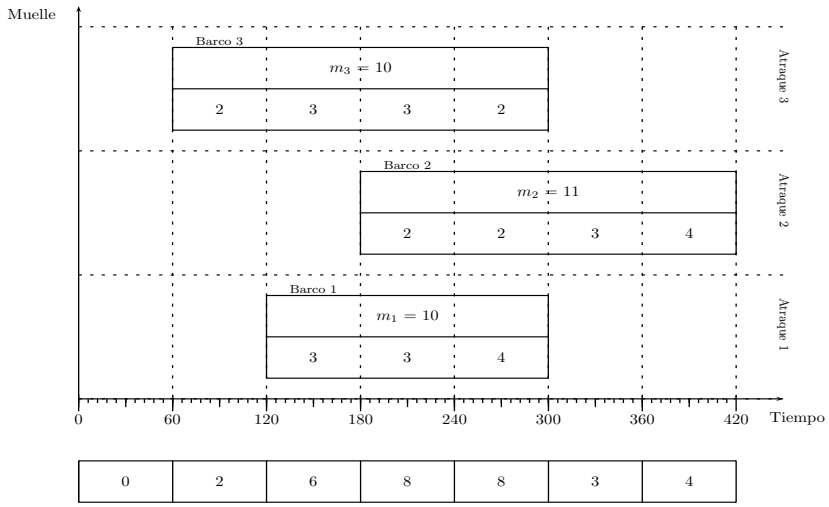


Figura 2: Ejemplo de planificación para el TBAP con 3 barcos y 3 atraques

reinsertados en otro atraque b', sujeto a $b \neq b'$, (b) basada en el movimiento de intercambio $N_b(S)$, este consiste en cambiar el barco $v \in V$ asignado al atraque $b \in B$ con otro barco $v' \in V$ asignado en el atraque b', sujeto a $b \neq b'$.

La solución de comienzo de la VNS, S, se genera asignando el perfil $p \in P$ con el mayor valor c_p a cada barco. La posición de atraque de cada barco se selecciona de forma aleatoria. Además, el comienzo del servicio al barco será el más pronto posible de acuerdo a su ventana de tiempo y a la disponibilidad del atraque (línea 1). El valor del parámetro k se fija a 1 (línea 2). El proceso de agitación (línea 4) permite escapar de los óptimos locales encontrados durante la búsqueda utilizando la estructura de entorno N_a. El proceso de búsqueda del algoritmo VNS se basa en una Búsqueda por Entorno Variable Descendente (VND) (líneas 5−14). El funcionamiento de la VND se basa en, dada una solución S', explorar el entorno de esta solución utilizando un movimiento cada vez hasta encontrar un óptimo local. Para ello se utilizan las estructuras de entorno N_a y N_b, respectivamente. La aplicación de las estructuras de entorno en la VND se realiza de acuerdo con el valor del parámetro k_1, inicializado a 1 (línea 5). La primera estructura de entorno explorada es N_a y luego N_b. La mejor solución encontrada en términos de valor objetivo es denotada como S'. El valor objetivo de esta solución S' permite actualizar la mejor solución encontrada durante la búsqueda (denotada por S) y reiniciar k (líneas $15 - 17$). En caso contrario, el valor de k se incrementa (línea 19). Estos pasos son repetidos hasta que $k = k_{max}$ (línea 21).

Algoritmo 1 VNS for the TBAP

```
1: S ← Generar solución inicial
2: k ← 1
3: repeat
4:     S' ← Shake(S, k)
5:     k_1 ← 1
6:     repeat
7:         S'' ← Local Search(S, k_1)
8:         if f(S'') > f(S') then
9:             S' ← S''
10:            k_1 ← 1
11:        else
12:            k_1 ← k_1 + 1
13:        end if
14:    until k_1 = k_1max
15:    if f(S') > f(S) then
16:        S ← S'
17:        k ← 1
18:    else
19:        k ← k + 1
20:    end if
21: until k = k_max
```

3 Resultados Experimentales

En esta sección se presenta y discute la experiencia computacional desarrollada para evaluar el rendimiento de las técnicas presentadas. Toda la experimentación realizada en este trabajo ha sido llevada acabo en un ordenador equipado con una procesador Intel 3.16 GHz y 4 GB of RAM.

Las Tablas 1 y 2 muestran la comparativa realizada para el OBAP entre la heurística presentada por Cordeau et al. [3], el modelo exacto GSPP [2] implementado en CPLEX [1] y la heurística propuesta en este trabajo $\mathtt{T^2S^*}$ + PR para las instancias presentadas en [3] y [8], respectivamente. El fundamento de esta comparativa es determinar el rendimiento de $\mathtt{T^2S^*}$ + PR y, como se señala en la sección 2.1, determinar si este algoritmo es apropiado para su integración con otros problemas relacionados.

Las primeras tres columnas de las tablas 1 y 2 indican el número de barcos ($|V|$), número de atraques ($|B|$) y el problema (*inst.*). Los valores óptimos dados por la formulación matemática GSPP y el tiempo de ejecución *cpu(s.)* en segundos requerido por CPLEX para resolver este modelo es mostrado en las columnas 4 y 5 (los valores mostrados en la Tabla 1 se encuentran en [1]). Las columnas 6 y 7 presentan el valor objetivo (*obj.*) obtenido por $\mathtt{T^2S}$ y presentado en [3] y la desviación respecto al valor óptimo (*desv.*). Las siguientes tres columnas muestran el valor objetivo mínimo, el tiempo de ejecución en segundos proporcionado por $\mathtt{T^2S^*}$ + PR, y la desviación respecto al valor óptimo.

Los resultados mostrados en la Tabla 1 indican que $\mathtt{T^2S^*}$ + PR es capaz de encontrar soluciones óptimas o cercanas a la óptima, en tiempos de ejecución inferiores respecto al modelo matemático GSPP y $\mathtt{T^2S}$. En este sentido, es apreciable la diferencia de calidad de las soluciones entre ambos métodos heurísticos,

[1] http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/index.html

donde la desviación media en el peor de los casos es 0,06 para T^{2}S* + PR mientras que es 1,09 para T^{2}S. Otro aspecto relevante es la ventaja temporal de T^{2}S* + PR, siendo 18 y 12 veces inferior de media respecto el GSPP y T^{2}S para las instancias más grandes respectivamente (25 barcos, 10 atraques). Por otro lado, como se muestra en la Tabla 2, cuando las instancias presentan una mayor complejidad y dimensión, GSPP es incapaz de encontrar una solución factible dado que necesita una gran cantidad de memoria. Respecto a los métodos heurísticos empleados, la desviación de T^{2}S comparada con la desviación de T^{2}S* + PR muestra que T^{2}S* + PR es capaz de mejorar la calidad de las soluciones en un tiempo de ejecución inferior en todas las instancias. Estos resultados motivan y justifican su aplicación en integraciones donde el BAP esté inmerso y su resolución se realice de forma frecuente.

			GSPP		T^{2}S			T^{2}S* + PR		
$\|V\|$	$\|B\|$	*inst.*	*obj.*	*cpu(s.)*	*obj.*	*cpu(s.)*	*dev.*	*obj.*	*cpu(s.)*	*dev.*
25	7	1	657	3,62	667	4,00	1,52	657	0,23	0,00
		2	662	3,15	671	4,00	1,36	662	0,20	0,00
		3	807	4,28	823	4,00	1,98	807	0,23	0,00
		4	648	3,78	655	4,00	1,08	650	0,22	0,31
		5	725	3,85	728	4,00	0,41	725	0,22	0,00
		6	794	3,60	794	4,00	0,00	794	0,24	0,00
		7	734	3,54	740	4,00	0,82	734	0,23	0,00
		8	768	3,93	782	4,00	1,82	768	0,21	0,00
		9	749	3,73	759	4,00	1,34	749	0,20	0,00
		10	825	3,82	830	4,00	0,61	825	0,24	0,00
				3,73		**4,00**	**1,09**		**0,22**	**0,03**
	10	1	713	5,83	717	4,00	0,56	713	0,35	0,00
		2	727	6,99	736	4,00	1,24	727	0,34	0,00
		3	761	6,12	764	4,00	0,39	764	0,34	0,39
		4	810	5,38	819	4,00	1,11	811	0,33	0,12
		5	840	6,77	855	4,00	1,79	841	0,32	0,12
		6	689	5,57	694	4,00	0,73	689	0,34	0,00
		7	666	5,83	673	4,00	1,05	666	0,36	0,00
		8	855	5,87	860	4,00	0,58	855	0,30	0,00
		9	711	5,38	726	4,00	2,11	711	0,34	0,00
		10	801	5,96	812	4,00	1,37	801	0,33	0,00
				5,97		**4,00**	**1,09**		**0,33**	**0,06**

Tabla 1: Comparativa entre GSPP, T^{2}S y T^{2}S* + PR. Instancias propuestas en [3].

			GSPP		T^{2}S		T^{2}S* + PR		
$\|V\|$	$\|B\|$	*inst.*	*obj.*	*cpu(s.)*	*obj.*	*cpu(s.)*	*obj.*	*cpu(s.)*	*dev.*
40	7	1	-	-	1489	8,16	1460	1,11	-1,95
		2	-	-	1423	8,26	1375	1,32	-3,37
		3	-	-	2149	8,13	2119	1,17	-1,40
		4	-	-	1618	7,99	1597	1,78	-1,30
		5	-	-	1885	8,37	1847	1,45	-2,02
		6	-	-	2104	8,11	2080	1,37	-1,14
		7	-	-	1863	7,91	1841	1,56	-1,18
		8	-	-	2040	8,16	2026	1,70	-0,69
		9	-	-	1901	8,01	1880	1,48	-1,10
		10	-	-	1922	7,89	1892	1,59	-1,56
						8,01		**1,45**	**-1,57**

Tabla 2: Comparativa entre GSPP, T^{2}S y T^{2}S* + PR. Instancias propuestas en [8].

En la Tabla 3 se presenta la comparativa realizada para el TBAP. Se muestran los resultados obtenidos con el modelo matemático implementado en CPLEX, una Búsqueda Tabú combinada con un método de Branch and Price (TS-BP) desarrollado por Giallombardo et al. [5] y la VNS propuesta. La primera columna (*Inst.*) muestra las instancias utilizadas. Para cada instancia, se muestra el nombre (*Name*), el número de barcos (V) y atraques (B) y el número de perfiles de grúa disponibles para cada barco (P). La columna *CPLEX* muestra los resultados obtenidos por CPLEX con un tiempo de ejecución límite de 2 horas. Las columnas (*TS-BP* y *VNS*) muestran los resultados obtenidos por el TS-BP y VNS con un $k_{max} = 5$ y $k_{1max} = 2$, respectivamente. En cada caso, el valor objetivo es escalado hasta 100 con respecto al límite superior proporcionado por CPLEX, UB_{CPLEX}.

Inst.				*CPLEX*	*TS-BP*		*VNS*	
Name	V	B	P	UB_{CPLEX}	f_{BP-TS}	t. (s)	f_{VNS}	t. (s)
A_{p10}	20	5	10	1383614	97.26	81	97.25	4.86
A_{p20}	20	5	20	1384765	97.19	172	96.99	10.49
A_{p30}	20	5	30	1385119	97.37	259	96.94	12.23
B_{p10}	30	5	10	1613252	95.68	308	96.66	12.46
B_{p20}	30	5	20	1613769	95.12	614	96.20	26.55
B_{p30}	30	5	30	1613805	—	920	96.33	59.71
C_{p10}	40	5	10	2289660	97.41	1382	97.50	68.47
C_{p20}	40	5	20	2290662	97.37	3144	97.62	99.04
C_{p30}	40	5	30	2291301	96.60	4352	96.72	223.1

Tabla 3: Comparativa entre CPLEX, TS-B&P [5] y la VNS propuesta.

Los resultados experimentales mostrados en la Tabla 3 indican que la VNS propuesta es bastante competitiva para un amplio rango de instancias. Esta provee los mejores resultados en tiempos de ejecución inferiores, menos de 225 segundos en el peor de los casos.

4 Conclusiones y Trabajos Futuros

En el presente trabajo se ha analizado la aplicación de técnicas de optimización avanzadas para resolver el Problema de Asignación de Atraques en los distintos niveles de planificación de una terminal portuaria. Las metaheurísticas presentadas han mostrado, a través de la experiencia computacional, ser apropiadas para abordar de forma satisfactoria el Problema de Asignación de Atraques en los niveles operacional y táctico. Estos resultados motivan el uso de técnicas de optimización como las propuestas en este trabajo para resolver de forma efectiva estos problemas y ser consideradas como herramientas de apoyo en la toma de decisiones durante la planificación de atraques de una terminal de contenedores. En este sentido, los métodos propuestos presentan una ventaja temporal relevante lo cual aconseja su aplicación en marcos integrados donde el BAP sea tratado de forma conjunta con otros problemas logísticos dentro del puerto.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el Ministerio de Economía y Competitividad de España (project TIN2012-32608). Eduardo Lalla Ruiz agradece

al Gobierno de Canarias su apoyo financiero recibido a través de su beca de doctorado.

Referencias

1. Katja Buhrkal, Sara Zuglian, Stefan Ropke, Jesper Larsen, and Richard Lusby. Models for the discrete berth allocation problem: a computational comparison. *Transportation Research Part E*, 47(4):461–473, 2011.
2. C.G. Christensen and C.T. Holst. Berth allocation in container terminals. Master's thesis, Department of Informatics and Mathematical Modelling, Technical university of Denmark, 2008. In Danish.
3. J. F. Cordeau, G. Laporte, P. Legato, and L. Moccia. Models and tabu search heuristics for the berth-allocation problem. *Transportation Science*, 39:526–538, 2005.
4. Jean-Franois Cordeau, Manlio Gaudioso, Gilbert Laporte, and Luigi Moccia. The service allocation problem at the gioia tauro maritime terminal. *European Journal of Operational Research*, 176(2):1167 – 1184, 2007.
5. G. Giallombardo, L. Moccia, M. Salani, and I. Vacca. Modeling and solving the tactical berth allocation problem. *Transportation Research Part B: Methodological*, 44(2):232–245, 2010.
6. Pierre Hansen and Nenad Mladenović. Variable neighborhood search. In Edmund K. Burke and Graham Kendall, editors, *Search Methodologies*, pages 211–238. Springer US, 2005.
7. Christopher Expósito Izquierdo, José Luis González Velarde, Belén Melián Batista, and J. Marcos Moreno-Vega. Hybrid estimation of distribution algorithm for the quay crane scheduling problem. *Applied Soft Computing*, (0):–, 2013.
8. E. Lalla-Ruiz, B. Melián-Batista, and J.M. Moreno-Vega. Artificial intelligence hybrid heuristic based on tabu search for the dynamic berth allocation problem. *Engineering Applications of Artificial Intelligence*, 25(6):1132–1141, 2012.
9. Rajeeva Moorthy and Chung-Piaw Teo. Berth management in container terminal: the template design problem. *OR Spectrum*, 28(4):495–518, 2006.
10. Matthew E.H. Petering. Decision support for yard capacity, fleet composition, truck substitutability, and scalability issues at seaport container terminals. *Transportation Research Part E: Logistics and Transportation Review*, 47(1):85 – 103, 2011.
11. Iris F.A. Vis and René de Koster. Transshipment of containers at a container terminal: An overview. *European Journal of Operational Research*, 147(1):1 – 16, 2003.

MOAMP-Tabu Search and NSGA-II for a Bi-objective Urban Waste Collection Problem: A Head to Head Comparison

Joaquín Pacheco, José Rubén Gómez and Hernán Gonzalo-Orden

{jpacheco,jrgomez,hgonzalo}@ubu.es

Universidad de Burgos, Spain

Abstract. A bi-objective model for the collection of waste bins in rural areas within a planning period is analyzed. This model is a real problem raised by local authorities in a rural region of northwestern Spain. The two objectives under consideration are: minimization of transport costs and improvements to the level of service. Specifically, the level of service depends on the frequency of waste collection at each point over the planning period. A solution method for this problem was developed by applying tabu search within the framework of MOAMP (Multiobjective Adaptive Memory Programming) and the results were compared with an implementation of NSGA-II, a well-known approach to multi-objective optimization.

Keywords: Urban Waste Collection, Waste Managemente, Bi-objective Problems, Routes, Service, Tabu Search, MOAMP, NSGA-II

1 Introduction

1.1 Literature review on waste collection problems in rural areas

Among the routing problems, a very interesting and actual class is the set of problems that deal with the collection of urban waste (UW). In general, the literature describes two principal ways in which a real UW problem can be formulated. The first way is to formulate the problem as an "arcs problem", that is, as the Capacitated Arc Routing Problem (CARP) or some of its variants. The second way is to formulate the problem as a "nodes problem"; such as the Capacitated Vehicle Routing Problem (CVRP), or some of its variants. This second type of problem occurs principally in rural areas. Over the past few years, groups of small villages (geographically nearby) have jointly managed different

services, such as the collection of UW. In this case, the UW are collected in each small village (node).

In [1] is described a dynamic model based on Markov's process and a real application to the city of Chicago which examined real waste collection problems formulated as CVRP. In [2] are described a method based on GVNT (Guided Variable Neighborhood Thresholding) metaheuristic strategy and an application in eastern Finland. In [3] is described a problem with time windows in Texas, USA. In [4] is proposed professional software for optimizing routes. This software was applied in the regions of Turkey, achieving cost savings of 24%. More recently, in [5], is proposed a system for the waste collection (including both, organic and recyclable residuals) in the district of Hampshire, UK.

1.2 Description of a real bi-objective waste collection problem

In this work, a bi-objective model for the planning of waste collection is analyzed. This model is based on a real problem proposed to the authors of this work by the local authorities of a set of small villages in the southeast of the province of Burgos, northwestern Spain.
Regional and local authorities are interested in "the application of the best available techniques" in order to rationalize their waste collection services: reducing the cost and improving the service level offered to the public. Specifically it is a matter of collecting waste (accumulated in containers or bins) in each of the villages and of transporting these waste bins to a central deposit. Monthly planning periods are considered. Planning decisions involve selecting the days on which to visit each point and designing the corresponding daily routes. In addition to the typical restrictions of the problems of routes (vehicle capacities, a vehicle must visit every point, etc.) a further restriction must be borne in mind: the number of days between two consecutive visits to each small village must not exceed a preset quantity ("maximum separation"). This maximum separation depends on the size of each village. Next, both objectives (cost and quality) are analyzed.

The cost of the service is the total sum of the transport costs for each route throughout the whole planning period. The transport cost of each route is the sum of a fixed cost and a variable cost. The fixed cost depends on the class of vehicle (whether a local authority fleet or a contracted fleet) and on the type of day (Labour day, Saturday and Holidays including Sundays). The variable cost is proportional to the travelled distance.

The local authorities measure the service (or quality) level in the following way: the service level in each population is worse when more UW accumulates at this population in time. In this regard, sanitary criteria (degradation of the residues, bad smells, appearance of rodents, and so on) should be taken into account, especially in the summer. Therefore, an "intuitive" measurement of service "non-quality" (accepted by the authorities), for every small village, is the sum of the quantity of accumulated UW “at all times” throughout the whole planning period: i.e. the integral of the function of the quantity of accumulated UW in each small village within the planning period. Figure 1 shows an example of this function. More specifically, it shows the accumulated UW in a small village where these UW are collected on days 0 (start of the planning period), 1, 3 and 6. There are therefore 3 spaces between consecutive visits (lasting 1, 2 and 3 days respectively) and 3 sub-areas (namely S_1, S_2 and S_3).

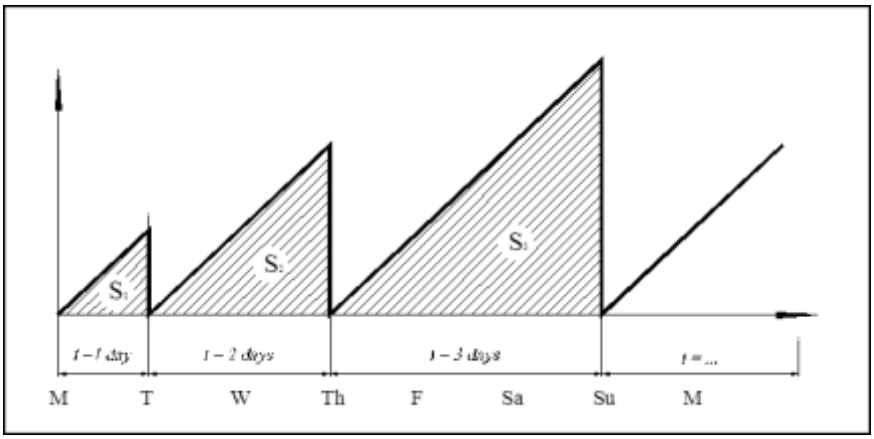


Fig. 1. Function of the quantity of accumulated UW in a small village

Improving service quality in a small village is equivalent to reducing the integral of the function of the quantity of its accumulated UW (see figure 1, $S_1 + S_2 + S_3$). Each of these surfaces S_i is proportional to the size of the population and to the square of the number of days that pass between each of the corresponding visits. The total "non-quality" of the service is defined as the sum of the integrals corresponding to all of the small villages. The more the number of visits to collect UW in a small village, the smaller its corresponding integral (i.e. a better service level) and the higher the transportation cost.

To the best of our knowledge this problem has not been tackled in the past so far. Our goal is to develop a solution procedure capable of approximating the efficient frontier of this bi-objective problem (BUWCP — Bi-objective Urban Waste Collection Problem). The frontier would enable a decision maker to analyze the tradeoff between cost and service. In this work we developed solution method for the BUWCP that is is based on MOAMP (Multiobjective Adaptive Memory Programming).

To assess the merit of our method, we compare its performance against an adaptation of the well-known NSGA-II procedure. NSGA-II is a general-purpose genetic algorithm that has enjoyed a high-level of success when applied to multi-objective optimization problems in a variety of settings. In addition, for real-data instances, we compared our solutions with the solutions that were used by the local authorities at the time.

The work is organized as follows: the following section introduces certain definitions and notation that help to describe the problem and the methods used for its solution. Sections 3 provide a description of our adaptation of the MOAMP method to the problem. Section 4 sets out the adaptation of the NSGA-II method and the results of some computational experiments are explained in section 5. Finally, the conclusions are developed in section 6.

2 Notation and Definitions

The following notation is useful to explain our work:

n : Number of collection points, where 0 represents the central depot (origin and end of routes) and $\{1, \dots, n\}$ is the set of villages
HT : Number of days of planning period
q_i : Quantity (in kilograms) of UW generated in point i every day
sp_i : Maximum separation (in days) allowed between 2 consecutive collections at point i
d_{ij}, t_{ij} : Respectively, distance and time between location i and location j
cte_{vp} : Fixed cost of vehicles of class v in days of type p (in monetary units, m.u.)
m_v : Number of vehicles of class v
m : Total number of vehicles (Obviously $m = \sum_{v=1}^{ntv} m_v$)
Q_l : Capacity of vehicle l
ckm : Cost for every traveled kilometer (in m.u.)
$f_1(S)$: Total cost in solution S
$f_2(S)$: "Non-quality" function value of solution

As previously pointed-out, the problem consists of determining the collection days at every point over the planning period and designing the corresponding daily routes. When a point (a small village) is visited, all the accumulated UW since the previous visit are collected. Therefore, the quantity of UW collected in every small village i is q_i multiplied by the number of days that have elapsed since the last visit. Planning should ensure that no more than sp_i days elapse between any two consecutive visits at every point i.

The fixed cost of every route is the fixed cost of the vehicle that is used. This cost (cte_{vp}) depends on the type of day p (Labor days, Saturday or Sunday) and on the type of vehicle v (local authority fleet or contracted fleet). Drivers of the local authority fleet have permanent contracts as public authority employees. These drivers have a fixed monthly salary but they also earn an extra salary for every route performed. This extra salary, the fixed cost of the route, depends on the type of the day (usu-

ally more on holidays or Saturdays than on Labor days). Sometimes, because of the small number of vehicles in the local authority fleet, the authorities contract vehicles and drivers from private firms (Contracted Fleet). The fixed cost of using these contracted vehicles depends on the type of day and is more expensive that the corresponding fixed cost of the local authority fleet. Our goal is to find a set of non-dominated solutions that approximates the set of efficient solutions (i.e., the efficient frontier).

3 Solution Method

Our proposed solution procedure is based on MOAMP (Multi-objective Adaptive Memory Programming). MOAMP [6] is a framework that distinguishes itself among numerous heuristic multi-objective optimization methods by emphasizing neighborhood search over mechanisms for evolving a population of solutions. The framework consists of three phases: 1) linked searches guided by the objective functions taken one at a time; 2) a search guided by a distance metric in the objective function value space that is based on compromise programming; and, 3) neighborhood searches around solutions for the best approximation of the efficient frontier. The first two phases result in an approximation of the efficient frontier and this set of non-dominated solutions is used as the initial point of departure for the third phase. This is accomplished with a mechanism that checks the status of each solution (i.e., dominated or non-dominated) visited during the first two phases. The non-dominated set is updated during the last phase as new non-dominated solutions are identified.

Our main contribution is the effective application of the MOAMP framework to the BUWCP. We point out that MOAMP is not a "plug-and-play" system that would require only a few changes to apply it to a problem. Instead, MOAMP is a conceptual structure that defines what to do (i.e., the three steps described above) but not how.

4 The NSGA-II method

To assess the merit of our proposed procedure, we have adapted the NSGA-II algorithm ([7])) that is universally accepted in the multiobjective optimization literature. NSGA-II is an improvement over NSGA (non-dominated sorting genetic algorithm) that deals with three major drawbacks of the original approach: 1) high computational cost of sorting, 2) lack of elitism and 3) lack of a parameter-free diversity-preservation mechanism. The NSGA-II computer code is available free of charge at http://www.iitk.ac.in/kangal/codes.shtml and although the software is set up for general use, certain context-dependent elements must be provided[1].

5 Computational experiences

We conducted our tests on a set of 20 artificial instances (but, as may be observed, with some value parameters that are similar to real instances) and four real instances. We considered 4 combinations of n and HT values and 5 instances for every combination. These combinations are shown in table 1.

combinations	n	HT
1	20	10
2	30	15
3	40	20
4	50	30

Table 1. Combinations of n and HT values for artificial instances

The rest of the parameters are defined and/or generated as follows:

[1] For more details about MAOMP and NSGA adaptations for BUWCP you can write to authors.

q_i : In kilograms. Takes a randomly generated integer value in the interval $[1, 100]$;

sp_i : sp_i takes a value 2 if $i \in \{1, \dots, {}^{n}/_{2}\}$ or otherwise 3;

d_{ij}, t_{ij} : Every point is assigned its x-y coordinates that takes a randomly generated integer value in the interval $[-100, +100]$ if $i \neq 0$. Then, d_{ij} is defined as the corresponding rounded Euclidean distance and $t_{ij} = d_{ij}$

The rest of the parameters are defined as follows: $m_1 = m_2 = 2$, thus $m = 4$; $Q_l = 1200\ \forall l$ and $ckm = 1$. Finally the fixed costs cte_{vp} for every class of vehicle and for every type of day are shown in table 2.

cte_{vp}	Labor	Saturday	Holiday
Local Auth. Fleet	0	100	200
Contracted Fleet	100	225	375

Table 2. Fixed costs of vehicles for every class of vehicle and for every type of day

We executed both methods with the 20 previously defined instances. So, both methods used the same computational time for each instance. All tests were performed on a Corel i7 2600 computer with a 3.4 GHz processer. The mean of results for every size are shown in table 3. Specifically for every method it is shown the number of non-dominated solutions (ND), the number of solutions found by this method that are dominated by a solution obtained by the other method (dom), and the best values found for both objectives (f_1^{min} and f_2^{min}).

Instances Size	MOAMP				NSGA-II			
	ND	dom	f_1^{min}	f_2^{min}	*ND*	dom	f_1^{min}	f_2^{min}
20-10	725	55.6	4622.8	10221.2	103.4	90.8	4944.6	12024
30-15	1906.2	0	9077.2	23872	72	71.6	9934.2	31371.6
40-20	4457.4	0	14484.8	40962.6	53	53	16558	58241.4
50-30	10979.4	0	30916	79391.6	30.2	30.2	35920.2	120166.2

Table 3. Means for each method and each instance size

MOAMP procedure performs better than the NSGA-II adaptation: MOAMP obtains a much denser set of solutions; almost none of the

solutions obtained by MOAMP are dominated by a solution of NSGA-II; in contrast, almost all solutions obtained by NSGA-II are dominated by a solution obtained by MOAMP. Moreover, MOAMP performs better if we consider the initial (cost and quality) objectives individually. These differences increase with the size of the problem. Figures 2 show a graphical representation of the approximations of the efficient frontier for instance 40-20-2 in order to illustrate the performance.

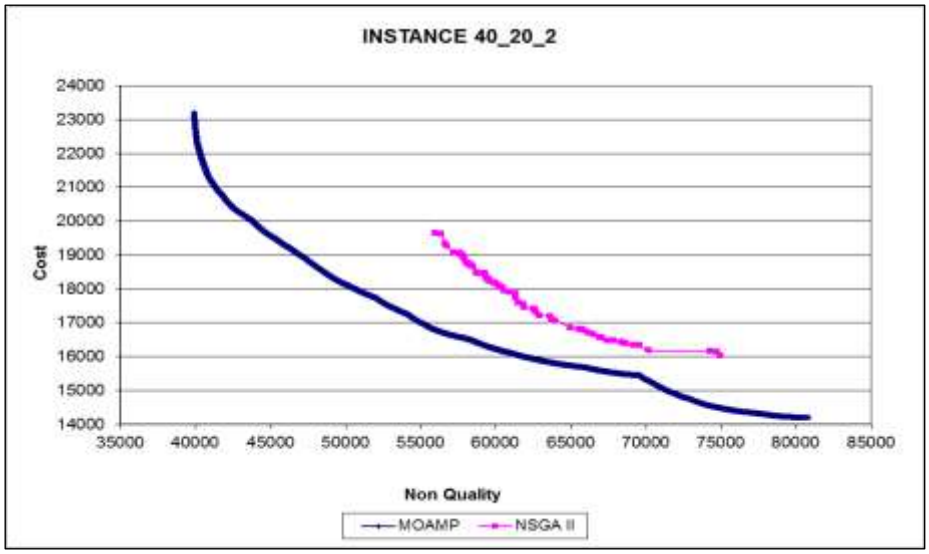


Fig. 2. Approximation of the efficient curve in instance 40-20-2

6 Conclusions

In this work, a real bi-objective residual collection problem in rural areas has been addressed. This problem required a schedule of daily routes over a lengthy planning period. Two decision levels were considered: 1) the villages to be visited every day and 2) the design of the corresponding daily routes. The two objectives under consideration were: minimization of transport cost and maximization of service level. This second objective dealt with the frequency of visit to each point. An approximation of the efficient frontier gives the decision-maker the ability to compare solutions and to analyze the trade-off between service-and

cost-related aspects of the problem. It should also be remarked that this is a high-dimensional problem (almost fifty villages over a period of one month). Our main contribution is the adaptation of the MOAMP framework to this problem in order to identify non-dominated solutions and to approximate the efficient frontier. We have performed experiments in order to compare the performance of our MOAMP adaptation and the well-known NSGA-II procedure. The experiments emphasized the merits of our approach and solidified the merit of MOAMP as a viable solution framework in the area of multi-objective optimization.

References

1. Eisenstein, D. D., and A. V. Lyer. (1997). "Garbage collection in Chicago: a dynamic scheduling model". *Management Science* 43 (7) pp. 922-933.

2. Kytojoki, J., T. Nuortio, O. Braysy, and M. Gendreau. (2007). "An efficient variable neighborhood search heuristic for very large scale vehicle routing problems". *Computers & Operations Research* 34 (9) pp. 2743-2757.

3. Kim, B. I., S. Kim, and S. Sahoo. (2006). "Waste collection vehicle routing problem with time windows". *Computers and Operations Research* 33 (12) pp. 3624-3642.

4. Apaydin, O., and M. T. Gonullu. (2008). "Emission control with route optimization in solid waste collection process: A case study". *Sadhana* 33 (2) pp. 71-82.

5. McLeod, F., and T. Cherrett. (2008). "Quantifying the transport impacts of domestic waste collection strategies". *Waste Management* 28 (11) pp. 2271-2278.

6. Caballero, R., J. Molina, and V. Rodríguez-Uría. 2003. MOAMP: Programación Multiobjetivo Mediante un Procedimiento de Búsqueda Tabú. Paper read at II Congreso Español de Metaheurísticas y Algoritmos Evolutivos y Bioinspirados (MAEB), at Gijón.

7. Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan. (2002). "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II". *IEEE Transactions on Evolutionary Computation* 6 (2) pp. 182–197.

Aplicación de retraso en tierra y secuenciación de vuelos para la gestión de incidencias en aterrizaje

A. Agustín[1], A. Alonso[2], A. Juan[3], J. Faulin[1]

[1]Universidad Pública de Navarra, Pamplona (Navarra), España
{albamaria.agustin,javier.faulin}@unavarra.es
[2]Universidad Rey Juan Carlos University, Móstoles (Madrid), España
antonio.alonso@urjc.es
[3]Universitat Oberta de Catalunya, Barcelona (Cataluña), España
ajuanp@uoc.edu

Resumen. El aumento de demanda en el tráfico aéreo contribuye a planificar una programación de vuelos muy ajustada. Y cuando se produce cualquier imprevisto, respecto a disminuir el número de pistas disponibles, probablemente la planificación inicial resultará inviable. En este trabajo se presenta un algoritmo multi-inicio que realiza pequeñas modificaciones al plan de vuelo inicial obteniéndose una nueva solución factible. El algoritmo propuesto hace uso de la combinación de dos importantes problemas en el control del tráfico aéreo (CTA), estos son, el problema multi-aeropuerto para la asignación del retraso en tierra y el problema de programación y secuenciación del aterrizaje de vuelos. En cada re-inicio, el algoritmo ordena la lista de vuelos conforme prioridades y aplica aleatorización sesgada. A continuación, y siguiendo esta ordenación de vuelos, realiza el proceso de programación y secuenciación de vuelos garantizando que los tiempos de seguridad entre el aterrizaje de los vuelos se verifican. Para el caso en que no se puedan mantener los tiempos de seguridad, se asignará al vuelo retraso en tierra. Algunos resultados computacionales contribuyen a validar nuestro algoritmo obteniéndose un gran ahorro en los costes, en pocos segundos de computación.

Palabras clave: Planificación de tráfico aéreo, Algoritmos aleatorizados, Metaheurísticas.

1 Introducción

En los últimos años, se ha experimentado un importante crecimiento del tráfico aéreo, haciendo uso intensivo de los aeropuertos y de sus pistas de aterrizaje y despegue cuyo crecimiento, sin embargo, no ha sido aumentado en consecuencia. Por lo tanto, el control de tráfico aéreo (CTA) realiza programaciones de vuelos cada vez más ajustadas y que son especialmente sensibles a la disponibilidad de las pistas. Luego, un incidente inesperado (derrame de petróleo, un vehículo atravesado, la reducción de visibilidad...) provoca de forma inmediata la congestión de los vuelos y, en consecuencia, una importante falta de seguridad.

En general, las técnicas más efectivas en problemas multi-aeropueto aplican retrasos en tierra de los vuelos (MAGHP) para resolver un conflicto de congestión. En 1973, de acuerdo con la primera crisis del petróleo, se asumen costes elevados para los retrasos en el aire y la política del Retraso en Tierra (GH) tuvo una amplia aplicación para la solución de este problema. En algunos casos, el problema de MAGHP ha sido resuelto con métodos exactos obteniéndose la solución óptima.

Por otra parte, los aviones deben seguir una programación y secuencia de aterrizaje (ALSS) que garantice la seguridad entre llegadas. Es decir, será necesario un tiempo mínimo de separación entre el aterrizaje de una aeronave y el aterrizaje de las aeronaves sucesivas. Por lo general, las consideraciones aerodinámicas de la aeronave en cuestión determinarán estos tiempos de separación. Diferentes heurísticas han sido desarrolladas para resolver este problema NP-completo Harikiopoulo y Neogi (2011).

En este trabajo se propone una meta-heurística que combina estos dos problemas, MAGHP y ALSS, y proporciona una respuesta inmediata para la congestión de vuelos cuando se reduce la disposición de pistas de aterrizaje. Con el algoritmo MAGHP-ALSS se obtiene una buena solución que minimiza los costes de retraso en tierra y los costes del adelanto o retraso en el proceso de aterrizaje, y a su vez satisface las restricciones de seguridad entre llegadas, es decir, asigna la secuencia de las aeronaves. En otras palabras, nos aseguraremos de que el vuelo no despegue hasta haber garantizado su aterrizaje con las condiciones de seguridad. Además, hemos incluido una política de cancelación para aquellos problemas en los no se puede garantizar la llegada de todos sus vuelos y que está asociada a una gran penalización. Estos problemas inviables resultan, por ejemplo, de un exceso de retrasos en tierra no admisible por el vuelo o del cierre de los aeropuertos, es decir, con cero pistas operativas.

Nuestro algoritmo considera la bien conocida tabla de prioridades y el llamado algoritmo ABG en Andreatta et al (1997), que proporcionará asignación de retraso en tierra, y el límite superior descrito en Beasley et al. (2000) para comprobar el tiempo de seguridad entre la llegada de los vuelos. Por otra parte, se propone un algoritmo multi-inicio con aleatorización sesgada en el orden de prioridades de los vuelos y con criterio de parada el tiempo (Juan et al. 2010). En otras palabras, vamos a evaluar diferentes ordenaciones para la lista de vuelos, para las que los vuelos con prioridades más altas serán más propensos a aterrizar primero que los que tienen menor prioridad.

El documento está estructurado de la siguiente manera: La sección 2 analiza algunos de los artículos más interesantes en MAGHP y ALSS. En la sección 3 del problema MAGHP-ALSS se describe formalmente. La sección 4 muestra el diagrama de flujo de nuestro enfoque y explica el algoritmo. En la sección 5 se presentan algunas tablas de la experiencia computacional, y pone de relieve los resultados más interesantes. La sección 6 resume las conclusiones e incluye investigación futura.

2 Estado del arte

El tráfico aéreo continuará aumentando hasta 2016 (Eurocontrol 2011) para lo que deberán considerarse diferentes estrategias según sea el horizonte temporal de planificación (Agustín et al, 2010 y Jakobovits, 2007). El problema MAGHP-ALSS perte-

nece a una planificación táctica proporcionando una respuesta para los incidentes que se producen en el día.

Conforme al problema MAGHP, los sistemas iniciales se basaron la experiencia de los controladores aéreos para tomar decisiones o incluso en la planificación Primero-Programado-Primero-Servido. Más tarde, la aplicación de la Investigación Operativa a este problema ha desarrollado de metodologías de toma de decisiones en las que la política de GH es bien reconocida. En la literatura, podemos destacar las primeras formulaciones del modelo matemático para este problema en Bertsimas y Stock (1998), que plantea la optimización lineal entera mixta 0-1, y el descrito en Andreatta y Brunetta (1998). Sin embargo, a finales de los 90's los métodos exactos aún necesitaban tiempos computacionales muy largos, y los algoritmos heurísticos que no requieren de un tiempo de respuesta tan elevado eran preferibles. Entonces, Andreatta et al. (1997) propuso la ABG-heurística basada en prioridades para el problema de congestión de vuelos en redes multi-aeropuerto. Navazio y Romanin-Jacur (1998) presentan otra versión del algoritmo de ABG que, así como con el ABG inicial, fueron probadas en una gran cantidad de casos en los que los resultados para ambas heurística eran muy similares (Brunetta et al. 1998).

Hoy en día, según se describe en Bertsimas et al. (2011) y Agustín et al. (2012), modelos complejos para la gestión del flujo de tráfico aéreo pueden ser resueltos en tiempos computacionales razonables, conforme al progreso de la informática en las últimas décadas. Además, los métodos heurísticos y meta-heurísticos complementan en aquellas características del tráfico aéreo que aún son difíciles de resolver de forma óptima. Zang et al. (2007) han desarrollado un algoritmo genético co-evolutiva que se aplica en los datos reales correspondientes a los Centros de CTA en Aviación Civil de China para Beijing, Shangai y Guangzhou. Wanke y Greenbaum (2007) presentan un modelo de simulación heurística que considera el enfoque estocástico del problema.

Conforme al problema ALSS, la política de CTA a menudo programa el aterrizaje de los aviones asumiendo que el primero que llega, será el primero que aterriza (FCFS). Desde el punto de vista de la investigación operativa, es necesario asumir las medidas de seguridad respecto al aterrizaje en pista. Algunas formulaciones de programación entera mixta 0-1 se han presentado para el problema ALSS. En Ernst et al. (1999) para una sola pista y Beasley et al. (2000) para múltiples pistas. Sin embargo, como los modelos tienen que evaluar tantas secuencias como el factorial de los vuelos, sólo podrían evaluarse los casos con pocos aviones. Dear y Sherif (1991) presentan la heurística Constraint Position Shifting para una sola pista del problema ALSS estático. Hu et al. (2009) proponen un algoritmo genético con cruzamiento uniforme para sistemas con múltiples pistas, y donde se consideran diferentes tiempos de llegada para cada aeronave según sea la pista de llegada.

Según nuestro conocimiento, ningún trabajo previo en la literatura tiene integrados estos dos problemas, el MAGHP y el ALSS. En nuestra opinión, considerar una incidencia que disminuye la disponibilidad pistas de aterrizaje en los aeropuertos es una hipótesis realista que no ha recibido suficiente atención en la literatura existente. En particular, en este trabajo se presenta un algoritmo para resolver el MAGHP-ALSS donde la capacidad de los aeropuertos viene dada en términos de pistas disponibles, con el fin de garantizar la seguridad entre las llegadas de los vuelos.

3 Descripción del problema MAGHP-ALSS

El objetivo del problema MAGHP-ALSS consiste en mejorar la toma de decisiones frente a situaciones inesperadas que afectan a la planificación inicial del vuelo. En particular, si se reduce el número de pistas de aterrizaje disponibles en los aeropuertos, entonces se produce congestión en los vuelos. Por lo tanto, la solución del MAGHP-ALSS implica la modificación del plan de vuelo inicial, de tal manera que se garantizan tiempos de seguridad entre las llegadas de los vuelos. Por otra parte, la solución requerida debe minimizar el coste de retraso en tierra y del avance o retraso en el proceso de aterrizaje. También es necesario disponer de la solución de manera inmediata, conforme a una planificación estratégica a corto plazo.

Del mismo modo, la planificación de los vuelos que garantice seguridad debe además realizar modificaciones dentro de lo permisible en cada vuelo. En la descripción MAGHP implícitamente se consideran las limitaciones siguientes: i) hay un máximo de periodos de tiempo para retraso en tierra, digamos que cada periodo son 15 minutos, y ii) cualquier vuelo continuado tiene que esperar hasta que el vuelo predecesor ha llegado y se ha completado tiempo de preparación. También en la descripción ALSS resultan limitaciones implícitas: i) para cada vuelo hay una ventana temporal para el aterrizaje, pues el proceso de aterrizaje no debe ser ni demasiado lento ni demasiado rápido.

Por último, se incluye la cancelación de vuelos para aquellos que no pueden ser modificados dentro de los límites establecidos, por ejemplo, un vuelo que sobrepasa el tiempo máximo permitido de retraso en tierra. Nótese que la cancelación tiene asociada una gran penalización, pero además siempre garantiza una solución factible.

En la literatura MAGHP se relaciona por lo general la capacidad del aeropuerto respecto al número de vuelos que pueden aterrizar en cada período de tiempo. Sin embargo, cuando la capacidad viene dado según las pistas de aterrizaje disponibles se mejora en eficacia, pues obtiene un punto de vista más realista y en seguridad, ya que garantizamos los tiempos de seguridad entre las llegadas de los vuelos. Estos tiempos de seguridad, para la separación entre el aterrizaje de los vuelos, están delimitados por las consideraciones aerodinámicas de los aviones.

4 El algoritmo propuesto

En este trabajo se ha desarrollado un algoritmo multi-inicio para resolver el MAGHP-ALSS, cuyo criterio de parada es el tiempo (MaxTime). Nuestro enfoque se basa en la tabla de prioridad (Andreatta et al 1997) que da prioridad a las llegadas de vuelos con vuelos continuados y de vuelos con menor número de retrasos en tierra permitidos, el algoritmo de ABG (Andreatta et al 1997) que asigna los retrasos en tierra y el límite superior ALSS (Beasley et al 2000) para comprobar que se cumplen las restricciones de seguridad entre llegadas. Además, hemos incluido la aleatorización sesgada en el orden de prioridades de los vuelos (Juan et al. 2010), que contribuye a escapar de un posible mínimo local. Por tanto, la solución MAGHP-ALSS será de entre todas las que se obtienen, la que presenta el costo más bajo.

La Figura 1 muestra el diagrama de flujo para este algoritmo, y puede describirse con los siguientes procedimientos:

Inicialización: Se considera un plan de vuelo inicial con la información de los vuelos programados, así como el número de pistas disponibles para los aeropuertos en cada período de tiempo. En particular, se conoce el horizonte temporal, el número de vuelos, los aeropuertos de salida y de llegada, los tiempos de salida y llegada, el máximo permitido de períodos de tiempo para retraso en tierra, antecesor y sucesor para los vuelos continuados, tiempos de preparación de vuelos continuados, ventana temporal para el aterrizaje, pistas de llegada operativas en los aeropuertos en cada período de tiempo, y los costes para el retraso en tierra, para la desviación en el proceso de aterrizaje y para la cancelación.

Heurística ABG: Por cada período de tiempo, el algoritmo ABG realiza los siguientes pasos: 1) crea una lista con los vuelos que son candidatos para aterrizar, es decir, vuelos cuyo período de tiempo de llegada es menor o igual al que se evalúa. Para el caso de vuelos continuados, también es necesario que verifique que su predecesor ya ha aterrizado y que se completó el tiempo de preparación correspondiente. 2) de la lista de vuelos generada y, siguiendo el orden que estable la tabla de prioridades, se toma un para programar su aterrizaje, 3) se realiza la programación del aterrizaje del vuelo, y por lo tanto que verifique las restricciones de seguridad ALSS (véase *Procedimiento ALSS*). Como resultado, el algoritmo obtiene como solución una planificación de vuelos y se calculan los costes de las modificaciones incorporadas, esto es, costes por el retraso en tierra, por el avance o retraso en el proceso de aterrizaje y por la cancelación.

Procedimiento ALSS: En cada período de tiempo, consideramos que el primer vuelo que aterriza es el último que ya aterrizó en el período anterior. Por otra parte, todos los vuelos que se programen a continuación deberán satisfacer las limitaciones establecidas en la descripción del problema ALSS, es decir, el tiempo en que aterriza el vuelo pertenece a su ventana temporal, y además tienen que garantizarse los tiempos de seguridad entre los vuelos. Los vuelos evaluados que no cumplan estas condiciones incrementarán en un período de tiempo su retraso en tierra, y serán evaluados de nuevo en el período de tiempo siguiente.

Asignación aleatoria: Para la lista de los vuelos generada, nuestro enfoque incluirá aleatorización sesgada en el orden de prioridades. Recordemos que la ordenación que toma la lista de vuelos inicialmente corresponde a la prioridad del vuelo en función de la tabla de prioridades (como se menciona en la *heurística ABG*). Al tratarse de aleatorización sesgada las listas de vuelo ordenadas por prioridad serán diferentes, sin embargo se mantiene la lógica de la regla de prioridad, es decir, vuelos de mayor prioridad tienen más probabilidades de ser seleccionados que los que tienen menor prioridad. Luego, la aleatorización contribuye a escapar de un posible mínimo local, pues se obtienen diferentes soluciones. En otras palabras, la aleatorización permite definir un algoritmo de multi-inicio donde para cada re-inicio se obtiene una solución.

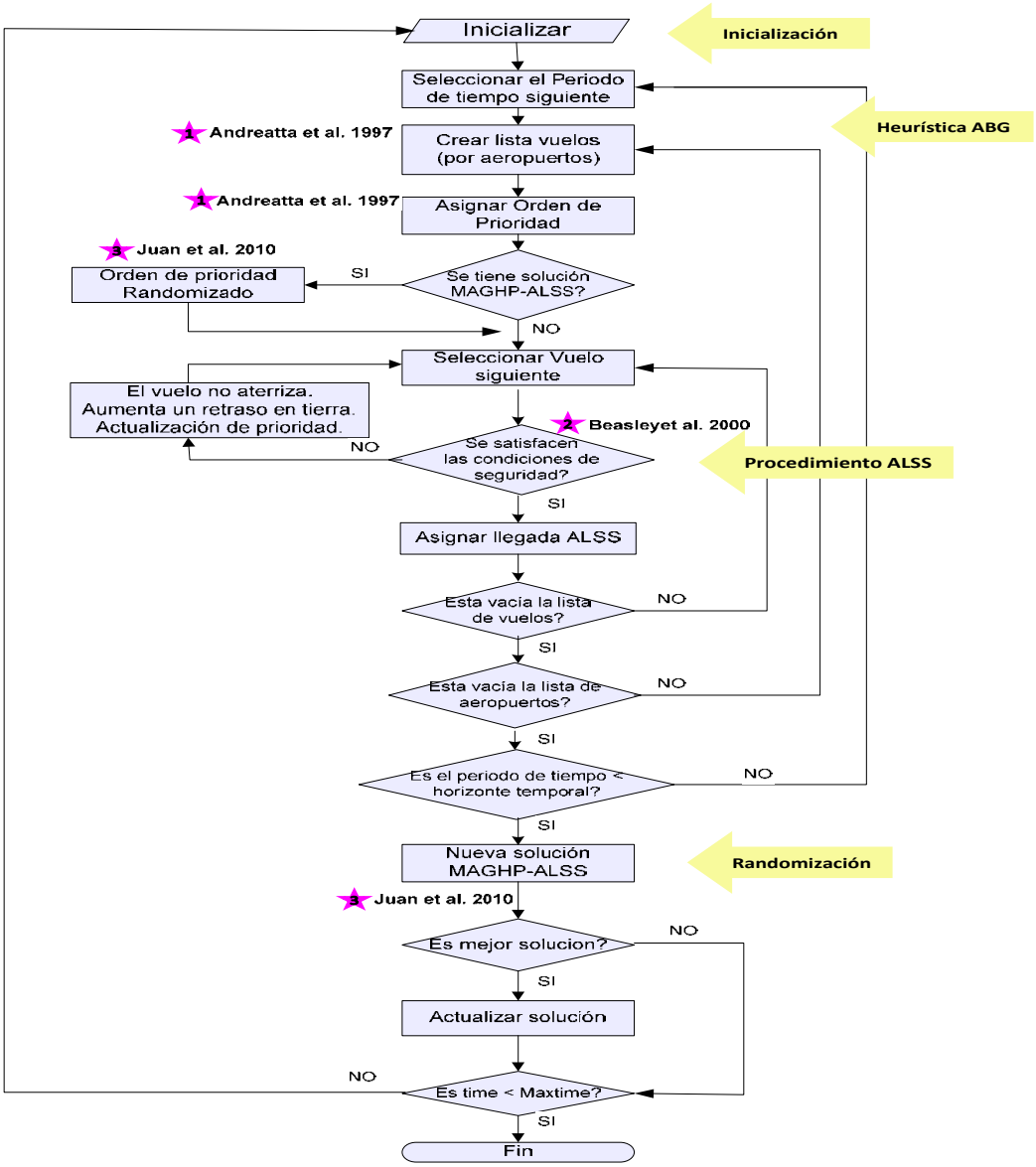


Fig. 1. Diagrama de flujo de la aproximación propuesta

5 Experiencia computacional

El algoritmo multi-inicio MAGHP-ALSS se ha implementado en la plataforma Netbeans IDE para C + + en Ubuntu. Los cálculos se llevaron a cabo en un Intel Core 2 Duo de 2,00 GHz y 4 GB de RAM.

Se han utilizado varios casos, con datos generados de forma aleatoria, para evaluar el algoritmo propuesto. La Tabla 1 muestra las dimensiones de cada caso. Los encabezados corresponden a el número de aeropuertos (K), vuelos (F), períodos de tiempo (T), vuelos continuados (C).

Tabla 1. Dimensiones de los casos.

Casos	K	F	T	C
c01	2	100	22	44
c02	3	100	25	35
c03	3	250	17	116
c04	6	500	18	185

Nótese que en cada vuelo se han incluido de forma aleatoria los datos siguientes: i) programación inicial del vuelo, es decir, los aeropuertos y los tiempos de salida y de llegada ii) las características de vuelo, es decir, el tiempo de preparación, las unidades de retraso en tierra permitidas, los minutos de adelanto o retraso en que puede variar el proceso de aterrizaje y ii) características de aeronave, es decir el tamaño de la aeronave a elegir entre dos posibles opciones.

Y, asumiendo disponibilidad de sólo una pista por cada aeropuerto, se han resuelto tres instancias para cada caso:

- *_a* que considera el tiempo de seguridad de 3 minutos entre el aterrizaje de los vuelos, en cualquier caso. El máximo de períodos de tiempo para el retraso en tierra se asigna aleatoriamente entre 0 y 4 (caso realista), y los minutos permitidos para adelanto y retraso en el proceso de aterrizaje son ambos 3 minutos.
- *_b* que considera el tiempo de seguridad de 3 minutos entre el aterrizaje de los vuelos, en cualquier caso. El máximo de períodos de tiempo para el retraso en tierra es siempre 10 (grande) para todos los vuelos y por lo tanto T también se incrementa 10 períodos. Finalmente los minutos permitidos para adelanto y retraso en el proceso de aterrizaje son ambos 3 minutos.
- *_c* que considera el tiempo de seguridad de 3 minutos entre el aterrizaje de vuelos para aviones de diferentes tamaños y de 2 minutos cuando los aviones son del mismo tamaño. El máximo de períodos de tiempo para el retraso en tierra se asigna aleatoriamente entre 0 y 4 (caso realista), y los minutos permitidos para adelanto y retraso en el proceso de aterrizaje son ambos 3 minutos.

Tabla 2 muestra los costos al modificar la planificación inicial de vuelo conforme a las técnicas antes mencionadas. Luego, se penaliza con 1000 la cancelación de un vuelo, con 10 cada período de retraso en tierra, y 0.1 cada minuto que se adelante o retrase el vuelo en el proceso de aterrizaje. La metodología FCFS que se aplica generalmente asigna retrasos en tierra y los retrasos en el proceso de aterrizaje de acuerdo a los tiempos de llegada de la programación inicial. La solución MAGHP-ALSS asignará retrasos en tierra y adelantos o retrasos en el proceso de aterrizaje conforme a la

tabla de prioridades. Por último, nuestra mejor solución (OBS) se obtiene con 10 segundos de aleatorización sesgada en el orden de las prioridades, es decir, asignar orden aleatorio a vuelos con la misma prioridad.

Tabla 2. Resultados para el valor de la función objetivo

Cases	FCFS	MAGHP-ALSS	OBS	GAP_1	GAP_2
c01_*a*	12139.7	1311.5	1300.8	-0.89	-0.01
c01_*b*	542.5	450.3	401.6	-0.26	-0.11
c01_*c*	6087.6	118.2	108.2	-0.98	-0.08
c02_*a*	11200.1	6290.5	5300.6	-0.53	-0.16
c02_*b*	592.2	481.6	481.3	-0.19	0.00
c02_*c*	5178.7	4211.0	3201.4	-0.38	-0.24
c03_*a*	137408.9	94225.1	87395.9	-0.36	-0.07
c03_*b*	80578.9	16203.5	11686.8	-0.85	-0.28
c03_*c*	95744.6	68184.0	67215.4	-0.30	-0.01
c04_*a*	190595.8	153190.9	152130.9	-0.20	-0.01
c04_*b*	78429.9	20075.2	20075.2	-0.74	0.00
c04_*c*	140501.8	96885.4	95813.9	-0.32	-0.01

En la Tabla 2 observamos que la política FCFS presenta mayor coste en todos los casos, ya que no tiene en cuenta la tabla de prioridades, es decir, no da prioridad a los vuelos que tienen vuelos continuados y con menor número retrasos en tierra permitido. Por ello, la diferencia de costes se ve reducida en _b donde el número permisible de retrasos en tierra es grande y en _c pues al disminuir los tiempos de seguridad permite aterrizar más vuelos en cada período de tiempo. Por otro lado, la aleatorización proporciona también rendimientos (gaps) negativos, lo que significa que se reducen costes.

La Tabla 3 determina las modificaciones al plan de vuelo modificado de acuerdo a las distintas estrategias FCFS, MAGHP-ALS y OBS. Los encabezados indican lo siguiente: FC, número de vuelos cancelados; GH, número de retrasos en tierra, y MD, número de minutos de retraso; MA, número de minutos de antelación.

En la Tabla 3 se observa que la falta de previsión en la política FCFS nunca aprovechar la estrategia para avanzar minutos (MA) en el proceso de aterrizaje, siendo siempre cero. Además, la estrategia FCFS cancelará una gran cantidad de vuelos. Finalmente observamos que la aleatorización es capaz de reducir retrasos en tierra mediante la desviación de minutos en el proceso de aterrizaje.

Tabla 3. Información de la solución

Cases	FCFS				MAGHP-ALSS				OBS			
	FC	GH	MD	MA	FC	GH	MD	MA	FC	GH	MD	MA
c01_a	12	13	97	0	1	30	83	32	1	29	85	23
c01_b	0	53	125	0	0	44	78	25	0	39	94	22
c01_c	6	8	76	0	0	11	57	25	0	10	56	26
c02_a	11	19	101	0	6	28	64	41	5	29	66	40
c02_b	0	58	122	0	0	47	79	37	0	47	78	35
c02_c	5	17	87	0	4	20	76	34	3	19	81	33
c03_a	137	39	189	0	93	120	189	62	86	137	192	67
c03_b	76	455	289	0	8	817	265	70	3	865	272	96
c03_c	95	72	246	0	67	116	171	69	66	119	188	66
c04_a	189	154	558	0	150	314	386	123	149	308	374	132
c04_b	68	1035	796	0	6	1400	583	169	6	1400	583	169
c04_c	139	144	618	0	94	283	400	154	93	276	397	142

6 Conclusiones y trabajo futuro

El algoritmo MAGHP-ALSS propuesto es capaz de integrar dos importantes estrategias cuando se produce una incidencia en un sistema multi-aeropuerto, como son, el retraso en tierra y la secuenciación de los aterrizajes, garantizando que se mantiene la seguridad. Además, el algoritmo es relativamente simple (sin parámetros), rápido, y de fácilmente paralelizable.

Conforme a mejorar la meta-heurística propuesta, se podrían incluir además otras técnicas de uso habitual al problema ALSS, tales como el intercambio de vuelos, el método de cross-over…, para ser evaluadas en instancias más flexibles. Por ejemplo, instancias que consideren diferentes tiempos para adelantar y retrasar los vuelos en el proceso de aterrizaje, considerar mayor número de opciones para el tamaño de las aeronaves, asumir múltiples pistas de aterrizaje disponibles. Además, sería conveniente resolver casos de mayor dimensión y publicar en una página web los datos para las instancias y los resultados.

Otra línea de investigación a considerar sería incluir más características del tráfico aéreo. Resulta inmediato suponer el proceso similar en los aeropuertos de salida, es decir, poder hacer frente a reducciones en el número de pistas de despegue disponibles. Por otra parte, se debería permitir que el vuelo experimente minutos de adelanto o retraso, no sólo en el proceso de aterrizaje, sino a lo largo de todo el vuelo.

Referencias

1. **Agustin, A., Alonso-Ayuso A., Escudero L.F. and Pizarro C.**: Combinatorial optimization in practice. In Hermann, editor, *Mathematical Optimization models for Air Traffic Flow Management: A review*, pages 141–184. Studia Informatica Universalis 8.2 (2010).
2. **Agustin, A., Alonso-Ayuso A., Escudero L.F. and Pizarro C.**: On air traffic flow management with rerouting. Part I: Deterministic case. *European Journal of Operational Research*, 219:156-166 (2012).
3. **Andreatta, G., and Brunetta L.**: Multi-Airport Ground Holding Problem: a Computational Evaluation of Exact Algorithms. *Operations Research*, 46(1):57-64 (1998).
4. **Andreatta, G., Brunetta L. and Guastalla G.**: Multi-airport ground holding problem: A heuristic approach based on priority rules. In Bianco L., Dell'Olmo P., and Odoni A.R., editors, *Modeling and Simulation in Air Traffic Management*, pages 71–87. (1997).
5. **Beasley, J.E., Krishnamoorthy M., Sharaiha Y.M. and Abramson D.**: Scheduling aircraft landings - the static case. *Transportation Science* 34(2):180–197 (2000).
6. **Bertsimas, D., and Stock S.**: The air traffic flow management problem with enroute capacities. *Operations Research*, 46:406–422 (1998).
7. **Bertsimas, D., Lulli G., and Odoni A.**: An integer optimization approach to large-scale air traffic optimization flow management. *Operations Research*, 59(1):211–227 (2011).
8. **Brunetta, L., Guastalla G. and Navazio L.**: Solving the multi-airport Ground Holding Problem. *Annals of Operations Research*, 81:271–287 (1998).
9. **Dear, R.G., and Sherif Y.S.**: An algorithm for computer assisted sequencing and scheduling of terminal area operations. *Transportation Research Part A* 25: 129–139 (1991**).**
10. **Ernst, A.T., Krishnamoorthy M. and Storer R.H.**: Heuristic and exact algorithms for scheduling aircraft landings. *Networks* 34(3): 229-241 (1999).
11. **EUROCONTROL**: Medium-Term Forecast Update, Flight Movements. *Intern Report* (2012).
12. **Harikiopoulo D. and Neogi N.**: Polynomial-time feasibility condition for multiclass aircraft sequencing on a single-runway airport. *IEEE Transactions on Intelligent Transportation Systems* 12(1): 2-14 (2011).
13. **Hu, X., and Di Paolo E.**: An efficient genetic algorithm with uniform crossover for air traffic control. *Computers & Operations Research* 31: 245-259 (2009).
14. **Jakobovits, R., Kopardekar P., Burke J. and Hoffman R.**: *Algorithms for managing sector congestion using the airspace restriction planner*. 7th USA/European ATM seminar. Barcelona, Spain (2007).
15. **Juan, A., Faulin J., Ruiz R., Barrios B. and Caballé S.**: The SR-GCWS hybrid algorithm for solving the capacitated vehicle routing. *Applied Soft Computing* 10: 215-224 (2010).
16. **Navazio, L., and Romanin-Jacur G.**: The multiple connections multi airport ground holding problem: Models and algorithms. *Transportation Science* 32(3): 268–276 (1998).
17. **Wanke, C., and Greenbaum D.**: *Incremental, probabilistic decision making for en route traffic management*. Proceedings 7th USA/European Air Traffic Management R&D Seminar. Barcelona, Spain (2007).
18. **Zhang, X., Zhou Y., Liu B. and Wang Z.**: *The air traffic flow management with dynamic capacity and co-evolutionary genetic algorithm*. IEEE Transactions on Intelligent Transportation System Conference. Seattle (WA, USA), pp. 580 – 585 (2007).

Hybrid Approach combining Insertion Heuristic and Biased Random Sampling for the Vehicle Routing Problem with Time Windows

José Cáceres-Cruz[1], Daniel Riera[1], Angel A. Juan[1], Silvia Padrón[2]

[1] Open University of Catalonia - IN3, Barcelona, Spain.
{drierat,ajuanp,jcaceresc}@uoc.edu

[2] Universitat Autònoma de Barcelona, Bellaterra, Spain.
silvia.padron@uab.cat

Abstract. The Vehicle Routing Problem with Time Windows is a NP-hard routing problem where the demand is located in nodes and should be completely served fulfilling certain scheduling constraints for delivering. This paper presents a randomized algorithm which combines a classical heuristic with biased random sampling in order to solve this type of problem. This new algorithm is compared with one version of the Insertion heuristic of Solomon, reaching results which outperform it. As discussed in the paper, the methodology presented is flexible, can be easily parallelized and it does not require any complex fine-tuning process. Some preliminary tests show the potential of the proposed approach as well as its limitations.

Keywords. Vehicle Routing Problem with Time Windows, combinatorial optimization, randomized algorithms, metaheuristics, simulation.

1 Introduction

In last decades, optimization routing problems have been the target of many studies **(Golden et al., 2008)**. The Vehicle Routing Problem with Time Windows (**VRPTW**) is probably one of the most developed research lines inside of the classical Vehicle Routing Problem **(Potvin & Bengio, 1996**; **Potvin et al., 1996)**. On this, a set of vehicles must deliver the goods to a set of customers. Unlike the original problem, VRPTW must to respect some delivery time windows on each customer and considering arrival, waiting and service times among others. The objective of this study is to adapt one of the VRPTW heuristics proposed in **Solomon (1987)**. This heuristic is known as *Insertion* and basically consists on the iterative construction routes with the insertion of proper customers. Therefore our main idea is to apply the randomization concepts presented in **Juan et al. (IN PRESS)** in order to generate a new promising metaheuristic algorithm. A concrete example of this randomized process can be found in **Juan et al. (2010)**.

The structure of this paper is as follows. First, the VRPTW problem is stated, establishing some assumptions and the basic notation. Section 3 visits the existing literature to revise the state of the art. Later, in section 4, the proposed algorithm is introduced, presenting the modifications on the classic heuristic. Section 5 displays and discusses some results, and finally, section 6 states some conclusions and future work.

2 VRPTW

The Vehicle Routing Problem (**VRP**) has been studied for over more than 50 years **(Laporte, 2009)**. The simplest version is known as the Capacitated Vehicle Routing Problem (**CVRP**), defined by **Dantzig & Ramser (1959)**. In CVRP, a directed graph $G = (V, A)$ is given, where $V = \{0, 1, \ldots, n\}$ is the set of $n + 1$ nodes and A is the set of arcs. *Node 0* represents the depot, while the remaining nodes $V' = V \setminus \{0\}$ correspond to the n customers. Each customer i in V' requires a known supply of q_i units, i.e., its demand, from a single depot (assume $q_0 = 0$). This demand is going to be served by exactly one visit of a single vehicle. The associated cost (distance) of an arc (i, j) is denoted by d_{ij}. In this basic form, there is an unlimited homogeneous fleet of m identical vehicles with capacity Q to serve these n customers. Each vehicle has also a time limit L for their single trip. A vehicle's trip is a sequence of customers, whose total demand cannot exceed Q that starts from and finishes at the depot with duration no greater than L (used to be a really big value in order to ignore its effect). CVRP aims at finding m trips (vehicles) so that all customers are serviced and the total distance traveled by the fleet is minimized.

In VRPTW, the objective function is the same but some delivery time windows must be considered. So the traveling time between a pair of customers (t_{ij}) is an important element on this study. The scheduling constraint is denoted by a predefined time interval, given as an earliest start time (e_i) and latest start time (l_i) at the customer i. The vehicles leave the depot, at time e_0, at the earliest and must return to the depot by time l_0, at the latest. There is also a given service duration time (f_i) on each customer of a route for considering unloading time of goods. Therefore vehicles must arrive at the customers not later than the latest start time. If vehicles arrive earlier than the earliest start time, then a waiting occurs. So after the routing planning is defined, an effective delivery service at customer i begins at a given time (b_i) within the defined customer time window. In (1-2) we define the basic relations of time windows and sequential customers' visits using commented variables.

$$e_i \leq b_i \leq l_i \quad / \quad \forall i \in V' \tag{1}$$

$$b_j = \text{Max}[e_j , b_i + f_i + t_{ij}] \quad / \quad \forall i \in V ; \forall j \in V' ; \; i \neq j \tag{2}$$

3 Literature Review

Different approaches to the VRPTW have been explored during the last decades **(Cordeau et al., 2001)**. These approaches range from the use of pure optimization methods, such as linear programming, for solving small-size problems with relatively simple constraints to the use of heuristics and meta-heuristics that provide near-optimal solutions for medium and large-size problems. One of the most promising frameworks is presented in **Cordeau et al. (2001b; 2004)** which is based on a Tabu Search technique. Notice that this framework combines the time windows constraint with other routing restrictions. Another Tabu Search algorithm is parallelized in **Badeau et al. (1997)**. A guided local search is proposed by **Kilby et al. (1999)**. Some greedy approaches have been presented in **Ioannou et al. (2001); Kontoravdis & Bard (1995).** Comprehensible recent surveys of algorithms and metaheuristics for the VRPTW can be found in **Bräysy & Gendreau (2005; 2005b)**. Also several variants of this problem have been studied: VRPTW minimizing route duration **(Savelsbergh, 1992)**, Dial-a-Ride problems with time windows **(Diana & Dessouky, 2004)**, VRPTW with a limited vehicle fleet **(Lau et al., 2003)**, Robust VRPTW **(Agra et al., 2013)**, among others. Since others approaches could generate better results **(Hu et al., 2013)**, they are also certainly more complex to implement and understand. Therefore the main advantage of the approach proposed on this study is its simplicity. Our method is focused to randomize a well-known heuristic.

Particularly, **Solomon (1987)** proposes six heuristics for the VRPTW. In between, we can find three sequential building heuristics based on the insertion of clients. These Insertion heuristics have been widely used in the research community **(Tan et al., 2001; Berger & Barkaoui, 2004; Campbell & Savelsbergh, 2004; Diana & Dessouky, 2004**). In fact, **Potvin & Rousseau (1993)** propose a parallel version of the first insertion heuristic. Likely, the work of **Ioannou et al. (2001)** is based on Solomon's Insertion heuristic Framework for solving theoretical instances and a real-life case inspired in a Food Company.

4 Our Approach

Here we focus on the Insertion Solomon heuristic so called *I3*. On this, the author initializes every route construction using one sorting criteria of a set to be described later. After initializing a current route, the method uses two criteria, $Sc_1(i, u, j)$ and $Sc_2(i, u, j)$, to iteratively insert a new customer u into the current partial route, between two adjacent customers i and j on the route. One by one, until time windows and capacity constraints do not allow to add more clients. For each unrouted customer, we first compute its best feasible insertion place in the emerging route (Sc_1). Next, the best unrouted customer to be inserted in the route is selected as the one for which Sc_2 is optimum and feasible. When no more customers with feasible insertions can be found, the method starts a new route, unless it has already routed all customers. As the same author states, "*this class of heuristics is a generalization of the time-oriented, nearest-neighbor heuristic, in that we allow insertion of an unrouted cus-*

tomer in any feasible location between a pair of customers on the route, rather than only at the end of the route". Formally, the value criteria that Solomon proposes for the *I3* are presented next (3-8). In summary, these criteria consist in a weighted addition of sub-elements where each represents an important routing-scheduling feature. Using four parameter values (λ; α_1; α_2; α_3), each feature is then related in the next expression of $Sc_1(i, u, j)$.

$$Sc_1(i, u, j) = \alpha_1 \cdot Sc_{11}(i, u, j) + \alpha_2 \cdot Sc_{12}(i, u, j) + \alpha_3 \cdot Sc_{13}(i, u, j) \tag{3}$$

subject to:

$$Sc_{11}(i, u, j) = d_{iu} + d_{uj} - \lambda \cdot d_{ij}\,, \quad \lambda \geq 0 \tag{4}$$

$$Sc_{12}(i, u, j) = b_{ju} - b_j \tag{5}$$

$$Sc_{13}(i, u, j) = l_u - b_u \tag{6}$$

$$\alpha_1 + \alpha_2 + \alpha_3 = 1, \quad \alpha_1 \geq 0; \quad \alpha_2 \geq 0; \quad \alpha_3 \geq 0 \tag{7}$$

$$Sc_2(i, u, j) = Sc_1(i, u, j) \tag{8}$$

Where b_{ju} is the new start time for service at customer j, given than u is on the route.

Our algorithm is implemented as described next. We propose to randomize in two points of the algorithm the original *I3* of **Solomon (1987)**. First, we apply a uniform randomization over the selection of the sorting criteria for the list of customers (explained in the next section). On each iteration, the combined effect of sorting with different criteria will create an intensive and promising search guide inside of the solution space. Second, a geometric (biased) distribution is used to pick up the next client over the sorted list. Thus, the clients at the top of the list will be more likely to be selected than others. This kind of double randomization has been previously applied in **González-Martín et al. (2013)** with good results.

```
procedure RandI3(inputs, λ, α1, α2, α3, β)
begin
  nodes := computeSolomonValues(inputs, λ, α1, α2, α3)
  while stopping criterion is not satisfied do
    unroutedList := nodes
    route := createNewRoute()
    while unroutedList is not empty do
      sortCriteria := selectSortCriteria()
      unroutedList := sortClients(unroutedList, sortCriteria)
      client := selectNext(unroutedList, β)
      unroutedList := remove(unroutedList, client)
      posCandidate := computePositions(route, client,λ,α1,α2,α3)
      if posCandidate is empty then
        solution := addRoute(solution, route)
        route := createNewRoute(client)
      else
```

```
        posCandidate := sortPositions(posCandidate)
        route := insertInBestPosition(route, client)
      end if
    end while
  end while
  return solution
end
```

5 Computational Results

Our algorithm RandI3 was implemented as a Java application and used to run instances on an Intel Xeon E5603 at 1.60 Ghz and 8 GB RAM. The implementation uses some state-of-the-art pseudo-random number generator. In particular, some classes from the SSJ library **(L'ecuyer & Buist, 2005)** were implemented. For preliminary experiments, we use a 100-customers test-bed also proposed by **Solomon (1987)**. All instances are represented by Euclidian distance, and the speed of all vehicles is assumed to be equivalent to the travel unit.

As in the original work, we have used the same four initialization sorting criteria and the parameters values for Solomon criteria values. The parameters values are (λ; α_1; α_2; α_3) = {(1; 0.5; 0.5; 0), (1; 0.4; 0.4; 0.2), (1; 0; 1; 0)}. The uniformly selected initialization sorting criteria: (*a*) the farthest unrouted customer, (*b*) the unrouted customer with the earliest deadline, (*c*) the unrouted customer with the minimum equally weighted combination of direct route-time and distance, and (*d*) actual heuristic criterion value. For each set of values (λ; α_1; α_2; α_3), 900 iterations per each instance were executed –i.e. 2,700 total iterations per each instance– which selects the best one. The biased random selection of customer is done using a Geometric distribution with β = 0.1.

Table 1 shows the routes and costs in the Best Known Solution (BKS) from **Tan et al. (2001)**; the routes and costs using the original *I3*; the routes and costs obtained using our approach *RandI3*, and finally the gaps between our approach and commented benchmarks. The *I3* solutions were obtained with a Java implementation following indications on original article **Solomon (1987)**. In general, the solutions are obtained in less than 5 minutes per each instance.

Table 1. Results obtained with Solomon instances.

	BKS		***I3***		***RandI3***			
Instance	Routes	Cost (1)	Routes	Cost (2)	Routes	Cost (3)	*Gap* (2-3)	*Gap* (1-3)
c101-100	10	829	10	855	10	829	-3.04%	**0.00%**
c102-100	10	827	11	1376	10	971	-29.43%	17.41%
c103-100	10	828	11	1162	10	969	-16.61%	17.02%
c104-100	10	824	11	1290	11	958	-25.74%	16.15%
c105-100	10	829	10	855	10	829	-3.04%	**0.00%**
c106-100	10	827	10	910	10	861	-5.38%	4.11%

c107-100	10	829	10	1027	10	830	-19.18%	**0.12%**
c108-100	10	827	10	993	10	865	-12.89%	4.59%
c109-100	10	829	10	1063	10	911	-14.30%	9.89%
				1059			***-14.40%***	***7.70%***
c201-100	3	590	3	590	3	590	**0.00%**	**0.00%**
c202-100	3	590	4	840	3	638	-24.05%	8.14%
c203-100	3	591	4	1109	3	727	-34.45%	22.90%
c204-100	3	590	4	1158	3	707	-38.95%	19.71%
c205-100	3	589	3	657	3	612	-6.85%	3.90%
c206-100	3	588	3	660	3	634	-3.94%	7.82%
c207-100	3	588	3	727	3	617	-15.13%	4.93%
c208-100	3	588	3	662	3	635	-4.08%	7.99%
				800.375			***-15.93%***	***9.42%***

Our approach outperforms the original version of the heuristics with an average improvement of around 14% [column "Gap (2-3)"]. Notice that some BKS are found. However, there is still a positive average gap of almost 10% with the BKS [column "Gap (1-3)"] that can be improved.

6 Conclusions and Future work

From the obtained results it can be seen that the classical Solomon Insertion heuristic is outperformed by its new randomized version. Furthermore, competitive results are obtained in medium sized instances with none special fine-tuning process.

Future work in this research will be to replace the uniform selection of sorting customer criteria by a biased selection based on the most promising criteria. This improvement can be interpreted as a “learning technique” based on the criterion that has generated better solutions. This new feature combined with a brief fine-tuning phase will help to improve the found values. Also due to the independence of all the generated solutions, the algorithm could easily be parallelized in order to improve its performance when attempting to solve larger instances of VRPTW problems.

7 Bibliography

1. Agra, A., Christiansen, M., Figueiredo, R., Hvattum, L.M., Poss, M., Requejo, C.: The robust vehicle routing problem with time windows. Computers & Operations Research, 40(3):856-866 (2013).
2. Badeau, P., Guertin, F., Gendreau, M., Potvin, J.Y., Taillard, E.: A parallel tabu search heuristic for the vehicle routing problem with time windows. Transportation Research Part C: Emerging Technologies, 5(2):109-122 (1997).
3. Berger, J., Barkaoui, M.: A parallel hybrid genetic algorithm for the vehicle routing problem with time windows. Computers & Operations Research, 31(12):2037 – 2053 (2004).

4. Bräysy, O., Gendreau, G.: Vehicle routing problem with time windows, part I: Route construction and local search algorithms. Transportation science, 39(1):104-118 (2005a).
5. Bräysy, O., Gendreau, G.: Vehicle routing problem with time windows, part II: Metaheuristics. Transportation science, 39(1):119-139 (2005b).
6. Campbell, A.M., Savelsbergh, M.: Efficient insertion heuristics for vehicle routing and scheduling problems. Transportation science, 38(3):369-378 (2004).
7. Cordeau, J.F., Desaulniers, G., Desrosiers, J., Solomon, M.M., Soumis, F.: VRP with time windows. In: Toth, P., Vigo, D. (eds.) The Vehicle Routing Problem, vol. 9, pp. 157-193. SIAM (2001).
8. Cordeau, J.F., Laporte, G., Mercier, A.: A unified tabu search heuristic for vehicle routing problems with time windows. Journal of the Operational research society, pages 928-936 (2001b).
9. Cordeau, J.F., Laporte, G., Mercier, A.: Improved tabu search algorithm for the handling of route duration constraints in vehicle routing problems with time windows. Journal of the Operational Research Society, 55(5):542-546 (2004).
10. Dantzig, G., Ramser, J.H.: The truck dispatching problem. Management Science, 6(1):80-91 (1959).
11. Diana, M., Dessouky, M.M.: A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows. Transportation Research Part B: Methodological, 38(6):539 – 557 (2004).
12. Golden, B.L., Raghavan, S., Wasil, E.A.: The vehicle routing problem: latest advances and new challenges, volume 43. Springer (2008).
13. González-Martín, S., Juan, A., Riera, D., Castellà, Q., Muñoz, R., Pérez, A.: Development and assessment of the SHARP and RandSHARP algorithms for the arc routing problem. AI Communications, 25(2):173-189 (2012).
14. Hu, W., Liang, H., Peng, C., Du, B., Hu, Q.: A Hybrid Chaos-Particle Swarm Optimization Algorithm for the Vehicle Routing Problem with Time Window. *Entropy*, *15*(4), 1247-1270 (2013).
15. Ioannou, C., Kritikos, M., Prastacos, G.: A greedy look-ahead heuristic for the vehicle routing problem with time windows. The Journal of the Operational Research Society, 52(5):523-537 (2001).
16. Juan, A., Faulin, J., Ruiz, R., Barrios, B., Caballè, S.: The SR-GCWS hybrid algorithm for solving the capacitated vehicle routing problem. Appl. Soft Comput., 10(1):215-224 (2010).
17. Juan, A.; Cáceres-Cruz, J.; González-Martín, S.; Riera, D.; Brian, B.: Biased-Randomization of Classical Heuristics. In: Wang, J. (ed.) Encyclopedia of Business Analytics and Optimization, IGI Global, USA (IN PRESS).
18. Kontoravdis, G., Bard, J.F.: A grasp for the vehicle routing problem with time windows. ORSA journal on Computing, 7(1):10-23, (1995).
19. Kilby, P., Prosser, P., Shaw, P.: Guided local search for the vehicle routing problem with time windows. In: Vo, S., Martello, S., Osman, I., Roucairol, C. (eds.), Meta-Heuristics, pp. 473-486. Springer USA (1999).
20. Laporte, G.: Fifty years of vehicle routing. Transportation Science, 43(4):408-416 (2009).
21. Lau, H.C., Sim, M., Teo, K.M.: Vehicle routing problem with time windows and a limited number of vehicles. European Journal of Operational Research, 148(3):559-569 (2003).
22. L'ecuyer, P., Buist, E.: Simulation in java with SSJ. In Winter Simulation Conference, pp. 611-620 (2005).

23. Potvin, J.Y., Rousseau, J.M.: A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. European Journal of Operational Research, 66(3):331-340 (1993).
24. Potvin, J.Y., Bengio, S.: The vehicle routing problem with time windows part II: genetic search. INFORMS journal on Computing, 8(2):165-172 (1996).
25. Potvin, J.Y., Kervahut, T., Garcia, B.L., Rousseau, J.M.: The vehicle routing problem with time windows part I: tabu search. INFORMS Journal on Computing, 8(2):158-164 (1996).
26. Savelsbergh, M.: The vehicle routing problem with time windows: Minimizing route duration. ORSA Journal on Computing, 4(2):146-154 (1992).
27. Solomon, M.M.: Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations research, 35(2):254-265 (1987).
28. Tan, K.C., Lee, L.H., Zhu, Q.L., Ou, K.: Heuristic methods for vehicle routing problem with time windows. Artificial Intelligence in Engineering, 15(3):281-295 (2001).

Un Algoritmo Heurístico para el Uncapacitated r-Allocation p-Hub Median Problem

Juanjo Peiró, Ángel Corberán, Rafael Martí

Departament d'Estadística i Investigació Operativa. Universitat de València
juanjo.peiro@uv.es, angel.corberan@uv.es, rafael.marti@uv.es

Resumen En este trabajo proponemos un algoritmo heurístico para el Uncapacitated r-Allocation p-Hub Median Problem. Éste es un problema de optimización, de la familia de problemas de localización de concentradores (hubs). Conocidas las distancias entre los nodos de una red, así como unos flujos de tráfico a transportar entre ellos, estos problemas consisten básicamente en seleccionar p de dichos nodos como hubs y enviar a través de ellos todos los flujos, de tal manera que se minimicen los costes totales de transporte en la red.
La versión r-allocation, propuesta recientemente en [1], tiene la particularidad de que cada nodo es asignado a r de los p hubs ($r \leq p$), permitiendo únicamente enviar los flujos desde cada nodo a través de sus r hubs asociados.
Como es usual en este tipo de problemas, el método que proponemos tiene tres fases: localización de hubs, asignación de cada nodo a r de los p hubs y transporte de flujos. Presentamos para ello un algoritmo heurístico basado en la metodología GRASP, en el cual consideramos dos procedimientos de búsqueda local.
Los resultados computacionales sobre 435 instancias muestran que sólo las más pequeñas pueden ser resueltas con métodos exactos y que nuestro heurístico es capaz de encontrar soluciones de alta calidad para instancias más grandes en tiempos de computación pequeños.

1. Introducción al problema

El problema de la localización de concentradores, conocido como p-Hub, es un problema de optimización combinatoria clásico [2]. Dado un conjunto de nodos N y las distancias (d_{ij}) entre ellos, así como unos flujos de tráfico que han de ser transportados (t_{ij}) entre los nodos, el problema consiste en seleccionar como hubs p de dichos nodos y enviar a través de ellos todos los flujos, de tal manera que se minimicen los costes totales de transporte en la red. Generalmente se asume que no es posible transportar directamente los tráficos entre nodos no-hub, y que los tráficos t_{ij} viajan a través del camino $i \to h_i \to h_j \to j$, donde h_i y h_j son uno de los hubs asignados a i y a j, respectivamente.

Transportar t_{ij} unidades a través del camino $i \to h_i \to h_j \to j$ tiene un coste asociado $c_{ij}(h_i, h_j)$, calculado generalmente como $c_{ij} = t_{ij}(d_{ih_i} + d_{h_ih_j} + d_{h_jj})$, donde d_{ih_i} es la distancia entre i y h_i (similarmente para $d_{h_ih_j}$ y d_{h_jj}). Por lo

tanto, una solución para el problema se determina por un conjunto de hubs, las asignaciones de nodos a hubs, y los caminos utilizados para transportar los tráficos para cada par de nodos.

Existen dos versiones del problema que han sido ampliamente estudiadas y cuya diferencia está basada en la estrategia de asignación de clientes a hubs: la versión *single allocation* y la versión *multiple allocation*. En la versión *single* cada nodo es asignado a un hub, permitiendo enviar y recibir tráficos a través de ese único hub. En cambio, en la versión *multiple* cada nodo es asignado a p hubs, permitiendo enviar y recibir tráficos a través de cualquiera de ellos.

Una nueva versión del problema, propuesta recientemente en [1] y conocida como Uncapacitated r-Allocation p-Hub Median Problem (UrApHMP), tiene como particularidad que cada nodo es asignado a r de los p hubs ($r \leq p$), permitiendo únicamente transferir los flujos de cada nodo a través de sus r hubs asociados. Muchas situaciones reales pueden ser modelizadas mejor con la versión *r-allocation* que con las anteriormente descritas, ya que la versión *single allocation* es demasiado restringida como para aplicarla a muchas situaciones reales, mientras que la versión *multiple allocation* acaba produciendo modelos con altos costes fijos y redes densas y complicadas. Además, la versión *r-allocation* generaliza las otras versiones, ya que cuando $r = 1$ obtenemos la versión *single allocation* y cuando $r = p$ estamos en la versión *multiple allocation*.

En [1], la autora propone una formulación del Uncapacitated r-Allocation p-Hub Median Problem como problema de Programación Lineal Entera y discute distintas versiones del problema. La formulación, aunque muy interesante, usa un gran número de variables enteras, lo que hace que su uso directo no sea posible excepto para problemas con un número pequeño de nodos.

2. GRASP

La metodología GRASP fue desarrollada a finales de los años 80 ([3]) y el acrónimo fue acuñado en [4]. En esencia, cada iteración del GRASP consiste en construir una solución para un problema de optimización a través de un procedimiento *voraz aleatorizado* y luego aplicar una búsqueda local a la construcción realizada previamente. La fase de construcción es iterativa, aleatorizada, voraz y adaptativa. Este proceso se repite hasta que algún criterio de parada se satisfaga. El mejor óptimo local encontrado durante el proceso es devuelto como la solución del procedimiento heurístico. En [5] se puede encontrar un resumen reciente sobre la metodología GRASP.

A continuación describimos los distintos elementos del GRASP que proponemos para la resolución del Uncapacitated r-Allocation p-Hub Median Problem (UrApHMP).

2.1. Algoritmo constructivo

Para obtener una solución posible para el UrApHMP proponemos un algoritmo constructivo consistente en tres fases:

1. Se seleccionan p hubs. Para ello usamos una función de evaluación greedy aleatoria adaptativa que contempla los tráficos desde cada nodo. Representamos por $H = \{h_1, h_2, ..., h_p\} \subseteq N$ al conjunto de hubs seleccionados.
2. Se determinan las asignaciones de cada nodo a r de los p hubs. En este caso utilizamos una función de evaluación de asignaciones que calcula el coste de asignar un nodo a cada uno de los hubs y seleccionamos los r hubs con mejor evaluación. H_i representa el conjunto de r hubs asignados a cada nodo i.
3. Para cada par de nodos, identificamos los hubs que utilizaremos para enviar los tráficos. Dados dos nodos i y j, sean $h_i \in H_i$ y $h_j \in H_j$ dos hubs de entre los asignados a i y j. El problema en esta fase consiste en determinar el h_i y el h_j que minimizan el coste $c_{ij} = t_{ij}(d_{ih_i} + d_{h_ih_j} + d_{h_jj})$. Puesto que r suele ser un número pequeño (generalmente $2 \leq r \leq 7$), realizamos una exploración exhaustiva de las posibles combinaciones de h_i y h_j para encontrar la solución óptima de esta fase.

2.2. Representación de una solución

Representamos una solución $\mathbf{S}$ mediante un conjunto de hubs H, una matriz de asignaciones A y dos matrices de hubs $\mathcal{H}^1$ y $\mathcal{H}^2$ que especifican las rutas que siguen los tráficos. En particular, $\mathbf{S} = (H, A, \mathcal{H}^1, \mathcal{H}^2)$, donde:

$$H = \{h_1, h_2, ..., h_p\} \subseteq N,$$

$$A = [a_{ij}]_{i=1,...,n;j=1,...,r}, a_{ij} \in H_i,$$

$$\mathcal{H}^1 = \left[\mathfrak{h}^1_{ij}\right]_{i=1,...,n;j=1,...,n}, \mathfrak{h}^1_{ij} \in H_i, \qquad \mathcal{H}^2 = \left[\mathfrak{h}^2_{ij}\right]_{i=1,...,n;j=1,...,n}, \mathfrak{h}^2_{ij} \in H_j,$$

Notar que cada fila i de la matriz A contiene los r hubs asignados al nodo i, H_i. Puesto que para cada par de nodos i y j necesitamos representar el camino $i \to h_i \to h_j \to j$ usado para enviar el tráfico desde i a j, la matriz $\mathcal{H}^1$ proporciona el primer hub en el camino y la matriz $\mathcal{H}^2$ la segunda, i.e. $\mathcal{H}^1(i,j) = h_i$ y $\mathcal{H}^2(i,j) = h_j$.

2.3. Búsquedas Locales

Dado que el proceso construtivo consiste en tres fases y la última de ellas queda resuelta de forma óptima, proponemos aplicar dos tipos de mejoras a través de búsquedas locales: cambiar los nodos seleccionados como hubs y cambiar las asignaciones de hubs a clientes.

El primer vecindario, N_H, implementa un intercambio clásico donde un hub h_i es eliminado de H y un nodo no hub $h_i^{'} \in N \setminus H$ pasa a ser hub.

El segundo vecindario, N_A, únicamente considera modificar las asignaciones de nodos a hubs sin modificar la selección de éstos. Para un nodo i cuyas asignaciones a hubs son $H_i = \{h_{i1}, ..., h_{ia}, ..., h_{ir}\} \subseteq H$, este vecindario intercambia un hub asignado por otro hub no asignado. En términos matemáticos, reemplazamos $h_{ia} \in H_i$ por $h_{ia}^{'} \in H \setminus H_i$, obteniendo $H_i^{'} = \{h_{i1}, ..., h_{ia}^{'}, ..., h_{ir}\}$.

3. Resultados computacionales

En esta sección describimos las pruebas computacionales realizadas para contrastar la eficiencia del algoritmo descrito anteriormente. Los procedimientos han sido implementados en C y los resultados han sido obtenidos con un ordenador Intel i7 @ 2.7 GHz y 4GB de RAM.

3.1. Instancias

El algoritmo GRASP propuesto se ha probado sobre dos conjuntos de instancias:

- *CAB*: Este conjunto consiste en un archivo de datos para un grafo de 25 nodos. A partir de este archivo se han creado 75 instancias con diferentes características.
- *AP*: Este conjunto consiste en un archivo de datos para un grafo de 200 nodos, basado en datos reales del servicio postal de Australia. Se pueden obtener instancias más pequeñas a través de un código de la ORLIB. Hemos generado 360 instancias con diferentes tamaños y diferentes combinaciones de parámetros.

3.2. Comparación con valores óptimos

Cuando ha sido posible, hemos comparado los resultados obtenidos por el GRASP con los valores óptimos de cada instancia, los cuales han sido obtenidos utilizando la formulación matemática del problema propuesta en [1] y el solver CPLEX 12.4. Únicamente instancias con hasta 50 nodos han podido resolverse óptimamente, ya que CPLEX no ha sido capaz de resolver instancias de mayor tamaño. El Cuadro 1 muestra los resultados obtenidos en estas instancias. La

Cuadro 1. Desviaciones del GRASP respecto de los valores óptimos.

n	p	# instancias	Opt	Dev	CPU
25	1	5	5	0.00 %	0.05
	2	10	10	0.00 %	0.21
	3	15	15	0.00 %	0.49
	4	20	15	0.17 %	1.17
	5	25	24	0.03 %	2.14
	Total	75	69	0.06 %	1.15
50	1	1	1	0.00 %	0.50
	2	2	2	0.00 %	1.85
	3	3	3	0.00 %	5.22
	4	4	4	0.00 %	11.46
	5	5	5	0.00 %	19.77
	Total	15	15	0.00 %	10.97

primera columna proporciona el número de nodos de la instancia, la segunda muestra el número de hubs a seleccionar, el número de instancias con esas características generadas se muestra en la columna tres, mientras que la columna cuatro proporciona el número de instancias para las que el GRASP ha encontrado la solución óptima. Finalmente, la desviación porcentual con respecto a la mejor solución encontrada (u óptima, si se conoce) y el tiempo de computación utilizado por el algoritmo (en segundos) son mostrados en las dos últimas columnas.

Como se puede observar, tanto el número de óptimos obtenidos como la desviación respecto del óptimo en el resto de instancias muestran el excelente comportamiento del GRASP.

3.3. Comparación con asignaciones óptimas

Con el objetivo de medir la calidad de los resultados del GRASP en instancias de tamaño medio y grande, y considerando que no existen algoritmos heurísticos previos para el UrApHMP, hemos estudiado el comportamiento del algoritmo únicamente en lo que se refiere al subproblema de asignación de nodos a hubs y al de determinar los caminos por los que se enviarán los tráficos. Para ello, utilizamos el heurístico GRASP para seleccionar el conjunto de p hubs H. Una vez H es fijado, comparamos los resultados del GRASP con los resultados obtenidos con CPLEX al resolver únicamente los subproblemas de asignación y transporte. Este experimento muestra si nuestro algoritmo GRASP es capaz de

Cuadro 2. Desviaciones del GRASP respecto del valor óptimo de las asignaciones y transporte para las instancias AP.

Tamaño	n	**# instancias**	**Best**	**Dev**
Medianas	70	35	35	0.00 %
	75	35	35	0.00 %
	80	35	35	0.00 %
	85	35	35	0.00 %
	90	35	35	0.00 %
	95	35	35	0.00 %
	Total	210	210	0.00 %
Grandes	100	36	36	0.00 %
	150	36	36	0.00 %
	200	36	36	0.00 %
	Total	108	108	0.00 %

obtener, o no, soluciones óptimas o casi óptimas para estos dos subproblemas. Los resultados obtenidos están recogidos en el Cuadro 2. También en este caso puede observarse que los resultados obtenidos por el GRASP son excelentes. Notar que, dada una selección inicial de hubs, la asignación de nodos a hubs

realizada por nuestro algoritmo y el transporte de tráficos son óptimos en las instancias probadas.

4. Conclusiones

En este trabajo hemos presentado un método heurístico basado en la metodología GRASP que proporciona soluciones de alta calidad para el Uncapacitated r-Allocation p-Hub Median Problem. El algoritmo ha sido comparado con los resultados óptimos obtenidos en instancias de tamaño pequeño/medio utilizando el solver CPLEX 12.4 para la resolución de su formulación como problema de Programación Lineal Entera. En problemas de tamaño medio/grande hemos comparado los resultados del GRASP con los obtenidos al resolver óptimamente, de nuevo con CPLEX, las fases de asignación de nodos a hubs y de transporte de tráficos, una vez determinada una elección inicial de hubs. Los experimentos se han realizado sobre 435 instancias de la literatura. Los resultados muestran el mérito del procedimiento propuesto.

Agradecimientos: Los autores desean agradecer la ayuda recibida del Ministerio de Economía y Competitividad (a través de los proyectos TIN2009-07516, MTM2009-14039-C06-02, TIN2012-35632-C02, y MTM2012-36163-C06-02) y de la Generalitat Valenciana (proyecto PROMETEO/2013/049).

Referencias

1. Yaman, H.: Allocation strategies in hub networks. European Journal of Operational Research **211**(3) (2011) 442–451
2. O'kelly, M.E.: A quadratic integer program for the location of interacting hub facilities. European Journal of Operational Research **32**(3) (1987) 393–404
3. Feo, T.A., Resende, M.G.C.: A probabilistic heuristic for a computationally difficult set covering problem. Operations Research Letters **8**(2) (1989) 67–71
4. Feo, T.A., Resende, M.G.C.: Greedy randomized adaptive search procedures. Journal of Global Optimization **6**(2) (1995) 109–133
5. Festa, P., Resende, M.G.C.: Grasp: Basic components and enhancements. Telecommunication Systems **46**(3) (2011) 253–271

Meta-Heuristic Algorithm based on Particular Swarm Optimization Algorithm and Project Scheduling Problem for the Traveling Salesman Problem

Fuentes-Penna Alejandro[1], Ruiz-Vanoye, Jorge A. [2], Diaz-Parra Ocotlán[3], Domínguez-Narvaez Joel A.[4] and Alonso-López Guillermo A. [5]

[1, 4, 5] Escuela Superior de Tlahuelilpan – Universidad Autónoma del Estado de Hidalgo (ESTl – UAEM)
[1]alexfp10hotmail.com, [4]alejandro_dn@yahoo.com.mx, [5]cuerbogeb@hotmail.com
[2, 3]Universidad Autónoma de Ciudad del Carmen
Jorge@ruizvanoye.com[2], ocotlan@diazparra.net[3]

Abstract. The main target of Traveling Salesman Problem (TSP) is to construct the path with the lowest time between different cities, visiting every one once. The Scheduling Swarm Optimization (SWO) Algorithm propose a way to solve TSP adding three aspects: time, cost effort and scope, where the scope is the pollen count, the effort is calculated multiplying time, distance and pollen count factors and dividing by the sum of them and optimizing the best way to visit the nodes graph. This development enables to have possible alternatives for future solutions in the same nodes pattern.

Keywords: Project Scheduling Problem, Particular Swarm Optimization, Traveling Salesman Problem, Scheduling Swarm Optimization (SWO).

1 Introduction

In a colony of social insects each kind of insect performs its own tasks independently from other members of the colony but with complementary activities; this tasks performed by different insects are related to each other in such a way that the colony, as a whole, is capable of solving complex problems through cooperation [2]. The collective behavior which emerges from a group of social insects has been called "swarm intelligence" [2]. The problems solutions requiring high computing performance (NP problems) researchers have been developing algorithms that mimic the behavior of nature.

Project Scheduling Problem (PSP) is an artificial intelligence technique that aims to optimize the use of resources in problems which involve cost and time variables where the cost is based on the characteristics of the problem such as: performance, economic cost, etc.

This paper discusses the incorporation of swarm intelligence and Project Scheduling Problem (PSP) to solve the traveling salesman problem as a new paradigm. Section 2 presents a review of Project Scheduling Problem as a generic name given to a whole class of problems in which it is necessary scheduling of optimum way the time, cost and resources of projects [3]. The section 3 involves the Travel Salesman Problem (TSP) description as the problem to solve with Particular Swarm Optimization (PSO) proposal. The section 5 describes the PSO as a new paradigm involving PSP and SWO, where the benefits are:

- An initial nodes evaluation: the recognition bees visit every node once; the initial exploration reduces the resources costs because there is a little bees group for identify the search space dimension.
- A continuous improvement: The optimization loop aims is to analyze the actual performance and suggest a new path if there is a better one.

2 Project Scheduling Problem

Huang et al, 2009 [4] defines a software project as a directed acyclic graph G = (V, A, S, E) where V = (1,…, n) is the set of nodes representing the events (figure 1). A is the set of arcs representing the activities, (i, j) $\in$ A is the arc acyclic graph G from node i to j, and there is only one directed arc (i, j) from i to j, S $\in$ V is the start node, and E $\in$ V is the end node. Every project has a set of activities involving a target and a set of restrictions. Project Scheduling Problem (PSP) is a way to solve this kind of problems [3].

Ruiz-Vanoye et al [5] presented an instance set of Project Scheduling Problem for projects of software development. There are many types of problems where PSP can be applied: Technical elements [6], Elements of the administration [6], Groups of industry [6], Bio–inspiration elements to choose the best option, Biotechnology elements to choose the best way to obtain the best elements combination, Scholar elements (Timetabling Scheduling, scholar administration, etc.), Health elements (nursing process, patient care management, etc.), PSP with Linear Responsibility Chart (LRC) to generate an initial workload and a project initial total cost [24], etc. We can classify the main variants of PSP [23] as: Resource-constrained Project Scheduling Problem (RCPSP), Multi-Mode Resource-Constrained Project Scheduling Problem (MRCPSP), Construction Project Scheduling Problem (construction PSP), Project Scheduling Problem for Software Development (software PSP), Payment Scheduling Problem and Time/cost trade-off Problem (TCTP).

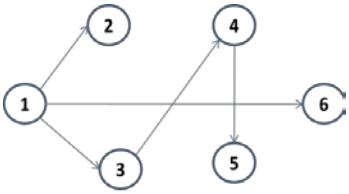


Fig. 1. Project representation

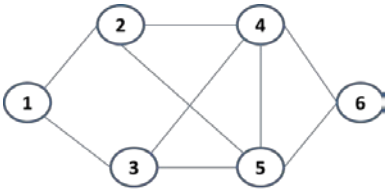


Fig. 2. Graph TSP representation

3 Traveling Salesman Problem

The traveling salesman problem (TSP) [10] models the situation of a travelling salesman who is required to pass through a number of cities without visiting twice the same city, so that the total travelling distances be minimal (Figure 2). If we define a TSP as Combinational Optimization Problem (COP) [11] P = (*S, f*) it can be an optimization problem in which are given a finite set of objects S and an objective function – maximize or minimize the journey time – f:S →R+ assigning a positive cost value to each of the objects s ∈ S. COP can be modelled as a discrete optimization problem in which the search space is defined over a set of decision variables Xi i = 1, …, n, with discrete domains. The main deterministic solutions to TSP can be: Linear programming [26], Dynamic programming formulation for an exact solution of TSP [27], the branch and bound technique [28] and Graphs [29]. In the case of non-deterministic solutions to TSP can be: Nearest neighbor algorithm [30], Insertion algorithms [30], K-Opt Heuristics [30] and Greedy Non-Deterministic Solution [25]. There are many practical real life uses of the TSP where each node can be presented as a city, geographical location, buildings, electronic points, etc.; the most common of which are [31]: transportation routing problems, salespersons, postal delivery person, etc.

4 Particle Swarm Optimization (PSO)

Poli et al [11] describes PSO as a number of simple entities (particles) placed in the search space of problem, and each particle evaluates the objective function at its current location minimizing the exploration time. The bees behavior is represented at figure 3. The PSO Algorithm consists on planed a number of simple particles (bees) in the search space of some problem, and each evaluates the objective function [11].

Fig. 3. Bees as PSO

Some authors propose different solutions for TSP using PSO like Dorigo [11], Stutzle and Hoos [16], Tsai and Tsai [18], Junjie and Dingwei [19], Salami [20], Dorigo et al [11], Machado and Lopes [32], Kaewkamnerdpong and Bentley [33] and others.

5 The Scheduling Swarm Optimization (SWO) Algorithm

The proposed algorithm is combined well distribution strategy of PSO and PSP to assign the best way. The recognition group explores the different paths according to PSO to visit every node. The main loop consists on Estimate the effort's cost between

the beehive and the nodes estimating the objective solution for each node, selecting the node with minimal effort cost (the node with minor path, max pollen and minor distance), and updating the visited nodes. The SWO main target is determine the best way (optimization loop) to travel between the beehive and the nodes selecting the optimal node based on the recognition loop and the main loop. The proposed algorithm is described as follows:

```
//Initialization
Set parameters, initialize particles
m = particles (bees)
D = search space dimension
Vk = particles velocity
nodei = Seti of flowers
costij = Travel cost between nodei and nodej
W = Total Weight
Wt = Total package's weight to deliver
Wj = package's weight taken on nodej
distanceij = Distance between nodei and nodej
N = Total of bees
k = random number of bees N
timeij = time between nodei and nodej
beep = actual ant
// Functions
max_pollen()
        Compare k –pollen from nodep and select the nodep with max pollen
End max_pollen
min_pathp()
        Compare k – pathsp and select the pathp with min total_distance
End min_path
Estimate cost_effortij(timeij, distanceij, Wt)
   If (timeij > 0, distanceij > 0, Wt > 0)
            cost_effortij = ((timeij* distanceij * Wt)/( timeij + distanceij + Wt))
End Estimate
Trace_best_way()
 selecting nodej to visit where (max_pollenp() and min_cost_effortij() and min_pathij())
   Select the best_way with (max_pollenp(), min_cost_effortij(), min_pathij())
End trace_best_way
new_best_node()
for each bee from z group
            Select random nodej to visit
                    Do
                        Estimate timeij    Estimate distanceij
                        Update nodes visited
                        Assign pathp = pathp + distanceij
                        nodei= nodej
                        Select random nodej to visit
                      While nodej was not visited
end for
end new best node()
// Recognition loop
Allocate h = random number of k on beehive (node 0)
```

```
for each bee_p from k group
        Select random node_i to visit
                Do
                    Estimate time_ij
                    Estimate distance_ij
                    Update nodes visited
                    Assign path_p = path_p + distance_ij
                    node_i= node_j
                    Select random node_j to visit
                While node_j was not visited
end for
//Main loop n=N-k
For each bee from n group
// second bee group
  Allocate n = N-k bees on initial node 0
End For
Do
        Estimate cost_effort_ij(time, distance) from neighbor nodes
        Select node_j to visit where ((max_pollen() and min_path_p() and min_cost_effor_ij())
        Update nodes visited()
        i=j
While not_visited_all_nodes()
  //Optimization loop
initial (or random) node_i // beehive
do
  Send h = random number of k from beehive (node 0)
  Select bee_z group
     Send bee_z group to new_best_node()
    trace_best_way() selecting node_i to visit where (max_pollen_p() and min_cost_effort_ij() and min_path_ij())
        i=j
  While (not finish_pollen or new_best_node())
If (finish_pollen)
   new_best_node()
```

5.1 SWO Algorithm description

An SWO algorithm differs from the algorithms proposed; it uses a Project Scheduling Problem (PSP) to enhance performance. It consists of three main sections: initialization, main loop and optimization loop. The main loop runs for a user defined number of iterations. These are described below:

Initialization

a. Set the initial parameters: variables, initial states, functions, inputs, outputs, initial trajectory, output trajectories, initial set of nodes (flowers groups)
b. Set the travel costs and distances between node 0 (beehive) and nodes
c. Define the total bees
d. Select a bees random number from total bees

Functions

a. The max pollen function compares the total pollen to extract from flowers between the node 0 with other nodes and choose the place with the largest number of pollen.

b. Min_path():This function aims to compare the different paths used by bees recognition group and select the minimal distance path.
c. Cost_effort: Each path between the node 0 and other nodes laid out by the time and distance. The effort is calculated from these factors multiplying them and dividing the result by the sum.
d. Trace_best_way():Analyzes the different path efforts between two cities, from this analysis, we choose the path of least distance, less effort and maximum pollen.

Recognition loop

a. A random bees number at node 0 to initialize the paths search (recognition group)
b. Each bee will visit every node randomly estimating the time and distance between the node 0 and other nodes estimating the total pollen

Main loop

a. Set the initial parameters with the results of initialization and recognition loop.
b. Each bee is individually placed on initial state with empty memory.
c. The rest of the bees is allocated on the same initial node from recognition group
d. Construct Bee Solution while each bee has not visited some node: Each ant constructs a path by applying the transition function between node (0,j) based on the data from recognition group and the new data from actual bee; the probability of moving from node to node depend on the parameters described before.
    a. Apply Local Search
    b. Estimate the cost effort from actual node to neighbor nodes based on time and distance estimated by recognition group
    c. Select the next node to visit based on minimize the cost effort, selecting the node with maximum pollen and minimizing the path's distance
    d. The operations are applied to individual(s) selected from the population with a probability based on fitness. Darwinian Reproduction, Structure-Preserving Crossover or Structure-Preserving Mutation.

Optimization loop

a. Set the initial parameters with the results of initialization, recognition and main loops.
b. Each node has a pollen level detected from N group.
c. Construct the best Solution: Each bee constructed a path by applying the transition function node (0,j).
d. the best way is calculated by multiplying the distance, time, the amount of pollen and divided by the sum of these parameters:
    a. While bee are visiting nodes
        i. max_pollen(): Analyze the nodes and select one with the maximum pollen level
        ii. min_cost_effort(): Analize the paths (i,j) minimizing the cost_effort
        iii. $min_path_{ij}()$: select the path (i,j) minimizing the distance

5.2 SWO Simulation

The 100 cities tour simulation [22] to be solved with Aleatorial Solution (AS), Ants Colony Optimization (ACO) and Particular Swarm Optimization (PSO) to compare the solutions. At the initial tours we presented 100 nodes to visit (figure 4). Each city is represented by a node with a random distance. TSP can be solved applying an Alea-

torial Solution (AS), where the solutions present crossing ways, a mayor path distances and a mayor effort cost for the recognition group (figure 5). The ACO Solution (Figura 6) does not present crossing ways and the results are better than AS; the time solutions decrease almost 30% comparing to AS and the solution reduced the 20% distance approximately. The effort cost is widely better if the AS effort cost is 100%, PSO presents de 85% averaged. The SWO solution (Figure 7) used less memory than AS and ACO, the time solution is similar to ACO, but the effort cost is 20% averaged lowest than AS and in some cases, presents a 13% lowest from ACO. The recognition loop identifies the distance between the beehive and the nodes estimating the total pollen for each node. The main loop established the distances between the nodes for future visits and knowing the travel time. The Optimization loop established the optimal way to visit every node at the minor time, the minor effort and identifying the pollen distribution to future recollections.

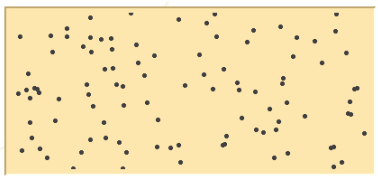


Fig. 4. Initial tour

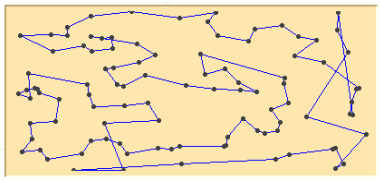


Fig. 5. Aleatorial solution

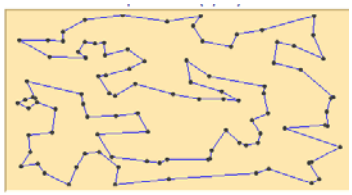


Fig. 6. ACO solution

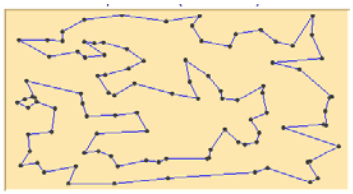


Fig. 7. SWO solution

5.3 Comparing solutions

The table 1 shows the comparison between AS, ACO and SWO results from one instance. This instance has a hundred nodes randomly distributed, ten neighbor nodes maximum, thirty-three bees from recognition group and random pollen per node. The table 2 shows the results from 30 instances where the best algorithm is SWO with the minimal effort cost estimation. In this table we assume the 100% effort cost is the AS traced path.

Table 1. Comparative table

Initialization loop		AS	ACO	SWO
m = 100 nodes $cost_{ij}$ = random function between 0…1 N = Random Total bees	Memory used	29.3 Mb	29.2 Mb	29.2 Mb
	Time solution	0.5 Sec	1.86 Sec	1.01 Sec
	Node swaps	NA	21 nodes	14 nodes
	Crossing ways	7	0	0

k = Random number of bees < N $time_{ij}$ = 0.001 seconds * distance * 1000 Neighbors per city: 10 Max iterations: 33	Total cost	NA	100%	98.2%
	Total distance	95.647	86.14	84.65
	Max Neighbors per city	10	10	10
	Max recognition bees	33%	33%	33%

Table 2. 30 effort per instance comparative table

Instance	ACO	SWO	Instance	ACO	SWO	Instance	ACO	SWO
1	85.04%	83.56%	11	87.80%	82.60%	21	83.25%	80.03%
2	98.66%	83.56%	12	89.91%	86.64%	22	92.77%	88.24%
3	97.52%	81.07%	13	94.25%	87.05%	23	85.93%	82.16%
4	97.76%	89.95%	14	88.23%	86.54%	24	84.64%	82.65%
5	88.19%	82.60%	15	91.21%	85.25%	25	97.22%	92.41%
6	92.88%	86.64%	16	88.04%	85.80%	26	90.33%	86.25%
7	94.36%	83.56%	17	93.15%	85.12%	27	88.23%	80.70%
8	93.33%	82.97%	18	86.22%	84.96%	28	81.80%	77.80%
9	90.45%	81.07%	19	87.96%	82.76%	29	91.69%	85.12%
10	93.32%	89.95%	20	91.17%	83.72%	30	90.40%	85.89%

6 Conclusions

SWO algorithm involves non probabilistic steps because it based on select the best way maximizing the resources and minimizing the efforts. The AS solution has several problems to solve TSP, the Ants Colony Optimization (ACO) algorithm may produce redundant states in earlier steps, but if we combine PSO with Project Scheduling Problem (PSP) adding a cost_effort function, the algorithm is better minimizing the time and effort's cost to enhance the behavior of the inducted system. A bees swarm moves through system states randomly based on the pollen detection but this way not always is the best solution. If we apply PSP, the bees moves are focused on reduce the redundant states by selecting the optimum way to solve the problem. On simulation results SWO has a better performance selecting the best way based on the minimal path, the minimal time and these parameters benefits minimizing the carrying pollen. The SWO Algorithm can be applied to solve different problem related to TSP as individual or multiple vehicles: School Bus Optimization Problem, Military vehicle positioning, Stuff Delivery Problem, Convoy routing Problem, Multiple Vehicle Routing Problem, Tour Planning Problem, Etc.

This is the first work introducing PSP with PSO and TSP, in future works the researches can improve the effort_cost function, introduce new paradigms or new meta heuristics. This solution can be applied on several problems with similar requirements. The SWO algorithm has been modeled on AIMMS [22], in this case we compared the solutions implementing the ACO algorithm and AS algorithm in this program. In future works SWO can be programming on C, C++, Java, etc. to implement the TSP benchmarks.

7 References

1. Dorigo M, Optimization, learning and natural algorithms. PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy, 1992 [in Italian].
2. E. Bonabeau, M. Dorigo and G. Theraulaz, Swarm Intelligence: From Natural to Artificial Systems. New York, NY: Oxford University Press, 1999.
3. Fuentes-Penna A., Ruiz-Vanoye J., Pathiyamattom-Joseph S. and Fernández-Medina M. A. Linear Responsibility Chart for PSP. 2011 Electronics, Robotics and Automotive Mechanics Conference
4. Huang, W., Ling, L., Wen, B., Cao, B. (2009). Project Scheduling Problem for Software Development with Random Fuzzy Activity Duration Times. Advances in Neural Networks – ISNN 2009. 6th International Symposium on Neural Networks, ISNN 2009. Wuhan, China, May 2009 Proceedings, Part II. Springer. LNCS 5552. ISSN: 0302-9743. ISBN-10: 3-642-01509-3.
5. Ruiz-Vanoye Jorge A., Pathiyamattom-Joseph Sebastian. y Fernández-Medina Miguel A. and Fuentes-Penna A. Project Scheduling Problem for Software Development Library – PSPSWDLIB. Revista de Ciências da Computação, 2010, nº5.
6. Jorge A. Ruiz-Vanoye, Ocotlán Díaz-Parra, José C. Zavala-Díaz, Alejandro Fuentes-Penna, Juan C. Olivares-Rojas. A survey of Project Scheduling Problems (PSP). Proceedings of the 15th Annual International Conference on Industrial Engineering Theory, Applications and Practice, pp. 460-472, Mexico City, Mexico October 17-20, (2010) ISBN 97809652558-6-8.
7. Jorge A. Ruiz-Vanoye, Ocotlán Díaz-Parra, Alejandro Fuentes-Penna, José C. Zavala-Díaz. Productos de Innovación Tecnológica obtenidos de imitar la Naturaleza. Revista de Educación y Divulgación Científica y Tecnológica de la UDLAP (ALEPH ZERO). Número 57, Julio - Septiembre (2010).
8. Jorge A. Ruiz-Vanoye, Ocotlán Díaz-Parra, Felipe Cocón, Andrés Soto, Ma. De los Ángeles Buenabad Arias, Gustavo Verduzco-Reyes, Roberto Alberto-Lira. Meta-Heuristics Algorithms based on the Grouping of Animals by Social Behavior for the Traveling Salesman Problem. International Journal of Combinatorial Optimization Problems and Informatics, Vol. 3, No. 3, pp. 104-123, Sep-Dec (2012). ISSN:2007-1558.
9. Jorge A. Ruiz-Vanoye and Ocotlán Díaz-Parra. An Overview of the Theory of Instances Computational Complexity.International. Journal of Combinatorial Optimization Problems and Informatics, Vol. 2, No. 2, May-Aug, 2010
10. Lawler E, Lenstra JK, Rinnooy Kan AHG, Shmoys DB. The travelling Salesman problem. New York: John Wiley & Sons; 1985
11. Poli R., Kennedy J., and Blackwell T. Particle Swarm Optimization an overview. Swarm Intell (2007) 1: 33–57. DOI 10.1007/s11721-007-0002-0
12. Papadimitriou CH, Steiglitz K. Combinatorial optimization—Algorithms and complexity. New York: Dover; 1982.
13. S. Goss, S. Aron, J. L. Deneubourg, and J. M. Pasteels, "Self-organized shortcuts in the argentine ant," Naturwissenschaften, vol. 76, pp. 579–581, 1989.
14. B. H¨olldobler and E. O. Wilson, The Ants. Berlin: Springer-Verlag, 1990.
15. R. Beckers, J., Deneubourg, and S. Goss, "Trails and U-turns in the selection of the shortest path by the ant Lasius Niger," J. Theoretical Biology, vol. 159, pp. 397–415, 1992

16. Thomas Stutzle and Holger Hoos. MAX-MIN Ant System and Local Search for the Traveling Salesman Problem. 0-7803-3949-5/97/$10.00 0 1997 IEEE
17. Marco Dorigo and Gianni Di Caro. Ant Colony Optimization: A New Meta-Heuristic. 0-7803-5536-9/99/$10.00 01999 IEEE
18. Cheng-Fa Tsai and Chun- Wei Tsai. A New Approach for Solving Large Traveling Salesman Problem Using Evolutionary Ant Rules.
19. Pan Junjie and Wang Dingwei. An Ant Colony Optimization Algorithm for Multiple Travelling Salesman Problem. Proceedings of the First International Conference on Innovative Computing, Information and Control (ICICIC'06). 0-7695-2616-0/06 2006 IEEE.
20. Nada M. A. Al Salami. Ant Colony Optimization Algorithm. UbiCC Journal, Volume 4, Number 3, August 2009
21. Marco Dorigo and Thomas Stützle. Ant Colony Optimization Book. The MIT Press. Editorial Springer.
22. Optimization Software for Mathematical Programming. http://www.aimms.com. Last visited: 02/02/2013.
23. Jorge A. Ruiz-Vanoye, Ocotlán Díaz-Parra, José C. Zavala-Díaz, Alejandro Fuentes-Penna, Juan C. Olivares-Rojas. Models, resources and activities of project scheduling problems. International Journal of Industrial Engineering-Theory Applications and Practice. ISSN: 1943-670X
24. Fuentes-Penna A., Ruiz-Vanoye J.A., Pathiyamattom-Joseph S. and Fernández-Medina M.A. 2011 Electronics, Robotics and Automotive Mechanics Conference 978-0-7695-4563-9/11 $26.00 © 2011 IEEE. DOI 10.1109/CERMA.2011.80 461.
25. Sanchit Goyal. A Survey on Travelling Salesman Problem.
26. Dantzig GB, Fulkerson DR, Johnson SM (1954). Solution of a Large-scale Traveling Salesman Problem." Operations Research, 2, 393410.
27. Held M, Karp RM (1962). A Dynamic Programming Approach to Sequencing Problems". Journal of SIAM, 10, 196 - 210.
28. Little JDC, Murty KG, Sweeney WD, Karel C, "An Algorithm for the Traveling Salesman Problem," Operational Research, vol. 11, no. 6, pp. 972-989, 1963.
29. Eppstein, D. 2003b. The traveling salesman problem for cubic graphs. In Proceedings of the 8th Workshop on Algorithms and Data Structures. Lecture Notes in Computer Science, vol. 2748. Springer-Verlag, New York, 307–318.
30. Hahsler, Michael; Hornik, Kurt (2007), "TSP Infrastructure for the Traveling Salesperson Problem
31. Leonardo Zambito. The Traveling Salesman Problem: A Comprehensive Survey. Submitted as a project for CSE 4080, Fall 2006.
32. Machado T. R and Lopes H. S. A hybrid Particle Swarm Optimization model for the traveling salesman problem. Adaptive and Natural Computer Algorithms. Springer Computer Science. Springer Wien New York. Pp. 255 – 258. ISBN: 3-211-24934-6. 2005
33. Kaewkamnerdpong B. and Bentley P. J. Perceptive Particle Swarm Optimisation. Adaptive and Natural Computer Algorithms. Springer Computer Science. Springer Wien New York. Pp 259 – 263. ISBN: 3-211-24934-6. 2005

Resolución del problema EDP para Grafos No Dirigidos utilizando algoritmos MSGA y GRASP

Bernardo Martín[1], Abraham Duarte[2] y Ángel Sánchez[2]

[1]INECO. Paseo de la Habana, 138. 28036 Madrid
bernardo.martin@ineco.es
[2]Dpto. Ciencias de la Computación, Universidad Rey Juan Carlos.
28933 Móstoles (Madrid)
{abraham.duarte, angel.sanchez}@urjc.es

Abstract. En este trabajo se comparan diferentes estrategias implementadas basadas en algoritmos voraces multi-arranque (MSGA) y en algoritmos GRASP para el problema *Edge Disjoint-Paths* (EDP), que consiste en encontrar el mayor número de caminos con arcos disjuntos entre pares de nodos terminales dados, en grafos no dirigidos. Los resultados se han evaluado sobre grafos de *benchmarks* estándares definidos para este problema.

Keywords. Problema EDP (*Edge-Disjoint Paths*), metaheurísticas, algoritmos voraces multi-arranque, GRASP, *k-shortest paths*, redes de comunicaciones.

1 Introducción

El problema de encontrar el mayor número de caminos disjuntos en grafos no dirigidos (conocido como *Edge-Disjoint Paths problem*, o abreviadamente EDP), presenta una importante utilidad práctica para redes de comunicaciones [2][7]. En efecto, en este tipo de redes una operación fundamental consiste en el establecimiento de rutas de conexión (es decir, caminos en el grafo correspondiente), que resulten ser simultáneas y disjuntas, para el intercambio de información entre pares de nodos terminales de la red. Ello obedece a que las peticiones de conexión pueden llegar simultáneamente y, también, se trata de establecer en mayor número de rutas entre nodos terminales de manera que no interfieran entre ellas mediante la compartición de *switches* (nodos) y/o de links (aristas) entre los caminos del grafo. Este hecho implicaría el cumplimiento de unas restricciones técnicas y también un mayor grado de paralelismo en la comunicación [1][2].

Este trabajo presenta nuevos algoritmos para la resolución de una versión del problema EDP donde los caminos no pueden compartir aristas. Otras versiones del problema consideran que no se pueden compartir nodos del grafo [2]. Las soluciones planteadas se basan en versiones de algoritmos voraces multi-arranque (*Multi-Start Greedy Algorithms*, abreviadamente MSGA) y de algoritmos GRASP [3]. El problema puede describirse más formalmente como sigue. Sean $G=(V, E)$ un grafo no diri-

gido, que representa a la red de comunicaciones, y sea $T=\{(s_i,t_i) \mid (i = 1, 2, \ldots, |T|) \cap (s_i, t_i \in V)\}$ una colección predefinida de *commodities* o pares de nodos terminales de G que pueden ser conectados a través de un camino. Un conjunto de *commodities* es realizable si existe una colección de caminos entre cada nodo terminal s_i con cada nodo terminal t_i, tales que los caminos correspondientes no comparten arcos. Se considera aquí el problema combinatorio de establecer el mayor número de conexiones entre los dos nodos componentes de *commodities* de T tal que dicho conjunto sea realizable en el grafo original G. Por lo tanto, el valor de la función objetivo $f(S)$ de una solución S factible del problema EDP, se define como: $f(S) =|S|$. Se trata de maximizar el número de pares terminales conectados tal que resulten realizables en G. El problema de decisión relativo a si un conjunto de *commodities* dado T es o no realizable sobre un grafo G es NP-completo [4]. Otras variantes del problema EDP para árboles parciales de orden k [8] o para grafos serie-paralelos [6] también son problemas NP-completos. En la literatura se han propuesto principalmente soluciones basadas en colonias de hormigas (ACO) para el problema considerado [7][1][2].

2 Algoritmos voraces multi-arranque (MSGA) implementados

2.1 Descripción de estrategias plantadas

La resolución del problema mediante los algoritmos voraces multi-arranque se ha realizado a partir de las implementaciones previas realizadas por los autores en [5], pero, en esta ocasión, se usan grafos no dirigidos. En aquel estudio, se implementó el algoritmo voraz siguiendo tres criterios a la hora de buscar soluciones en cada una de las iteraciones a ejecutar:

- Estrategia *sp-MSGA* (*shortest paths MSGA*): Se obtiene el camino entre cada par terminal $(s_i,t_i) \in T$ buscando el camino mínimo por medio del algoritmo de Dijkstra.
- Estrategia *dod-MSGA* (*decreasing out-degree MSGA*): Para crear el camino entre pares terminales $(s_i,t_i) \in T$, toma para cada nodo que pertenezca al camino, el nodo adyacente con mayor grado de adyacencia posible, y lo añade a dicho camino.
- Estrategia *iod-MSGA* (*increasing out-degree MSGA*): Similar al anterior, en este caso para construir el camino se elige el nodo con menor grado de adyacencia posible.

Además de las estrategias anteriores, en este trabajo se presenta la opción de precarga o precálculo *off-line* de resultados mediante el algoritmo *k-shortest paths* (que llamaremos estrategia *ksp-MSGA*).

2.2 Implementación del algoritmo según la estrategia *ksp-MSGA*

Teniendo en cuenta que los algoritmos de las estrategias *sp-MSGA*, *dod-MSGA* e *iod-MSGA* son similares a los desarrollados para los grafos dirigidos presentados en

[5], en este caso se detallará únicamente la estrategia *ksp-MSGA*. Básicamente, el algoritmo se divide en dos partes bien diferenciadas. Por un lado, se encuentra la función general `MSGA(N,solKSP)` que realiza las labores de diversificación de las tareas de búsqueda, creando para cada arranque una lista ordenada y aleatoria de los pares terminales a conectar. Por otro lado, se usa la función auxiliar `obtenerSolucionParcial(G,`L_{tabu}`,s,`t_1`)` que realiza las labores de búsqueda local para cada par terminal (s_i, t_i). Esta función toma los posibles caminos asignables a dicho par de la información que le proporciona el algoritmo *k-shortest paths*. A continuación, se muestra el pseudocódigo de ambas funciones.

```
function MSGA(N,solKSP=AlgoritmoKSP(G, T, K))
  input:
    N= número de permutaciones de T para multi-arranque
    G=(V,A), grafo dirigido
    T={(si,ti)| ∀i∈1..m: si,ti ∈ V pares terminales
    K número caminos calculados por ksp para cada par terminal(si,ti).
    solKSP= {c1..cm}|ci={d1..dr} | ∀i∈1..m /
         si,ti ∈ V, pares terminales de ci
         d1..dr ∈ ci, lista de caminos con longitud creciente |r| < K
  output: mejor solución Smejor al problema EDP con MSGA
  var
    Sj: solución parcial iteración j-ésima
    Π: conjunto de permutaciones sobre elementos de T
    f: valor de la función objetivo para una solución
    Pi: camino entre los nodos terminales si y ti
    fmejor: valor de la función objetivo para Smejor
  begin
    Smejor=nulo;
    fmejor=-infinito;
    Π={Π1..Π|k|| Πi permutación de {(si,ti)|i∈1..k}∈T};
    for j=1..N do // primeras N iteraciones de Π
      Sj=nulo; // inicializa solución parcial para iter. j
      for l=1..Q do // ∀(si,ti)∈T ^ Q=|V|
        Pi= obtenerSolucionParcial(solKSPi);
        Sj=Sj+Pi; // incluir solución parcial al total
        Â= Â-{a|a∈Pj}; //eliminar de Â los arcos de Pi
      end for
      if f(Sj) > f(Smejor) then
        Smejor = Sj // actualiza la mejor solución
      end if
    end for
end function
```

```
function obtenerSolucionParcial(c_l)
  input: c_l, caminos solución ksp en longitud creciente del par (s_l,t_l)
  output: camino P_l con menor longitud entre s_l y t_l
  var L_tabu, lista tabú de caminos
  begin
       L_tabu = nulo;
       do
          d_i= primerCamino(c_l) | d_i ¬∈ L_tabu;
          if d_i ∈ Â then
            return d_i;
          else
            L_tabu=L_tabu+d_i;
       while (d_i!=nulo);
end function
```

3 Algoritmos GRASP implementados

3.1 Implementaciones según estrategias *sp-GRASP*, *dod-GRASP* e *iod-GRASP*

Los algoritmos GRASP según las estrategias *sp-GRASP*, *dod-GRASP* e *iod-GRASP*, estrategias descritas en la sección anterior, se han implementado como algoritmos voraces multi-arranque, con la particularidad de incluir aleatoriedad en la elección de los "mejores nodos", según el valor dado a α que es el parámetro usado para el cálculo de la lista restringida RCL_α en el GRASP.

En el pseudocódigo presentado a continuación, se muestra solamente la parte de búsqueda local del algoritmo para el criterio *sp-GRASP*. Como se puede apreciar, para esta estrategia la variable CL es una lista de caminos ordenados por longitud creciente. Los algoritmos para las estrategias *dod-GRASP* e *iod-GRASP* son similares al anterior. Para estos casos, CL es una lista de caminos, ordenados por la suma del grado de adyacencia de los nodos que forman el camino, en orden decreciente y creciente, respectivamente.

```
function obtenerSolucionParcial(G,L_tabu,s,t_l)
  inputs:
    G=(V,A), grafo no dirigido; s, nodo actual
    L_tabu, lista tabú de nodos
    t_l, nodo final del par terminal (s_l,t_l)∈T
  output: camino más corto p_l entre s y t_l
  var
    a: conjunto de nodos adyacentes a uno dado
    CL: Lista de caminos, ordenados por longitud
       creciente, candidatos a generar solución parcial
    RCL_α: Lista restringida de los α mejores caminos de CL
```

```
  begin
    L_tabu= L_tabu+s;
    if (s= t_1) then
      return t_1; //camino encontrado
    else if s ∈ nodoFinal(G) then //nodo sin sucesores
      return nulo; //no encuentra camino
    else // algoritmo de Dijkstra de caminos mínimos
      a=nodosAdyacentes(s);
      for i=1..|a| do
        if a_i ┐∈ L_tabu then // si no se encuentra en L_tabu
          P_1= obtenerSolucionParcial(G,L_tabu,a_i,t_1);
          CL=CL+P_i;
        end if
      end for
      P_1=obtenerCaminoAleatoriamente(RCL_α);
      //actualiza L_tabu, eliminando caminos de la lista excepto escogida
      L_tabu= L_tabu-{n|n ∈ (RCL-P_1)};
      return P_1;
    end if-else
end function
```

3.2 Implementación con estrategia de precarga (*ksp-GRASP*)

Para el caso de la estrategia *ksp-GRASP*, se aplica el algoritmo GRASP sobre una lista CL obtenida a partir de los resultados obtenidos de la ejecución previa del algoritmo *k-shortest paths*. Dicha lista contiene los caminos en orden creciente de longitud. De esta manera, será más sencillo obtener el umbral y la lista restringida RCL_α.

```
function MSGA(N,CL=ordenacionCaminosPorLongitud(solKSP=
                  AlgoritmoKSP(G, T, K)))
  inputs:
    N= número de permutaciones de T para multi-arranque
    G=(V,A), grafo dirigido
    T={(s_i,t_i)| ∀i∈1..m: s_i,t_i ∈ V pares terminales
    K es una constante que indica el número de caminos
   calculados por KSP para cada par terminal (s_i,t_i).
    solKSP= {c_1..c_m}|c_i={d_1..d_r} | ∀i∈1..m /
         s_i,t_i ∈ V, pares terminales de c_i
         d_1..d_r ∈ ci, lista de caminos en orden de longitud
        creciente |r| < K
    CL= { d_1..d_u | ∀d_i∈{c_1..c_m} /  |d_i-1|≤ |d_i|≤ |d_i+1|
         lista de caminos solución ordenados de KSP
          ordenados por longitud creciente
```

```
  output: mejor solución S_mejor al problema EDP con GRASP
  var
    S_j: solución parcial iteración j-ésima
    f: valor de la función objetivo para una solución
    f_mejor: valor de la función objetivo para S_mejor
  begin:
    S_mejor=nulo;
    f_mejor=infinito;
    for j=1..N do // primeras N iteraciones de Π
      S_j=nulo; // inicializa solución para it. j
      S_j= obtenerSolucionIteracion(CL);
      if f(S_j) > f(S_mejor) then
        S_mejor = S_j // actualiza la mejor solución
      end if
    end for
end function

function obtenerSolucionIteracion (CL)
  input:
    CL: { d_1..d_u | ∀d_i∈{c_1..c_m} / |d_i-1|≤ |d_i|≤ |d_i+1|
  output: conjunto de caminos S_1 obtenidos del conjunto CL
          a partir de una posición inicial en la lista Π_i
  var
    RCL_α: Lista restringida de los α mejores caminos de CL
    umbral: longitud máxima para caminos c ∈ CL / c ∈ RCL_α
    Π={Π_1..Π_|CL|| Π_i permutación de RCL_α}
    L_tabu, lista tabú de caminos
    S_j: solución parcial iteración
  begin
      S_j = nulo; // iniciar solución de la iteración
      L_tabu  = nulo; // iniciar L_tabu
      do
      // Calcular umbral: camino long. mín y máx de CL.
      Umbral= |d_1| + α*( |d_u|- |d_1|)/ d_1,d_u ∈ CL;
      //calcular RCL_α
      RCL_α= c_x ∈ CL / |c_x|< Umbral;
        //tomar camino aleatorio de RCL_α
        d_i =random({d_1..d_u} ∈ RCL_α) ^ d_i ┐∈ L_tabu;
        // si el camino es factible: de momento no
        // pertenece, a la solución, ni ésta contempla los
        // pares terminales del camino.
```

```
            if (di ┐∈ Sj) ^ ((si,ti)∈T)^((si,ti)∈ di)/
               ((si,ti) ┐∈ Sj) then
               Sj = Sj + di; // se añade a la solución de la it.
               CL = CL – di; // se elimina camino de CL. Se
                  //actualiza dl y du si se diese el caso.
            end if
            //se actualiza la lista para impedir que se tome
            //varias veces el mismo camino
            Ltabu  =Ltabu+ di;
         // mientras que no se haya seleccionado todos los
         // elementos de RCLα
         while ┐(∀di ∈ RCLα | di ∈ Ltabu);
         return Sj;
end function
```

4 Resultados

Los grafos utilizados en las pruebas de los algoritmos desarrollados, que corresponden a un subconjunto de los presentados en los trabajos [1] y [2], son los siguientes: *graph3.bb* (renombrado abreviadamente en las sucesivas tablas de resultados como *G1*), *graph4.bb* (renombrado como *G2*), *AS-BA.R-Wax.v100e217.bb* (renombrado como *G3*), *bl-wr2-wht2.10-50.sdeg.bb* (renombrado como *G4*) y *mesh25x25.bb* (renombrado como *G5*, respectivamente). El parámetro *#c* representa el número de *commodities* o pares terminales establecidos para cada grafo.

Para cada estrategia algorítmica implementada aparecen tres columnas:

- La columna q_{max} es el valor máximo de función objetivo obtenida mediante la aplicación del algoritmo sobre un grafo dado para el total de las *commodities* o instancias de pares terminales.
- La segunda de las columnas, q_{med}, muestra la media de los valores de función objetivo obtenida mediante la aplicación del algoritmo sobre un grafo dado para el total de las *commodities* o instancias de pares terminales.
- La tercera de las columnas *t(s)* indica el valor medido del tiempo de ejecución del algoritmo (en segundos) para un conjunto o instancia de pares terminales y un grafo concreto.

Para todos los experimentos se han usado los siguientes valores de parámetros: N=100 y K=10, respectivamente, donde N representa el número de permutaciones de T para la ejecución multi-arranque y K es una constante que indica el número de caminos precalculados por el algoritmo *k-shortest paths* para cada par terminal (s_i,t_i). Los experimentos se han realizado en un ordenador Intel Pentium Dual-Core E5800 a 3,20GHz y con 4GB RAM.

4.1 Algoritmos MSGA

Resultados de las estrategias de distancia mínima *sp* y de precarga *ksp*

		sp-MSGA			*ksp-MSGA*		
Grafo	*#c*	q_{max}	q_{med}	*t(s)*	q_{max}	q_{med}	*t(s)*
	16	16	15	5,46	16	14	0,02
G1	41	29	25	6,30	30	25	0,03
	65	37	29	6,67	37	32	0,04
G2	43	38	35	91,93	35	30	0,17
	10	9	7	0,55	9	7	0,01
G3	25	15	12	0,64	15	12	0,02
	40	19	16	0,66	20	16	0,02
	50	24	19	25,10	28	20	0,26
G4	125	39	33	28,85	43	36	0,30
	200	54	43	31,73	57	46	0,36
	62	26	20	37,16	42	34	0,41
G5	156	32	27	39,19	61	56	0,47
	250	34	30	39,67	75	68	0,56

Table 1. Resultados para las estrategias *sp-MSGA* y *ksp-MSGA*

Resultados de las estrategias de máxima y mínima adyacencia

		dod-MSGA			*iod-MSGA*		
Grafo	*#c*	q_{max}	q_{med}	*t(s)*	q_{max}	q_{med}	*t(s)*
	16	11	9	2,13	16	15	11,43
G1	41	16	12	2,33	30	26	12,96
	65	17	14	2,44	37	30	13,44
G2	43	18	14	19,12	41	37	190,01
	10	9	6	0,49	10	8	1,15
G3	25	13	10	0,61	15	12	1,35
	40	16	12	0,67	19	16	1,48
	50	19	15	20,10	26	20	54,43
G4	125	30	24	24,98	45	35	69,44
	200	35	29	28,18	60	47	80,74
	62	15	12	22,62	29	24	122,18
G5	156	21	17	25,06	38	32	127,36
	250	26	20	26,29	42	37	128,68

Table 2. Resultados para las estrategias *dod-MSGA* e *iod-MSGA*

4.2 Algoritmos GRASP

Resultados de las estrategias de distancia mínima *sp* y de precarga *ksp*

		sp-GRASP			*ksp-GRASP*		
Grafo	*#c*	q_{max}	q_{med}	*t(s)*	q_{max}	q_{med}	*t(s)*
	16	12	10	0,89	15	13	1,08
G1	41	15	13	1,04	29	24	12,91
	65	18	15	1,10	35	31	47,88
G2	43	18	15	8,17	33	29	15,32
	10	8	6	0,22	8	6	0,33
G3	25	13	10	0,27	15	12	3,33
	40	16	13	0,29	20	16	12,07
	50	17	13	6,35	26	20	23,10
G4	125	28	22	8,63	42	35	331,71
	200	35	28	10,13	56	45	1360,53
	62	17	13	11,70	40	32	42,87
G5	156	21	18	13,28	61	54	647,15
	250	25	21	14,24	73	68	2695,39

Table 3. Resultados para las sstrategias *sp-GRASP* y *ksp-GRASP*

Resultados de las estrategias de máxima y mínima adyacencia

		dod-GRASP			*iod-GRASP*		
Grafo	*#c*	q_{max}	q_{med}	*t(s)*	q_{max}	q_{med}	*t(s)*
	16	13	10	0,90	13	10	1,70
G1	41	16	13	1,91	17	13	1,92
	65	18	15	2,00	18	15	2,04
G2	43	17	15	15,37	19	15	15,45
	10	9	7	0,44	9	7	0,43
G3	25	13	10	0,63	13	10	0,55
	40	18	13	0,64	18	13	0,62
	50	21	16	17,78	21	16	17,91
G4	125	32	26	22,90	31	26	22,90
	200	36	31	26,69	36	32	26,67
	62	18	13	14,66	20	14	14,76
G5	156	23	18	16,94	22	18	17,11
	250	25	21	18,83	27	22	19,07

Table 4. Resultados para las estrategias *dod-GRASP* e *iod-GRASP*

Comparando los resultados para estos grafos con los presentados por Blesa y Blum en [2], usando un algoritmo ACO extendido, puede resumirse que nuestros resultados en relación con el valor de la función objetivo (valores q_{max} y q_{med}) son ligeramente inferiones a los presentados en [2]. Sin embargo, los correspondientes tiempos de ejecución conseguidos *t(s)* por nuestros algoritmos para dichos grafos mejoran considerablemente (en promedio, en un 98,94%) a los presentados por Blesa y Blum.

5 Conclusión

Este trabajo compara experimentalmente diferentes versiones de algoritmos de tipo MSGA y GRASP para el problema EDP. Para decidir cuál es el mejor algoritmo desarrollado, hay que tener en cuenta el valor alcanzado por cada algoritmo para la función objetivo y también el tiempo de ejecución correspondiente. Con respecto al primer criterio, se puede apreciar que para los algoritmos MSGA, los mejores resultados corresponden a las estrategias *iod* y *ksp*. La estrategia *iod*, funciona bien para todos los grafos analizados, excepto para la instancia *mesh25×25*. La estrategia *ksp* obtiene buenos resultados para todos los grafos analizados. En el caso de los algoritmos GRASP, la estrategia *ksp* es la que genera los valores de la función objetivo más altos. Se presentan buenos resultados en todas las instancias, mostrando los mejores resultados para los grafos *AS-BA.R-Wax.v100e217* y *mesh25×25*. En cambio, GRASP no funciona bien si se decide por utilizar alguna de las estrategias restantes. En resumen, con respecto a la función objetivo, MSGA obtiene los mejores resultados cuando siguen las estrategias *ksp* o *iod*. Entre los GRASP, la estrategia *ksp* produce los mejores resultados. Si se introduce el tiempo en la decisión de mejor resultados, entonces *ksp-MSGA* aparece como ganador absoluto, obteniendo el mejor tiempo para todos los grafos seleccionados. Como futuro trabajo se plantea abordar el problema *Directed Edge-Disjoint Paths* (DEDP) usando un enfoque similar al propuesto en este trabajo.

Agradecimientos

Este trabajo ha sido financiado parcialmente por el proyecto TIN2011-29827-C02-01 del Ministerio de Ciencia e Innovación.

Bibliografía

1. Blesa, M. y Blum C.: Ant Colony Optimization for the Maximum Edge-Disjoint Paths Problem, G.R. Raidl et al. (eds.): *EvoWorkshops 2004*, *LNCS 2005*, págs. 160-169, 2004.
2. Blesa, M. y Blum C.: Finding edge-disjoint paths in networks by means of artificial ant colonies*, Journal of Mathematical Modelling and Algorithms* 6(3), págs. 361-391, 2007.
3. Blum C. y Roli A.: Metaheuristics in combinational optimization: Overview and conceptual comparison, *ACM Computing Surveys* 35(3), págs. 268-308, 2003.
4. Karp, R.: Complexity of computer computations*. Chapter Reducibility among Combinatorial Problems*, Plenum Press, New York, págs. 85-103, 1972.
5. Martin B., Sánchez A. y Duarte A.: Resolución del problema de los caminos disjuntos en grafos dirigidos usando técnicas metaheurísticas*. CEDI 2007. JAEM 2007*, 2007.
6. Nishizeki, T., Vygen J. y Zhou, X.: The edge-disjoint paths problem is NP-complete for series–parallel graphs. *Discrete Applied Mathematics* 115, págs. 177–186, 2001.
7. Nowé, A., Verbeeck, K. y Vrancx, P.: Multi-Type Ant Colony: The Edge Disjoint Paths Problem. M. Dorigo et al. (eds.): *Ants 2004, LNCS 3172*, págs. 202-213, 2004.
8. Zhou, X. y Nishizeki, T.: The Edge-Disjoint Paths Problem is NP-Complete for Partial k-Trees, *Proceedings of the 9th International Symposium on Algorithms and Computation* (ISAAC'98), págs. 417-426, 1998.

SISTEMA DE AUTOADAPTACIÓN DE LOS PARÁMETROS DEL ALGORITMO CLEARING PARA PROBLEMAS COMBINATORIOS: Aplicación al *job shop*

Pérez, E.[1] Posada, M.[1] y Herrera, F.[2]

[1] Dpto. Organización de Empresas y C.I.M., Universidad de Valladolid, 47011.
{elena, posada}@eii.uva.es
[2] Dpto. Ciencias de la Computación e I.A., Universidad de Granada, 18071
herrera@decsai.ugr.es

Resumen. Los algoritmos de optimización multimodal se utilizan en problemas en cuyo espacio de búsqueda existen múltiples óptimos locales y globales. Sin embargo, su utilización se ve dificultada por los parámetros que utilizan, principalmente, el radio de nicho. En este trabajo se ha desarrollado un sistema autoadaptativo de los dos parámetros que utiliza el algoritmo Clearing. El nuevo algoritmo, denominado Auto-Clearing, utiliza un conjunto externo para almacenar las soluciones óptimas que se van encontrando y, en base a la información proporcionada por estas soluciones, se obtiene los valores más apropiados de sus parámetros. La experimentación realizada muestra que tanto la eficacia como la eficiencia para encontrar el mayor número posible de óptimos globales en cada ejecución son mejores en el algoritmo Auto-Clearing que en el algoritmo Clearing.

Palabras clave: Optimización Multimodal, Algoritmos Genéticos, Algoritmo Clearing.

1 Introducción al problema multimodal

Los problemas de optimización multimodales han atraído la atención de los investigadores debido a su complejidad. Se caracterizan en que en su espacio de búsqueda existen múltiples óptimos globales y/o locales. En la Fig. 1 se muestra dos ejemplos de funciones multimodales clásicas [1]. También nos encontramos con multimodalidad en la mayoría de los problemas de secuenciación, por ejemplo en los del tipo *job shop* [2,3].

La multimodalidad empezó a ser estudiada con algoritmos genéticos (AG) a finales de los años 80. Los AG se basan en la mejora evolutiva de una población de soluciones. Aplicando técnicas de diversificación, lo que se conoce como AG multimodales o de nichos, se divide la población en grupos de soluciones con características

similares. Cada grupo de soluciones evoluciona hasta converger a un óptimo del problema, permitiendo obtener diferentes óptimos globales y/o locales.

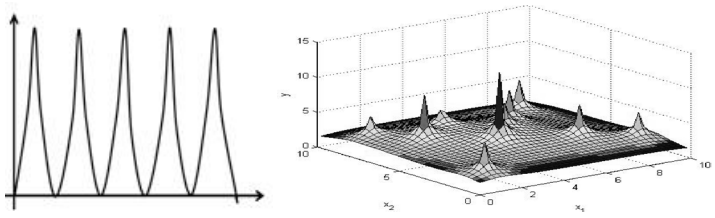


Fig. 1. Ejemplo de problemas multimodales (Izq. 5 óptimos globales y 0 locales, Der. 1 óptimos global y 9 locales)

El primer AG multimodal se basaba en el concepto de penalización o *sharing* [4]. A raíz de este estudio surgieron otros muchos algoritmos, entre los que destacamos los siguientes:

- Adaptive niche hierarchical genetic algorithm [5].
- Clearing algorithm [6]
- Niche identification techniques with fitness sharing [7].
- Deterministic crowding [8].
- Multi-niche crowding genetic algorithm [9].
- Restricted competition selection method [10].
- Restricted competition selection with pattern search method [11].
- Species conserving genetic algorithm [12].
- Adaptive species conservation genetic algorithm [13]
- Quick hierarchical fair competition [14].

El principal problema de todos estos algoritmos es que para dividir la población en grupos o nichos es necesario utilizar una serie de parámetros, principalmente el radio que define un nicho (σ_{share}). El valor más adecuado para este parámetro es difícil de establecer ya que depende de la orografía del espacio de búsqueda. Si establecer el valor más adecuado es difícil en problemas de codificación real (Fig. 1), más lo es en problemas combinatorios como el *job shop*, donde los óptimos no pueden ser representados en un espacio de soluciones, ni se puede definir una distancia euclídea entre ellas. Además, establecer el valor más adecuado supone un gran esfuerzo computacional mediante prueba y error. Esta complejidad hace que los AG multimodales apenas se utilicen en los problemas industriales, a pesar de la gran potencialidad que presentan.

El algoritmo Clearing [6] utiliza dos parámetros para la diversificación. El primer parámetro es el radio de nicho (σ_{share}). Define la máxima distancia que puede haber entre dos soluciones para que ambas pertenezcan al mismo nicho. El segundo parámetro es el número de ganadores por nicho (kappa). Define el máximo número de soluciones permitidas en cada nicho. En este trabajo presentamos un método que permite eliminar estos dos parámetros. Para ello, creamos un conjunto externo semejante al

que se utilizan en los algoritmos de optimización multi-objetivo (como por ejemplo, el conjunto élite del algoritmo SPEA2 [15]). En este conjunto externo se almacenan todas las soluciones de mayor calidad que se van encontrando a lo largo de las iteraciones. La distancia entre estas soluciones definirá el radio de nicho y nos permitirá calcular el valor del parámetro kappa. Estos valores irán cambiando según las necesidades de la búsqueda en un proceso auto-adaptativo. A este algoritmo lo hemos denominado Auto-Clearing.

Este trabajo se organiza de la siguiente forma. En la sección 2 se introduce el concepto de distancia y radio de nicho, parámetros clave en la mayoría de AG multimodales. En la sección 3 se describe brevemente el algoritmo Clearing. En la sección 4 se desarrolla la propuesta de esta contribución: el algoritmo Auto-Clearing. En la sección 5, se muestran los resultados de la experimentación realizada y el análisis estadístico. Finalmente, en la sección 6 se presentan las conclusiones.

2 El parámetro radio de nicho

En la mayoría de los algoritmos clásicos de optimización multimodal (los denominados algoritmos de nichos o *niching algorithms* en inglés), existe un parámetro que define la pertenencia de una solución a un determinado nicho en base a una distancia. Se define la distancia entre dos soluciones como la medida que nos permite cuantificar lo diferentes que son dichas soluciones [4]. En problemas de codificación real la distancia entre dos soluciones ($x_1, x_2, \in R^n$) se determina mediante la distancia Euclídea:

$$d(x_1, x_2) = \sqrt{\sum_{j=1}^{n} \left(x_1^j - x_2^j\right)^2} \quad (1)$$

En problemas combinatorios, determinar la distancia entre dos soluciones es más complejo. En concreto, para el problema *job shop* se ha utilizado el número de piezas que se secuencien en distinto orden [2,3] (Tabla 1).

Tabla 1. Ejemplo de distancia en el problema del *job shop*.

Soluciones	Distancia
(1 3 2)(2 3 1)(3 2 1)	5
(3 2 1)(1 3 2)(3 2 1)	

Una vez definido el concepto de distancia, se define el concepto de nicho. Nicho es el conjunto de soluciones cuya distancia entre si es menor que un umbral máximo que se denomina radio de nicho. En la Fig. 2 se presentan tres nichos diferentes en un espacio de búsqueda. Se observa que la distancia entre las soluciones de un mismo nicho puede ser, como mucho, el radio de nicho, mientras que la distancia entre soluciones de dos nichos diferentes será mayor que el radio de nicho. Este concepto se utiliza para penalizar las soluciones de una población cuando hay muchas soluciones que pertenecen a la misma zona, y beneficia a las soluciones que pertenecen a áreas menos pobladas. De esta forma, el algoritmo mantiene la diversidad, permite la su-

pervivencia de soluciones en diferentes zonas y logra que cada zona evolucione convergiendo a su óptimo.

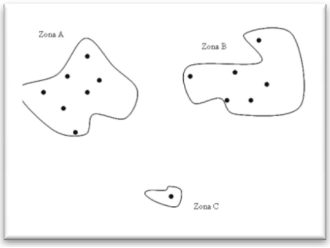


Fig. 2. Identificación de nichos.

3 Algoritmo Clearing

En la naturaleza los individuos de una misma especie son los que luchan entre sí para lograr los recursos limitados existentes en la naturaleza, sin embargo esta lucha no es frecuente entre individuos de especies con diferentes características. Así es posible que convivan animales tan dispares como el león y la hormiga. No tienen que luchar por los mismos recursos y, por ello, sobreviven. No hay competencia entre ellos. El algoritmo *Clearing* utiliza el concepto de recursos limitados para eliminar la competencia entre soluciones que son muy diferentes [6]. Se utiliza el parámetro radio de nicho (σ_{share}) para identificar las soluciones que pertenecen al mismo nicho que el mejor individuo de cada área. El parámetro kappa (k) define el número máximo de soluciones que pueden sobrevivir en cada nicho. El proceso se describe a continuación (Fig. 3).

La población inicial (*P(t)* de tamaño *N*) se modifica para formar otra población (*P'(t)* de tamaño *N'*) desde la que se hace la selección. Las soluciones de la población de partida *P(t)* se ordenan de mayor a menor calidad (en un problema de maximización). En primer lugar, se selecciona la primera solución de la ordenación que pasa directamente a la población *P'(t)*. Después, se calcula la distancia entre cada solución de la población inicial *P(t)* y la seleccionada. Aquellas soluciones cuya distancia es menor que el radio de nicho son las que pertenecen al mismo nicho. De este conjunto se reservan las *k* mejores soluciones (las de mayor calidad) mientras que a las restantes se les asigna una calidad nula. De esta forma, se limita la competencia dentro del nicho. En este punto, se tiene una solución en la población *P'(t)*, mientras que en la población de partida *P(t)* se mantiene las *N* iniciales pero algunas de ellas con calidad cero. El proceso continua volviendo a seleccionar la siguiente solución de la ordenación cuya calidad sea distinta de cero. El proceso termina cuando todas las soluciones que quedan por seleccionar en *P(t)* tienen calidad cero.

Del conjunto *P'(t)* se selecciona la población de padres. Sobre la población de padres se realiza el cruce y la mutación obteniéndose la población de hijos. Para terminar el ciclo, el algoritmo Clearing implementa la siguiente forma de elitismo. De la población resultante *P'(t)* se seleccionan y pasan a la siguiente población *P(t+1)* las

soluciones cuya calidad es superior a la media de la calidad original de la población *P(t)*. El resto de soluciones, hasta completar las *N* necesarias para formar la población *P(t+1)*, se obtiene de las mejores de la población de hijos.

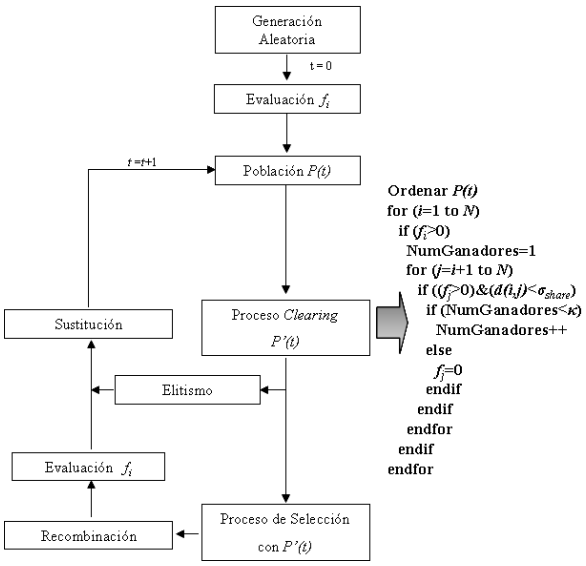


Fig. 3. Algoritmo Clearing

4 Algoritmo Auto-Clearing

El algoritmo Auto-Clearing se propone con el objetivo de evitar estimar (mediante prueba y error) los valores de los parámetros del algoritmo Clearing: el radio de nicho y kappa. La razón es que conocer a priori el valor de ambos parámetros es complejo y dependiente de la función a optimizar (más aún, en problemas combinatorios).

El esqueleto del algoritmo es el mismo que el del algoritmo Clearing, salvo que se añade un conjunto externo donde se archivarán las mejores soluciones que se van encontrando en el proceso evolutivo. Cada vez que la calidad de las soluciones almacenadas en el conjunto externo mejora, el conjunto se borra y entran las nuevas mejores soluciones. Cada vez que hay un cambio en el conjunto externo se calcula la distancia entre las soluciones archivadas. El radio de nicho es una función de esta distancia. El proceso completo para un problema de maximización se realiza mediante el siguiente proceso:

1. **Establecimiento de los parámetros del proceso**:
   - Tamaño de la población *N* y *t*amaño del conjunto externo (*ExtSet*) *N**.
   - El número de soluciones que hay dentro del conjunto *ExtSet* se denomina *NumSol*. Inicialmente es 0 porque está vacío.
   - Criterio de parada: número de evaluaciones máximo (*NumEval*).
   - Se rellena la primera población *P(1)* con soluciones aleatorias.

2. **Evaluación de la población *P(1)***.
3. **Inicialización del Conjunto *ExtSet***. Todas las soluciones, distintas entre sí, con la máxima calidad de *P(1)* entran en el conjunto *ExtSet*. A esta calidad la denominaremos $F_{maxExtSet}$.
4. **Cálculo del radio de nicho y k**.
   (a) Si en el conjunto *ExtSet* sólo hay una solución, se establece un radio de nicho igual a 0 y un valor de k=1. Con estos valores, el proceso del Clearing sólo eliminará las soluciones que son copias.
   (b) Si en el conjunto *ExtSet* hay más de una solución (*NumSol*), el radio de nicho es la mitad de la distancia mínima entre las soluciones del conjunto *ExtSet*:

$$Radio\ de\ Nicho = min\{d(x_i,x_j)/2 \quad \forall\, x_i, x_j \in ExtSet\} \tag{2}$$

$$k = min\ (5, N^*/NumSol) \tag{3}$$

   Con este valor de k no permitimos que se generen grandes nichos, sino muchos nichos pero de pequeño tamaño.
   (c) Si durante *0.1xNumEval/N* evaluaciones el conjunto *ExtSet* no se ha actualizado (con nuevas soluciones o con soluciones de mejor calidad), se activa el proceso de diversificación modificando el radio de nicho y el valor de k para favorecer la exploración. El radio de nicho es un cuarto de la distancia máxima entre las soluciones del conjunto *ExtSet*:

$$Radio\ de\ Nicho = max\{d(x_i,x_j)/4 \quad \forall\, x_i, x_j \in ExtSet\} \tag{4}$$

$$k = 1 \tag{5}$$

5. **Obtención de *P'(t).*** Clearing sobre *P(t)* con los valores del radio de nicho y k calculados en el paso anterior.
6. **Selección y reproducción sobre *P'(t)***. Se obtiene la población hijos *Q(t)*.
7. **Actualización del Conjunto *ExtSet***.
   (a) Si la máxima calidad de *Q(t)* es mejor que $F_{maxExtSet}$, el conjunto *ExtSet* es borrado y se guardan en el todas las soluciones, distintas entre si, de *Q(t)* con la máxima calidad. Dicha calidad será $F_{maxExtSet}$.
   (b) Si la máxima calidad de *Q(t)* es igual que $F_{maxExtSet}$, el conjunto *ExtSet* se actualiza con las soluciones de *Q(t)* con dicha calidad que sean distintas de las ya almacenadas en el conjunto *ExtSet*. Recordemos que el conjunto *ExtSet* tiene un tamaño máximo de N^*. Una vez que esté lleno, la actualización no se realizará
   (c) Si la máxima calidad de *Q(t)* es menor que $F_{maxExtSet}$, el conjunto *ExtSet* no se actualizará.
8. **Sustitución con elitismo** según Clearing. Formación del nuevo *P(t),* con $t=t+1$. Volver al paso 4.

5 Experimentación

Para comprobar el comportamiento del algoritmo Auto-Clearing se ha seleccionado ejemplos muy conocidos del problema de secuenciación tipo *job shop* disponibles en la siguiente web: http://people.brunel.ac.uk/~mastjjb/jeb/orlib/jobshopinfo.html.

Son los denominados la[01-05], de tamaño 10x5 (10 trabajos x 5 máquinas), la[06-10] de 15x5, la [16-20] y mt10 de 10x10, la[11-15] de 20x5 y mt20 de 20x5.

Por un lado, de los AG se busca que sean eficaces. Para ello, la medida de eficacia que se ha utilizado es minimizar el makespan (*Cmax*), definido como el momento en el que termina de ejecutarse todos los trabajos en el taller. Es una medida de eficacia estándar para el problema *job shop* y cuyo valor para los ejemplos utilizados es conocido (Tabla 2). Por otro lado, al estar ante un problema multimodal resuelto con un AG multimodal, se busca que el AG sea eficiente, es decir, que alcance el objetivo con el menor consumo de recursos posibles. En este problema multimodal se define eficiencia como el número de óptimos globales encontrados por ejecución (*Nmo*).

Tabla 2. Valor óptimo del *Cmax* para los ejemplos utilizados

ejemplo	la01	la02	la03	la04	la05	la06	la07	la08	la09	la10	la11
Cmax	666,0	655,0	597,0	590,0	593,0	926,0	890,0	836,0	951,0	958,0	1222,0
ejemplo	**la12**	**la13**	**la14**	**la15**	**la16**	**la17**	**la18**	**la19**	**la20**	**mt10**	**mt20**
Cmax	1039,0	1150,0	1292,0	1207,0	945,0	784,0	848,0	842,0	902,0	930,0	1165,0

Los parámetros generales del algoritmo genético son:

- Tamaño de población y del conjunto externo = 100 individuos
- Número de evaluaciones: según el tamaño del problema, 100.000 (para tamaño 10x5), 300.000 (para tamaño 15x5) y 500.000 (para tamaños 10x10 y 20x5).
- Sistema de selección: Torneo binario.
- Operador Cruce: OX (*Order crossover*) [16]
- Probabilidad de cruce: 0.8
- Operador Mutación: OBM (*Order based mutation*) [17]
- Probabilidad de mutación: 0.2
- Cada ejecución se ha realizado 30 veces.
- Medidas de estudio:
  - Eficacia, *makespan* medio (*Fm*). Calculado como la media de la mejor solución encontrada en cada ejecución.
  - Eficiencia, número medio de las soluciones óptimas contenidas en el conjunto *ExtSet* (*Nmo*). Calculado como la media del número de óptimos globales localizados en cada ejecución. Si en una ejecución no se localiza el óptimo global conocido, se considera que el número de óptimos es 0 en esa ejecución.

El algoritmo Clearing se ha ejecutado con 5 radios de nicho diferentes (0, 10, 20, 30 y 40) y 3 valores de *k* (1, 5 y 10). Los resultados se muestran en la Tabla 3. En negrita están marcados los mejores valores de *Fm* y *Nmo*. Se comprueba que, en cada ejemplo, el mejor valor de *Fm* se obtiene con diferentes valores del radio de nicho y de *k*. Es decir, la parametrización de estos valores depende de la orografía o distribución de los óptimos en el espacio de búsqueda. La diversidad de valores de los parámetros obtenidos confirma la dificultad de determinar a priori el valor más adecuado para cada ejemplo.

Tabla 3. Resultados comparativos de los algoritmos Clearing y Auto-Clearing en problemas de *job shop*. Las celdas con el símbolo --- indican que no se han encontrado ningún óptimo global.

	Kappa	1	5	10	1	5	10	1	5	10	1	5	10	1	5	10	AutoClearing
	σshare	0	0	0	10	10	10	20	20	20	30	30	30	40	40	40	
la01	Fm	**666,0**	667,6	671,1	**666,0**	**666,0**	**666,0**	**666,0**	**666,0**	**666,0**	667,1	**666,0**	666,0	675,7	670,1	668,2	**666,0**
	Nmo	**79,93**	37,92	35,17	65,30	37,27	35,93	7,27	8,37	11,43	1,25	2,87	3,76	1,18	1,41	1,80	**100,00**
la02	Fm	671,6	674,4	675,8	666,7	670,0	667,8	**663,9**	666,4	665,4	674,2	665,1	665,6	698,3	684,8	681,3	**663,1**
	Nmo	**79,75**	28,60	19,33	2,67	2,67	2,80	1,25	2,50	2,33	1,00	1,29	1,50	---	1,00	---	**100,00**
la03	Fm	616,6	622,6	621,8	**611,9**	614,0	615,1	618,0	615,9	613,7	624,5	620,8	617,3	639,1	630,0	627,1	**609,0**
	Nmo	**71,00**	---	---	---	---	1,00	---	---	1,20	---	---	1,00	---	2,00	---	**88,00**
la04	Fm	607,4	608,8	609,3	606,5	**603,3**	603,5	607,4	606,2	604,4	609,5	605,5	605,3	623,4	613,8	612,3	**600,1**
	Nmo	---	**32,00**	---	---	---	---	---	---	---	---	---	2,00	---	---	---	**80,20**
la05	Fm	**593,0**	**593,0**	**593,0**	**593,0**	**593,0**	**593,0**	**593,0**	**593,0**	**593,0**	**593,0**	**593,0**	**593,0**	**593,0**	**593,0**	**593,0**	**593,0**
	Nmo	**82,20**	51,43	47,13	78,83	50,00	48,23	60,17	46,77	47,07	8,53	14,63	21,20	2,50	3,20	5,50	**100,00**
la06	Fm	**926,0**	**926,0**	**926,0**	**926,0**	**926,0**	**926,0**	**926,0**	**926,0**	**926,0**	**926,0**	**926,0**	**926,0**	**926,0**	**926,0**	**926,0**	**926,0**
	Nmo	**84,70**	42,60	34,50	82,13	39,70	37,10	79,33	41,90	34,50	73,67	33,77	32,83	48,67	23,10	19,60	**100,00**
la07	Fm	**890,0**	890,8	892,4	890,1	**890,0**	890,1	890,1	890,3	890,2	890,7	891,3	891,0	891,3	891,1	891,2	**890,0**
	Nmo	**82,47**	31,88	24,63	71,79	36,23	24,18	26,76	21,31	13,45	4,26	6,00	5,50	1,89	3,00	4,00	**100,00**
la08	Fm	**863,0**	**863,0**	863,2	**863,0**	**863,0**	**863,0**	**863,0**	**863,0**	**863,0**	**863,0**	**863,0**	**863,0**	**863,0**	**863,0**	**863,0**	**863,0**
	Nmo	**82,80**	38,63	30,03	80,77	41,07	36,87	74,73	37,20	29,37	41,77	20,10	15,47	18,40	11,20	7,17	**100,00**
la09	Fm	**951,0**	**951,0**	**951,0**	**951,0**	**951,0**	**951,0**	**951,0**	**951,0**	**951,0**	**951,0**	**951,0**	**951,0**	**951,0**	**951,0**	**951,0**	**951,0**
	Nmo	**82,73**	40,37	36,53	82,70	44,03	39,67	78,70	39,03	36,60	71,87	36,57	32,30	42,50	19,50	18,07	**100,00**
la10	Fm	**958,0**	**958,0**	**958,0**	**958,0**	**958,0**	**958,0**	**958,0**	**958,0**	**958,0**	**958,0**	**958,0**	**958,0**	**958,0**	**958,0**	**958,0**	**958,0**
	Nmo	**83,57**	49,40	45,17	83,10	49,90	47,57	82,00	51,50	46,00	79,53	48,03	43,93	64,07	41,70	41,70	**100,00**
mt10	Fm	1008,6	1019,2	1011,2	997,3	1004,6	1017,0	**996,3**	1007,6	1004,2	1004,8	1002,7	1009,2	1012,2	998,1	1009,1	**990,4**
	Nmo	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
la16	Fm	989,9	992,8	998,6	991,7	993,3	999,1	990,0	993,8	992,6	**989,0**	994,0	992,0	992,8	990,9	998,6	**987,2**
	Nmo	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
la17	Fm	798,6	802,9	802,9	**795,4**	799,9	800,9	796,2	800,6	798,8	801,3	799,5	798,6	807,7	799,1	801,1	**793,6**
	Nmo	---	**25,00**	---	1,00	---	---	---	---	---	---	---	---	---	---	1,00	**100,00**
la18	Fm	874,2	889,0	885,8	868,3	878,4	878,2	**868,3**	869,8	872,8	876,4	869,1	878,1	886,2	876,2	874,6	**866,3**
	Nmo	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	**100,00**
la19	Fm	883,4	891,8	894,6	**878,6**	880,4	889,7	881,3	884,4	886,5	887,4	882,2	888,0	895,0	887,0	881,8	**870,0**
	Nmo	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
la20	Fm	931,2	944,1	953,9	925,0	932,7	947,6	**924,3**	930,0	936,7	934,9	927,6	936,5	942,1	935,2	936,1	**919,4**
	Nmo	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	**51,00**
mt20	Fm	1242,9	1237,3	1259,0	1248,3	1247,6	1249,5	1256,6	**1234,6**	1253,3	1266,9	1259,7	1255,9	1267,8	1254,0	1274,6	1235,0
	Nmo	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
la11	Fm	**1222,0**	**1222,0**	**1222,0**	**1222,0**	**1222,0**	**1222,0**	**1222,0**	**1222,0**	**1222,0**	**1222,0**	**1222,0**	**1222,0**	**1222,0**	**1222,0**	**1222,0**	**1222,0**
	Nmo	**83,67**	40,83	33,03	82,07	40,63	35,70	80,87	41,80	37,57	78,77	38,77	35,43	75,70	37,03	32,90	**100,00**
la12	Fm	**1039,0**	**1039,0**	**1039,0**	**1039,0**	**1039,0**	**1039,0**	**1039,0**	**1039,0**	**1039,0**	**1039,0**	**1039,0**	**1039,0**	**1039,0**	**1039,0**	**1039,0**	**1039,0**
	Nmo	**84,43**	37,67	34,77	81,63	40,83	34,77	80,37	41,43	33,80	80,67	41,07	33,03	76,53	37,67	33,93	**100,00**
la13	Fm	**1150,0**	**1150,0**	**1150,0**	**1150,0**	**1150,0**	**1150,0**	**1150,0**	**1150,0**	**1150,0**	**1150,0**	**1150,0**	**1150,0**	**1150,0**	**1150,0**	**1150,0**	**1150,0**
	Nmo	**85,00**	40,40	33,07	81,20	39,57	32,93	79,80	42,07	33,87	78,13	38,60	37,07	74,60	34,37	30,93	**100,00**
la14	Fm	**1292,0**	**1292,0**	**1292,0**	**1292,0**	**1292,0**	**1292,0**	**1292,0**	**1292,0**	**1292,0**	**1292,0**	**1292,0**	**1292,0**	**1292,0**	**1292,0**	**1292,0**	**1292,0**
	Nmo	**84,03**	53,00	47,73	82,97	52,43	49,37	83,20	55,93	50,03	82,07	52,10	52,20	80,47	53,57	51,00	**100,00**
la15	Fm	1212,2	1214,5	1217,2	1209,7	1210,0	1209,9	1214,9	1210,2	1211,9	1216,7	1210,6	**1209,1**	1218,9	1213,5	1215,0	**1207,5**
	Nmo	**80,87**	33,88	25,17	71,05	34,19	28,25	25,20	27,96	30,33	3,09	20,40	20,27	5,50	8,25	7,06	**100,00**

Los resultados obtenidos con el algoritmo Auto-Clearing se muestran en la última columna de la Tabla 3. El algoritmo Auto-Clearing obtiene mejor *Fm* que el algoritmo Clearing con la mejor combinación de valores de sus parámetros en todos los ejemplos, salvo para el mt20 y en los ejemplos en los que siempre se consigue el óptimo del ejemplo en todas las ejecuciones donde se obtiene el mismo valor de *Fm*. En cuanto al valor *Nmo,* el número de óptimos encontrados por el algoritmo Auto-Clearing es superior al que proporciona el algoritmo Clearing para cualquier combinación de valores del radio de nicho y de *k*.

Se realiza el estudio estadístico no paramétrico utilizando para ello los test de Friedman y Holm [18]. En primer lugar se analiza la eficacia con el valor *Fm*. En la

Tabla 4 se muestra el ranking que hace el test de Friedman de los mejores algoritmos (el mejor es el que obtiene el menor valor y el peor algoritmo es el de mayor valor). Con esta información se realiza una preselección de los 7 mejores algoritmos, aplicando sobre ellos el test de Holm (Tabla 5). Los resultados obtenidos muestran que el algoritmo AutoClearing es el mejor de todas las diferentes configuraciones del algoritmo Clearing con un *p*-valor bastante bajo, el mayor es 0.06961 para la configuración 10K1. En la Tabla 6 se muestra el test de Holm para el valor de la eficiencia, *Nmo*. En este caso, igualmente el algoritmo AutoClearing es el mejor de todos los demás algoritmos Clearing con *p*-valor menor que en el caso anterior.

Tabla 4. Resultados de la clasificación según Friedman, para *Fm*

Algoritmo	Ranking	Algoritmo	Ranking	Algoritmo	Ranking
σ_{share} 0 k1	7.273	σ_{share} 20 k1	6.682	σ_{share} 40 k1	12.227
σ_{share} 0 k5	10.523	σ_{share} 20 k5	7.454	σ_{share} 40 k5	9.545
σ_{share} 0 k10	12.000	σ_{share} 20 k10	7.363	σ_{share} 40 k10	10.636
σ_{share} 10 k1	5.909	σ_{share} 30 k1	9.682	**AutoClearing**	**4.318**
σ_{share} 10 k5	7.182	σ_{share} 30 k5	7.932		
σ_{share} 10 k10	8.886	σ_{share} 30 k10	8.387		

Table 5. *p*-valor ajustado del test de Holm, con AutoClearing como algoritmo de control, *Fm*

Algoritmo	p-valor ajustado
σ_{share} 0 k1	0.00898
σ_{share} 20 k10	0.00898
σ_{share} 20 k5	0.00898
σ_{share} 10 k5	0.01574
σ_{share} 20 k1	0.02172
σ_{share} 10 k1	0.06961

Table 6. *p*-valor ajustado del test de Holm, con AutoClearing como algoritmo de control, *Nmo*

Algoritmo	p-valor ajustado
σ_{share} 20 k10	3.94E-9
σ_{share} 20 k5	2.61E-7
σ_{share} 10 k5	1.68E-6
σ_{share} 20 k1	8.79E-6
σ_{share} 10 k1	0.00480
σ_{share} 0 k1	0.04671

6 Conclusiones

Los algoritmos genéticos adaptados para optimizar problemas multimodales ofrecen una alternativa a los algoritmos clásicos de optimización. Sin embargo, debido a la utilización de parámetros que son muy dependientes del problema a optimizar, su utilización en problemas reales (por ejemplo, los combinatorios como el Job Shop) es muy reducida. En este trabajo hemos presentado una modificación del algoritmo Clearing, denominado Auto-Clearing, que realiza un autoajuste de sus dos parámetros (radio de nicho y *k*). El algoritmo mantiene un conjunto externo de soluciones óptimas. La distancia entre dichas soluciones permite parametrizar el radio de nicho y *k* de forma auto-adaptativa. Se ha comprobado que tanto la eficacia y como la eficiencia para encontrar el mayor número posible de soluciones óptimas globales en cada ejecución se mejora con el algoritmo presentado.

Bibliografía

1. Hedar, A.R.: Test Functions for Unconstrained Global Optimization. http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO_files/Page364.htm
2. Pérez, E., Herrera, F., Hernández, C.:. Finding multiple solutions in job shop scheduling by niching genetic algorithms. *Journal of Intelligent Manufacturing* 14:3-4 (2003) 323-339.
3. Pérez, E., Posada, M., Herrera, F.: Analysis of new niching genetic algorithms for finding multiple solutions in the job shop scheduling.. *Journal of Intelligent Manufacturing* 23:3 (2012) 341-356.
4. Goldberg, D. E., Richardson, J.: Genetic algorithms with sharing for multimodal function optimization. In *Proc. of the 2nd international conference on genetic algorithms,* 41-49 (1987).
5. Dunwey, G., Fengping, P., Shifan, X.: Adaptive niche hierarchy genetic algorithm. *In Proc. of IEEE TENCON*, Pp. 39-42 (2002).
6. Pétrowski, A.: A clearing procedure as a niching method for genetic algorithms. In *Proc. IEEE International conference on evolutionary computation*. Japan, 798-803 (1996).
7. Lin, C., Wu, W.: Niche identification techniques in multimodal genetic search with sharing scheme. *Advances in Engineering Software* (33). Elsevier, 779–791 (2002).
8. Mahfoud, S. W.: Crowding and preservation revisited. *Parallel problem solving form nature II*. Männer, R. and Manderick, B. (Eds.). 27-36 (1992).
9. Cedeño, W., Vemuri, V. R.: Analysis of speciation and niching in the multi-niche crowding GA. *Theoretical Computers Science*, (229), 177-197 (1999).
10. Lee, Ch., Cho, D., Jung, H.: Niching genetic algorithm with restricted competition selection for multimodal function optimization. *IEEE transactions on magnetics*, doi: 10.1109/20.767361, (1999).
11. Kim, J., Cho, D., Jung, H., Lee, C.: Niching genetic algorithm adopting restricted competition selection combined with pattern search method. *IEEE Transactions on magnetic*, doi: 10.1109/20.996257, (2002).
12. Li, J., Balazs, M., Parks, G. T., Clarkson, P. J.: A species conserving genetic algorithm for multimodal function optimization. *Evolutionary computation*, doi:10.1162/106365602760234081, (2002).
13. Dunwey, G., Fengping, P., Shifan, X.: Adaptive niche hierarchy genetic algorithm. *In Proc. of IEEE TENCON*, Pp. 39-42, (2002).
14. Hu, J. J., Goodman, E. D.: Robust and efficient genetic algorithms with hierarchical niching and a sustainable evolutionary computation model. In Deb, K et al., (Eds). *GECCO*, 1220-1232, (2004).
15. Zitzler. E. y Thiele,L.: Multiobjective evolutionary Algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4), 257 – 271, (1999).
16. Davis, L.: Adapting operators probabilities in genetic algorithms. *In* Schaffer, J. D. (Ed.). *Proc. of the 3th international conference on genetic algorithms.* Kaufmann. San Mateo, 375-378, (1989).
17. Mattfeld, D.C.: *Evolutionary search and the job shop. Investigations on genetic algorithms for production scheduling*. Springer. Berlin, (1995).
18. García S, Molina D, Lozano M, Herrera F.: A Study on the Use of Non-Parametric Tests for Analyzing the Evolutionary Algorithms' Behaviour: A Case Study. CEC'2005 Special Session on Real Parameter Optimization. Journal of Heuristics 15 (2009) 617-644.

A simple heuristic approach for generating real-time solutions to real-life crew pairing problems

Jesus M. Puentes, Christine A. Oketch, Angel A. Juan

Computer Science Dept., Open University of Catalonia,
08018 Barcelona, Spain
{jpuentesg, coketch0, ajuanp}@uoc.edu

Abstract. The airline industry is one of the most competitive industries existing today. The Crew Scheduling Problem involves the process of assigning a crew to the operation of a designated route. Several approaches to solve this complex problem exist in the literature. In this paper, we propose a heuristic and fast approach to complete the crew pairing. Our approach aims at obtaining efficient flight pairings with short computer processing times, while satisfying all the airline or collective agreement constraints.

Keywords: Airline management, crew scheduling and pairing, heuristics.

1 Introduction

Being an extremely competitive industry, one of the current trends in airline companies is the reduction of daily operation costs. Just after the cost of the fuel, personnel costs represents one of the main costs of daily operations in most airline companies, about 14% if both cabin and pilots crews are considered together (Figure 1).

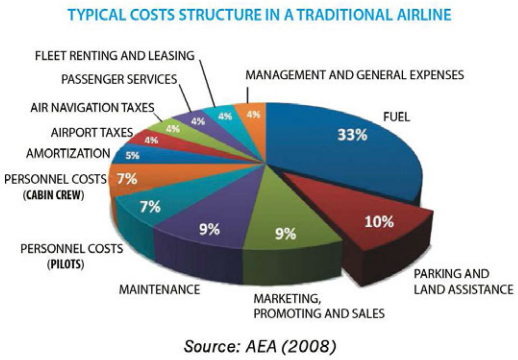


Fig. 1. Main expenses in an airline (Source: AEA [1]).

Personnel expenses are probably the most important area for potential savings in airlines. Most airlines use a sequential procedure to plan their operations. First, they create a schedule that defines origin and destination cities, as well as departure and arrival times for each flight leg. Then, they assign an aircraft type to each flight leg trying to maximize expected profits. After setting aircraft routes and side constraints, airlines configure crew assignments and schedules. With this in mind, airlines are striving to reduce the costs by implementing efficient and effective solutions to solve the so-called Crew Scheduling Problem (CSP). The CSP is a combinatorial optimization problem in which crew personnel must be assigned to different working days (flight pairings), depending on the city they are placed and achieving several constraints –e.g.: rest periods prior to working days, maximum daily or monthly flight hours, maximum number of flights, etc. Due to its intrinsic difficulty, it is usual to divide the CSP into two subproblems, the Crew Pairing Problem (CPP) and the Crew Rostering Problem (CRP).

A crew pairing is a sequence of flight legs or segments that begin and end at a crew base –i.e., in each sequence of legs the arrival city of a flight leg coincides with the departure city of the next flight leg. Inefficient or incomplete crew pairing can be very costly to an airline [2]. Crew pairing problems can be divided into two categories: cock-pit crew pairing and cabin crew pairing. Each category has its own rules and, usually, cock-pit crew pairing is more complicated than cabin crew because each pilot is qualified to fly only one type of aircraft [3]. The Crew Pairing Problem is usually approached by first generating a huge number of pairings. From this huge collection of legal pairings, a set of pairings that has minimal –or as low as possible– costs and which ensures that each flight is manned by exactly one crew is selected. Once final crew pairings have been obtained, they are not addressed to individual crew members because they are part of the next phase: the Crew Rostering Problem. It represents the process of assigning individual crew members to crew pairings –which have been obtained in the previous phase– on a monthly basis. Therefore, the objective is to assign crew pairings to particular crew members in order to make an efficient use of human resources. While assigning the crew to flight pairs, several factors have to be taken into consideration depending on airlines: number of days off per week, number of hours per week, number of nights per week in the case of layover flights, age restriction on operating some legs in the case of pilots, crew dead heading etc.

The purpose of this paper is to propose a heuristic algorithm that provides, in almost ‘real-time’, feasible and good solutions that can be useful in solving the aforementioned crew scheduling problem.

2 Related Work

Crew cost is the largest expenditure an airline can control, therefore effective assignment of crews to flights is a very important aspect of airline planning [4]. Inefficient crew pairing can be very costly to an airline. Deng et al. [2], propose an Ant Colony Optimization (ACO) algorithm to solve the crew scheduling problem.

Likewise, Yan et al. [5], propose a column generation model for minimizing crew costs throughout proper cockpit crew pairing. Unlike cabin crew pairing –where a crew can be qualified for operate in several aircraft types–, cockpit crew can only be qualified to fly one type of aircraft, thus making the generation of the cockpit crew scheduling more complicated.

There are several factors of stochastic nature that can disrupt the smooth flow of operations –e.g. weather conditions, flight cancellations, delays, etc. These factors, in turn, can imply crew scheduling disruptions, which are costly for the company. Schaefer et al. [6] used Monte Carlo simulation to face the crew scheduling problem under uncertainty. In Mercier et al. [7], the authors combine Benders's decomposition and column generation. Benders decomposition is used to reformulate the problem so as to reduce the number of variables at the expense of an increase in the number of constraints. Other authors propose creating and assigning crew working plans in just one step, instead of doing it in a sequential way [8]. These authors provide solution techniques based on simple tree searches, column generation, and the shortest-path algorithm. This way, when making modifications, their algorithm undoes part of the job done during the crew scheduling creation phase. Yu et al. [9] and Wei et al. [10] discuss disruptions in the original pairing assignments due to a flight cancellation or delays, and propose methods for crew-pairing recovery/repair. Medard and Sawhney [11] integrate the pairing and rostering models, and apply a methodology that integrates both pairing construction and pairing assignment in one step. Similarly, Weide et al. [12] consider the aircraft routing problem together with both crew scheduling problems, one for the technical crew and another one for the cabin crew. Instead of solving the integrated model, they propose to solve both problems iteratively. Thus, they start with a low cost crew pairing solution which does not take aircraft routings into account. The, at each iteration they solve the individual aircraft routing problem first, taking into account the current crew pairing solution.

A stochastic crew scheduling model and a solution methodology for integrating disruptions in the evaluation of crew schedules was devised in [13]. The goal was to develop a robust solution able to support moderate disruptions. A stochastic integer programming model was used to develop a branching algorithm to identify expensive flight connections and find alternative sources. Finally, it is also worthy to mention the hybrid genetic algorithm for airline crew scheduling proposed by Levine [14].

3 Our Approach

Our algorithm takes the existing flight legs and then creates pairings of legs for crew members. Obviously, these pairs will have to respect a set of constraints. Figures 2 and 3 show the pseudo-codes associated with our heuristic. The pseudo-code in Figure 2 describes how to generate a set composed of basic routes of paired legs. At the beginning (line 1), we do not have any values in our list yet. The available legs are sorted by their starting time (line 2). The starting time is generated from the origin airport leg time/date as well as the arrival time/date of the destination airport. Then, we verify if the leg is not empty (line 3), and a new empty route will be created

(line 4). The next leg is extracted (line 5), as well as the inverse leg (line 6). The next leg and the inverse leg are merged –if feasible– to create the new route (line 7). This new route is then be added to the basic route (line 8) and the new basic route created is added to the list of paired legs.

```
procedure generateBasicRoutesOfPairedLegs(legs)

% This procedure generates basic roundtrip routes composed of pairs of
% inverse legs, e.g.: (BCN-JFK) + (JFK-BCN) = BCN-JFK-BCN
01  basicRoutes <- emptyList
02  legs <- sortByStartingTime(legs)
03  while {legs is not empty} do
04    newRoute <- emptyRoute
05    leg <- extractNextLeg(legs)
06    invLeg <- extractInverseLeg(leg, legs)
07    newRoute <- merge(leg, invLeg)
08    basicRoutes <- add(newRoute)
09  end while
10  return basicRoutes

end procedure
```

Fig. 2. Pseudo-code for generating basic routes of paired legs.

```
procedure generateInitialSol(legs, airports)

% This procedure generates an inital solution by merging basic routes
% (pairs of legs) starting at each airport,
% e.g.: (BCN-MAD-BCN) + (BCN-VAL-BCN) = BCN-MAD-BCN-VAL-BCN
01  globalSol <- emptySol
02  basicRoutes <- generateBasicRoutesOfPairedLegs(legs)
03  for each {airport in airports} do
04    basicRoutes(airport) <- getBasicRoutes(basicRoutes, airport)
05    airportSol <- emptySol
06    while {basicRoutes(airport) is not empy} do
07      newRoute <- extractNextStartingRoute(basicRoutes(airport))
08      for each {route in basicRoutes(airport)} do
09        tentativeRoute <- merge(newRoute, route)
10        if {tentativeRoute is feasible} then
11          basicRoutes(airport) <- delete(route, basicRoutes(airport))
12          newRoute <- tentativeRoute
13        end if
14      end for
15      airportSol <- add(newRoute, airportSol)
16    end while
17    globalSol <- add(airportSol, globalSol)
18  end for
19  return globalSol

end procedure
```

Fig. 3. Pseudo-code for generating an initial solution.

Once the basic routes have been generated, we will create the initial solution by merging the basic routes at each airport (Figure 3). The idea behind this would be to balance the crew routes and return a working and feasible crew route combination. Thus, for each airport we consider its associated set of the basic routes (lines 3 and 4). Then we check if the basic route is empty (line 6). If it is not, we extract the next starting route (line 7) associated with the current airport. For each route in the list, we merge it with the new route in order to create a tentative route (line 9). At this stage, feasibility of the tentative route is checked (line 10). If feasible, we accept the tentative route as valid (line 12) and add it to the global solution. At the end of the iterative process, a global feasible solution is returned.

4 Numerical Experiments

In order to test our approach, we used real data provided by an airline company. Also, the following realistic constraints were considered during the process of legs merging: *(a)* the destination airport of a leg must be the initial airport in the next linking leg –i.e. the aircraft and the crew must be in the airport where next leg begins to be able to continue with the assigned pairing; *(b)* there must be a minimum of one hour between two linked legs in order to allow the aircraft to complete the turnaround process; *(c)* the maximum number of legs allowed for each pairing is five; and *(d)* the maximum number of flight hours in a day made by a single crew is eight.

Notice that in our tests we have just considered domestic flights and daily pairings. International (long-distance) flights have a different set of values for the model constraints. Table 1 shows the actual solution employed by airline as well as two alternative solutions obtained with our approach. The columns in the table contain the following outputs: *(i)* solution; *(ii)* number of legs in the solution; *(iii)* real flight hours (RFH); *(iv)* obtained flight hours (OFH); *(v)* flight hours without working (FHW); *(vi)* number of pairings (NOP); *(vii)* mean value of legs per pairing (MVL); *(viii)* number of pairings with maximum number of legs (NPM); *(ix)* mean value of flight hours per pairing (MVFH); and *(x)* flight hours average (FHA).

Table 1. Font sizes of headings. Table captions should always be positioned *above* the tables.

Solutions	N Legs	RFH	OFH	FHW	NOP	MVL	NPM	MVFH	FHA
Airline	44	69.61	63.07	3.29	16	3	1	4.33	4.35
Our sol 1	44	69.61	69.61	__	16	3	1	4.33	4.35
Our sol 2	44	69.61	69.61	__	15	3	4	5.41	4.64

From the numerical results, the following conclusions can be concluded: *(i)* the three solutions are quite similar in all their parameters; (*ii*) while the airline needed several hours to generate its solution –including human fine-tuning–, our heuristic was able to generate it in just a few seconds; (*iii*) one pairing is saved in only 44 treated legs,

saving a complete crew as well; (*iv*) thanks to rises obtained in air routes per pairing, better balanced pairings are reached.

5 Conclusions

We have proposed a fast heuristic for providing solutions to the crew pairing problem considering real-life constraints. Currently, most airlines are still generating their solutions using either expensive software or slow and tedious processes –some of them implying manual steps. Our simple heuristic approach has shown to be able to provide equivalent solutions to those employed in some airlines with low computing times, even when considering a large number of daily flights.

Although our heuristic is able to solve the main problem that is related to crew route generation and balancing, it still needs further improvements in order to generate near-optimal solutions. Among other research lines for future work, we are considering the use of biased randomization techniques which can contribute to significantly enhance our approach.

Acknowledgments

This work has been partially supported by the Spanish Ministry of Science and Innovation (TRA2010-21644-C03) and by the Ibero-American Programme for Science, Technology and Development (CYTED2010-511RT0419) in the context of the ICSO-HAROSA Programme of the IN3 (http://dpcs.uoc.edu).

References

1. AEA (Association of European Airlines), http://www.aea.be.
2. Deng, G., Lin, W.: Ant Colony optimization-based algorithm for air-line crew scheduling problem. Expert Systems with Applications. 38(5), 5787--5793 (2011)
3. Yan, S., Chang, J.: Airline cockpit crew scheduling. European Journal of Operational Research, 136(3), 501--511 (2002)
4. Gopalakrishnan, B., Johnson, E.J.: Airline Crew Scheduling: state-of-the-Art. Annals of Operations Research 140, 305--337 (2005)
5. Yan. S., Chang, J.: Airline cockpit crew scheduling, European Journal of Operational Research, 136(3), 501--511 (2002)
6. Schaefer, A.J., Johnson, E.L., Kleywegt, A.J., Nemhauser, G.L.: Airline crew scheduling under uncertainty. Transportation Science, 39(3), 340--348 (2005)
7. Mercier, A., Cordeau, J.F., Soumis, F.: A computational study of Benders decomposition for the integrated aircraft routing and crew scheduling problem. Computers & Operations Research, 32(6), 1451-1476 (2005)
8. Medard, C.P., Sawhney, N.: Airline Crew Scheduling from planning to operations. European Journal of Operational Research, 38(3), 1013-1027 (2007)
9. Yu, G., Arguello, M., Song, G., McCowan, S., White, A.: A new era for crew recovery at Continental Airlines. Interfaces, 33(1), 5-22 (2003)

10. Wei, G., Yu, G., Song, M.: Optimization model and algorithm for crew management during irregular operations. Journal of Combinatorial Optimization, 1, 80-97 (1997)
11. Medard, C.P., Sawhney, N.: Airline Crew Scheduling from planning to operations. European Journal of Operational Research, 38(3), 1013-1027 (2007)
12. Weide, O., Ryan, D., Ehrgott, M.: An iterative approach to robust and integrated aircraft routing and crew scheduling. Computers & Operations Research, 37(5), 833-844 (2010)
13. Yen, JW, Birge, JR: A Stochastic programming approach to the airline crew scheduling problem. Transportation Science, 40(1), 3-14 (2006)
14. Levine, D.: Application of a hybrid genetic algorithm to airline crew scheduling. Computers & Operations Research, 23(6), 547-558 (1996)

Biclustering Evolutivo mediante Identificación y Explotación de Regiones Prometedoras

Fco. Javier Gil-Cumbreras, Raúl Giráldez, Federico Divina y Jesús S. Aguilar-Ruiz

Escuela Politécnica Superior, Universidad Pablo de Olavide
Ctra. Utrera s/n, 41013, Sevilla
{fjgilcum, giraldez, fdiv, aguilar}@upo.es

Resumen Las técnicas de biclustering representan una de las alternativas más utilizadas en el análisis automático de microarrays. El biclustering es un tipo de problema que se caracteriza por tener cierto nivel de incertidumbre inherente. En este trabajo se presenta un nuevo tipo de codificación para algoritmos evolutivos, denominada *codificación probabilística*, la cual extiende la codificación binaria tradicional sustituyendo los bits por números reales que representan la probabilidad de que un determinado rasgo esté presente en la solución. De esta forma, un individuo probabilístico no representa una única solución, sino una región de soluciones, lo que permite modelar la incertidumbre anteriormente mencionada, además de incrementar el poder de exploración del algoritmo. La codificación probabilística ha sido incorporada en un algoritmo de biclustering evolutivo multiobjetivo y multimodal, cuyos resultados experimentales confirman la validez de nuestra propuesta.

Keywords: Codificacón Probabilística, Algoritmos Evolutivos, Biclustering.

1. Introducción

En el contexto del análisis de datos de expresión génica, la tecnología microarray permite estudiar los niveles de expresión de un elevado grupo de genes frente a diversas condiciones experimentales, generando una matriz numérica bidimensional en la que cada elemento se corresponde con el nivel de expresión de un determinado gen bajo una condición experimental específica. Estas matrices numéricas permiten el análisis computacional de dichos datos, lo cual facilita el diagnóstico y descubrimiento de nuevos subtipos de enfermedades [1].

La aplicación de técnicas de biclustering sobre microarrays ha experimentado un gran auge en los últimos años [6]. El biclustering es una variante de las técnicas de clustering en la que la búsqueda se realiza simultáneamente sobre las filas y columnas en la matriz. De esta manera, en el contexto del análisis de microarrays, es posible agrupar genes que muestren un comportamiento similar frente a subconjuntos de condiciones. A estos agrupamientos en dos dimensiones se les denomina biclusters.

Entre las numerosas técnicas utilizadas para descubrir biclusters destacan los Algoritmos Evolutivos (AE) [2,4,7,8]. Un aspecto crítico en el diseño de un AE

es definir la codificación de los individuos genéticos. Existen muchos tipos de codificación estándar estudiados en la literatura, los cuales han sido aplicados a diversos tipos de problemas. La codificación binaria es uno de los esquemas más simples y utilizados, en la que cada individuo se representa mediante una cadena de bits. La codificación binaria es adecuada para problemas cuyas soluciones pueden ser modeladas mediante un conjunto de variables booleanas, como es el caso del biclustering. Sin embargo, a menudo es necesario abordar problemas caracterizados por la incertidumbre. En tal caso, la codificación binaria puede no ser el mejor enfoque, ya que no permite representar dicha incertidumbre. En este sentido, el biclustering es un problema que se caracteriza por una cierta cantidad de incertidumbre implícita, la cual puede ser debida a diversas razones, por ejemplo, valores ausentes o ruido en la matriz de expresión. Este hecho ha supuesto la principal motivación para desarrollar un nuevo tipo de codificación, llamada *codificación probabilística*, y cuya aplicación al biclustering presentamos en este trabajo. Esta nueva representación extiende la codificación binaria tradicional permitiendo representar la incertidumbre que puede existir en ciertos problemas de optimización. Además, la codificación probabilística puede a su vez incrementar el poder de exploración de los AE. Asimismo, con el fin de probar la efectividad de la codificación probabilística en la tarea de biclustering, se presenta un nuevo método de biclustering, denominado IEPR (*Identification and Exploitation of Promising Regions*), el cual incorpora la codificación probabilística en un AE multiobjetivo y multimodal.

2. Medidas de Calidad de los Biclusters

La técnica más extendida para el análisis de los microarrays de expresión génica es el biclustering, que permite un análisis flexible de los microarrays y posibilita localizar agrupaciones, denominadas biclusters, que se definen en función de los genes que la forman y las condiciones donde dichos genes se co-expresan. En general, el objetivo del biclustering es encontrar biclusters con ciertas propiedades consideradas interesantes desde el punto de vista biológico.

Existen en la literatura muchas medidas de evaluación de biclusters que cuantifican la bondad de los mismos. Una de las medidas más utilizadas en la literatura es el residuo cuadrático medio (MSR *Mean Squared Residue*) [3], el cual cuantifica la coherencia numérica presentada por los genes y condiciones de un bicluster. Cuanto menor es el MSR, mayor es la coherencia entre los valores de expresión de los genes contenidos en el bicluster y, por tanto, esta medida debe ser minimizada. Un valor cero en el MSR indica que los genes del bicluster muestran un comportamiento idéntico frente a las condiciones de dicho bicluster. Debe tenerse en cuenta que este caso también incluye biclusters de tipo constantes o triviales, donde no hay fluctuación alguna de los genes frente a las distintas condiciones. Obviamente, este tipo de biclusters no tienen interés y, por tanto, deben evitarse. Para ello, se hace uso de otras medidas en combinación con el MSR, como son la varianza media de filas (MRV, *Mean Row Variance*) [3] y el volumen (tamaño del bicluster). Es el caso del algoritmo IERP, cuyo objetivo es obtener biclusters con mínimo MSR, máxima MRV y máximo volumen.

3. Codificación Probabilística

En algunos problemas de optimización, cada solución candidata puede ser modelada como una evaluación particular de un conjunto simple de variables booleanas (BDV, *Boolean Decision Variables*). Formalmente, un problema de optimización Φ, definido en un espacio de búsqueda S es un problema de optimización BDV, si y sólo: $\exists\ V = \{v_1, ..., v_n\} \mid \forall s \in S,\ s \equiv \{v_1 = b_1^s, ..., v_i = b_i^s, ...v_n = b_n^s\}$; donde s es solución candidata en S, v_i $(1 \leq i < n)$ son variables booleanas y $b_i^s \in \{false, true\}$. Esta definición implica que, en un problema BDV, la presencia de un rasgo o atributo particular se indica mediante la variable v_i. Una solución se determina mediante la selección de un conjunto de rasgos presentes (para los que $v_i = true$) y otros ausentes (para los que $v_i = false$). Los problemas de optimización BDV son el contexto adecuado para aplicar la codificación binaria estándar como representación, ya que cada bit en un genotipo indicaría la presencia o ausencia del rasgo asociado en la solución representada. Sin embargo, existen problemas que se caracterizan por la presencia de cierto nivel de incertidumbre [10]. Esto normalmente implica que la función objetivo tenga limitada su capacidad para evaluar la calidad de las soluciones y, en consecuencia, que el algoritmo pueda no ser capaz de distinguir entre la calidad real de algunas soluciones.

La *codificación probabilística* es desarrollada para abordar el tipo de problemas anteriormente descritos, reemplazando los bits de la codificación binaria por números reales que representan la probabilidad de que un determinado rasgo esté presente en la solución. De esta forma, un individuo probabilístico no representará un solo punto del espacio de búsqueda, como ocurre con la codificación binaria, sino una región de dicho espacio. Por tanto, un individuo probabilístico puede representar varias soluciones candidatas diferentes. Intuitivamente, esto le dará más poder de exploración a un AE que adopte tal representación, pero también hará que la evaluación del individuo sea más difícil. Formalmente, un genotipo probabilístico puede ser visto como una función de probabilidad de rasgos p_{traits} que asocia un valor de probabilidad a cada variable de decisión v_i del problema. Así, la probabilidad $P(s)$ de cada solución $s \equiv \{v_1 = b_1^s, ..., v_i = b_i^s, ...v_n = b_n^s\}$ del espacio de búsqueda puede ser calculada en base a las probabilidades establecidas por un individuo para cada uno de los rasgos (p_{traits}) de la siguiente forma:

$$P(s) = P(v_1 = b_1^s) \times ... \times P(v_i = b_i^s) \times ... \times P(v_n = b_n^s)$$
$$= \left[\prod_{i|b_i^s = true} p_{traits}(v_i) \right] \times \left[\prod_{j|b_j^s = false} (1 - p_{traits}(v_j)) \right]$$

Un AE que adopte esta codificación debe evolucionar la población hacia individuos en los que la probabilidad asociada a un rasgo mide su frecuencia de aparición en la región óptima que intenta descubrirse. Si suponemos que las soluciones pertenecientes a esta región óptima tienen un conjunto importante de características comunes, es decir, comparten y evitan un mismo conjunto

importante de características, estos rasgos deben ver como sus probabilidades convergen a 1 y a 0, respectivamente. Para los rasgos restantes, cuyo interés no puede ser evaluado de manera cierta, el algoritmo da una tasa de interés estimada, que es la relación de soluciones óptimas que exhiben esos rasgos. Como ejemplo, supongamos que el problema es encontrar la combinación óptima de tres objetos A, B y C. Si se utiliza una representación binaria, un ejemplo de individuo podría ser "1 1 0", que representa la solución $\{A, B\}$. En cambio, si se utiliza la codificación probabilística, un ejemplo de individuo podría ser "0,95 0,52 0,04". Si un AE converge a una población compuesta de genotipos similares a la que se presenta aquí, eso indica que la solución óptima debe contener el punto A ($P(A = true) = 0{,}95$), pero no el punto C ($P(C = true) = 0{,}04$), mientras que hay un alto nivel de incertidumbre acerca de la presencia de B en la solución ($P(B = true) = 0{,}52$). La región óptima individualizada por el algoritmo se compone de dos soluciones $\{A\}$ y $\{A, B\}$, ya que tienen una probabilidad mucho mayor de ser representadas por el genotipo.

Independientemente de la conveniencia de su uso en problemas con incertidumbre, la codificación probabilística también puede ofrecer otra ventaja sobre la codificación binaria: una mejor capacidad de exploración del espacio de búsqueda. Como cada individuo probabilístico puede representar una región de potenciales soluciones puntuales, el algoritmo puede explorar en cada generación una parte más amplia del espacio de búsqueda. Al final del proceso evolutivo, el AE convergerá hacia lo que denominamos una *región prometedora*, es decir, una región donde es probable que se encuentre una solución óptima. Esto hace que sea necesario aplicar alguna estrategia de búsqueda posterior que explore dicha región en busca de una única solución óptima, si la hubiere, como se detalla en la Sección 4.

4. Algoritmo IEPR

En esta sección se presenta el algoritmo IEPR (*Identification and Exploitation of Promising Regions*), una herramienta de biclustering evolutivo que genera un conjunto de biclusters (por defecto 100) en cada ejecución. Para ello se aplica un AE multiobjetivo y multimodal, en el que los biclusters son representados mediante la codificación probabilística, con una probabilidad asociada a cada uno de los genes y de las condiciones en el conjunto de datos analizados. Los principales componentes algorítmicos que IEPR incorpora para adaptar la codificación probabilística al problema del biclustering son: los operadores genéticos, el mecanismo de evaluación y el algoritmo de búsqueda local que explorará las regiones prometedoras.

El mecanismo de inicialización de la población selecciona, para cada individuo, una proporción de genes y condiciones al azar, para asignarles valores altos de probabilidad, mientras que los restantes reciben un valor bajo de probabilidad. Dicha proporción de genes y condiciones, así como las probabilidades asignadas a cada gen o condición seleccionada, son obtenidas al azar entre unos umbrales mínimo y máximo configurables, lo que permite controlar el tamaño de las regiones creadas, así como el tamaño de las sub-matrices en esas regiones.

Por otro lado, el fitness de un individuo se basa en el concepto de dominación de Pareto, teniendo tres objetivos a optimizar: MSR, MRV y Volumen de una región, descritos en la Sección 2. La selección se aplica utilizando el mecanismo de selección *Niched Pareto* propuesto en [5]. Este operador selecciona al azar dos individuos candidatos de la generación actual, y devuelve uno de ellos mediante un torneo en dos etapas. Durante la primera etapa del torneo, se evalúa la calidad de los dos individuos usando las relaciones de dominación en la generación actual. Si la primera etapa del torneo conduce a un empate, se utiliza la estrategia *Continuously Updated Sharing* para calcular el *niche count* de cada uno de los dos individuos en la próxima generación. A continuación, se selecciona el individuo que presenta el *niche count* más bajo. A diferencia de [5], el *sharing* se aplica directamente en el espacio de búsqueda. Esto se debe a que nuestro objetivo no es mantener la diversidad a lo largo del frente óptimo de Pareto, sino localizar diferentes biclusters en una única ejecución. La función de distancia d entre dos genotipos (que codifican los rasgos de dos funciones de probabilidad p^1_{traits} y p^2_{traits}) se define como una medida estimada de la tasa de solapamiento entre los biclusters típicos que cada genotipo representa:

$$d(p^1_{traits}, p^2_{traits}) = 1 - overlap_{genes} \times overlap_{conditions}$$

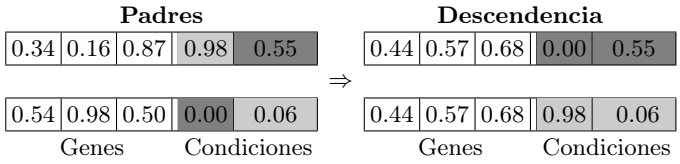
donde

$$overlap_{genes} = \frac{|\{i \in I|\ (p^1_{traits}(i) \geq 0{,}5 \wedge p^2_{traits}(i) \geq 0{,}5)\}|}{|\{i \in I|\ (p^1_{traits}(i) \geq 0{,}5 \vee p^2_{traits}(i) \geq 0{,}5)\}|}$$

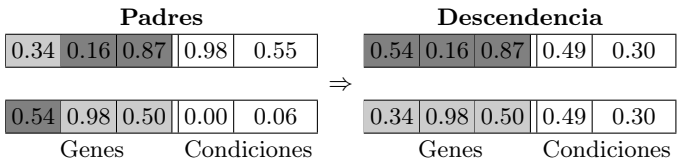
$$overlap_{conditions} = \frac{|\{j \in J|\ (p^1_{traits}(j) \geq 0{,}5 \wedge p^2_{traits}(j) \geq 0{,}5)\}|}{|\{j \in J|\ (p^1_{traits}(j) \geq 0{,}5 \vee p^2_{traits}(j) \geq 0{,}5)\}|}$$

Respecto a los operadores genéticos, se aplican tres operadores de cruce y dos de mutación: cruce uniforme clásico, cruce promedio de genes, cruce promedio de condiciones, mutación de genes y mutación de condiciones. El cruce promedio de genes asigna a ambos descendientes la media de las probabilidades asignadas a los genes de los padres, mientras que para las condiciones aplica un cruce uniforme clásico. El cruce promedio de condiciones es similar pero asignando las probabilidades medias a las condiciones y aplicando el cruces uniforme a los genes. La Figura 4 ilustra esta idea. La mutación tanto de genes como de condiciones sustituye un valor de probabilidad de un gen o condición seleccionado al azar, por un número aleatorio en $[0, 1]$.

Una vez el AE concluye, devuelve una población de regiones prometedoras de biclusters. Ahora es necesario buscar biclusters con un MSR bajo en cada una de las regiones localizadas. Para ello, IEPR aplica Simulated Annealing (SA) [9] para realizar la búsquea local. Nuestra implementación de SA comienza a procesar el bicluster que contiene todos los genes y las condiciones con una probabilidad $\geq 0{,}5$ en el genotipo que codifica la región explorada. A continuación, intenta mejorar el MSR añadiendo o eliminando repetidamente un gen o una condición al bicluster, de acuerdo con el esquema de SA. Los genes y las condiciones con



(a) Cruce promedio de genes.



(b) Cruce promedio de condiciones.

Figura 1. Operadores de Cruce.

una probabilidad en el genotipo mayor que un umbral no pueden ser eliminados del bicluster durante la búsqueda. Del mismo modo, los genes y las condiciones con una probabilidad en el genotipo inferior a un umbral no se pueden añadir al bicluster durante la búsqueda.

5. Experimentos

Con el fin de validar nuestra propuesta, hemos probado IEPR en ocho conjuntos de datos. La lista de los conjuntos de datos, junto con sus características, se muestran en la Tabla 1.

Tabla 1. Conjuntos de Datos [4].

Dataset	Nombre	#Genes	#Condiciones
Yeast	Yeast *Saccharomyces cerevisiae* cell cycle	2884	17
Human	Human B–cells	4026	96
Colon	Colon Cancer	2000	62
Malaria	Malaria *Plasmodium parasites* life cycle	3719	16
Embryonal	Embryonal tumors of the central nervous syst.	7129	60
Leukemia	Leukemia	7129	72
RatCNS	Rat Central Nervous System	112	9
Steminal	Steminal Cells	26127	30

La Tabla 2 muestra la parametrización utilizada tanto para el AE adoptado en la primera fase de IEPR, como para la búsqueda SA aplicada la fase de explotación. Estos parámetros se determinaron después de varios ensayos preliminares.

Se han comparado los resultados obtenidos por IEPR con los obtenidos por otro AE multiobjetivo aplicado en biclustering de microarrays denominado SMOB [4], el cual optimiza los mismo objetivos que IEPR. SMOB adopta una codificación binaria clásica para representar biclusters, donde un bit se asocia a cada gen y condición. Por tanto, en el caso de SMOB, un individuo representa un

Tabla 2. Parámetros.

AE		SA	
Parameter	Value	Parameter	Value
Generaciones	1000	P_M	0.95
Tamaño de Población size	600	P_m	0.15
Min-max % genes (en inicialización)	0.05–4	Temp. Inicial Filas	2.0
Min-max % conditiones (en inicialización)	90-100	Temp. Inicial Columnas	30
P_M (umbral de probabilidad máximo)	0.7	Temp. Final Filas	0.0375
P_m (umbral de probabilidad mínimo)	0.3	Temp. Final Columnas	0.562
NI (% de individuos nuevos en reinicialización)	5	Ratio de decremento Temp.	0.90
Tamaño de la Muestra	50		
Tamaño del Conjunto Comparación	510		
Radio del Nicho	0.85		
Factor de Escala	1		
Probabilidad de Cruce	0.85		
Probabilidad de Mutación	0.10		

Tabla 3. Valores de δ/Δ usados en SMOB-δ y SMOB-Δ.

Dataset	δ en SMOB-δ	δ en SMOB-Δ
Yeast	300	3400
Human	1200	23000
Colon	500	3300
Malaria	600	19000
Embryonal	1800	10000
Leukemia	1800	23130700
RatCNS	5	11
Steminal	10	130

bicluster específico. Por otra parte, SMOB utiliza una estrategia secuencial para cubrir distintas áreas del microarray. El algoritmo evolutivo se ejecuta varias veces (100 en este caso), y cada vez se extrae un único bicluster. Una característica importante de SMOB, es el uso de un umbral δ, con el fin de limitar el MSR aceptable de un bicluster de igual manera que en [3]. Puesto que IEPR no utiliza ningún umbral para limitar el MSR, también hemos incluido los resultados obtenidos por una versión de SMOB en la que hemos relajado dicho umbral. Por lo tanto, hemos utilizado dos configuraciones de SMOB: SMOB-δ, que utiliza un umbral estricto; y SMOB-Δ, que utiliza un umbral relajado. La Tabla 3 presenta los umbrales utilizados por las dos configuraciones de SMOB en cada conjunto de datos.

Las Tablas 4, 5 y 6 muestran, respectivamente, el promedio de MSR, de la varianza media de fila y del volumen de cien biclusters obtenidos por IEPR, SMOB-δ y SMOB-Δ, en los ocho conjuntos de datos. Un símbolo menos/más $(-/+)$ junto a un resultado indica que la diferencia es significativamente menor/mayor que el resultado obtenido por IEPR, de acuerdo con un test estadístico t-test con un nivel de confianza de un 1 %.

Como se puede observar en la tabla 4, IEPR obtuvo un MSR significativamente menor en cinco conjuntos de datos con respecto a SMOB-δ. Los dos algoritmos obtuvieron resultados similares en el conjunto de datos Steminal, y sólo en Colon y Leukemia SMOB-δ obtuvo mejores resultados. Este resultado se explica por el hecho de que el umbral adoptado por SMOB-δ en estos dos conjuntos de datos limita el MSR de los biclusters descubiertos hasta 500, en el caso de Colon, y hasta 1800, en el caso de Leukemia. Esto significa que cualquier

Tabla 4. MSR medio.

Dataset	MSR				
	IEPR	SMOB-δ		SMOB-Δ	
Yeast	$107.0_{\pm 12,2}$	$272.0_{\pm 25,0}$	+	$1062.1_{\pm 274,8}$	+
Human	$295.0_{\pm 218,7}$	$1103.6_{\pm 86,7}$	+	$10742.6_{\pm 2321,3}$	+
Colon	$916.3_{\pm 125,0}$	$455.3_{\pm 45,5}$	−	$2107.2_{\pm 306,3}$	+
Malaria	$311.5_{\pm 267,3}$	$449.7_{\pm 174,5}$	+	$12095.5_{\pm 2298,1}$	+
Embryonal	$9.1_{\pm 0,5}$	$398.2_{\pm 344,2}$	+	$803.0_{\pm 1243,1}$	+
Leukemia	$23538.4_{\pm 9498,6}$	$1533.2_{\pm 169,2}$	−	$4929089.4_{\pm 2938946,0}$	+
CnsRat	$0.6_{\pm 0,2}$	$1.6_{\pm 1,3}$	+	$2.2_{\pm 2,0}$	+
Steminal	$4.1_{\pm 0,5}$	$4.1_{\pm 2,1}$		$13.1_{\pm 12,9}$	+

Tabla 5. Varianza Media de Fila (MRV).

Dataset	MRV				
	IEPR	SMOB-δ		SMOB-Δ	
Yeast	$527.6_{\pm 99,2}$	$408.6_{\pm 77,2}$	−	$1245.2_{\pm 294,0}$	+
Human	$982.6_{\pm 354,1}$	$1412.1_{\pm 168,6}$	+	$11412.7_{\pm 2476,3}$	+
Colon	$3707.4_{\pm 345,6}$	$2836.8_{\pm 1026,7}$	−	$4876.2_{\pm 548,4}$	+
Malaria	$4454.6_{\pm 4562,0}$	$2667.4_{\pm 3765,7}$	−	$16707.7_{\pm 2914,7}$	+
Embryonal	$9.9_{\pm 0,5}$	$428.8_{\pm 366,0}$	+	$849.4_{\pm 1304,1}$	+
Leukemia	$23959.9_{\pm 9640,9}$	$2391.4_{\pm 660,1}$	−	$5269867.7_{\pm 3099017,9}$	+
CnsRat	$0.7_{\pm 0,3}$	$2.1_{\pm 1,6}$	+	$2.7_{\pm 2,3}$	+
Steminal	$4.1_{\pm 0,5}$	$4.5_{\pm 2,3}$		$13.8_{\pm 13,3}$	+

bicluster con un MSR mayor que el especificado, quedaría descartado por SMOB. IEPR no limita el MSR de los biclusters descubiertos en modo alguno. Esta es una ventaja importante de IEPR, ya que el uso de un valor erróneo de δ puede provocar que el algoritmo no localice biclusters interesantes, además de que no existe un procedimiento estándar para ajustar δ. También podemos notar que si el umbral se relaja, el MSR promedio de los biclusters descubiertos por SMOB es significativamente mayor que los obtenidos por IEPR en todos los conjuntos de datos.

Respecto a MRV, podemos observar en la Tabla 5, que en cuatro conjuntos de datos (Yeast, Colon, Malaria y Leukemia) IEPR obtuvo mejores resultados que SMOB-δ. Por otra parte, en Human, Embryonal y CnsRat, SMOB-δ obtuvo resultados significativamente mejores. Sin embargo, si analizamos estos tres casos, podemos observar que el MSR promedio obtenido por SMOB-δ en estos tres conjuntos de datos fue significativamente mayor que el obtenido por IEPR. En general, a valores más altos de MSR corresponden valores más altos de la varianza media de fila [3], ya que MSR y la varianza media de fila son objetivos que normalmente entran en conflicto. También podemos observar que el MSR medio obtenido por IEPR en los conjunto de datos Yeast y Malaria es menor que el obtenido por SMOB-δ en los mismos conjuntos de datos. Si el umbral utilizado por SMOB se relaja, entonces MRV es significativamente mayor que MRV obtenida por IEPR en todos los conjuntos de datos. Sin embargo, estos resultados se justifican por el hecho de que en todos los casos el MSR obtenido por SMOB-Δ es significativamente mayor.

Finalmente, observando la Tabla 6, destaca que IEPR siempre localiza biclusters caracterizados por un volumen significativamente mayor que los biclusters

Tabla 6. Volumen Medio.

Dataset	Volume		
	IEPR	SMOB-δ	SMOB-Δ
Yeast	$759.0_{\pm 166,4}$	$226.1_{\pm 77,3}$ −	$461.5_{\pm 95,0}$ −
Human	$363.8_{\pm 668,0}$	$362.2_{\pm 106,8}$ −	$1128.3_{\pm 260,4}$ +
Colon	$4790.4_{\pm 1519,8}$	$197.6_{\pm 85,1}$ −	$1172.5_{\pm 334,5}$ −
Malaria	$452.2_{\pm 674,8}$	$54.7_{\pm 31,5}$ −	$462.0_{\pm 108,2}$ +
Embryonal	$230266.7_{\pm 14973,8}$	$1456.2_{\pm 577,3}$ −	$1423.6_{\pm 519,3}$ −
Leukemia	$139609.4_{\pm 32628,0}$	$441.7_{\pm 141,2}$ −	$1132.2_{\pm 220,4}$ −
CnsRat	$353.2_{\pm 65,9}$	$128.5_{\pm 81,0}$ −	$128.6_{\pm 69,0}$ −
Steminal	$291095.0_{\pm 57256,9}$	$1312.5_{\pm 371,8}$ −	$1286.7_{\pm 399,4}$ −

encontrados por SMOB-δ. Este resultado es aún más significativo, si consideramos que en cinco casos, el MSR medio obtenido por IEPR es significativamente inferior a la media de MSR obtenido por SMOB-δ. El volumen puede entrar en conflicto con el MSR. Sin embargo, IEPR es capaz de encontrar biclusters que presentan un MSR bajo y un volumen alto. Incluso cuando se relaja δ, SMOB sólo es capaz de descubrir biclusters que presentan un mayor volumen en dos casos. Como en el caso de la varianza, debemos señalar también para el volumen, que el MSR promedio de los biclusters localizados por SMOB-Δ es mayor que el obtenido por IEPR en todos los casos. Esta observación explica por qué SMOB-Δ localiza en dos casos biclusters con volúmenes más altos.

Al tener en cuenta los tres objetivos en su conjunto, podemos concluir que los resultados obtenidos por IEPR, en los ocho conjuntos de datos, son muy satisfactorios. En general, los biclusters descubiertos por IEPR se caracterizan por un MSR inferior y por una varianza media de fila y un volumen superior que los biclusters descubiertos por SMOB-δ. Si relajamos el umbral, podemos realizar una comparación más equitativa de los dos algoritmos, puesto que IEPR no utiliza ningún umbral para limitar el MSR de los biclusters. En ese caso, IEPR presenta prestaciones superiores con respecto a SMOB. De hecho, SMOB-Δ obtiene mejores resultados sólo en lo que se refiere a la varianza media de fila. Sin embargo, los otros dos objetivos son significativamente peores que los obtenidos por IEPR.

6. Conclusiones

En este trabajo se presenta una nueva codificación para AE, llamada codificación probabilística, y su aplicación al problema del biclustering sobre microarrays mediante la implementación del algoritmo IEPR. La codificación probabilística sustituye los bits de la codificación binaria tradicional por probabilidad. Esta nueva representación permite tratar con la incertidumbre inherente a ciertos problemas de optimización, como es el caso del biclustering. Asimismo, el uso de esta codificación en lugar de la binaria proporciona a los algoritmos que la utilizan una alta potencia de exploración del espacio de búsqueda, puesto que cada individuo representa una región de dicho espacio, en vez de una única solución.

IEPR opera en dos fases: en una primera fase, el algoritmo aplica un AE multiobjetivo y multimodal para obtener las regiones prometedoras; en una segunda

fase, las regiones son explotadas por medio de un procedimiento de búsqueda local (Simulated Annealing). Los resultados obtenidos por nuestra propuesta se han comparado con los obtenidos por dos configuraciones de SMOB, un AE multiobjetivo que utiliza una representación binaria clásica. El análisis experimental de los ocho conjuntos de datos de expresión génica muestra que IEPR proporciona excelentes resultados en comparación con SMOB, especialmente, explorando el espacio de búsqueda cuando se están estudiando varios objetivos. La codificación probabilística revela un alto potencial para su uso en la resolución de problemas complejos con técnicas evolutivas.

Como trabajo futuro, la codificación probabilística abre una nueva línea de investigación para desarrollar mejor los fundamentos de la codificación, sus operadores, así como sus múltiples aplicaciones en otros problemas de optimización y aprendizaje automático.

Agradecimientos

Este trabajo ha sido subvencionado por el Ministerio de Economía y Competitividad con el proyecto TIN2011-28956-C02.

Referencias

1. D. P. Berrar, W. Dubitzky, and M. Granzow. *A Practical Approach to Microarray Data Analysis*. Springer Publishing Company, Incorporated, 2003.
2. S. Bleuler, A. Prelić, and E. Zitzler. An EA Framework for Biclustering of Gene Expression Data. In *Congress on Evolutionary Computation*, pages 166–173, Piscataway, NJ, 2004. IEEE.
3. Y. Cheng and G. M. Church. Biclustering of expression data. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, pages 93–103. AAAI Press, 2000.
4. Divina, F., Aguilar-Ruiz, J.S.: A multi-objective approach to discover biclusters in microarray data. In: Proceedings of the 16th Genetic and Evolutionary Computation Conference (GECCO-2007), pp. 385–392. ACM (2007)
5. J. Horn, N. Nafpliotis, and D. E. Goldberg. A niched pareto genetic algorithm for multiobjective optimization. In *IEEE World Congress on Computational Intelligence*, pages 82–87. IEEE, 1994.
6. S.C. Madeira and A.L. Oliveira. Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM Trans. Comp. Biol. Bioinformatics*, 1(1):24–45, 2004.
7. S. Mitra and H. Banka. Multi-objective evolutionary biclustering of gene expression data. *Pattern Recogn.*, 39(12):2464–2477, 2006.
8. B. Pontes, R. Giráldez, and J.S. Aguilar-Ruiz. Configurable pattern-based evolutionary biclustering of gene expression data. *Algorithms for Molecular Biology*, 8(1):4, 2013.
9. B. Suman and P. Kumar. A survey of simulated annealing as a tool for single and multiobjective optimization. In *Journal of the Operational Research Society* (2006):57, pages 1143–1160, 2006.
10. S. Yang, Y.S. Ong, and Y. Jin. Evolutionary Computation in Dynamic and Uncertain Environments, *Studies in Computational Intelligence*, (51). Springer, 2007.

Optimización basada en Malla Variable para funciones multimodales

Daniel Molina[1], Amilkar Puris[2], Rafael Bello[2], and Francisco Herrera[3]

[1] Dpto. de Ciencias de la Computación e Ingeniería Universidad de Cadiz, España
daniel.molina@uca.es
[2] Dpto. de Ciencias de la Computación Universidad de Las Villas, Cuba
apuris@frominter.net; rbello@uclv.edu.cu
[3] Dpto. de Ciencias de la Computación e Inteligencia Artificial Universidad de Granada, España herrera@decsai.ugr.es

Resumen Los algoritmos de optimización basados en nichos han sido propuestos para resolver problemas multimodales. En este trabajo presentamos una variante del algoritmo VMO que, como el original, utiliza una malla de soluciones que se expande en el espacio de búsqueda creando nuevas soluciones. Posteriormente, un operador de limpieza (*clearing*) elimina las soluciones cercanas a otra de mejor calidad, utilizando para ello un umbral de distancia dinámico. La nueva propuesta introduce la exploración en los entornos de los óptimos globales encontrados. Incorpora también un proceso de búsqueda local en zonas prometedoras (soluciones distantes con buena calidad). Además, combina lo anterior con una memoria externa que almacena los óptimos encontrados para evitar una posible pérdida durante la búsqueda. Por último es evaluada y comparada usando un conjunto de funciones multimodales presentadas en el CEC'2013. La propuesta obtiene muy buenos resultados en comparación con otros algoritmos, mostrándose como un algoritmo de nichos muy competitivo para problemas multimodales.

Keywords: Algoritmos de Nichos, Algoritmos Evolutivos, Optimización continua.

1. Introducción

Las Meta-heurísticas (MHs) [3] basadas en población son muy apropiadas para resolver problemas multimodales[4] . Estos algoritmos utilizan una población (conjunto de soluciones) para explorar el espacio de búsqueda. De esta forma, pueden concentrar subpoblaciones en los entornos de las soluciones de buena calidad, no utilizando únicamente la información de la mejor solución.

Hay MHs especialmente orientadas a buscar los distintos óptimos posibles, y no únicamente hacia uno de ellos. Los algoritmos con esta forma de exploración son conocidos como *algoritmos basados en nichos*. Los Algoritmos Genéticos [4]

[4] problemas con múltiples óptimos, donde el objetivo es encontrarlos todos

fueron de las primeras MHs para nichos, pero han ido apareciendo versiones para los algoritmos de optimización basada en Enjambre de Partículas (PSO) [1] o la Evolución Diferencial [8].

En un trabajo anterior propusimos la Optimización basada en Malla Variable (VMO) [7] como una MH eficiente para problemas de optimización continua. VMO utiliza una población conocida como malla para explorar el espacio de búsqueda. Esta malla se expande (incorporando nuevas soluciones) utilizando distintas formas de generación de soluciones. Finalmente, la malla es reducida utilizando un operador de limpieza adaptativo (*clearing*) que elimina todo nodo o solución que se encuentre cerca de otro de mejor calidad (*fitness*). Este operador de *clearing* permite mantener las soluciones más representativas de las zonas exploradas. Esta característica hace que el VMO pueda ser utilizado en la búsqueda de problemas multimodales.

En este trabajo presentamos una adaptación de VMO orientada a explorar espacios multimodales. Difiere del modelo original en tres aspectos fundamentales. Primero, utiliza una forma de generación de nuevas soluciones que involucra no sólo a la mejor solución encontrada. si no también a aquellas con una calidad o *fitness* similar a la mejor encontrada. Además, se incorpora un método de búsqueda local (BL) con el objetivo de explorar detalladamente los entornos de las zonas más prometedoras y hacerlas más representativas. Por último, utiliza una memoria externa para almacenar las mejores soluciones encontradas durante la ejecución, permitiendo que el algoritmo pueda no tener que mantenerlas en la población durante toda la búsqueda, con las limitaciones que eso puede suponer en la búsqueda.

Para validar la propuesta presentamos un estudio comparativo con otros algoritmos modernos de nichos, utilizando como funciones de prueba las presentadas en Sección Especial *Métodos de Nichos para Problemas Multimodales* del Congreso de Computación Evolutiva (CEC2013) [5].

Este trabajo ha sido organizado de la siguiente manera: en la Sección 2, se describe el algoritmo VMO original enfatizando en las características que utiliza nuestra propuesta. En la Sección 3, describimos detalladamente los cambios introducidos a VMO para explorar espacios multimodales. Luego en la Sección 4, mostramos las características y resultados de los experimentos. Y por último, en la Sección 5 presentamos las principales conclusiones de este trabajo.

2. Optimización basada en Mallas Variables

VMO es una Meta-heurística basada en población en la que ésta se distribuye como una malla de nodos $(n_1, n_2, \ldots, n_P)$ que representan soluciones en el espacio de búsqueda. La exploración se lleva a cabo por dos procesos, de expansión y de contracción [7]. El proceso de expansión genera nuevos nodos utilizando la población en la iteración actual, por medio de los pasos siguientes:

Fase 1 (*Generación de nodos hacia los mejores vecino*). Para cada solución de la malla (n_i) se buscan los k nodos más cercanos[5]. Se selecciona el vecino de

[5] utilizaremos la distancia euclidiana en todo el trabajo

mejor calidad (*fitness*) (n_i^*). Si n_i^* es mejor que n_i se genera un nuevo nodo entre ambos.
Fase 2 (*Generación de nodos hacia la mejor solución*). Se genera un nuevo nodo para cada n_i en dirección al nodo con el mejor valor de *fitness* (n_g, óptimo global) de la malla actual.
Fase 3 (*Generación de nodos utilizando los nodos fronteras de la malla*). Los nodos fronteras se definen como las soluciones que se encuentran más cerca y más lejos del centro de la población. Los nuevos nodos se generan desplazando cada nodo frontera en función de un valor de desplazamiento (w, ver definición en [7]).
Fase de Clearing. En esta fase, se seleccionan las soluciones más representativas entre la malla actual y los nuevos nodos generados en el proceso de expansión. Para ello, evita seleccionar soluciones demasiado cercanas entre sí. Durante el *clearing* se define una distancia ξ_j como distancia mínima entre soluciones, y se ignora todo nodo que tenga a una distancia inferior un nodo con mejor o igual calidad (*fitness*). Dicha distancia se inicializa a un 25 % del rango de búsqueda, y va reduciéndose durante la ejecución del algoritmo (más información en [7]). Durante esta fase no se usa información sobre la distancia entre los óptimos, por lo que se podrían eliminar soluciones óptima.

3. Optimización basada en Malla Variables para problemas multimodales

El algoritmo VMO obtiene buenos resultados en problemas clásicos de optimización continua, pero no es totalmente aplicable a problemas multimodales por la siguientes razones: primero, la fase 2, orienta la búsqueda alrededor del la mejor solución encontrada. Esto supone que en el caso de más de un óptimo, el resto serán ignorados. Por último, VMO puede perder buenas soluciones encontradas por el operador de clearing.

En esta sección describimos en detalle los elementos incorporados a VMO para problemas multimodales. La Figura 1 muestra la descripción general de la propuesta.

3.1. Exploración hacia los óptimos globales

En la **Fase 2** de VMO se explora solamente en dirección a la solución de mejor calidad en la población (óptimo global). Para problemas multimodales es importante orientar también la búsqueda hacia las distintas soluciones óptimas (con calidad semejante). Para obtener este conjunto, consideramos que el *i-ésimo* nodo es un óptimo global si la diferencia (calidad) con la mejor solución actual no excede a un valor de ϵ ($\epsilon = 10^{-6}$). La siguiente ecuación muestra la forma de obtener el conjunto de óptimos:

$$lg = lg \cup n_i,\ si\ (Calidad(n_i) - Calidad(n_g) < \epsilon) \tag{1}$$

Inicial Generación aleatoria de la malla inicial M con distribución uniforme.
$M^{temp} \leftarrow M$.
Identificar la mejor solución $n_g \in M$.

Fase 1 Para cada $n_i \in M$, n_i no es óptimo local:
1.1. Obtener el vecino de mejor calidad n_i^*.
1.2. $n_i^{fase1} = combinar_1(n_i, n_i^*)$.
1.3. $M^{temp} \leftarrow M^{temp} \cup_i n_i^{face1}$.

Fase 2' Obtener lg (Ecuación1). Para cada $n_i \in M$:
2.1. $ng_i \leftarrow$ el más cercano de lg a n_i .
2.2. $n_i^{ph2} = combinar_2(n_i, ng_i)$ por Ecuación 2.
2.3. $M^{temp} \leftarrow M^{temp} \cup_i n_i^{fase2}$.

Fase 3 Expandir a través de las fronteras:
3.1. n_s con los nodos más externos.
3.2. $M^{temp} \leftarrow M^{temp} \cup_i (n_i^s + w_j \cdot sign(n_i^s))$.
3.3. n_u con los nodos más internos.
3.4. $M^{temp} \leftarrow M^{temp} \cup_i (n_i^u - w_j \cdot sign(n_i^u))$.

BL Aplicar la Búsqueda Local (Figura3).
4.1. Obtener el conjunto candidato $Cand$ desde soluciones nuevas, o que fueron mejoradas anteriormente por la BL .
4.2. $S_{BL} \leftarrow$ el N_{BL} como las mejores soluciones y más distantes del conjunto $Cand$.
4.3. Para cada solución $n_i \in S_{LS}$ aplicar la BL.

Memoria Actualizar la memoria (Figura4).
5.1. Obtener el conjunto lg a partir de M^{temp} usando Ecuación 1.
5.2. Para cada $n_i \in lg$.
a) $distancia_{min_i}$ = distancia mínima(n_i,n_j) $\forall n_j \in OptimaM$.
b) si $distancia_{min_i} \geq umbral$ entonces $OptimaM \leftarrow n_i$.

Clearing Eliminar los peores nodos de M^{temp}:
4.1. Ordenar M^{temp} por la calidad.
4.2. Para cada $n_i, n_k \in M^{temp}$ calcular la distancia ($Calidad(n_i)$ mejor que $Calidad(n_k)$).
4.3. Eliminar n_k si $distancia(n_i, n_k) \leq \xi_j$, donde ξ_j se define en [7].

Actualiza Malla *M* Crear la próxima M utilizando M_{temp}.
5.1. Si $|M| < |M^{temp}|$ $M \leftarrow$ los mejores nodos de M^{temp}.
5.2. Si $|M| > |M^{temp}|$ $M \leftarrow M^{temp} \cup$ generar soluciones aleatorias.
5.3. $M^{temp} \leftarrow M$.
5.4. Identificar la mejor solución actual $n_g \in M$.

Parada Ir a **Paso 1** si no cumple la condición de parada.

Figura 1. Descripción general de VMO para problemas multimodales

donde lg representa el conjunto de óptimos globales de la población actual incluyendo a n_g. Para explorar las regiones representadas en este conjunto, presentamos en la Figura 2 los paso a seguir, donde ξ_j es la distancia de *clearing*.

$$v_j^x = \begin{cases} n_j^i + U[0,1] * (n_j^i - n_j^{g_i}) \text{ si } n_j^i < n_j^{g_i} \\ n_j^i - U[0,1] * (n_j^{g_i} - n_j^i) \text{ otros casos} \end{cases} \quad (2)$$

2.1. Obtener el conjunto lg desde M por la Ecuación 1.
2.2. Para cada nodo $n_i \in M$:
  2.2.1 Seleccionar el óptimo más cercano (n_{g_i}) desde el conjunto lg usando la distancia euclidiana ($D_{euclidiana}$)
  2.2.2 si $D_{euclideana} > \xi_j$ crear nuevo nodo por la Ecuación 2 usando n_i y n_{g_i}

Figura 2. Nueva fase 2 de VMO

3.2. Aplicación de la Búsqueda Local

Hoy en día es muy común encontrar algoritmos de BL aplicados en etapas de MHs poblacionales para mejora la exploración local de estas [6]. Con este objetivo, proponemos aplicar el conocido algoritmo de BL Solis Wets [9] antes del operador de contracción. La mejora introducida por la BL depende de tres aspectos fundamentales: la selección de las soluciones a mejorar, el esfuerzo invertido y el tamaño de salto inicial. La Figura 3 describe este proceso.

Primero, se crea el conjunto de candidatos $Cand$ con el conjunto de N_{LScand} soluciones de mejor calidad que no han sido seleccionadas, o han sido mejoradas anteriormente por la BL. La idea es evitar perder evaluaciones de la función objetivo por aplicar la BL varias veces al mismo óptimo.

Luego, se crea el conjunto N_{LS} que inicialmente incluye la mejor solución n^g, y se va añadiendo, hasta completar el conjunto, en cada paso la solución más distante a las ya incluídas.

Por último, se le aplica la BL a cada solución de N_{LS} utilizando LS_{eval}/N_{LS} evaluaciones (LS_{eval} es un nuevo parámetro). Cuando aplicamos la BL a una solución el tamaño del salto inicial es el mínimo entre un 2 % del dominio y el valor de la distancias euclidiana a su vecino más cercano dividida por la dimensión. Esta idea es para adaptar el proceso de BL en función de la distancia entre soluciones y asegurarse que la BL se dedica efectivamente a la exploración local de la soluciones seleccionadas.

3.3. Memoria externa para óptimos globales

Como fue mencionado anteriormente, el modelo VMO puede perder soluciones óptimas porque la distancia ξ_j del operador de limpieza no conoce infor-

3.1. Obtener $Cand \subset M$ con las mejores soluciones.
3.2. Para cada n_i de $Cand$, chequear si fue seleccionado anteriormente
- Si no fue seleccionado, mantener en el conjunto.
- Si fue seleccionado y no mejoró o $mejora < 10^{-6}$ eliminar n_i de $Cand$.

3.3. $S_{BL} = \{n_g\}$.
3.4. Mientras $|S_{BL}| < N_{BL}$:
  3.4.1 Obtener $n_{bl} \in Cand$ que minimiza $\sum_{c=1}^{|S_{BL}|} distancia(n_{bl}, n_C)$, $\forall n_C \in S_{BL}$.
  3.4.2 $S_{BL} \leftarrow S_{BL} \cup n_{bl}$.
3.5. Para cada $n_s \in S_{BL}$:
  3.5.1 Obtener la *distancia* a su vecino más cercano.
  3.5.2 $I_{paso} = min(distancia, 0{,}02 \cdot (b-a))$, donde $[a, b]$ es el dominio de búsqueda.
  3.5.3 $n_s^* = BL(n_s)$ con tamaño de paso inicial I_{paso} durante BL_{eval}/N_{BL} evaluaciones.
  3.5.4 $M^{temp} \leftarrow M^{temp} \cup n_s^*$.

Figura 3. Proceso de selección de la BL

mación del problema (distancia entre óptimos). Por este motivo antes de aplicar el proceso de contracción, el nuevo algoritmo utiliza una memoria $OptimaM$ para almacenar soluciones óptimas encontradas. Para mantener un tamaño razonable de la memoria, esta se actualiza en función de la mejor solución encontrada, almacenando los nodos de lg cuya distancia euclídea al mejor, sea mayor que un valor umbral determinado. La Figura 4 muestra el proceso. Al finalizar el algoritmo, esta memoria contiene las mejores soluciones encontradas.

for all $n_i \in M^{temp}$ **do**
  $calidad_i \leftarrow calidad(n_i)$
  if $calidad_i$ es mejor que $calidad_g$ (calidad del n_g) **and** $abs(calidad_i - calidad_g) > 10^{-6}$ **then**
    $OptimaM \leftarrow \{n_i\}$ (borrar memoria anterior).
  else if $abs(calidad_i - calidad_j) \leq 10^{-6}$ **then**
    encontrar n_{cerca} tal que $\forall n_j \in OptimaM$ distancia$(n_{cerca}, n_i) \leq distancia(n_j, n_i)$.
    if $distancia(n_{cerca}, n_i) \geq valor_umbral$ **then**
      $OptimaM \leftarrow OptimaM \cup n_i$.
    end if
  end if
end for

Figura 4. Generación de la memoria óptima

4. Experimentos

En esta sección, hemos preparado un estudio experimental para probar el desempeño de nuestra propuesta de VMO en la solución de problemas multimodales (algoritmo al que notamos como NVMO). Las funciones de prueba utilizadas son las presentadas en la Competición de Método de Nichos en el CEC'2013 (puede consultarse una información más detallada en [5]). Estas 12 funciones tienen diferentes dimensiones y cantidad de óptimos, por lo que contamos con 20 combinaciones diferentes para el estudio.

Todos los experimentos, tanto de la propuesta como de los algoritmos de referencia, se han llevado a cabo siguiendo las especificaciones de las funciones de prueba. Los resultados se obtuvieron con 50 ejecuciones independientes para cada una de las 20 combinaciones de funciones. La condición de parada del algoritmo se define por una cantidad máxima de evaluaciones (*MaxEvals*): 5.0E+04 para $F_1 - F_5$, 2.0E+05 para $F_6 - F_{11}$ (en 2D), y 4.0E+05 para $F_6 - F_{12}$ (3D o superior).

4.1. Descripción de los Parámetros

La Tabla 1 muestra la descripción y los valores utilizado para los parámetros usados en la experimentación.

Tabla 1. Parámetros del algoritmo NVMO

Parámetro	**Descripción**	**Valor**
P	Tamaño de la malla inicial	10+cantidad de óptimos
k	Cantidad de nodos que definen la vecindad	3
C	Condición de Parada (número de evaluaciones)	MaxEvals
ϵ	Valor de umbral para detectar óptimos	10^{-6}
Y	Cantidad de nodos agenerar en la fase 3	2
$N_{LS_{cad}}$	Cantidad de soluciones candidatas para BL	12
N_{LS}	Cantidad de nodos seleccionados por iteración para BL	3
I_{step}	Evaluaciones consumida por cada BL	150
LS_{freq}	Cada cuantas evaluaciones se aplica la BL	150

4.2. Resultados

En este trabajo utilizamos como medida la media del porcentaje de óptimos encontrados durante las 50 ejecuciones, PR [2]. Dicho valor es medido para diferentes valores de exactitud o ϵ (10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, y 10^{-5}). Se considera que se ha alcanzado el óptimo con un valor ϵ de exactitud si $abs(fitness(sol) - fitness_optimo)$ es menor o igual que ϵ

La Tabla 2 muestra los resultados obtenidos por nuestra propuesta para PR. Podemos observar que NVMO obtiene los óptimos para $\epsilon = 10^{-1}$ en 15 de las 20 combinaciones. También mantiene un buen comportamiento para menores valores de exactitud. En el caso especifico de $\epsilon = 10^{-5}$ el modelo identifica la mitad de los óptimos en 12 funciones.

Tabla 2. Comparación entre NVMO y los algoritmos DEs utilizando la medida PR (DE = DE/nrand). Para cada función y dimensión se indica el número de óptimos

Exactitud	F1(1D/2)			F2(1D/5)			F3(1D/1)			F4(2D/4)			F5(2D/2)		
	NVMO	DE	CDE	NVMO	DE	CDE	NVMO	DE	CDE	NVMO	DE	CDE	NVMO	DE	CDE
1.E-01	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.E-02	**1.000**	**1.000**	0.710	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.E-03	**1.000**	**1.000**	0.090	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.E-04	**1.000**	**1.000**	0.020	1.000	1.000	1.000	1.000	1.000	1.000	**1.000**	**1.000**	0.995	1.000	1.000	1.000
1.E-05	**1.000**	**1.000**	0.000	1.000	1.000	1.000	1.000	1.000	1.000	**1.000**	**1.000**	0.420	1.000	1.000	1.000
Exactitud	F6(2D/18)			F7(2D/36)			F6(3D/81)			F7(3D/216)			F8(2D/12)		
	NVMO	DE	CDE	NVMO	DE	CDE	NVMO	DE	CDE	NVMO	DE	CDE	NVMO	DE	CDE
1.E-01	**1.000**	0.450	**1.000**	**1.000**	0.347	0.703	0.412	0.108	**0.847**	**1.000**	0.097	0.271	1.000	1.000	1.000
1.E-02	**1.000**	0.438	0.999	**1.000**	0.346	0.724	0.294	0.105	**0.835**	**0.683**	0.095	0.272	1.000	1.000	1.000
1.E-03	0.940	0.440	**0.972**	**0.945**	0.349	0.715	0.270	0.113	**0.716**	**0.399**	0.099	0.274	**1.000**	0.998	**1.000**
1.E-04	**0.670**	0.434	0.107	**0.901**	0.337	0.709	0.198	0.112	**0.290**	**0.275**	0.095	0.274	1.000	1.000	1.000
1.E-05	0.000	0.000	0.000	**0.806**	0.333	0.716	0.027	**0.113**	0.038	0.192	0.094	**0.270**	0.968	**1.000**	**1.000**
Exactitud	F9(2D/6)			F10(2D/8)			F11(2D/6)			F11(3D/6)			F12(3D/8)		
	NVMO	DE	CDE	NVMO	DE	CDE	NVMO	DE	CDE	NVMO	DE	CDE	NVMO	DE	CDE
1.E-01	**1.000**	0.683	0.937	0.848	**0.855**	0.380	**1.000**	0.667	0.837	**1.000**	0.667	0.683	**1.000**	0.522	0.730
1.E-02	0.667	0.673	**0.690**	0.745	**0.837**	0.055	0.667	0.667	**0.683**	0.667	0.667	0.667	**0.713**	0.535	0.690
1.E-03	0.667	**0.683**	0.667	0.725	**0.815**	0.007	0.667	0.667	0.667	0.667	0.667	0.667	**0.668**	0.507	0.627
1.E-04	0.667	**0.673**	0.667	0.713	**0.815**	0.007	0.667	0.667	0.667	0.667	0.667	0.667	**0.623**	0.502	0.490
1.E-05	0.667	**0.670**	0.667	0.565	**0.777**	0.002	0.663	**0.667**	**0.667**	0.637	**0.667**	**0.667**	0.390	**0.507**	0.375
Exactitud	F11(5D/6)			F12(5D/8)			F11(10D/6)			F12(10D/6)			F12(20D/8)		
	NVMO	DE	CDE	NVMO	DE	CDE	NVMO	DE	CDE	NVMO	DE	CDE	NVMO	DE	CDE
1.E-01	**1.000**	0.677	0.697	**1.000**	0.345	0.567	**0.987**	0.403	0.517	**0.340**	0.303	0.000	0.000	0.130	**0.502**
1.E-02	**0.703**	0.663	0.667	**0.475**	0.325	0.425	**0.483**	0.343	0.250	0.133	**0.223**	0.000	0.000	**0.127**	0.013
1.E-03	0.653	0.663	**0.667**	**0.440**	0.295	0.280	**0.470**	0.323	0.200	0.133	**0.203**	0.000	0.000	**0.130**	0.000
1.E-04	0.653	0.663	**0.667**	**0.413**	0.290	0.115	**0.470**	0.270	0.173	0.130	**0.167**	0.000	0.000	**0.125**	0.000
1.E-05	0.633	0.657	**0.667**	**0.320**	0.287	0.047	**0.360**	0.250	0.170	0.103	**0.170**	0.000	0.000	**0.123**	0.000

4.3. Comparación con otros algoritmos

Para comparar hemos evaluado, usando el mismo benchmark y bajo las mismas condiciones, los resultados de dos variantes de Evolución Diferencial (DE) para nichos, el conocido Crowding DE/rand/1/bin (CDE) [10] y otro recientemente propuesto DE/nrand/1/bin (DE/nrand) [2], que utiliza información sobre la distancia mínima entre óptimos.

La Tabla 2 muestra los resultados comparativos entre NVMO y los algoritmos DEs para nichos. Se puede apreciar que NVMO obtiene buenos resultados en más funciones que DE/nrand y CDE para los distintos valores de exactitud. En muchas funciones VMO obtiene los mejores resultados y cuando esto no ocurre los resultados son similares a los del algoritmo DE/nrand (el mejor de los DEs). Por otra parte, mientras DE/nrand y CDE presenta buenos resultados en algunas funciones y en otras no, NVMO obtiene buenos resultados en la mayoría de ellas.

En conclusión, NVMO ofrece los mejores resultado y es el más robusto de los algoritmos comparados.

Para hacer un análisis más general de los resultados mostramos en la Tabla 3 la media de los valores de PR para cada nivel de exactitud. Se puede apreciar claramente que NVMO presenta mejor comportamiento que los otros algoritmos para los distintos niveles de exactitud utilizados. Se puede observar que, en cuanto a la media del total de óptimos encontrados, NVMO detecta cerca del 70 % (69.8) del total mientras que el mejor de los algoritmos de referencia obtiene un ratio cercano al 60 % (hay una diferencia de un 10 %).

Tabla 3. Valor medio de PR para cada algoritmo

Nivel de exactitud	NVMO	DE/nrand	CDE
1.0E-1	0.879	0.619	0.734
1.E-02	0.712	0.602	0.634
1.E-03	0.682	0.598	0.577
1.E-04	0.652	0.591	0.492
1.E-05	0.567	0.566	0.435
valor medio	0.698	0.594	0.575

5. Conclusiones

En este trabajo presentamos una variante del algoritmo VMO [7] para problemas multimodales NVMO. Que difiere del modelo original en lo siguiente:

- Posee un método de combinación que crea soluciones entre cada nodo de la población y su óptimo global más cercano. Consideramos que un nodo es óptimo global si tiene calidad semejante a la mejor solución actual. Este método de generación reemplaza la fase 2 del VMO original, donde solamente se orientaba la búsqueda hacia la mejor solución. Esta forma de generación no utiliza información especifica del problema (sobre los óptimos o su distancia entre ellos).
- Cada cierta iteraciones se aplica un método de BL a un conjunto de soluciones seleccionadas. Estas soluciones son los buenas soluciones distantes entre sí. La misma solución puede ser seleccionada en más de una ocasión, siempre y cuando haya su aplicación anterior de la BL haya mejorado sustancialmente el *fitness*. Al aplicar la BL se adapta el tamaño inicial de salto en función de la distancia a su vecino más cercano. De esta manera, podemos obtener mayor exactitud en la etapa final del algoritmo.
- Se utiliza una memoria externa para almacenar los óptimos encontrados en el proceso de búsqueda. Este elemento es implementado en NVMO porque

la versión original VMO puede perder soluciones óptimas encontradas como consecuencia del proceso de *clearing* (al no usar información sobre los óptimos). Por tanto, nuestra propuesta, al no utilizar información del problema como la distancia mínima entre óptimos, puede ser utilizada en en problemas reales en donde esta información raramente se conoce.

Los resultados muestran que NVMO obtiene mayor cantidad de óptimos globales en relación a los otros algoritmos comparados. Además, para cada uno distintos valores de exactitud considerados, nuestra propuesta de VMO mejora los resultados de los otros algoritmos. En conclusión, hemos presentado un algoritmo basado en nichos muy robusto y competitivo,

6. Agradecimientos

Este trabajo ha sido financiado por los proyectos de investigación TIN2012-37930-C02-01 y P08-TIC-04173.

Referencias

1. R. Brits, a.P. Engelbrecht, and F. van den Bergh. Locating multiple optima using particle swarm optimization. *Applied Mathematics and Computation*, 189(2):1859–1883, 2007.
2. M.G. Epitropakis, V.P. Plagianakos, and M.N. Vrahatis. Finding multiple global optima exploiting differential evolution's niching capability. In *Differential Evolution (SDE), 2011 IEEE Symposium on*, pages 1–8, 2011.
3. F. Glover. *Handbook of meta-heuristics*. Massachusetts: Kluwer Academic Publishers, 2003.
4. David E. Goldberg and Jon Richardson. Genetic algorithms with sharing for multimodal function optimization. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*, pages 41–49, Hillsdale, NJ, USA, 1987. L. Erlbaum Associates Inc.
5. X. Li, A. Engelbrecht, and M.G. Epitropakis. Benchmark Functions for CEC'2013 Special Session and Competition on Niching Methods for Multimodal Function Optimization. Technical report, Evolutionary Computation and Machine Learning Group, RMIT University, Australia, 2013.
6. P. A. Moscato. *Memetic Algorithms: a Short Introduction*. D. Corne, M. Dorigo and F. Glower, editors. McGraw-Hill, London, 1999.
7. Amilkar Puris, Rafael Bello, Daniel Molina, and Francisco Herrera. Variable mesh optimization for continuous optimization problems. *Soft Computing*, 16(3):511–525, 2012.
8. B. Y. Qu, P.N. Suganthan, and J.J. Liang. Differential evolution with neighborhood mutation for multimodal optimization. *Evolutionary Computation, IEEE Transactions*, 16(5):601–614, 2012.
9. F. J. Solis and R. J. Wets. Minimization by Random Search Techniques. *Mathematical Operations Research*, 6:19–30, 1981.
10. R. Thomsen. Multimodal optimization using crowding-based differential evolution. In *Evolutionary Computation, 2004. CEC2004. Congress on*, volume 2, pages 1382–1389, 2004.

Detección de puntos de comienzo y fin de la traducción génica mediante la combinación evolutiva de modelos basados en máquinas de vectores soporte.

Javier Pérez-Rodríguez and Nicolás García-Pedrajas

Departamento de Informática y Análisis Numérico, Universidad de Córdoba
javier.perez@uco.es, npedrajas@uco.es
http://www.cibrg.org/

Resumen La localización de sitios funcionales se considera una de las tareas fundamentales a la hora de abordar el reconocimiento de genes en secuencias de ADN. Concretamente, de vital importancia es determinar los puntos del gen donde comienza y termina el proceso de traducción dentro de la síntesis proteica. Estos puntos son los llamados respectivamente Sitios de inicio de traducción (TIS), y Codones de parada (Stop-codon). Desde el punto de vista computacional, y teniendo en cuenta los distintos factores biológicos, se puede considerar esta tarea como un problema de clasificación, en el cual las Máquinas de Vectores Soporte (SVM) con un kernel de cadenas representan una herramienta idónea para obtener resultados bastante satisfactorios. Este trabajo muestra como la combinación de distintos modelos de SVM con kernel de cadenas generados a partir de una selección adecuada – y completamente novedosa – de los patrones de los conjuntos de entrenamiento, puede mejorar los resultados de los mejores enfoques con este propósito existentes hasta la fecha.

1. Introducción

Todos los procesos biológicos que se experimentan en los seres vivos vienen determinados por las proteínas, las cuales son resultado directo de sus genomas. Sin embargo, sólo algunas partes del genoma, las denominadas secuencias codificantes (CDS), participan en la síntesis de las proteínas. De este modo, determinar qué zonas de las secuencias ADN son funcionalmente relevantes es un problema central en bioinformática y en biología computacional [5].

Dependiendo de su naturaleza, las técnicas existentes de reconocimiento automático de secuencias codificantes pueden ser de dos tipos. Por un lado se encuentran los enfoques basados en homologías, los cuales tratan de buscar similitudes en las bases de datos de secuencias anotadas. La enorme cantidad de información sobre genomas y los genes conocidos en ellos dotan a este tipo de técnicas del potencial que ostentan. Sin embargo, debido a su carácter, cuentan con la desventaja de sólo predecir elementos con características similares a las

ya descubiertas. Por otro lado, se han desarrollado una serie de métodos capaces de discernir entre las partes funcionalmente relevantes de las que no sin la necesidad de utilizar información de la búsqueda por homología. Las denominadas técnicas intrínsecas extraen propiedades estadísticas discriminantes de la propia secuencia de nucleótidos.

CONTRAST [3], uno de los sistemas punteros en lo que a reconocimiento de genes se refiere, incorpora un modelo para reconocimiento de Sitios de inicio de traducción (TIS), Codones de parada (Stop-codon) y sitios de corte y empalme (splice-sites) en secuencias basado en Máquinas de Vectores Soporte (SVM). Concretamente el modelo que utiliza este sistema viene implementado mediante una SVM con una función de kernel de gausiano. Existen otros métodos intrínsecos que abordan el mismo problema ya sea desde una perspectiva probabilística o desde un punto de vista discriminativo. En el primero de los casos se encuentra GENESCAN, que utiliza un modelo basado en las frecuencias relativas posicionales de los nucleótidos que se encuentran alrededor de un sitio funcional [10]. En el segundo grupo están los métodos basados en redes neuronales, en funciones discriminantes lineales como ATGpr [9] y, fundamentalmente, en las ya mencionadas SVM. Actualmente, se considera este último, utilizando kernel de cadenas, como el método más preciso y eficaz para este tipo de tarea [8].

En este trabajo, se muestra como se puede superar el rendimiento de los métodos existentes hasta la fecha en lo que a reconocimiento de TIS y codones de parada se refiere, mediante la combinación de distintos modelos basados en máquinas de vectores soporte con kernel de cadenas. Particularmente, dicha combinación será optimizada mediante la aplicación de dos métodos heurísticos ampliamente utilizados en la literatura: la Evolución Diferencial y los Algoritmos Genéticos.

El resto del trabajo se organiza como sigue: La sección 2 presenta algunos conceptos preliminares sobre las técnicas y métodos empleados. Además se describe el modelo propuesto y las medidas de evaluación utilizadas. La sección 3 muestra el estudio experimental llevado a cabo. Por último, en la la sección 4 se presentan y analizan los resultados y se concluye el trabajo.

2. Reconocimiento de sitios mediante la combinación evolutiva de modelos basados en máquinas de vectores soporte.

2.1. Las SVM como reconocedores de sitios funcionales

Debido al excelente rendimiento que se ha visto especialmente en problemas de dos clases y un número de entradas alta, las Máquinas de vectores soporte se consideran un método muy atractivo a la hora de afrontar la predicción de sitios de inicio y fin de la traducción en secuencias de ADN. Las SVM aprenden a discriminar dos clases mediante la maximización del margen de clasificacién entre las dos clases. Las SVM utilizan funciones de kernel como un método

para separar las clases en espacios de alta dimensionalidad sin la necesidad de trabajar de forma explícita con las variables de dichos espacios. Para el caso de la clasificación de sitios en ADN, existen kernel que comparan pares de secuencias en términos de emparejamiento de subcadenas. Concretamente, un tipo de kernel de cadenas, el kernel WD [8], tiene en cuenta el emparejamiento de k-meros para todas las longitudes de k desde 1 a un determinado grado d, pero únicamente si estos emparejamientos ocurren en la misma posición en ambas secuencias.

2.2. Categorización de patrones negativos: optimizando el proceso de construcción de los conjuntos de datos

En una primera instancia, dentro de la cadena de nucleótidos de un cromosoma, los genes son las secuencias de nucleótidos funcionalmente relevantes a la hora de la síntesis de proteínas. Los genes se encuentran separados entre sí por regiones mucho más largas – comprenden la mayor parte de la secuencia de los genomas – cuya funcionalidad es objeto de investigación de la actualidad: el espacio intergénico. Dentro de los genes de organismos eucariotas, más concretamente, en la parte que se transcribe a ARN, las regiones pueden ser de dos tipos: exones e intrones. Los intrones son regiones que se transcriben, pero que son eliminados en el proceso de traducción para producir un ARNm, el cual está formado sólo por las bases encontradas en los exones, encargadas de transformarse en proteína.

En todos los métodos de clasificación de sitios en secuencias de ADN encontrados hasta el momento, los patrones que conformaban los conjuntos de datos de entrenamiento para generar los modelos correspondientes, podían ser de dos clases. Sin embargo, tal y como ha quedado demostrado en trabajos previos [7] , categorizar en más de una clase los patrones resulta una idea muy ventajosa a la hora de obtener mejores resultados de predicción de este tipo de puntos.

2.3. Combinación de modelos

Aunque se han desarrollado numerosas y sofisticadas propuestas de combinación de clasificadores para mejorar el desempeño en tareas de predicción, este trabajo pretende considerar una combinación simple de modelos de predicción de sitios para poder valorar el incremento de rendimiento con respecto a cada uno de los enfoques por separado. De un modo más específico, se plantea el encontrar una combinación óptima mediante la asignación un peso específico a cada uno de los modelos generados en pos de conseguir un mejor modelo de predicción. La búsqueda heurística puede resultar una herramienta excelente para conseguir este propósito. Por tanto, los algoritmos evolutivos resultan ideales para afrontar la tarea de optimización planteada, sobre todo ante el complejo espacio de búsqueda que se ha observado. Se propone la implementación de dos enfoques de esta naturaleza como son, por un lado la Evolución Diferencial, y por otro los conocidos Algoritmos Genéticos para atacar la búsqueda de una solución óptima.

2.4. Medidas de evaluación

La precisión no es una medida útil en los casos de enfrentarse a problemas de clases no están equilibradas, especialmente cuando el número de patrones perteneciente a la clase minoritaria está muy por debajo con respecto al de la mayoritaria. Si se presentara un caso en el que la proporción es de 1:100, un clasificador que asigne a todas las instancias la clase mayoritaria obtendrá una precisión del 99 %. En la clasificación de sitios de comienzo y fin de la traducción en secuencias de ADN, la proporción de puntos que efectivamente son funcionalmente activos con respecto a los que son simples candidatos está mucho más desequilibrada, haciéndose necesario usar otras medidas[11] para tener una idea del desempeño de los clasificadores.

De este modo, teniendo en cuenta la naturaleza de este tipo de problemas, se han desarrollado numerosas medidas. Partiendo de los conceptos de verdaderos positivos (TP), falsos positivos (FP), verdaderos negativos (TN) y falsos negativos (FN), se pueden definir las siguientes dos medidas básicas:

- Tasa de verdaderos positivos TP_{rate} o sensibilidad Sn:

$$TP_{rate} = R = Sn = \frac{TP}{TP+FN} \tag{1}$$

  Esta medida es relevante si estamos interesados solo en el rendimiento sobre la clase positiva.
- Tasa de verdaderos negativos TN_{rate} or especificidad Sp:

$$TN_{rate} = Sp = \frac{TN}{TN+FP} \tag{2}$$

Particularmente, en este trabajo se pretende una alcanzar unos resultados en el que exista un óptimo compromiso en la relación entre sensibilidad y especificidad. Una media geométrica (G-mean) de ambos valores se puede considerar que ilustra este objetivo.

$$G - mean = \sqrt{Sn \cdot Sp} \tag{3}$$

donde PP es el número de positivos predichos, AP los positivos reales, PN los negativos predichos y AN los negativos reales.

3. Diseño experimental

El proceso experimental llevado a cabo para este trabajo consta en dos pasos fundamentales. En una primera etapa, se ha tenido en cuenta el diseño experimental y su consiguiente ajuste de parámetros de trabajos previos [7]. En consecuencia, se han entrenado una serie de modelos clasificadores basados en Máquinas de vectores soporte con kernel de cadenas. En este proceso, para obtener cada uno de los modelos se utilizaron como conjunto de entrenamiento tanto patrones positivos, como una categoría determinada de patrones negativos, en función estos últimos de su localización y en consecuencia, del contenido y estructura de la información que poseen. Los modelos generados fueron:

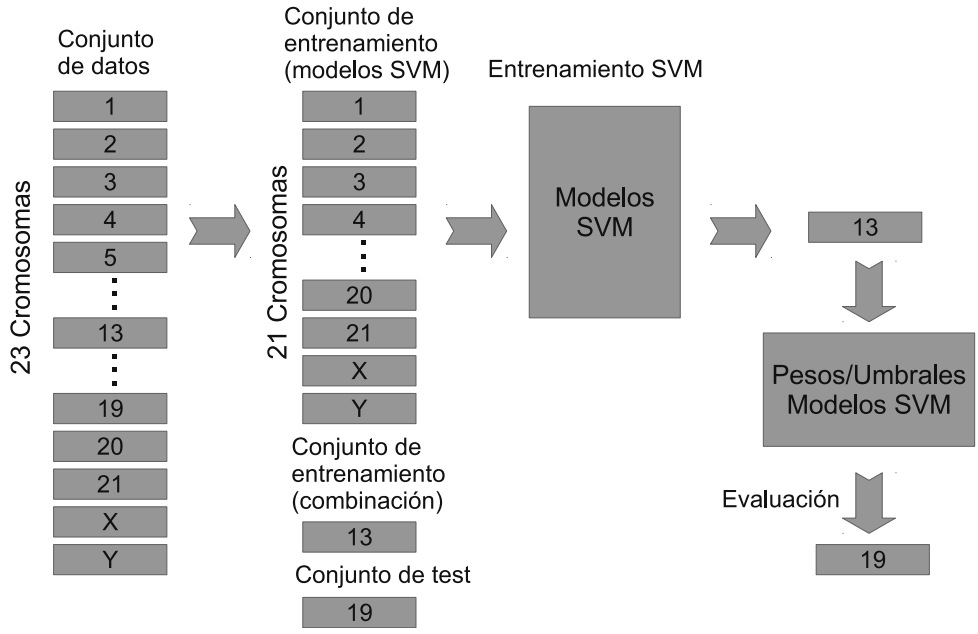


Figura 1: Diseño de experimento

- Modelo SVM con kernel de cadenas a partir de los patrones positivos y todos los patrones negativos. Este modelo es el originalmente propuesto en la bibliografía [8].
- Modelo SVM con kernel de cadenas a partir de los patrones positivos y los patrones negativos localizados en exones.
- Modelo SVM con kernel de cadenas a partir de los patrones positivos y los patrones negativos localizados en intrones.
- Modelo SVM con kernel de cadenas a partir de los patrones positivos y los patrones negativos localizados en regiones intergénicas.

Además, se incorpora un modelo de índole probabilística llamado Método de Codón de parada [11]. Este consiste en examinar la frecuencia de codones de parada en las posiciones de la cadena posteriores al TIS en cuestión. Los TIS representan el inicio del primer exon, lo que nos informa del marco de lectura del primer exón. Normalmente, los primeros exones tienen una longitud mínima en la que no se encontrarán codones de parada in-frame. Los pseudo TIS no tendrán esta restricción, lo que supone una característica discriminante entre los TIS reales y los que no lo son. A partir de este principio se puede construir un predictor bastante simple analizando la presencia de codones de parada in-frame en la región posterior a un TIS potencial.

Es de importancia destacar que el conjunto de patrones destinados a entrenamiento se divide en dos. Por un lado, un subconjunto, generalmente con mayor

número de patrones, se usa para obtener los distintos modelos de clasificación descritos. El segundo subconjunto se utilizará para obtener las salidas de estos modelos, las cuales servirán como entrada al siguiente paso en el proceso de experimentación. En esta etapa se lleva a cabo la optimización de pesos para la combinación de los distintos modelos planteados mediante una búsqueda heurística realizada a través de un algoritmo con carácter evolutivo. Como quedó mencionado en la sección 2.3, se han implementado dos enfoques ampliamente conocidos y utilizados para este tipo de tareas: un algoritmo de Evolución Diferencial (ED) y un Algoritmo Genético (AG). En ambos casos, se ha evolucionado una población de vectores en los que cada dimensión representaba un factor de ponderación para cada uno de los modelos de clasificación. Por tanto, en una primera aproximación, denominada OW, la salida de un individuo w de dimensión n – número de clasificadores – para un patrón p quedaría determinada según la expresión:

$$O(p) = \sum_{i=1}^{n} w_i o_i \tag{4}$$

La aptitud de los individuos ha sido determinada por la precisión que el modelo combinado resultante presentaba en clasificación, siempre en términos de media geométrica.

Con el objetivo de incrementar el rendimiento del proceso presentado, se ha implementado otro modelo de combinación denominado MTHR en el que además de los pesos de los distintos clasificadores, un individuo representaba en su genotipo un factor corrector f asociado a cada una de sus dimensiones. En esta nueva versión, la salida de un individuo w de dimensión n para un patrón p quedaría determinada según la expresión:

$$O(p) = \sum_{i=1}^{n} w_i (o_i - f_i) \tag{5}$$

Tanto el Algoritmo Genético como la Evolución Diferencial han evolucionado una población de 200 individuos durante 200 generaciones. El número de evaluaciones máximas consentidas en ambos casos se estableció en 20000. Los operadores utilizados han sido los ampliamente conocidos y utilizados en la bibliografía en los dos paradigmas.

El código fuente de estos métodos, en C y en Python, así como todos los conjuntos de datos utilizados se pondrán a disposición de cualquiera que lo solicite a los autores. Las SVM fueron implementadas con la biblioteca *libsvm*[2].

3.1. Conjuntos de datos

Para comprobar el desempeño de los métodos descritos, se ha utilizado un conjunto de datos extraído de las secuencias del genoma humano. Gracias a la secuenciación GRCh37.p9 y su correspondiente CCDS release 11 [4] – un proyecto colaborativo que supone un esfuerzo para compilar e identificar un conjunto referencia de genes humanos anotados de un modo preciso – se han

obtenido el global de secuencias correspondientes a los puntos de inicio y fin de traducción de cada uno de los genes organizados por cromosomas. Estos suponen todas las instancias positivas utilizadas. Para conseguir los patrones negativos se han procesado base a base la secuenciación de cada uno de los contigs de los cromosomas buscando las secuencias consenso de cada tipo. En el caso de los TIS la secuencia consenso es el codón ATG y para el caso del codón de parada son los tripletes TAA, TAG y TGA. Una vez localizado el sitio y comprobado que no se trataba de un patrón positivo, apoyándose en la información del CCDS cada instancia ha sido categorizada según la región en la que se encuentra. Dicho esto se podría especular con la posibilidad de que un patrón negativo perteneciese a dos clases al mismo tiempo debido al proceso de *splicing* alternativo que se experimenta en ciertas ocasiones en algunos genes. Esto ha sido tenido en cuenta a la hora de la categorización, eliminando aquellos patrones negativos de la clase mayoritaria.

Se han considerado unas secuencias de 500 bps arriba y abajo del punto en cuestión para cada patrón, obteniendo una longitud total de 1003 bps para cada uno de ellos.

El conjunto de patrones de entrenamiento utilizado ha sido extraído de todos los cromosomas del genoma humano a excepción del 13 en un caso y el 19 en otro, utilizados respectivamente para test. Como subconjunto de entrenamiento utilizado para optimizar los pesos y factores correctores, se consideró respectivamente los patrones del cromosoma 19 y del 13. El cromosoma 13 cuenta con 417 genes entre sus más de 95589 Kbps. El cromosoma 19 cuenta con 1767 genes entre sus más de 55858 Kbps.

4. Resultados

Para finalizar el estudio experimental se analizan los resultados en términos de precisión o acierto obtenido durante la clasificación de los patrones correspondientes al conjunto de datos de test. Como se ha mencionado en las secciones previas, el problema tiene una naturaleza no balanceada, por lo que el rendimiento de los modelos queda mejor expresado en función de la media geométrica de los valores de sensibilidad y especificidad. La tabla 1 muestra los resultados medios obtenidos para los dos conjuntos de test correspondientes a los patrones de TIS. Gráficamente, se puede observar una comparativa de rendimientos en la figura 2. De un modo análogo, la tabla 2 muestra los resultados medios obtenidos para los conjuntos de test correspondientes a los patrones de codones de parada de traducción. Igualmente, se puede observar una comparativa gráfica de rendimientos en la figura 2. En todos los casos y para que el estudio quede aún más completo, se incluyen los resultados de la combinación de modelos mediante suma no ponderada.

Analizando los resultados se han recogido de los distintos experimentos, se puede decir que, en general, la combinación ponderada de modelos de predicción tanto de TIS como de codones de parada en secuencias de ADN supera, con amplio margen, ya no sólo al rendimiento de clasificación del modelo general, si

Test set	Func.	DE Train	DE Test	AG Train	AG Test	Todos neg	Neg exon	Neg intr	Neg IG	Mét. SC	Comb. Suma
Chrom 13	OW	0,8871	0,8784	0,886	0,8764	0,8298	0,8616	0,786	0,8611	0,7619	0,8439
Chrom 13	MTHR	0,8894	0,8796	0,8897	**0,8864**	0,8298	0,8616	0,786	0,8611	0,7619	0,8439
Chrom 19	OW	0,8897	0,8789	0,8882	0,8666	0,8401	0,8584	0,7753	0,8576	0,6894	0,8589
Chrom 19	MTHR	0,8787	**0,8777**	0,9115	0,8689	0,8401	0,8584	0,7753	0,8576	0,6894	0,8589

Cuadro 1: Rendimiento – G-mean de Sp y Sn – de los diferentes modelos de reconocimiento de TIS en los cromosomas 13 y 19 del genoma humano.

Test set	Func.	DE Train	DE Test	AG Train	AG Test	Todos neg	Neg exon	Neg intr	Neg IG	Mét. SC	Comb. Suma
Chrom 13	OW	0,8994	0,8695	0,8967	0,8685	0,8158	0,7480	0,8324	0,8331	0,8522	0,8560
Chrom 13	MTHR	0,9076	**0,8895**	0,9032	0,8801	0,8158	0,7480	0,8324	0,8331	0,8522	0,8560
Chrom 19	OW	0,8903	0,8771	0,8963	0,8639	0,8410	0,7376	0,8517	0,8515	0,8562	0,8372
Chrom 19	MTHR	0,8933	**0,8987**	0,8948	0,8980	0,8410	0,7376	0,8517	0,8515	0,8562	0,8372

Cuadro 2: Rendimiento – G-mean de Sp y Sn – de los diferentes modelos de reconocimiento de codones de parada en los cromosomas 13 y 19 del genoma humano.

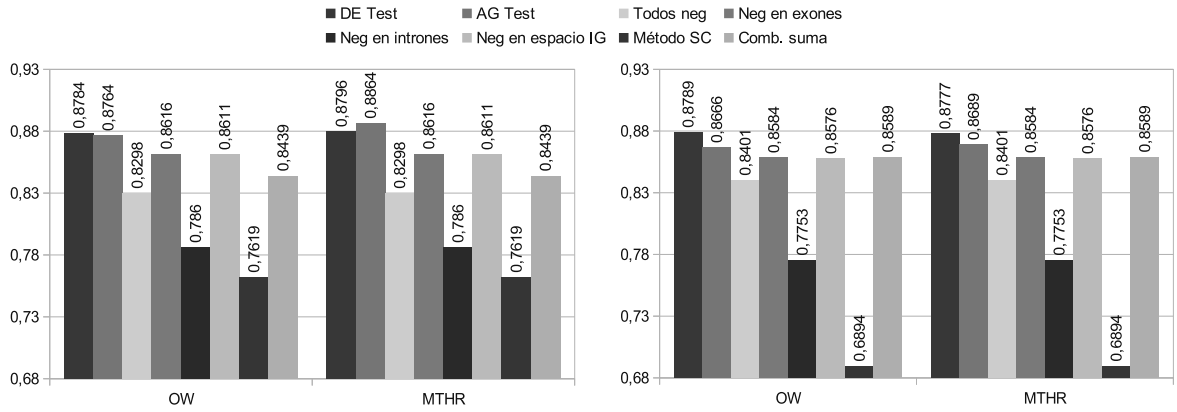


Figura 2: Comparativa del rendimiento de los diferentes modelos de reconocimiento de TIS en los cromosomas 13 y 19 del genoma humano.

no también, a cada uno de los modelos por separado y a la combinación de los mismos sin ponderar. En segundo lugar hay que notar que ambos paradigmas de búsqueda obtienen soluciones muy similares, quizá alcanzando mayores cotas de precisión el método de Evolución Diferencial. También es llamativo como la combinación con factores de corrección (MTHR) supera en todos los casos la combinación ponderada simple (OW). Por último, señalar como la mejora

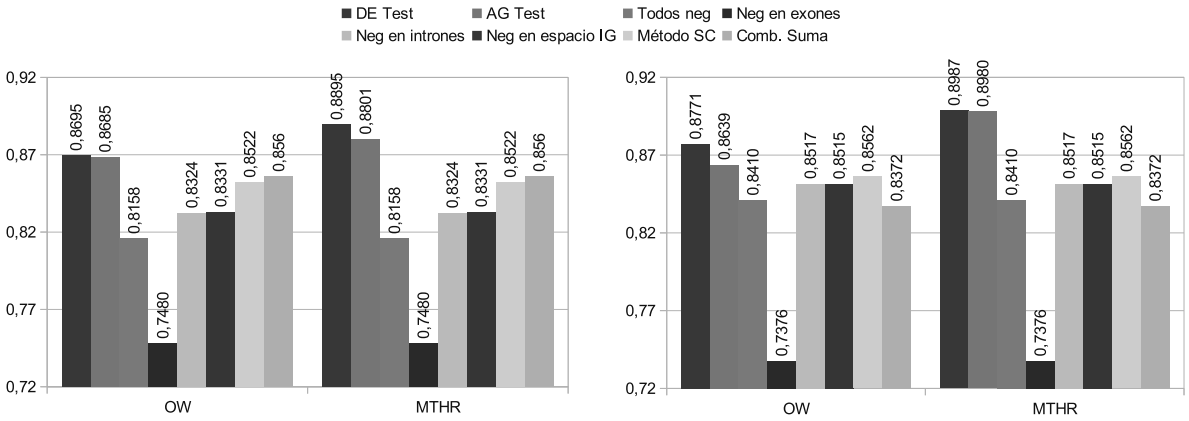


Figura 3: Comparativa del rendimiento de los diferentes modelos de reconocimiento de codones de parada en los cromosomas 13 y 19 del genoma humano.

experimentada en clasificación es mucho más notable en el caso del Codón de parada que en el caso del TIS.

5. Conclusiones

A modo de conclusión se puede decir que en este trabajo se ha presentado un modelo de detección de puntos de comienzo y fin de la traducción en secuencias del genoma humano mediante la combinación evolutiva de modelos basados en máquinas de vectores soporte con kernel de cadenas. Dicha combinación, optimizada mediante algoritmos de carácter evolutivo, supera significativamente en términos de rendimiento, según dictan los resultados experimentales, a todos los modelos originales y a otras propuestas anteriores.

Una extensión natural de este trabajo sería el estudio pormenorizado de otras vías de combinación más complejas que incrementasen el desempeño en las clasificación.

Referencias

1. Brent, M.R., Guigó, R.: Recent advances in gene structure prediction. Current Opinion in Structural Biology 14, 264–272 (2004)
2. C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, 2001, software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
3. Gross SS, Do CB, Sirota M, Batzoglou S. CONTRAST: A discriminative, phylogeny-free approach to multiple informant de novo gene prediction. Genome Biol. 2007;8:R269. doi: 10.1186/gb-2007-8-12-r269.
4. Harte RA, Farrell CM, Loveland JE, Suner MM, Wilming L, Aken B, Barrell D, Frankish A, Wallin C, Searle S, Diekhans M, Harrow J, Pruitt KD: Tracking and coordinating an international curation effort for the CCDS Project. Database 2012 Mar 20;2012:bas008. doi: 10.1093/database/bas008. PubMed: PMID: 22434842

5. Mount D. Bioinformatics: Sequence and Genome Analysis. Cold Spring Harbor Laboratory Press, New York, 2001
6. X.-Y. Liu, J. Wu, Z.-H. Zhou, Exploratory undersampling for class-imbalance learning, IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics 39 (2) (2009) 539–550.
7. Pérez-Rodríguez J, Arroyo-Peña G, García-Pedrajas N. Improving translation initiation site and stop codon recognition using more than two classes. Bioinformatics (2013). Submitted.
8. Sonnenburg S, Rätsch G, Schäfer C. Learning interpretable SVMs for biological sequence classification. BMC BIOINFORMATICS. 2005;3500:389–407.
9. Salamov, A. A., Nishikawa, T., Swindells, M. B. Assessing protein coding region integrity in cDNA sequencing projects. Bioinformatics, (1998). 14 (5), 384–390.
10. Salzberg, S. L. A method for identifying splice sites and translational start sites in eukaryotic mRNA. Comput Appl Biosci, (1997). 13 (4), 365–376.
11. Saeys, Y., Abeel, T., Degroeve, S., Van de Peer, Y. (2007) Translation initiation site prediction on a genomic scale: beauty in simplicity. Bioinformatics 23, 418-23.
12. R. Storn, K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous space, Journal of Global Optimization 11 (1997) 341–359.

Una Aproximación Multiobjetivo Basada en el Comportamiento de las Luciérnagas para la Inferencia Filogenética

Sergio Santander-Jiménez y Miguel A. Vega-Rodríguez

Univ. Extremadura. Dept. Tecnología de los Computadores y de las Comunicaciones, Escuela Politécnica. Campus Universitario s/n, 10003. Cáceres, España
{sesaji,mavega}@unex.es

Resumen A lo largo de los años se han publicado propuestas de diferente naturaleza para inferir historias evolutivas a partir de datos biológicos de especies. En este sentido, se ha hecho patente una creciente necesidad de resolver los problemas que surgen cuando distintos métodos filogenéticos dan como resultado hipótesis evolutivas conflictivas entre sí. En este artículo tratamos de afrontar el problema de la inferencia filogenética usando un algoritmo multiobjetivo inspirado en el comportamiento colectivo de las luciérnagas, implementando una representación del individuo basada en matrices de distancias genéticas. Por ello, es necesario examinar los diferentes métodos propuestos en la literatura para reconstruir árboles filogenéticos a partir de estas matrices. Los resultados obtenidos en cuatro bases de datos reales de nucleótidos ponen de manifiesto las ventajas e inconvenientes de cada una de las propuestas, valoradas en términos de rendimiento multiobjetivo y tiempos de procesamiento.

Keywords: Inteligencia de Enjambre, Optimización Multiobjetivo, Inferencia Filogenética, Algoritmo de las Luciérnagas

1. Introducción

La comprensión de los eventos evolutivos que han dado lugar a las especies actuales representa uno de los campos de investigación más destacados en bioinformática. El desarrollo de nuevas metodologías que definen la inferencia filogenética como un problema de optimización ha cobrado mayor importancia en los últimos años. En este sentido, se debe considerar una serie de cuestiones clave. En primer lugar, es necesario definir nuevas técnicas para afrontar el análisis de modernas bases de datos biológicas. Al aumentar el número de especies, se produce un crecimiento exponencial del número de topologías filogenéticas que impide la aplicación de métodos de búsqueda exhaustiva [1]. Por este motivo, diversas investigaciones [2] propusieron el uso de aproximaciones basadas en computación bioinspirada, demostrando ser una alternativa prometedora para obtener resultados de calidad en tiempos razonables.

En segundo lugar, existe una amplia variedad de criterios de optimalidad para inferir árboles filogenéticos, destacando los principios de parsimonia, verosimilitud y los métodos basados en distancias [3]. El problema que surge es que,

para una misma base de datos, distintos criterios pueden apoyar la inferencia de hipótesis conflictivas entre sí. Este hecho motivó la formulación de la inferencia filogenética como un problema de optimización multiobjetivo [4]. En este sentido, podemos destacar las investigaciones de Coelho et al. [5] y Cancino y Delbem [6], quienes aplicaron metaheurísticas multiobjetivo para optimizar el proceso de inferencia según diferentes principios, tales como mínima evolución, error cuadrático medio, máxima parsimonia y máxima verosimilitud.

En este artículo proponemos una aproximación multiobjetivo de inteligencia de enjambre para inferir árboles filogenéticos según los criterios de parsimonia y verosimilitud, el Multiobjective Firefly Algorithm (MO-FA). Dado que esta propuesta incorpora una representación del individuo basada en matrices de distancias genéticas, incluimos un estudio comparativo de los métodos más populares para reconstruir árboles filogenéticos a partir de dichas matrices: *neighbor-joining* (NJ), BIONJ, *unweighted* y *weighted-pair group methods with arithmetic means* (UPGMA y WPGMA) y *complete-linkage clustering*. Los resultados obtenidos sobre cuatro bases de datos biológicas serán evaluados empleando la métrica multiobjetivo de hipervolumen [7] y comparando su relevancia biológica con otras aproximaciones multiobjetivo y métodos biológicos del estado del arte.

Este artículo está organizado como sigue. En la Sección 2 introducimos los conceptos básicos del problema planteado. En la Sección 3 detallamos los diferentes métodos de reconstrucción basados en distancias examinados en este estudio. La Sección 4 presenta el algoritmo de inteligencia de enjambre propuesto. La Sección 5 muestra nuestra metodología experimental y los resultados obtenidos. Finalmente, la Sección 6 incluye conclusiones y líneas futuras de trabajo.

2. Inferencia Filogenética

Los métodos de inferencia filogenética analizan secuencias de datos biológicos que caracterizan a un conjunto de organismos con objeto de inferir sus relaciones evolutivas ancestrales [3]. Mediante su procesamiento, es posible reconstruir una hipótesis sobre la evolución de las especies estudiadas mediante una estructura arborescente $T = (V, E)$, siendo V el conjunto de especies ancestrales y actuales y E el conjunto de ramas que relacionan especies emparentadas. Existen diferentes principios para llevar a cabo la búsqueda de árboles filogenéticos, siendo dos de los más populares los criterios de máxima parsimonia y máxima verosimilitud.

Los métodos de máxima parsimonia tratan de reconstruir la hipótesis evolutiva que minimice el número de cambios a nivel molecular observados a lo largo de su topología. Dado un árbol filogenético $T = (V, E)$ inferido de un conjunto de datos compuesto por N secuencias de S caracteres, definimos el valor de parsimonia $P(T)$ del siguiente modo [8]:

$$P(T) = \sum_{i=1}^{S} \sum_{(a,b)\in E} C(a_i, b_i) \tag{1}$$

donde $(a, b) \in E$ representa una relación ancestral entre dos nodos a, $b \in V$, a_i y b_i son los valores en el carácter i-ésimo de las secuencias moleculares que

caracterizan a a y b, y $C(a_i, b_i)$ es el coste de evolución desde a_i a b_i. La hipótesis más parsimoniosa será aquel árbol filogenético que minimice el valor de $P(T)$.

Por su parte, el criterio de máxima verosimilitud busca encontrar la hipótesis evolutiva más probable de acuerdo al principio estadístico de verosimilitud, empleando modelos evolutivos que describen la probabilidad de observar eventos de mutación a nivel molecular. Dados un árbol filogenético $T = (V, E)$ generado a partir de un conjunto de N secuencias moleculares de longitud S, y m un modelo evolutivo, se define la verosimilitud $L[T, m]$ como sigue [1]:

$$L[T, m] = \prod_{i=1}^{S} \prod_{j=1}^{E} (r_i t_j)^{n_{ij}} \tag{2}$$

donde r_i es la probabilidad de mutación para el carácter i, t_j el tiempo evolutivo transcurrido entre los nodos emparentados mediante $j \in E$, y n_{ij} el número de cambios observados entre los nodos conectados por j para el carácter i-ésimo. Aquella solución que maximice la función de verosimilitud será considerada como la hipótesis más probable según los supuestos dados por el modelo m.

3. Métodos Basados en Distancias

Las aproximaciones basadas en distancias gozan de gran popularidad en filogenética. Las distancias describen una estimación del ratio de sustituciones en las secuencias moleculares, bajo los supuestos dados por un modelo evolutivo [3]. Dadas N secuencias a analizar, estos métodos calculan una estructura matriz M de tamaño NxN, donde cada entrada $M[i, j]$ contiene la distancia genética entre las secuencias que caracterizan a las especies (o grupos de especies) i y j. En un segundo paso, estos métodos construyen el árbol agrupando de manera iterativa aquel par de especies en M que optimizan un determinado criterio. El Pseudocódigo 1 muestra este proceso para un método de distancias genérico.

Pseudocódigo 1 Algoritmo de Distancias

```
repetir
    i, j ← Seleccionar los índices i, j que optimicen un determinado valor según M[i, j]
    n_ij ← Crear un nodo padre conectándolo a i, j
    /*Asignar valores de longitud de rama l_i y l_j de i y j hacia n_ij */
    n_ij.l_i, n_ij.l_j ← Calcular longitudes de rama (M[i, j])
    /*Calcular las distancias de n_ij al resto de los grupos*/
    para k = 1 to N hacer
        D[k] ← Calcular distancias (M, n_ij, k) /* k ≠ i, j */
    fin para
    M, N ← Actualizar matriz con las nuevas distancias (n_ij, D)
    T ← Añadir nuevo nodo (n_ij)
hasta que No existan grupos por procesar
```

Métodos de Clustering. Una de las aproximaciones tradicionales basadas en distancias son los métodos de clustering. Estas técnicas siguen el concepto de

reloj molecular, que considera que la tasa de cambios en las secuencias moleculares es constante [1]. Dada una matriz de distancias M, un método de clustering procede seleccionando aquellas entradas i,j que minimicen el valor de distancia genética. Se genera una topología parcial o clúster conectando i y j a un nuevo nodo n_{ij} que representa su ancestro común. Tras ello, se calculan las longitudes de rama como l_i, l_j=$M[i,j]/2$ y se actualiza M con las distancias al nuevo clúster. El cálculo de estas distancias depende del método de clustering utilizado, entre los que se encuentran UPGMA, WPGMA, y *complete-linkage* [3].

Sea r el número de especies en una filogenia parcial generada al agrupar i y j. UPGMA asigna los valores actualizados en M mediante el cálculo de la media de las distancias considerando el número de elementos en el nuevo clúster:

$$D[k] = \left(\frac{i.r}{i.r + j.r}\right) M[i,k] + \left(\frac{j.r}{i.r + j.r}\right) M[j,k] \tag{3}$$

Por su parte, el método WPGMA considera que todos los elementos del nuevo clúster contribuyen con el mismo peso en el cálculo de las distancias:

$$D[k] = \frac{M[i,k] + M[j,k]}{2} \tag{4}$$

Finalmente, en la técnica de *complete-linkage*, $D[k]$ hace referencia a la distancia más grande desde cualquier elemento del clúster a k:

$$D[k] = \text{máx}(M[i,k], M[j,k]) \tag{5}$$

Neighbor-joining. El método de *neighbor-joining* (NJ) [9] se diferencia de los métodos de clustering en el hecho de que NJ no asume el supuesto de reloj molecular. Al considerar que los ratios de cambios a nivel molecular pueden evolucionar de maneras diferentes entre distintos linajes, NJ realiza el análisis de bases de datos complejas considerando supuestos más realistas.

En cada iteración, NJ realiza el cálculo de unos factores de divergencia $u[i]$ desde cada grupo de especies i en M a todo grupo $j \neq i$, mediante el sumatorio de $M[i,j]/(N-2)$. Considerando estos valores, el algoritmo procede seleccionando aquel par de índices i, j que minimicen $M[i,j] - u[i] - u[j]$. A la hora de calcular el valor de longitud de las ramas al nuevo nodo padre n_{ij}, se utiliza la fórmula $n_{ij}.l_i = \frac{1}{2}M[i,j] + \frac{1}{2}(u[i] - u[j])$. La asignación de distancias del clúster al resto de grupos en M también difiere con respecto a los métodos anteriores:

$$D[k] = (M[i,k] + M[j,k] - M[i,j])/2 \tag{6}$$

Por su parte, BIONJ [10] extiende este diseño algorítmico mediante la introducción de un modelo de varianzas y covarianzas calculadas a partir de las distancias evolutivas con objeto de reducir el error estadístico [1]. Para ello, se calcula un factor λ que se aplica al cálculo de las distancias al resto de grupos (véase la Ecuación 7) para minimizar las varianzas de muestreo en cada iteración.

$$D[k] = \lambda M[i,k] + (1-\lambda)M[j,k] - \lambda n_{ij}.l_i - (1-\lambda)n_{ij}.l_j \tag{7}$$

Estos métodos serán analizados para tratar de seleccionar aquella propuesta que permita maximizar el rendimiento multiobjetivo de nuestra aproximación.

4. Multiobjective Firefly Algorithm

El Multiobjective Firefly Algorithm (MO-FA) es una adaptación multiobjetivo de un novedoso algoritmo de inteligencia de enjambre, el Firefly Algorithm (FA) [11]. El comportamiento de las luciérnagas se fundamenta en los conceptos de atracción y bioluminiscencia. Mediante la emisión de destellos de luz, las luciérnagas son capaces de atraer posibles parejas, que preferirán a aquellas luciérnagas que muestren los patrones más intensos de emisión de luz. Esta atracción se ve afectada por tres factores: la intensidad de la luz emitida, la distancia entre luciérnagas y el grado de absorción de luz por parte del entorno. En un contexto algorítmico, se puede identificar la intensidad lumínica con la calidad de una solución, pudiéndose emplear el conocido concepto de dominancia [4] para distinguir dicha calidad desde un punto de vista multiobjetivo.

Para adaptar este algoritmo al problema de la inferencia filogenética, usaremos una metodología basada en el empleo de matrices de distancias para la representación del individuo, aplicando métodos basados en distancias para generar las topologías. MO-FA toma los siguientes parámetros de entrada: *tamañoEnjambre* (tamaño de la población), *numGeneraciones* (número de generaciones), β_0 (factor de atracción), γ (coeficiente de absorción), y α (factor de aleatoriedad). La salida del algoritmo estará constituida por un conjunto de soluciones *Pareto* que supongan un compromiso entre los principios de parsimonia y verosimilitud. El Pseudocódigo 2 resume las tareas principales en MO-FA.

Pseudocódigo 2 Multiobjective Firefly Algorithm

```
Inicializar y evaluar la población (X, tamañoEnjambre)
para numIter = 1 to numGeneraciones hacer
    para r = 1 to tamañoEnjambre hacer
        para s = 1 to tamañoEnjambre hacer
            si X_s ≻ X_r entonces
                Mover X_r hacia X_s (X_r.M, X_s.M, β_0, γ, α)
            fin si
        fin para
        Aplicar un método de distancias para inferir el árbol X_r.T (X_r.M)
        Evaluar la solución (X_r.T)
    fin para
    Actualizar el conjunto de Pareto con las soluciones más prometedoras (X)
fin para
```

En primer lugar, se inicializan las luciérnagas mediante la selección aleatoria de topologías filogenéticas procedentes de un repositorio compuesto por 1000 árboles generados mediante técnicas de máxima parsimonia y máxima verosimilitud. Estos árboles iniciales son evaluados conforme a los dos criterios considerados, tras lo cual se computan sus matrices de distancias.

Las luciérnagas asociadas a soluciones dominadas serán actualizadas teniendo en cuenta la información proporcionada por las soluciones más prometedoras

generadas. Sea X_r una luciérnaga en el enjambre X dominada por X_s, y $X_r.M$, $X_s.M$ las matrices de distancias de tamaño NxN asociadas a X_r y X_s, respectivamente. En primer lugar, se calcula la distancia global δ_{rs} que separa X_r de X_s como $\delta_{rs} = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{i}(X_r.M[i,j] - X_s.M[i,j])^2}$. Tras ello, se actualizan las entradas en $X_r.M$ aplicando una fórmula de movimiento considerando el valor δ_{rs}, el factor de atracción β_0, el coeficiente de absorción γ y el factor aleatorio α. Este último parámetro introduce aleatoriedad con objeto de permitir al algoritmo realizar búsquedas en regiones aún no exploradas del espacio de búsqueda y aportar diversidad. La Ecuación 8 describe este paso de actualización.

$$\begin{aligned} X_r.M[i,j] = & X_r.M[i,j] + \beta_0 e^{-\gamma\delta_{rs}^2}(X_s.M[i,j] - X_r.M[i,j]) \\ & + \alpha(rand[0,1] - \frac{1}{2}) \end{aligned} \tag{8}$$

Tras la actualización de la matriz, se infiere la topología filogenética resultante mediante el empleo de uno de los métodos de distancias considerados en este estudio, calculando posteriormente sus valores de parsimonia y verosimilitud. Estos pasos se repiten hasta procesar todas las luciérnagas dominadas. En los pasos finales de una generación, se actualiza el conjunto de soluciones Pareto más prometedoras, optimizándose los puntos extremos aplicando una metodología de búsqueda topológica [8] para ampliar el rango del frente de Pareto.

5. Resultados Experimentales

En esta sección introducimos la metodología experimental que se ha empleado para evaluar el rendimiento de MO-FA, considerando distintos métodos para llevar a cabo el proceso de reconstrucción filogenética a partir de distancias.

En primer lugar, se ha realizado un estudio para configurar los parámetros de entrada del algoritmo β_0, γ and α. El resto de parámetros *tamañoEnjambre* y *numGeneraciones* han sido configurados mediante experimentos adicionales y considerando las propuestas de otros autores [6]. La Tabla 1 da cuenta de los diferentes valores examinados para β_0, γ and α. La Tabla 2 resume la configuración de valores adoptada que permitía al algoritmo maximizar el rendimiento desde un punto de vista multiobjetivo, según la métrica de hipervolumen.

Tabla 1. Valores del estudio paramétrico

Parámetro	Valores
β_0	{0,05, 0,1, 0,25, 0,5, 0,75, 1}
γ	{0,05, 0,1, 0,25, 0,5, 0,75, 1}
α	{0,05, 0,1, 0,25, 0,5, 0,75, 0,9}

Tabla 2. Configuración final

Parámetro	Valor final
numGeneraciones	100
tamañoEnjambre	100
β_0	1
γ	0,5
α	0,05

Hemos llevado a cabo experimentos bajo el modelo $HKY85 + \Gamma$ [1] sobre cuatro bases de datos reales de nucleótidos [6]: *rbcL_55*, 55 secuencias (1314

nucleótidos) de plantas verdes, $mtDNA_186$, 186 secuencias (16608 nucleótidos) de ADN mitocondrial humano, $RDPII_218$, 218 secuencias (4182 nucleótidos) de ARN procariota y $ZILLA_500$, 500 secuencias (759 nucleótidos) del gen rbcL.

Para cada par instancia-método de distancias, se han realizado 30 ejecuciones independientes del algoritmo, evaluando los resultados en términos de hipervolumen y tiempos de procesamiento. Los puntos de referencia usados para el cálculo de hipervolumen vienen dados en la Tabla 3. Por su parte, la Tabla 4 resume los resultados obtenidos. Se muestra para cada instancia y método los valores de parsimonia y verosimilitud correspondientes al árbol más parsimonioso (columnas 2-3) y verosímil (columnas 4-5) obtenidos en la ejecución que obtuvo el valor más cercano al hipervolumen medio global (columnas 6-7). La columna 8 muestra los tiempos de ejecución medios de cada configuración. Finalmente, la Figura 1 muestra una representación gráfica de los frentes de Pareto obtenidos.

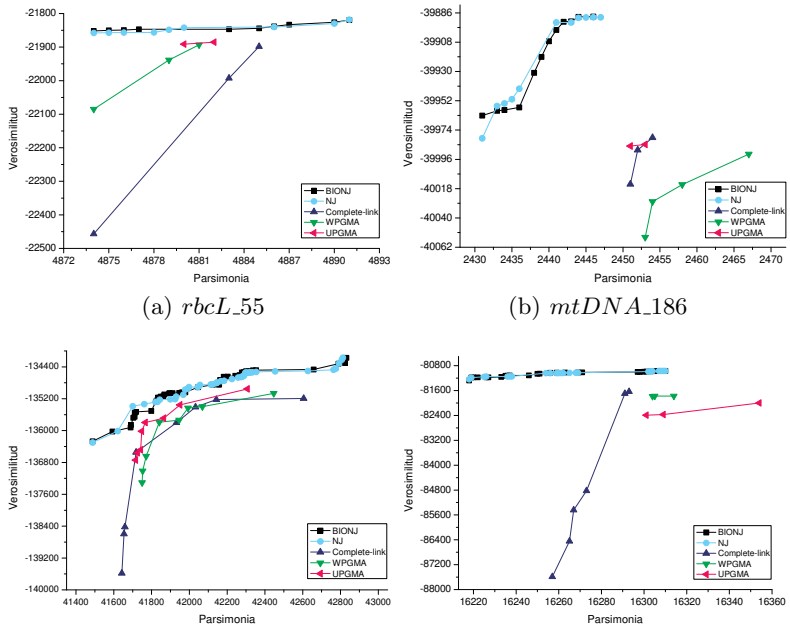
Los valores de hipervolumen muestran diferencias significativas según el método de distancias aplicado en MO-FA. NJ y BIONJ superan a las aproximaciones basadas en clustering en términos de rendimiento multiobjetivo y calidad de las soluciones en todas las bases de datos analizadas. Mientras que la selección de un método u otro implica variaciones de hipervolumen en torno al 3,419 % sobre $rbcL_55$, dicha elección tiene un impacto considerable sobre $ZILLA_500$, dando lugar a una diferencia de hipervolumen medio del 11,625 %. Estos resultados sugieren que los métodos basados en NJ se adecúan al proceso evolutivo mostrado en los datos de entrada, permitiendo al algoritmo generar árboles filogenéticos de calidad a partir de las matrices procesadas a lo largo de su ejecución. La métrica de hipervolumen muestra que BIONJ mejora ligeramente a NJ, reduciendo además la variación estadística de acuerdo a los valores de desviación estándar.

Tabla 3. Puntos de referencia hipervolumen

rbcL_55		
	Parsimonia	Verosimilitud
Ideal	4774	-21569,69
Nadir	5279	-23551,42
mtDNA_186		
	Parsimonia	Verosimilitud
Ideal	2376	-39272,20
Nadir	2656	-43923,99
RDPII_218		
	Parsimonia	Verosimilitud
Ideal	40658	-132739,90
Nadir	45841	-147224,59
ZILLA_500		
	Parsimonia	Verosimilitud
Ideal	15893	-79798,03
Nadir	17588	-87876,39

Tabla 4. Resultados experimentales obtenidos por MO-FA

rbcL_55							
	Árbol máx. parsimonia		Árbol máx.verosimilitud		Hipervolumen		Tiempos de
Métodos	Pars.	Veros.	Pars.	Veros.	Media	Desv. Est.	Ejecución (s)
UPGMA	4880	-21891,084	4882	-21885,359	66,488 %	0,352	287,613
WPGMA	4874	-22085,094	4881	-21893,387	66,916 %	0,481	269,203
Complete	4874	-22456,859	4885	-21898,369	66,491 %	0,619	**244,188**
NJ	4874	-21857,617	4891	-21818,107	**70,050 %**	**0,067**	395,729
BIONJ	4874	-21852,010	4891	-21819,188	70,039 %	0,070	399,001
mtDNA_186							
	Árbol máx. parsimonia		Árbol máx.verosimilitud		Hipervolumen		Tiempos de
Métodos	Pars.	Veros.	Pars.	Veros.	Media	Desv. Est.	Ejecución (s)
UPGMA	2451	-39985,988	2453	-39984,811	61,416 %	1,307	3476,906
WPGMA	2453	-40054,581	2467	-39992,076	61,215 %	0,827	3473,117
Complete	2451	-40014,640	2454	-39979,645	62,093 %	0,810	**2990,974**
NJ	2431	-39980,181	2447	-39889,267	69,658 %	0,012	3661,246
BIONJ	2431	-39963,162	2446	-39888,752	**69,659 %**	**0,005**	3707,444
RDPII_218							
	Árbol máx. parsimonia		Árbol máx.verosimilitud		Hipervolumen		Tiempos de
Métodos	Pars.	Veros.	Pars.	Veros.	Media	Desv. Est.	Ejecución (s)
UPGMA	41716	-136739,845	42307	-134953,526	65,779 %	0,602	5774,361
WPGMA	41750	-137306,291	42447	-135066,775	65,714 %	0,582	5455,675
Complete	41643	-138581,502	42605	-135193,930	65,593 %	0,686	4761,591
NJ	41488	-136297,204	42813	-134167,447	74,084 %	0,245	3888,291
BIONJ	41488	-136267,570	42831	-134173,576	**74,086 %**	**0,224**	**3862,010**
ZILLA_500							
	Árbol máx. parsimonia		Árbol máx.verosimilitud		Hipervolumen		Tiempos de
Métodos	Pars.	Veros.	Pars.	Veros.	Media	Desv. Est.	Ejecución (s)
UPGMA	16301	-82391,553	16354	-82001,417	55,409 %	0,797	4800,333
WPGMA	16304	-81786,190	16314	-81776,387	57,339 %	1,154	4796,018
Complete	16257	-87588,105	16293	-81632,727	59,588 %	1,029	**4774,700**
NJ	16218	-81236,179	16310	-80966,641	69,064 %	0,056	5974,231
BIONJ	16218	-81274,703	16310	-80966,525	**69,070 %**	**0,055**	5998,784



(a) $rbcL_55$ (b) $mtDNA_186$

(c) $RDPII_218$ (d) $ZILLA_500$

Figura 1. Frentes de Pareto obtenidos según el método de distancias aplicado

Con respecto a los tiempos de ejecución, se pueden observar diferencias entre los métodos de clustering y las aproximaciones de NJ que pueden explicarse en términos de complejidad algorítmica. Mientras que los procedimientos de clustering precisan n^2 operaciones para construir la topología, los métodos de NJ presentan una complejidad de orden $O(n^3)$ [1]. Para el caso de $RDPII_218$, los tiempos de procesamiento de NJ y BIONJ resultan mejores con respecto al análisis clustering. Dado que, en esta instancia, MO-FA genera un número significativo de soluciones no dominadas, el número de soluciones que se incluyen en el conjunto de Pareto crece a medida que evoluciona el algoritmo. Esto reduce el número de tareas a realizar (incluyendo llamadas a métodos de construcción y evaluación de soluciones), compensando la complejidad de estas aproximaciones.

En conclusión, aunque las técnicas de clustering precisan menores tiempos de procesamiento, BIONJ y NJ mejoran de manera significativa la calidad de las soluciones. Con respecto a los métodos de clustering, *complete-linkage* presenta los mejores tiempos de ejecución, mejorando los valores de hipervolumen obtenidos por UPGMA y WPGMA para $mtDNA_186$ y $ZILLA_500$. Por su parte, BIONJ destaca sobre todas las aproximaciones en términos de hipervolumen, reduciendo los tiempos de ejecución para $RDPII_218$. Los resultados experimentales muestran que una aproximación multiobjetivo y bioinspirada apoyada sobre un método estadísticamente consistente como BIONJ puede resultar de utilidad para llevar a cabo análisis filogenéticos en conjuntos de datos reales.

Comparativas con Otras Propuestas. Con objeto de evaluar la calidad biológica del algoritmo, presentamos a continuación una comparativa entre MO-FA (usando el método BIONJ) y las propuestas de otros autores. En primer lugar, la Tabla 5 introduce una comparativa a nivel de resultados con PhyloMOEA [6], un algoritmo multiobjetivo para reconstrucción filogenética por parsimonia y verosimilitud, usando el modelo evolutivo $HKY85+\Gamma$. Los resultados sugieren una mejora notable con respecto a los mejores valores de parsimonia y verosimilitud generados por PhyloMOEA en todas las instancias consideradas.

Tabla 5. Comparando con PhyloMOEA

	MO-FA			
	Árbol máx. parsimonia		Árbol máx. verosimilitud	
Instancia	Pars.	Veros.	Pars.	Veros.
rbcL_55	**4874**	-21852,01	4891	**-21819,19**
mtDNA_186	**2431**	-39963,16	2446	**-39888,75**
RDPII_218	**41488**	-136267,57	42831	**-134173,58**
ZILLA_500	**16218**	-81274,70	16310	**-80966,53**
	PhyloMOEA			
Instancia	Mejor parsimonia		Mejor verosimilitud	
rbcL_55	**4874**		-21889,84	
mtDNA_186	2437		-39896,44	
RDPII_218	41534		-134696,53	
ZILLA_500	16219		-81018,06	

Tabla 6. Comparando con TNT-RAxML

	MO-FA		
	Mejor parsimonia	Árbol máx. verosimilitud	
Instancia	Pars.	Pars.	Veros.
rbcL_55	**4874**	**4890**	**-21783,48**
mtDNA_186	**2431**	**2448**	**-39868,74**
RDPII_218	**41488**	**42833**	-134089,91
ZILLA_500	**16218**	**16305**	**-80608,46**
	TNT	RAxML ($GTR+\Gamma$)	
Instancia	Pars.	Pars.	Veros.
rbcL_55	**4874**	4893	-21791,98
mtDNA_186	**2431**	2453	-39869,63
RDPII_218	**41488**	42894	**-134079,42**
ZILLA_500	**16218**	**16305**	-80623,50

En segundo lugar, la Tabla 6 compara MO-FA con dos métodos biológicos populares: TNT [12] (para reconstrucción por máxima parsimonia) y RAxML [13] (máxima verosimilitud). La comparativa con TNT muestra que nuestra propuesta logra alcanzar los valores de referencia de TNT en todas las bases de datos. Con respecto a la comparativa de verosimilitud, dado que las últimas versiones de RAxML no incluyen el modelo $HKY85+\Gamma$, hemos realizado nuevos experimentos usando el modelo $GTR+\Gamma$ [1]. Bajo este modelo, nuestra topologías de máxima verosimilitud dominan los árboles generados por RAxML en *rbcL_55*, *mtDNA_186* y *ZILLA_500*, mejorando a su vez el valor de parsimonia para *RDPII_218*. Esta comparativa muestra la bondad de nuestra aproximación, la cual obtiene un rendimiento significativo desde un punto de vista biológico.

6. Conclusiones

En este artículo hemos propuesto un novedoso algoritmo multiobjetivo de inteligencia de enjambre para la inferencia filogenética, basado en el comportamiento de las luciérnagas. A su vez, hemos introducido un estudio del rendimiento obtenido sobre cuatro bases de datos biológicas al emplear diversos métodos de distancias para la reconstrucción de árboles a partir de las matrices procesadas durante la ejecución del algoritmo. Aunque los tiempos de procesamiento demuestran la complejidad reducida de las técnicas de clustering, las aproximaciones de NJ son las que permiten obtener los mejores resultados biológicos, especialmente cuando la complejidad de la instancia es considerable. La configuración basada en BIONJ permite a MO-FA obtener el mejor rendimiento

multiobjetivo, optimizando además los tiempos de ejecución para la instancia $RDPII_218$. Finalmente, la comparativa a nivel biológico con las propuestas de otros autores pone de manifiesto la bondad de la aproximación.

Como trabajo futuro, planteamos el desarrollo de nuevas metaheurísticas multiobjetivo aplicadas a filogenética. Los algoritmos que mejor se adapten al problema pueden cooperar mediante técnicas de programación paralela para crear un equipo de algoritmos que permitan mejorar la calidad de los resultados.

Agradecimientos

Este trabajo está parcialmente financiado por el Ministerio de Economía y Competitividad y el FEDER (Fondo Europeo de Desarrollo Regional), bajo el proyecto TIN2012-30685 (proyecto BIO). Sergio Santander-Jiménez es becario del programa FPU del Gobierno de España (FPU12/04101).

Referencias

1. Felsenstein, J.: Inferring phylogenies. Sinauer Associates, Sunderland, Massachusetts (2004)
2. Fogel, G. B.: Evolutionary computation for the inference of natural evolutionary histories. IEEE Connections 3(1), 11-14 (2005)
3. Lemey, P., Salemi, M., Vandamme, A.-M.: The Phylogenetic Handbook: a Practical Approach to Phylogenetic Analysis and Hypothesis Testing. Cambridge University Press, Cambridge (2009)
4. Coello, C., Veldhuizen, D. V., Lamont, G.: Evolutionary algorithms for solving multi-objective problems. En: Genetic Algorithms and Evolutionary Computation 5. Kluwer Academic Publishers (2002)
5. Coelho, G. P., da Silva, A. E. A., Von Zuben, F. J.: An Immune-Inspired Multi-Objective Approach to the Reconstruction of Phylogenetic Trees. Neural Computing and Applications 19(8), 1103-1132 (2010)
6. Cancino, W., Delbem, A. C. B.: A Multi-Criterion Evolutionary Approach Applied to Phylogenetic Reconstruction. En: Korosec, P. (ed.) New Achievements in Evolutionary Computation, pp. 135-156, InTech (2010)
7. While, L., Hingston, P., Barone, L., Huband, S.: A faster algorithm for calculating hypervolume. IEEE Transactions on Evolutionary Computation 10(1), 29-38 (2006)
8. Goëffon, A., Richer, J.M., Hao, J.K.: Progressive Tree Neighborhood Applied to the Maximum Parsimony Problem. IEEE/ACM Transactions on Computational Biology and Bioinformatics 5(1), 136-145 (2008)
9. Saitou, N., Nei, M.: The Neighbor-joining Method: A New Method for Reconstructing Phylogenetic Trees. Molecular Biology and Evolution 4(4), 406-425 (1987)
10. Gascuel, O.: BIONJ: An Improved Version of the NJ Algorithm Based on a Simple Model of Sequence Data. Molecular Biology and Evolution 14(7), 685-695 (1997)
11. Yang, X.-S.: Firefly Algorithm, Stochastic Test Functions and Design Optimisation. Int. J. Bio-Inspired Computation 2(2), 78-84 (2010)
12. Goloboff, P.A., Farris, J.S., Nixon, K.C.: TNT, a free program for phylogenetic analysis. Cladistics 24(5), 774-786 (2008)
13. Stamatakis, A.: RAxML-VI-HPC: Maximum Likelihood-based Phylogenetic Analyses with Thousands of Taxa and Mixed Models. Bioinformatics 22(21), 2688-2690 (2006)

Adaptando algoritmos evolutivos paralelos al lenguaje funcional Erlang

J. Albert-Cruz[1], L. Acevedo-Martínez[2], P.A. Castillo[3], M.G. Arenas[3], J.J. Merelo[3]

[1] Centro de Arquitecturas Empresariales, Universidad de Ciencias Informáticas, La Habana, Cuba
[2] Centro de Estudios de Matemática Computacional, Universidad de Ciencias Informáticas
[3] Dept. Arquitectura y Tecnología de los Computadores, Universidad de Granada, España
jalbert@uci.cu

Resumen En este artículo se describe el modelado e implementación de un esquema de computación evolutiva sobre los paradigmas de programación funcional y concurrente en el lenguaje Erlang. Aunque se particulariza en un lenguaje, la experiencia es posible extenderla a otros que poseen dichas características tales como Go, Scala y Haskell. El trabajo realizado expone las ventajas de estos paradigmas aplicados al modelado de un algoritmo genético paralelo híbrido, sobre una topología de isla. Se analizan algunas decisiones tomadas en el diseño y se muestran los resultados alcanzados en la solución de un caso de estudio.

Key words: Algoritmos Evolutivos, Lenguajes Funcionales, Lenguajes Concurrentes, Erlang, Implementación de Algoritmos, Modelación

1. Introducción

El campo de la Computación Evolutiva (CE) es rico en características, modelos y problemas pero no lo es igual en cuanto a paradigmas de programación a usar en la implementación de los algoritmos. Tecnologías como Java, C/C++ y Perl se mantienen como las más usadas y aunque se acepta la importancia de la implementación [1], no es mucho el interés de la comunidad científica en abordar nuevos lenguajes de programación por novedosos y prometedores que pudieran ser.

Dos paradigmas emergen actualmente en la industria del software como herramientas de abstracción ante viejos y nuevos problemas: el funcional y el concurrente. Aunque los lenguajes funcionales están a la mano desde hace buen tiempo (LISP es uno de los lenguajes de programación más longevos que existe) nunca han sido de mucho uso fundamentalmente por no disponerse de implementaciones eficientes. Esto ha ido cambiando, sus desventajas superándose y las ventajas llevando a que varios de sus conceptos sean incluidos en lenguajes modernos (ejemplo C#).

El desafío de los multi-núcleos [2], entendiéndolo como la necesidad de hacer paralelo el más simple de nuestros programas de manera que hagan uso de los modernos y potentes microprocesadores, lleva a crear (o retomar), patrones para el desarrollo de algoritmos paralelos. Dentro de dichos patrones algunos acaban como nuevas características en lenguajes de programación existentes y otros sirven como principio de diseño para lenguajes completamente nuevos. Entre los lenguajes emergentes con soporte al paradigma concurrente destacan Clojure, Go, Scala, Haskell y Erlang; en todos existen construcciones para manejar la concurrencia: facilitando razonar sobre ella y simplificando su sintaxis. El presente trabajo se encarga de explorar la idoneidad del lenguaje Erlang para la implementación de Algoritmos Genéticos Paralelos y pudiera servir como motivación para hacer lo mismo con los demás lenguajes mencionados.

El resto del trabajo se estructura como sigue: estado del arte, dentro del que se caracterizan los paradigmas funcional y concurrente y se describe el lenguaje Erlang que los incluye. A continuación se muestra la modelación e implementación de un algoritmo genético usando conceptos de los paradigmas antes expuestos (Sección 3); y finalmente, en la Sección 4, se presentan los resultados y conclusiones.

2. Estado del arte

Independientemente a la variedad de problemas a los que es aplicable la CE, en particular los Algoritmos Genéticos (AG), los conceptos en los que se basan son los mismos y así lo reflejan de una manera u otra las diferentes implementaciones realizadas. Lógicamente esto ha culminado en la creación de bibliotecas que encapsulan los elementos comunes según la variante arquitectónica que se necesite usar. A semejanza del resto del espectro de proyectos software, el paradigma dominante para la creación de tales bibliotecas es el Orientado a Objetos (OO).

2.1. Herramientas existentes

Son numerosas las herramientas desarrolladas para el trabajo con algoritmos evolutivos; en dependencia de su objetivo soportan diferentes plataformas, modelos de AE, integración con otros entornos y tamaños de los problemas. Entre las más representativas se encuentran: ECKit[4], JDEAL[5], ECJ[6], ParadisEO[7], EASEA[8] y MALLBA[9]. De estas, la más popular (al menos en número de usuarios y mención en la literatura científica) es ECJ, escrita en Java. Como se ve,

[4] http://cs.gmu.edu/ eclab/tools.html
[5] http://laseeb.isr.ist.utl.pt/sw/jdeal/home.html
[6] http://cs.gmu.edu/ eclab/projects/ecj/
[7] http://paradiseo.gforge.inria.fr/
[8] https://lsiit.u-strasbg.fr/easea/index.php/EASEA_platform
[9] http://neo.lcc.uma.es/software/mallba/index.php

en todos los casos se trata de lenguajes *tradicionales*, procedurales o, en su mayor parte, orientados a objetos. Una más completa revisión de estos entornos se puede consultar en [3].

ECJ ECJ consiste en un conjunto de herramientas para el trabajo con poblaciones, posee facilidades para la optimización paralela y la optimización multiobjetivo [4]. Entre otras formas de representación admite la programación genética. Está implementado en Java y soporta múltiples modelos de AG paralelos: modelo de islas y celular asíncrono, maestro-esclavo y distribución coevolutiva.

ParadisEO Marco de trabajo orientado a objetos escrito en C++ para el diseño de metaheurísticas paralelas y distribuidas, basado en Evolving Objects [5], incluye soporte para algoritmos evolutivos, búsquedas locales con soporte para los patrones paralelos y distribuidos más comunes [6]. Posee soporte solamente para los modelos clásicos de AG paralelos.

EASEA *EAsy Specification of Evolutionary Algorithms* es una plataforma de evolución artificial masivamente paralela desarrollada por SONIC (Stochastic Optimisation and Nature Inspired Computing). Posee un lenguaje propio en el que se se describe el algoritmo: estructura de los individuos, operador de inicialización, función de aptitud y los operadores de entrecruzamiento y mutación; a partir de esto la plataforma traduce a un conjunto de ficheros C++ que pueden además ser compilados para que usen las GPUs nVidia. Solo soporta el modelo de islas de pGA.

MALLBA MALLBA consiste en una biblioteca integrada de plantillas para optimización combinatoria: técnicas exactas, heurísticas e híbridas. Soporta ejecución tanto en ambientes secuenciales como paralelos, teniendo en cuenta el uso tanto de redes LANs como WANs [7]. Está implementada en C++ e incluye las versiones distribuidas y celulares de AG así como las técnicas CHC y μCHC.

2.2. Programación funcional

El paradigma de la programación funcional por su parte, aún cuando ofrece varias ventajas, no ha sido muy usado. Un tiempo atrás se exploraron en el campo de la Programación Genética [8,9,10], más recientemente en la neuroevolución [11]; sin embargo dentro de los AG ha sido poca su presencia [12].

La programación funcional se caracteriza por el uso de las funciones como datos (pasándolas por parámetros y devolviéndolas como resultados), en particular de las funciones puras: aquellas cuyo resultado solo depende de los parámetros de entrada, excluyendo los cambios de estado. Esto la hace particularmente adecuada para el desarrollo de algoritmos concurrentes pues estos tienen la primera fuente de errores y complejidad en la comunicación entre procesos, a través de cambios de estado.

El uso de listas, con implementaciones muy eficientes, es omnipresente en la programación funcional; en los AGs por su parte es una de las estructuras de datos más utilizadas, lo cual facilita el proceso de implementación de los diversos modelos de algoritmos evolutivos.

2.3. Erlang

Erlang es un lenguaje de programación funcional, concurrente y distribuido, adecuado para la construcción de sistemas que requieran grandes niveles de distribución [13], tolerancia a fallos y disponibilidad. Ha sido en más de una ocasión escogido por encima de C/C++ para el desarrollo de sistemas de uso intensivo de recursos [14] dada la eficiencia de su ejecución [15].

Utiliza el modelo *actor* para su implementación del paradigma de programación concurrente y posee facilidades para la integración con otros lenguajes tales como C/C++ y Java. Sus procesos son manejados por su máquina virtual (MV), que tiene un planificador por cada núcleo de la CPU, lo cual lo hacen ideal para la actual (y venideras) generación de procesadores multi-núcleos.

El que sea un lenguaje concurrente significa que posee entre sus tipos de datos el de proceso, siendo valores de primera clase, no facilidades proveídas por bibliotecas. Siendo en principio tan ligeros que una misma instancia de la MV puede tener a la vez millones en ejecución.

3. Diseñando AGs en un lenguaje funcional y concurrente

Para lograr una buena implementación de un algoritmo, cualquiera sea su naturaleza, es necesario tener en cuenta las características del lenguaje en el que se realizará así como cada concepto constituyente del dominio del problema. Se usará como caso de estudio un AG paralelo híbrido, sobre una topología de isla, en la que cada nodo será a su vez un AG concurrente basado en piscina. El lenguaje de programación será Erlang y el problema *OneMax*.

Los componentes del AG paralelo identificados como principales a la hora de diseñar la implementación aparecen listados en la Tabla 1. Las construcciones de Erlang utilizadas para la modelación son las expuestas en la Tabla 2 y el mapeo realizado es el mostrado en la Tabla 3.

Los principales conceptos concurrentes y funcionales utilizados son: actor, mensaje, función y lista. Su uso en la implementación de la arquitectura híbrida utilizada se basó en la flexibilidad y rapidez de implementación que permitiera. Para los procesos independientes existentes, ya fueran las islas o los evaluadores y reproductores se emplearon actores; para la comunicación entre ellos se usaron mensajes al ser estos el concepto indicado para comunicar actores. La lógica de funcionamiento del AG se expresó en funciones (medio funcional para la transformación de los datos) y las unidades de datos (cromosomas, poblaciones y configuraciones) se codificaron en listas y tuplas que son las estructuras de datos básicas del paradigma funcional.

Componente AG	**Papel**	**Descripción**
cromosoma	Representación de la solución al problema.	cadena binaria
cromosoma evaluado	Par {cromosoma, fitness}.	cantidad de valores 1
población	Conjunto de cromosomas.	lista
cruzamiento	Relación entre dos cromosomas que da por resultado otros dos nuevos.	función de cruzamiento
mutación	Modificación de un cromosoma.	función que altera un valor
selección	Criterio para obtener una sublista a partir de la población.	función de selección
pool	Población compartida entre las unidades de cálculo en un nodo.	población
isla	Nodo de la topología.	población
migración	Evento aleatorio de intercambio de cromosomas.	mensaje

Tabla 1: Componentes del AG paralelo.

Concepto Erlang	**Papel**
tupla	Tipo de datos para representar entes compuestos cuyas componentes sean de diferentes tipos y no varíen en el tiempo.
lista	Tipo de datos para representar entes compuestos cuyas componentes sean de igual tipo y varíen en el tiempo.
función	Relaciones entre datos, operaciones.
actor	Unidad de ejecución, proceso.
mensaje	Comunicación entre actores.
ets	Listado de cromosomas compartidos mediante el pool.
módulo random	Generación de números aleatorios.

Tabla 2: Construcciones de Erlang.

Concepto Erlang	**Concepto AG en el que se emplea**
tupla	cromosoma evaluado
lista	cromosoma y población
función	cruzamiento, mutación y selección
actor	isla, evaluador y reproductor
mensaje	migración
ets	pool
módulo random	naturaleza estocástica del AG

Tabla 3: Mapeo entre conceptos de Erlang y de AGs.

3.1. Caso de estudio

El objetivo de este trabajo es validar las facilidades de Erlang para implementar AGs, por ello hemos tomado un problema muy sencillo (*OneMax*): el simple conteo del número de unos en una cadena binaria, siendo la solución óptima la alcanzada cuando coincide la longitud de la lista con dicha cantidad. Dadas las características de la arquitectura diseñada y la posibilidad de correr nodos Erlang en disímiles dispositivos, la adaptación de este trabajo a otros problemas más complejos solamente requeriría redefinir las funciones de evaluación, cruce y mutación.

3.2. Biblioteca erlEA

En la realización del caso de estudio fue implementado, teniendo en cuenta los conceptos seleccionados anteriormente, un proyecto Erlang de varios módulos. El código se encuentra bajo la licencia AGPL, de código abierto, en la dirección: `https://github.com/jalbertcruz/erlEA/tree/MAEB`. Sus principales módulos y funciones son descritos a continuación.

Módulo reproducer Este es el módulo que selecciona la subpoblación a reproducir, los padres, realiza el cruzamiento y desencadena las migraciones. Como actor responde a los mensajes *evolve*: para realizar una iteración y *emigrateBest* para efectuar una emigración. Las funciones con las que logra esto aparecen enumeradas en la Tabla 4.

Función	**Descripción**
`extractSubpopulation(Table, N)`	A partir de una *ets* y una cantidad, selecciona de la *ets* un grupo de cromosomas.
`bestParent(Pop2r)`	Selecciona de una lista de cromosomas el mejor individuo.
`selectPop2Reproduce(Pop, N)`	Selecciona aleatoriamente un conjunto de pares de una lista de cromosomas.
`crossover(Ind1, Ind2)`	Realización de un cruce y mutación sobre el mismo, a partir de dos cromosomas.

Tabla 4: Funciones del módulo reproducer.

Módulo evaluator Este es el módulo que calcula el fitness: hace periódicas consultas sobre el pool para obtener individuos a los que calcularle el fitness. Está compuesto por la función *maxOnes/1* (función de evaluación), y por el mensaje *eval*, dicho mensaje es el que activa al evaluador para que calcule.

Módulo poolManager Este es el módulo encargado de inicializar el trabajo del pool así como enrutar los mensajes entre los evaluadores. Es el encargado de controlar la finalización del algoritmo una vez se ha encontrado la solución. Los mensajes a los que responde este actor aparecen enumerados en la Tabla 5.

Mensaje	Descripción
`evolveDone`	Finalización de una iteración de reproducción.
`evalDone`	Finalización de una iteración de evaluación.
`solutionReached`	Obtención de la solución.
`migration`	Realización de una inmigración.

Tabla 5: Mensajes a los que responde el actor del módulo poolManager.

Módulos auxiliares e interconexión Los módulos ya descritos contienen toda la lógica del AG, faltan sin embargo, para que sea operativo el software, algunos componentes no funcionales.

Dichos componentes son:

experiment – Encargado de iniciar una corrida del experimento.
configBuilder – Especificación de los parámetros de un experimento.
profiler – Análisis del comportamiento: tiempos de ejecución, cantidad de iteraciones, etc.
manager – Control del inicio y coordinación de la finalización adecuada de cada unidad de ejecución durante la corrida de un experimento.
report – Control de la secuencia de experimentos y emisión del reporte final resultado de la experimentación.

Los diferentes módulos interactúan de la siguiente manera durante una sesión de experimentación:

1. Configuración de los parámetros: tamaño de población y cromosomas, población inicial, conformación de la topología de las islas. Módulos *experiment* y *configBuilder*.
2. Lógica del AG: reproducción, selección de padres, emigraciones, cálculo del fitness. Módulos *evaluator* y *reproducer*.
3. Control del proceso y reportes: enruteo de mensajes entre islas y unidades de ejecución (actores), coordinación de los procesos de finalización de una corrida e inicio de otra, resumen y formateo de salida de los resultados. Módulos *poolManager*, *profiler*, *manager* y *report*.

4. Resultados, conclusiones y trabajo futuro

El diseño fue puesto a prueba con cromosomas de longitud 128, una población de 256 individuos por isla y con evaluadores y reproductores trabajando con

bloques de 50 cromosomas. Los experimentos fueron realizados en un ordenador con Windows 8 y 16 Gb de RAM así como con un procesador Core i7. Los resultados obtenidos aparecen en la Tabla 6, correspondiendo a evaluaciones independientes no promediadas cuya condición de parada es la obtención de la solución óptima.

No.	**Islas**	**Evaluadores**	**Reproductores**	**Migraciones**	**Reproducciones**	**Tiempo (s)**
1	2	5	10	215	416	11.62375
2	2	10	20	299	629	25.3412
3	4	5	10	347	664	10.672001
4	4	10	20	580	1223	27.09117
5	8	5	10	862	1635	16.375004
6	8	10	20	1333	2845	36.294951

Tabla 6: Resultados de los experimentos.

Aunque se trata solo de resultados iniciales, ya se puede ver que el escalado no es bueno, siendo en todo caso mejor cuanto menor es el número de evaluadores. Sin embargo, también se observa que el aumento en el tiempo se debe más al aumento en el número de reproducciones que al aumento en el número de islas, obteniendo, por ejemplo, en el caso de 8 islas un número de reproducciones superior al obtenido para 4 islas con un tiempo un 40 % aproximadamente menor. Esto apunta al hecho de que el escalado se debería medir más sobre la mejora en la obtención de soluciones que en el tiempo. Sin embargo, como se ha comentado en esta primera etapa de la investigación nos hemos centrado más en la adaptación del algoritmo que en las posibilidades de escalado, que se abordarán como trabajo futuro.

Con este trabajo se muestra la simplicidad de implementación de un modelo híbrido de AG, en su versión concurrente, normalmente muy compleja sencillamente por ser concurrente. En la arquitectura las unidades de ejecución fueron mapeadas a actores, las estructurales a módulos y las procedurales a funciones; quedando claramente identificadas y listas a futuras extensiones.

Como trabajo futuro queda la implementación de un experimento más complejo dónde se use además una arquitectura distribuida y heterogénea. Dado que el lenguaje tiene de manera nativa el soporte para la distribución de procesos y su MV está implementada para multitud de plataformas hacer esto sólo conllevaría cambios en las funciones que tratan con los cromosomas lográndose un elevado nivel de reutilización del diseño.

5. Agradecimientos

Este trabajo se está desarrollando gracias a la financiación de los proyectos EvOrq (P08-TIC-3903), CANUBE (CEI2013-P-14) y ANYSELF (TIN2011-

28627-C04-02). Es también respaldado por el Programa de Doctorado de la AUIP y por el proyecto 83 del Campus CEI BioTIC.

Referencias

1. Merelo-Guervós, J.J., Romero, G., García-Arenas, M., Castillo, P.A., Mora, A.M., Jiménez-Laredo, J.L.: Implementation matters: Programming best practices for evolutionary algorithms. In Cabestany, J., Rojas, I., Caparrós, G.J., eds.: IWANN (2). Volume 6692 of Lecture Notes in Computer Science., Springer (2011) 333–340
2. Herb Sutter and James R. Larus: Software and the concurrency revolution. ACM Queue **3**(7) (2005) 54–62
3. Parejo, J.A., Ruiz-Cortés, A., Lozano, S., Fernández, P.: Metaheuristic optimization frameworks: A survey and benchmarking. Soft Computing **16**(3) (2011) 527–561
4. Luke, S.: Essentials of Metaheuristics. Zeroth Edition (2010)
5. Keijzer, M., Merelo, J.J., Romero, G., Schoenauer, M.: Evolving objects: A general purpose evolutionary computation library. Artificial Evolution **2310** (2002) 829–888
6. Liefooghe, A., Jourdan, L., Talbi, E.: A software framework based on a conceptual unified model for evolutionary multiobjective optimization: ParadisEO-MOEO. European Journal of Operational Research (2010)
7. Alba, E., Almeida, F., Blesa, M., Cotta, C., Díaz, M., Dorta, I., Gabarró, J., León, C., Luque, G., Petit, J., Rodríguez, C., Rojas, A., Xhafa, F.: Efficient parallel LAN/WAN algorithms for optimization. the MALLBA project. Parallel Computing **32**(5-6) (2006) 415–440
8. Briggs, F., O'Neill, M.: Functional genetic programming and exhaustive program search with combinator expressions. Int. J. Know.-Based Intell. Eng. Syst. **12**(1) (2008) 47–68
9. Huelsbergen, L.: Toward simulated evolution of machine-language iteration. In: Proceedings of the First Annual Conference on Genetic Programming. GECCO '96, Cambridge, MA, USA, MIT Press (1996) 315–320
10. Walsh, P.: A functional style and fitness evaluation scheme for inducting high level programs. In Banzhaf, W., Daida, J., Eiben, A.E., Garzon, M.H., Honavar, V., Jakiela, M., Smith, R.E., eds.: Proceedings of the Genetic and Evolutionary Computation Conference. Volume 2., Orlando, Florida, USA, Morgan Kaufmann (1999) 1211–1216
11. Sher, G.I.: Handbook of Neuroevolution Through Erlang. Springer (2013)
12. Hawkins, J., Abdallah, A.: A generic functional genetic algorithm. In: Proceedings of the ACS/IEEE International Conference on Computer Systems and Applications. AICCSA '01, Washington, DC, USA, IEEE Computer Society (2001) 11–
13. Veldstra, H.: Why Erlang is awesome. (2013)
14. Cesarini, F., Thompson, S.: Erlang Programming. O'Reilly Media, Inc. (2009)
15. Katz, D.: Erlang, the language from the future? InfoQ (2012)

Desarrollo de servicios para una Arquitectura Orientada a Servicios para Algoritmos Evolutivos

P. García-Sánchez, M. García-Arenas, A. M. Mora, P.A. Castillo, C. Fernandes, J. González, G. Romero, J.J. Merelo y P. de Las Cuevas

[1] Dept. de Arquitectura y Tecnología de Computadores, Universidad de Granada
[2] LaSEEB-ISR-IST, Technical University of Lisbon (IST), Lisboa, Portugal
pgarcia@atc.ugr.es

Resumen Este artículo muestra el diseño e implementación de servicios para Computación Evolutiva dentro del paradigma de la Arquitectura Orientada a Servicios. Este paradigma proporciona independencia en el lenguaje y comunicación, pero el desarrollo requiere tratar con algunas cuestiones tecnológicas y de diseño, como un diseño lo más abstracto posible o la ejecución desordenada. Para resolverlas, se utiliza OSGi-Liath, una implementacion de una Arquitectura Orientada a Servicios para Algoritmos Evolutivos, para desarrollar nuevos servicios interoperables teniendo en cuenta esas restricciones.

1. Introducción

El paradigma Service Oriented Architecture (SOA) o de Arquitectura Orientada a Servicios (AOS) [1] se está convirtiendo en una importante tendencia en el desarrollo de software. Este paradigma permite la organización y distribución utilizando el concepto de *servicio*, es decir, una interacción en la que un *proveedor de servicio* publica *descripciones de servicio* (o interfaces) en el *registro de servicios*, para que los *consumidores de servicio* puedan encontrarlo y enlazarse con los proveedores para usarlo.

SOA permite independencia en el lenguaje de programación y los mecanismos de distribución y de comunicación, centrándose en la facilidad de extensión e integración, pero cuenta con las siguientes restricciones:

- Los servicios deben ser funciones de entrada/salida.
- Los servicios no deben tener estado (es decir, no usar variables globales).
- El orden de ejecución de los servicios no es fijo.
- Los servicios deben ser diseñados con un nivel de abstracción adecuado.

La computación distribuida ofrece la posibilidad de utilizar procesamiento paralelo y así conseguir un mayor poder de cómputo [2]. SOA también puede aplicarse a este área, utilizando plataformas basadas en Web Services [1], o nuevos estándares para este paradigma, como OSGi (Open Services Gateway Initiative) [3].

OSGi permite construir sistemas software de calidad considerando un alto nivel de modularidad. Además de los beneficios que los paradigmas clásicos de modularización ofrecen (como el modelado orientado a objetos) y las mejoras en testeo, reusabilidad, disponibilidad y mantenibilidad, es necesario explorar otras técnicas, como el desarrollo basado en plug-ins o el diseño SOA. Estos tipos de desarrollo simplifican aspectos como la complejidad, personalización, configuración, y coste del desarrollo. En el campo de las heurísticas de optimización los beneficios de usar este tipo de desarrollo tienen lugar en la creación de algoritmos, evaluación experimental y combinación de diferentes paradigmas de optimización [4].

En un trabajo anterior [5] presentamos una Arquitectura Orientada a Servicios para Algoritmos Evolutivos (SOA-EA), junto con guías y pasos para migrar del desarrollo tradicional de Algoritmos Evolutivos (AEs) a SOA. También se presentó una implementación específica, llamada OSGiLiath (*OSGi Laboratory for Implementation and Testing of Heuristics*): un entorno para el desarrollo de algoritmos distribuidos extensible con una arquitectura basada en plug-ins y basada en una especificación software ampliamente aceptada (OSGi). En este trabajo mostramos el desarrollo completo de un servicio utilizando la tecnología específica, en lugar del diseño abstracto presentado en el anterior trabajo.

El resto del trabajo se estructura como sigue: después del estado del arte, presentamos los principios de diseño para crear servicios para Computación Evolutiva (CE) (Sección 3). A continuación, se explica la tecnología de implementación OSGi en la Sección 4, utilizada para construir nuestro *framework* (descrito en la Sección 5). Después, se presentan los pasos para crear servicios dentro de este framework (Sección 6). Finalmente se presentan las conclusiones y trabajo futuro (Sección 7).

2. Estado del arte

Aunque SOA se usa ampliamente en el desarrollo de software no está muy aceptada en la comunidad científica de la optimización y metaheurísticas. La mayoría de los frameworks (ver [6] para un resumen) tienen carencia de la baja generalidad, ya que están enfocados en un campo específico, como por ejemplo, la Búsqueda Local. Otro problema común es que suelen ser simplemente librerías o módulos Perl [7], no tienen GUIs o son muy complicados de instalar y requieren muchas destrezas de programación.

Entre la gran cantidad de herramientas software que existen, queremos centrarnos en los frameworks de algoritmos evolutivos distribuidos más aceptados. ECJ [8], Evolutionary Computation in Java, es un conjunto de clases Java que pueden ser extendidas e incluyen varios módulos de comunicación. MALLBA [9] se basa en esqueletos software con una interfaz común pública. Cada esqueleto implementa una técnica de resolución para optimización en el campo de optimización exacta, heurística o híbrida. Proporciona capacidades de distribución utilizando MPI. Sin embargo estos dos frameworks no se basan en el desarrollo orientado a plug-ins, por lo que no pueden tomar las ventajas de característi-

cas como gestión del ciclo de vida, versionado o enlazado dinámico de servicios, como propone OSGi.

Otra plataforma importante es DREAM [10], un framework para Algoritmos Evolutivos basado en Java que define un modelo de isla y usa el protocolo Gossip y sockets TCP/IP para comunicación. Puede desplegarse en plataformas P2P y está dividido en cinco capas. Cada capa proporciona una interfaz de usuario y diferentes niveles de interacción y abstracción, pero añadir nuevas funcionalidades no es tan fácil, debido al hecho de que el sistema debe pararse antes de añadir nuevos módulos y la implementación de interfaces debe estar definida en el código fuente, por lo que necesita compilarse con cada elemento a añadir (como en ECJ). OSGi permite añadir nuevas funcionalidades solamente compilando las nuevas características y no las ya existentes. jMetal [11], también es un framework basado en Java, pero sin posibilidad de distribución y mayormente centrado en optimización Multi-Objetivo.

ParadiseEO [12] permite el diseño de AEs y Búsqueda Local con hibridación, proporcionando una gran variedad de operadores y funciones de evaluación. También implementa los modelos paralelos y distribuidos más comunes, y está basado en librerías estándar como MPI, PVM y Pthreads. Sin embargo, cuenta con los mismos problemas que los frameworks anteriores, no cuenta con gestión del ciclo de vida, ni programación orientada a servicios. GAlib [13] es muy similar y comparte las mismas características y problemas.

En el campo de los frameworks basados en plug-ins, HeuristicLab [14] es el ejemplo más avanzado. Permite además programación distribuida utilizando Servicios Web y una base de datos centralizada, pero no usa su propio diseño de plug-ins para la comunicación distribuida. Finalmente, el único framework orientado a servicios para optimización es GridUFO [15], pero sólo permite la modificación de la función objetivo y añadir nuevos algoritmos, sin poder combinar servicios existentes.

Los frameworks anteriores han sido diseñados para ser extensibles y reusables, pero sin tener en cuenta las restricciones de SOA para lograr más independencia y mejoras en el desarrollo. A continuación explicamos cuales deben ser los requisitos para poder cumplir con esas restricciones.

3. Diseñando servicios para una Arquitectura Orientada a Servicios para AEs

En [5] demostramos que es posible crear una arquitectura orientada a servicios para AEs utilizando una tecnología SOA específica. Esta arquitectura utiliza las capacidades que SOA ofrece. Para hacer esto, se diseñaron servicios débilmente acoplados para EAs (SOA-EA), y se implementaron utilizando una tecnología SOA y se compararon con otros frameworks. Estos servicios pueden combinarse de varias formas para obtener diferentes algoritmos (por ejemplo, de un Algoritmo Genético (AG) canónico se puede crear un NSGA-II simplemente añadiendo nuevos servicios). También se presentaron varias técnicas para combinar servicios de forma flexible.

3.1. Principios de diseño

Una de las restricciones principales de SOA, además de el enfoque en servicios abstractos es la naturaleza sin estado de los servicios. Por lo tanto, en SOA el diseño de servicios debe seguir algunas pautas.

Para empezar, debido a que los servicios no tienen por qué tener conocimiento de otros servicios, no debe haber variables globales en ninguna parte del código. Los servicios están escuchando y esperando a ser ejecutados. Por ejemplo, debe evitarse una función fitness con un contador que se incremente cada vez que se llama (para parar el algoritmo al llegar a un límite). Si varios (y diferentes) algoritmos están trabajando en paralelo, y llamando a esta función a la vez el contador no distinguiría entre algoritmos, ofreciendo resultados erróneos. Sin embargo, un servicio que mantiene algún tipo de estado se permite, por ejemplo, un servicio de estadísticas que lee eventos de todos los algoritmos que se ejecutan a la vez, pero debe ser administrado para evitar errores.

Además, un servicio debe ser indistinguible de ser ejecutado de forma local o remota en otro nodo de la red. Por lo tanto, cada etapa del algoritmo debe ser tratada como servicio a ser ejecutado en local o en remoto, incluso un servicio de *Población* o *Parámetros*. Deben proveerse mecanismos correctos para el intercambio de datos. Además, muchas implementaciones del mismo servicio pueden existir a la vez (por ejemplo, diferentes implementaciones del servicio *Crossover*), y deberían ser manejados y usados correctamente.

Un servicio es siempre una función solicitud-respuesta. Por ejemplo, el cálculo del fitness no debe ser un método de la implementación *Individuo*, como en la mayoría de los frameworks, sino una función que recibe una lista de individuos y devuelve una lista de fitness de esos individuos. Esto permite opciones como cálculo remoto del fitness y balanceo de carga distribuido, más difícil de realizar si el fitness es un método de la clase Individuo.

Pensar lo más abstractamente posible requiere separar conceptos como el orden de recombinación, y el propio crossover. Normalmente, después de la selección de los padres, los individuos se cruzan en orden. Sin embargo, si necesitamos un mecanismo de emparejamiento distinto (por ejemplo, usar más de un padre, o seleccionar un padre más de dos veces) se necesita una duplicación de esfuerzo para integrar nuevo código. Esta es la razón por la que se deben separar conceptos como *recombinación* de *crossover*.

Finalmente, no debemos asumir el orden de los servicios a ejecutar. Por ejemplo, servicios como *Recombinador* or *Mutator* deberían devolver los individuos con el fitness ya calculado. Normalmente este paso se realiza en la última etapa de la generación, pero podría ser necesario obtenerlo anteriormente para ser usados en otras tareas, por ejemplo una búsqueda local o un recolector de estadísticas para guiar al algoritmo.

3.2. Otras restricciones tecnológicas

En [5] presentamos las ventajas de usar SOA en algoritmos Evolutivos: para empezar, SOA encaja con las ventajas de generalidad en el desarrollo de AEs

explicado en [16], pero añade nuevas características, como independencia del lenguaje y mecanismos de distribución. También permite añadir y eliminar servicios en tiempo de ejecución sin alterar la ejecución general del algoritmo (es decir, no es obligatorio pararlo o añadir código extra para soportar nuevos operadores). Esto permite incrementar la interoperabilidad entre diferentes elementos software. Además, esto permite distribución del código de manera fácil: SOA no requiere de una implementación o librería de distribución concreta. Los servicios a desarrollar deben cumplir las siguientes restricciones tecnológicas:

- Estos servicios pueden ser enlazados dinámicamente para cambiar los aspectos necesarios del AE.
- El código fuente de los servicios básicos no debe ser reescrito o recompilado para conseguir esta tarea.
- Nuevos servicios pueden añadirse en tiempo de ejecución.
- No hace falta código específico para distribución, ni el código fuente de los servicios debe modificarse para este propósito. Es decir, cambiar las librerías de distribución no debe añadir código extra en los servicios existentes.

4. Tecnología de implementación

Esta sección explica algunas características técnicas de la tecnología escogida para implementar una SOA para AEs y así guiar al lector para entender el framework OSGiLiath y poder evaluar las ventajas de utilizar estas características en el desarrollo de algoritmos distribuidos que cumplan las restricciones anteriores.

La tecnología utilizada, OSGi, define una especificación para SOAs en máquinas virtuales. Proporciona características muy deseables, como abstracción de paquetes, administración del ciclo de vida o versionado, permitiendo reducción en el tiempo de desarrollo, soporte, y complejidad en el despliegue de las aplicaciones. Esta tecnología permite descubrimiento dinámico de nuevos componentes, para incrementar la colaboración y minimizar y administrar el acoplamiento entre módulos. Además, ya existen interfaces estándar para patrones muy usados, como servidores HTTP, configuración, registros, seguridad o administración de XML.

OSGi cuenta con un modelo orientado a capas, que explican su funcionalidad. Cada una se basa en la capa siguiente:

- *Bundles*: son los componentes OSGi creados por los desarrolladores (aplicaciones).
- Servicios: Esta capa conecta los bundles de forma dinámica ofreciendo un modelo de publicación-búsqueda-enlace.
- Ciclo de vida: La API para instalar, iniciar, parar, actualizar y desinstalar *bundles*.
- Módulos: Esta capa define cómo un bundle puede importar y exportar código.
- Seguridad: los aspectos de seguridad se administran en esta capa.
- Entorno de ejecución: Define qué metodos y clases están disponibles en una plataforma específica. Por ejemplo, los dispositivos móviles cuentan con menos clases Java debido a restricciones de rendimiento.

Para cumplir la restricción de que en un buen framework para EAs los servicios deben de ser indistinguibles de ser locales o remotos se han utilizado otras características OSGi. Se ha elegido ECF (Eclipse Communication Framework)[3] por ser la implementación más madura y aceptada [17], y porque soporta el mayor número de protocolos de transmisión, incluyendo comunicación síncrona y asíncrona. Proporciona una implementación modular del estándar de Servicios Remotos de OSGi V4.2 [4]. Esta especificación utiliza el registro de servicios de OSGi para exponer servicios como remotos (siendo indistinguibles de los locales). ECF también separa el código fuente del mecanismo de descubrimiento y transmisión, permitiendo que los usuarios apliquen la tecnología más adecuada a sus necesidades y proporcionando integración con aplicaciones existentes.

5. OSGiLiath

Utilizando tecnologías como OSGi o ECF se puede crear un entorno orientado a servicios. Esta sección explica la funcionalidad y diseño del entorno OSGiLiath, presentado en [5]. Este entorno es un framework para el desarrollo de aplicaciones de optimización utilizando metaheurísticas, no centrándose en un paradigma concreto, y cuyo principal objetivo es promover el uso de SOA y OSGi, y ofrecer a los programadores las siguientes características:

- Interfaces fáciles. Después del estudio de los frameworks anteriores se ha desarrollado una jerarquía de interfaces para usar.
- Envío/recepción de datos asíncronos. Gracias a las posibilidades de distribución con OSGi, este framework cuenta con distribución fácil de servicios, sin implementar funciones específicas para esta tarea, al contrario que otros frameworks de distribución, como MPI.
- Programación orientada a componentes. El framework está orientado a plugins, por lo que se pueden añadir nuevas mejoras de forma fácil sin modificar los módulos existentes. Añadir o modificar implementaciones de servicios se puede realizar sin recompilar el código fuente.
- Cliente/servidor o modelo distribuido. Todos los componentes del framework pueden comunicarse de forma bidireccional, por lo que no hace falta un administrador central si no se requiere.
- Independiente del paradigma. Este framework no está enfocado en un tipo de metaheurística.
- Servicios declarativos. Enlazar interfaces a distintas implementaciones se puede realizar sin modificar el código fuente existente. Los programadores no necesitan instanciar implementaciones de los servicios.
- Manejo de eventos remoto: utilizando las ventajas de OSGi, los usuarios pueden usar una herramienta muy potente para sincronizar o compartir datos entre servicios.

El código fuente está disponible en `http://www.osgiliath.org` bajo una licencia LGPL.

[3] `http://www.eclipse.org/ecf/`
[4] `http://www.osgi.org/Release4/Download`

6. Desarrollo de servicios en OSGiLiath

Esta sección explica los pasos para añadir servicios al núcleo de OSGiLiath. En esta sección se explica cómo añadir el problema del Vehicle Routing Problem (VRP).

6.1. Creación de un bundle

En OSGiLiath, los servicios se pueden añadir a bundles existentes o crear nuevos. Cada *bundle* incluye un fichero MANIFEST.MF . En este fichero se seleccionan los paquetes que importar (incluyendo interfaces y clases del núcleo de OSGiLiath) y exportar. La sección *Service-Description* en este fichero muestra la localización de Component Definitions para describir los servicios. En este caso se implementan dos interfaces: *TransportData* y *FitnessCalculator*. Otras clases relacionadas con el VRP se añaden, como *Route* o *Shop*.

6.2. Implementando servicios

Para implementar un servicio se crea una clase que implementa una interfaz. Por ejemplo, *VRPFitnessCalculator* implementa la interfaz *FitnessCalculator*. La relación entre estos dos elementos se hace en el Component Definition de la Figura 1. De esta forma, la implementación se anuncia a los otros servicios en el entorno, que pueden enlazarse o desenlazarse. Por ejemplo, la implementación *VRPInitializer* (implementando *Initializer*) requiere esta implementación para crear los individuos. Los servicios pueden enlazarse automáticamente con otros servicios con los métodos set/unset en el Component Definition. Otros servicios fuera del AE pueden añadirse (por ejemplo, en este bundle el servicio *TransportData*, que incluye información sobre distancias y tiempo a los nodos). Finalmente, se añaden *VRPMutation* and *VRPCrossover* siguiendo los consejos proporcionados en la Sección 3.

6.3. Añadiendo comunicación

Gracias a la especificación OSGi 4.2, los servicios pueden y deben ser indistinguibles de los existentes en un entorno OSGi respecto a uno externo (en la misma máquina o en la red). Para conseguir esto, se utiliza ECF para exportar los servicios. En este caso, hemos creado un servicio de migración. Este servicio tiene dos operaciones: enviar y leer. El primero se usa para enviar los individuos al exterior y el segundo para leer los recibidos. Normalmente cada isla deberá tener un migrador para recibir individuos, y referencias a los migradores de otras islas. En nuestro caso, la implementación de *Replacer* enlaza el *Migrator* local para escribir en él los individuos a enviar. Un ejemplo de implementación del migrador es *MigratorRingBuffer*: esta clase implementa esa interfaz y automáticamente enlaza con todos los migradores disponibles en el entorno (cuenta con un vector de referencias) gracias a los métodos *bind* y *unbind* de los servicios

```
<?xml version="1.0" encoding="UTF-8"?>
<scr:component xmlns:scr="http://www.osgi.org/xmlns/scr/v1.1.0"
name="VRPFitnessCalculator">
   <implementation class="es.ugr.osgiliath.vrp.VRPFitnessCalculator"/>
   <service>
      <provide interface="es.ugr.osgiliath.evolutionary.elements.FitnessCalculator"/>
   </service>
   <reference bind="setTransportData"
    unbind="unsetTransportData"
    cardinality="1..1"
    interface="es.ugr.osgiliath.vrp.TransportData"
    name="TransportData"
    policy="static"/>
   <property name="name" type="String" value="vrpfitnesscalculator"/>
</scr:component>
```

Figura 1: Service Description. Este documento indica que la implementación del servicio *FitnessCalculator* es *VRPFitnessCalculator*, pero no se puede activar hasta que todas sus referencias (otros servicios) se activen.

declarativos. Por lo tantos, los migradores pueden ser añadidos en tiempo de ejecución y no pasa nada si un nodo cae. El *MigratorRingBuffer* envía individuos al Migrador remoto cuya ID es inmediatamente superior que la ID local (un anillo). La Figura 2 muestra esta configuración. Se pueden añadir algunas propiedades al servicio para que ECF automáticamente anuncie la implementación a todos los nodos de la red, por lo que no hace falta cambiar el código para cambiar de un mecanismo de distribución a otro.

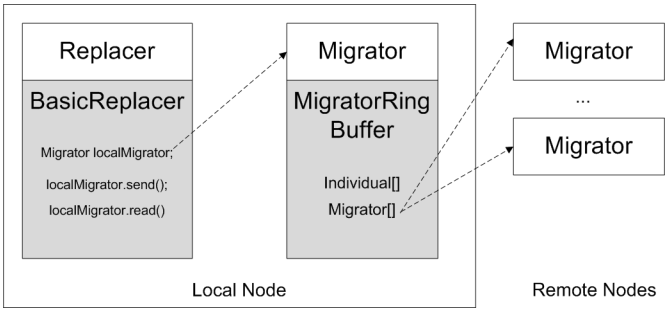


Figura 2: Utilizando el servicio *Migrator* para crear un AE distribuido basado en islas con topología en anillo (las cajas blancas son interfaces de los servicios y las grises son las implementaciones).

7. Conclusiones

Este trabajo muestra los requisitos para crear un framework orientado a servicios para AEs y la tecnología escogida para cumplir esos requisitos. Service

Oriented Architecture (SOA) ofrece independencia del lenguaje, mecanismos de distribución o incluso sistemas operativos, permitiendo la integración de diferentes elementos. Sin embargo, algunos elementos deben considerarse en el desarrollo: los servicios son funciones de entrada/salida sin estado, los servicios pueden aparecer o desaparecer en tiempo real, y el orden de ejecución no debe estar fijo. En el campo de los AEs los servicios deben desarrollarse teniendo en cuenta estas restricciones. Este trabajo muestra el diseño abstracto de los elementos de un AE, junto con los requisitos tecnológicos a resolver. Se ha utilizado la tecnología OSGi como ejemplo. Los elementos para crear una SOA para AEs han sido presentados junto con un ejemplo de desarrollo.

Como trabajo futuro, se pretende realizar un estudio sobre escalabilidad utilizando otros algoritmos (como GRASP, Scatter Search, Ant Colony Optimization y otros). Además está planeado incrementar el uso de las ventajas de OSGi, como la administración de eventos o la gestión de servicios de forma más profunda. Finalmente, se plantea desarrollar un portal o un repositorio Maven[5] para centralizar todas las implementaciones de problemas y algoritmos para permitir la distribución junto con la plataforma base. También se plantea un estudio sobre portar software existente (especialmente los frameworks escritos en Java, como DREAM o ECJ) a nuestro framework. Además, debido a la facilidad de enlazar implementaciones a interfaces, se plantea desarrollar la funcionalidad de elegir una u otra implementación dependiendo de varios parámetros, o por ejemplo, utilizar Programación Genética para evolucionar e hibridizar algoritmos.

Agradecimientos

Este trabajo se está desarrollando gracias a la financiación del proyecto FEDER de la Unión Europea con título "Sistema de Información y Predicción de bajo coste y autónomo para conocer el Estado de las Carreteras en tiempo real mediante dispositivos distribuidos" (SIPEsCa) del Programa Operativo FEDER de Andalucía 2007-2013 y la beca FPU AP2009-2942 y los proyectos EvOrq (TIC-3903), CANUBE (CEI2013-P-14) y ANYSELF (TIN2011-28627-C04-02).

Referencias

1. Papazoglou, M., van den Heuvel, W.J.: Service oriented architectures: approaches, technologies and research issues. The VLDB Journal **16** (2007) 389–415

[5] `http://www.maven.org`

2. Altunay, M., Avery, P., Blackburn, K., Bockelman, B., Ernst, M., et al.: A Science Driven Production Cyberinfrastructure-the Open Science Grid. Journal of GRID Computing **9**(2, Sp. Iss. SI) (2011) 201–218
3. OSGi Alliance: OSGi service platform release 4.2 (2010) Available at: `http://www.osgi.org/Release4/Download`.
4. Wagner, S., Winkler, S., Pitzer, E., Kronberger, G., Beham, A., Braune, R., Affenzeller, M.: Benefits of plugin-based heuristic optimization software systems. In Moreno Díaz, R., Pichler, F., Quesada Arencibia, A., eds.: Computer Aided Systems Theory - EUROCAST 2007. Volume 4739 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2007) 747–754
5. García-Sánchez, P., González, J., Castillo, P.A., Arenas, M.G., Guervós, J.J.M.: Service oriented evolutionary algorithms. Soft Comput. **17**(6) (2013) 1059–1075
6. Parejo, J.A., Cortés, A.R., Lozano, S., Fernandez, P.: Metaheuristic optimization frameworks: a survey and benchmarking. Soft Comput. **16**(3) (2012) 527–561
7. Merelo Guervós, J., Castillo, P., Alba, E.: Algorithm::evolutionary, a flexible Perl module for evolutionary computation. Soft Computing - A Fusion of Foundations, Methodologies and Applications **14** (2010) 1091–1109
8. Luke, S., et al.: ECJ: A Java-based Evolutionary Computation and Genetic Programming Research System (2009) Available at `http://www.cs.umd.edu/projects/plus/ec/ecj`.
9. Alba, E., Almeida, F., Blesa, M., Cotta, C., Díaz, M., Dorta, I., Gabarró, J., León, C., Luque, G., Petit, J., Rodríguez, C., Rojas, A., Xhafa, F.: Efficient parallel LAN/WAN algorithms for optimization. The MALLBA project. Parallel Computing **32**(5-6) (2006) 415–440
10. Arenas, M., Collet, P., Eiben, A., Jelasity, M., Merelo, J.J., Paechter, B., Preuß, M., Schoenauer, M.: A framework for distributed evolutionary algorithms. In: Parallel Problem Solving from Nature, PPSN VII. (2002) 665–675
11. Durillo, J.J., Nebro, A.J., Alba, E.: The jmetal framework for multi-objective optimization: Design and architecture. In: IEEE Congress on Evolutionary Computation. (2010) 1–8
12. Liefooghe, A., Jourdan, L., Talbi, E.: A software framework based on a conceptual unified model for evolutionary multiobjective optimization: ParadisEO-MOEO. European Journal of Operational Research (2010)
13. Wall, B.M.: A genetic algorithm for resource-constrained scheduling, Ph.D. thesis, MIT. (1996) Available at: `http://lancet.mit.edu/ga`.
14. Wagner, S., Affenzeller, M.: HeuristicLab: A generic and extensible optimization environment. In Ribeiro, B. and Albrecht, R.F. and Dobnikar, A. and Pearson, D.W. and Steele, N.C., ed.: Adaptive and Natural Computing Algorithms. Springer Computer Science (2005) 538–541 7th International Conference on Adaptive and Natural Computing Algorithms (ICANNGA).
15. Munawar, A., Wahib, M., Munetomo, M., Akama, K.: The design, usage, and performance of gridufo: A grid based unified framework for optimization. Future Generation Computer Systems **26**(4) (2010) 633 – 644
16. Gagné, C., Parizeau, M.: Genericity in evolutionary computation software tools: Principles and case-study. International Journal on Artificial Intelligence Tools **15**(2) (2006) 173
17. Petzold, M., Ullrich, O., Speckenmeyer, E.: Dynamic distributed simulation of DEVS models on the OSGi service platform. Proceedings of ASIM 2011 (2011)

VisualJCLEC: un entorno de trabajo gráfico para computación evolutiva

Juan Ignacio Jaén, José Raúl Romero y Sebastián Ventura

Dept. Informática y Análisis Numérico
{jijaen, jrromero, sventura}@uco.es,
Universidad de Córdoba
Córdoba, España

Resumen En este trabajo se presenta VisualJCLEC, un entorno de trabajo de carácter gráfico basado en el marco de computación evolutiva JCLEC. La arquitectura software sobre la que se ha construido está orientada a obtener un alto grado de flexibilidad y escalabilidad. VisualJCLEC permite la edición, gestión y ejecución, tanto secuencial como paralela, de múltiples experimentos de JCLEC. Con fines ilustrativos, se resuelve el problema de la mochila, como caso práctico de optimización clásica, haciendo uso de la herramienta presentada.

Palabras clave: computación evolutiva, herramientas software, JCLEC, GUI

1. Introducción

La Computación Evolutiva (CE) es una de las técnicas más comúnmente empleadas para abordar la resolución de problemas de optimización. Son numerosos los ejemplos de aplicación de CE en multitud de disciplinas, siendo el diseño industrial [2], el aprendizaje de las consultas booleanas [4], el aprendizaje de controladores en robótica [8] o la mejora de los sistemas *e-learning* [10] sólo una breve muestra de su idoneidad como ejemplos de aplicación para la resolución de este tipo de problemas. Sin embargo, la adopción de CE como técnica de trabajo implica, usualmente, la necesidad de codificar sofisticados algoritmos que requieren del usuario tanto conocimientos especializados en programación como considerables cantidades de tiempo y esfuerzo. Para aliviar estos inconvenientes, surgen los entornos de trabajo o *frameworks* para CE de propósito general (JCLEC[1] [11], ParadisEO [3], MALLBA [1], entre otros) cuya finalidad es descargar al usuario de la necesidad de escribir código reiterativo en los distintos modelos de CE, facilitando así la reutilización y permitiendo que el programador focalice todos sus esfuerzos en adecuar el desarrollo al dominio abordado codificando únicamente aquellas partes que dependen del problema, como pueden ser las funciones para la evaluación del *fitness* o los operadores de reproducción.

[1] Disponible para su descarga en http://jclec.sourceforge.net

Como se puede apreciar, la disponibilidad de este tipo de *frameworks* no permite acercar el uso de la CE a una audiencia no experta en programación. En este sentido, surge la necesidad de disponer de entornos gráficos que oculten a los usuarios finales, que generalmente son expertos en sus respectivos dominios pero que no disponen de los conocimientos específicos de programación de problemas de CE, los aspectos de codificación necesarios para la confección de sus experimentos.

A pesar de lo anterior, la mayoría de los entornos de trabajo para CE no disponen de este tipo de entorno visual (p.ej. EasyLocal [5]) o aportan una GUI (*Graphical User Interface*) muy rígida, consistente en formularios de configuración prefijados, difícilmente escalable o adaptable a nuevas propuestas (p.ej. ECJ [6]). Esta falta de flexibilidad implica que cualquier mínimo cambio que se desee realizar en el entorno (por ejemplo, introducir un nuevo algoritmo) acarree la necesidad de estudiar el modo en que se construyó la herramienta para poder reconstruir la interfaz con la nueva funcionalidad. Desde la perspectiva del diseño, la creación de un GUI con requisitos de adaptabilidad (tolerancia a cambios) y escalabilidad (incorporar nueva funcionalidad al sistema) implica ser muy cuidadoso en las primeras fases de diseño para tratar de mitigar los posibles problemas que puedan surgir posteriormente y que resulten ser una carga que impidan la futura usabilidad y mantenimiento.

Este trabajo presenta *VisualJCLEC*, un nuevo entorno gráfico extensible para el *framework* JCLEC. El principal objetivo consiste en acercar el uso de JCLEC a una audiencia no experta en CE y/o que no dispone de las habilidades requeridas en programación. Además, se pretende aportar una herramienta de soporte a investigadores que ayude a minimizar el esfuerzo necesario para el diseño de su experimentación. El resto del trabajo se estructura de la siguiente manera: en la Sección 2 se realiza un breve repaso sobre las características de la GUI provista por algunas herramientas para CE. Posteriormente, en la Sección 3, se recogen los detalles sobre la arquitectura que ha sido seguida para el desarrollo. En la Sección 4, como caso de estudio, se presenta la resolución del problema de la mochila haciendo uso de la herramienta presentada. Finalmente, en la Sección 5 se recogen algunas conclusiones y trabajo futuro.

2. Trabajo Relacionado

En [9] se presenta un marco general para la realización de comparativas entre distintos frameworks metaheurísticos de optimización. Para tal fin, los autores identifican las principales características requeridas desde distintos puntos de vista, tales como, metaheurísticas soportadas, codificación, adaptabilidad al problema y apoyo en el proceso de optimización. Es en esta última perspectiva donde los autores enmarcan, entre otros aspectos, las características deseables por la interfaz gráfica provista por la herramienta. De esta manera, se identifican: *(a)* la disponibilidad de ayuda integrada y usabilidad básica (menús, accesos directos, botones, etc.), *(b)* el soporte para la configuración y especificación de las técnicas metaheurísticas, *(c)* la capacidad para el modelado del problema e

importación de datos, *(d)* el soporte para la planificación de tareas de optimización, *(e)* la capacidad para recoger toda la información sobre las instancias del problema, técnicas empleadas y resultados obtenidos; y *(f)* la capacidad para la representación gráfica de los resultados mediante el empleo de diagramas y figuras.

La evaluación de la interfaz *GenLab*, que fue presentada como GUI para el *framework* JCLEC en [11], mediante el marco comparativo anterior, revela que ésta no cubre las características deseables en una herramienta de soporte para CE. Además, desde la perspectiva de la arquitectura, presenta importantes carencias de adaptabilidad y escalabilidad, debido a que meramente consistía en un conjunto de formularios rígidos que únicamente permitían manejar un número fijo de componentes. Dicho esto, queda de manifiesto que en el ámbito de la GUI *JCLEC* presenta un área significante de mejora.

3. Arquitectura de la Herramienta

Para potenciar la *escalabilidad* y *adaptabilidad* de la herramienta se ha utilizado una arquitectura en tres capas (Datos, Lógica y Presentación) separadas entre sí. Esta independencia garantiza que la necesidad de reflejar un cambio en alguna de ellas no afecte al resto. Junto con esta arquitectura de capas se ha seguido la filosofía del conocido patrón arquitectónico Modelo-Vista-Controlador (MVC), que permite separar la representación gráfica, con la que el usuario interactúa, de la información manejada. Brevemente, esta información constituye el *modelo* de datos que se está manejando, el *controlador* captura las entradas del usuario y genera las invocaciones adecuadas para el modelo y la vista y, finalmente, la *vista* es cualquier representación visual de los datos del modelo. En mayor detalle, en VisualJCLEC se han dispuesto dos tipos de vistas: *(a)* la vista de componentes en la que se ubican secuencias de elementos gráficos como: campos de texto, *comboboxes*, paneles, tablas, etc. y *(b)* la vista de árbol, en la cual todos los datos del modelo son representados de forma jerárquica en la que cada nodo representa un elemento del modelo. Dicho esto, es importante resaltar que el patrón MVC cohabita con la arquitectura de tres capas definida. Concretamente, mientras que la *vista* del MVC está desplegada sobre la capa de *Presentación*, el *controlador* y el *modelo* se ubican sobre la capa *Lógica* de la estructura de tres capas desarrollada. Finalmente, el acceso a los datos está encapsulado en la última capa.

3.1. Organización Módular

La Figura 1 muestra una visión general de los distintos módulos que conforman *VisualJCLEC*, así como las interacciones entre ellos. Esta organización se descompone en tres bloques principales: el módulo editor de experimentos, el módulo de generación de ficheros de configuración y el módulo ejecutor de experimentos. En algunos casos, la funcionalidad de los módulos está fragmentada

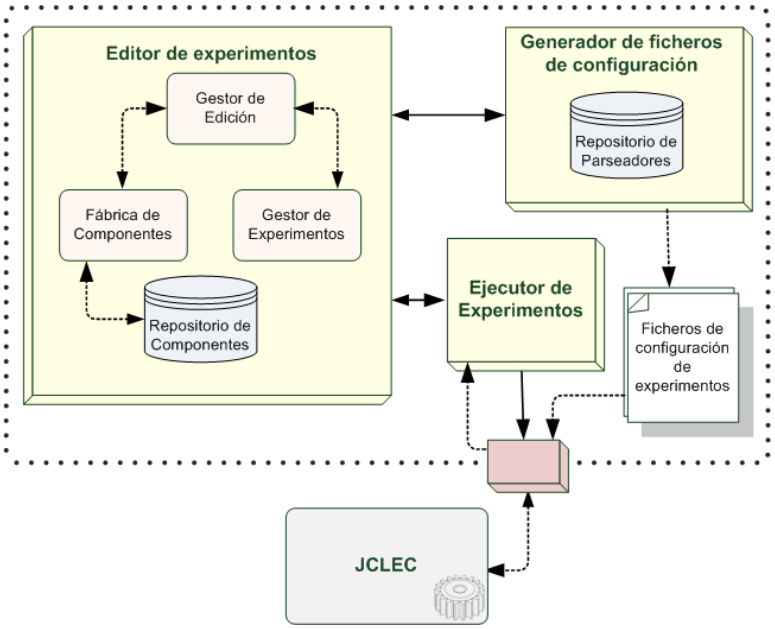


Figura 1. Esquema de módulos de VisualJCLEC

en subfuncionalidades independientes de menor entidad, como se detallarán a continuación.

La línea de puntos que rodea a los tres módulos principales de la Figura 1 delimita el contexto de la herramienta. De esta manera, las cajas externas a ella representan aplicaciones software de terceros que *VisualJCLEC* necesita invocar, como el propio *framework* de CE JCLEC. Esta separación permite que ambas herramientas puedan evolucionar independientemente. Por otro lado, las cajas representadas sobre el límite representan las interfaces implementadas para proveer el intercambio de datos y sincronización requerido, que hacen posible el trabajo entre *VisualJCLEC* y las herramientas de su entorno. Finalmente, las líneas punteadas entre las cajas representa intercambio de información y las líneas contínuas representan flujos de control e invocaciones.

El módulo *Editor de experimentos* se responsabiliza tanto de la gestión de los experimentos diseñados como de los elementos de interacción de la interfaz gráfica que son mostrados al usuario final. La combinación del módulo *Repositorio de Componentes* y el de *Fábrica de Componentes* permite garantizar una alta reusabilidad de código: por un lado, el repositorio almacena todos los descriptores de los distintos elementos que son manejados por la aplicación, tales como, especies, algoritmos, cruzadores, mutadores, etc. Estos descriptores están especificados usando un lenguaje propio denominado LDC, que será descrito en la Sección 3.2. Por otro lado, la *Fábrica de Componentes* se encarga de recoger las peticiones de componentes hecha por el módulo *Gestor de Edición*, el cual se

encarga de dirigir el proceso de edición, y representarlos en pantalla conforme éstos han sido especificados en sus descriptores correspondientes.

En el módulo *Generador de ficheros de configuración* se lleva a cabo la transformación de los distintos experimentos modelados gráficamente en los correspondientes ficheros de configuración internamente requeridos por el *framework* de CE JCLEC. Para realizar esta tarea, se habilita un *Repositorio de Parseadores* que almacenará un conjunto de generadores de ficheros de configuración para diferentes *frameworks* de CE. En este caso, la herramienta presentada dispone del correspondiente parseador para JCLEC. Finalmente, la tarea de ejecución, realizada en el módulo *Ejecutor de experimentos*, implica la invocación del motor de CE, en este caso JCLEC, empleando para ello los ficheros de configuración, en el formato apropiado, obtenidos por el módulo anterior. Una vez realizada la petición al motor de CE, se encargará de recoger y sincronizar con el módulo *Editor de experimentos* los distintos resultados devueltos por el motor de CE, con la finalidad de ir presentándoselos al usuario a través de la interfaz de usuario.

3.2. Lenguaje para la Definición de Componentes

Se ha definido un DSL (*Domaing Specific Language*) para describir los componentes (algoritmos, operadores, etc.) manejados por la aplicación. Este lenguaje, que denominamos Lenguaje de Definición de Componentes (LDC), nos permite una alta reusabilidad de código, una mejora sustancial de la flexibilidad y reducir el esfuerzo necesario para incorporar nuevos elementos a la herramienta.

Por un lado, la reusabilidad es obtenida debido a que cada componente de la interfaz de usuario es creado en tiempo de ejecución por la aplicación haciendo uso de los descriptores expresados formalmente en LDC. Referente a la ganancia de flexibilidad, para reflejar cualquier cambio sobre los componentes manejados únicamente se requerirá modificar estos descriptores LDC, en lugar de tener que modificar el código y recompilar la aplicación.

En la Figura 2 se recogen los constructores de LDC y sus relaciones. Como se puede apreciar, se ha establecido que un experimento (*Experiment*) sea definido en términos de una especie (*Species*), un algoritmo (*Algorithm*), un generador de aleatorios (*Randomizer*) y un informador (*Listener*). Opcionalmente, también puede constar de un operador de tipo selector de padres (*Selector*), cruzador (*Crossover*), mutador (*Mutator*), proveedor (*Provider*) y/o evaluador (*Evaluator*). Esto es debido a que es una práctica muy común que algunos algoritmos fijen internamente sus propios operadores y, por lo tanto, el usuario no debe indicarlos. Además de lo anterior, los componentes *Species*, *Algorithm* y los distintos constructores en los que se especializa el elemento *Operator* pueden definir parámetros que podrán ser simples o complejos, esto es, un parámetro que contiene a su vez a otros parámetros. Con fines ilustrativos, el siguiente código muestra un ejemplo de definición de la especie *bin-array* conforme a la sintaxis textual definida para el lenguaje de definición de componentes.

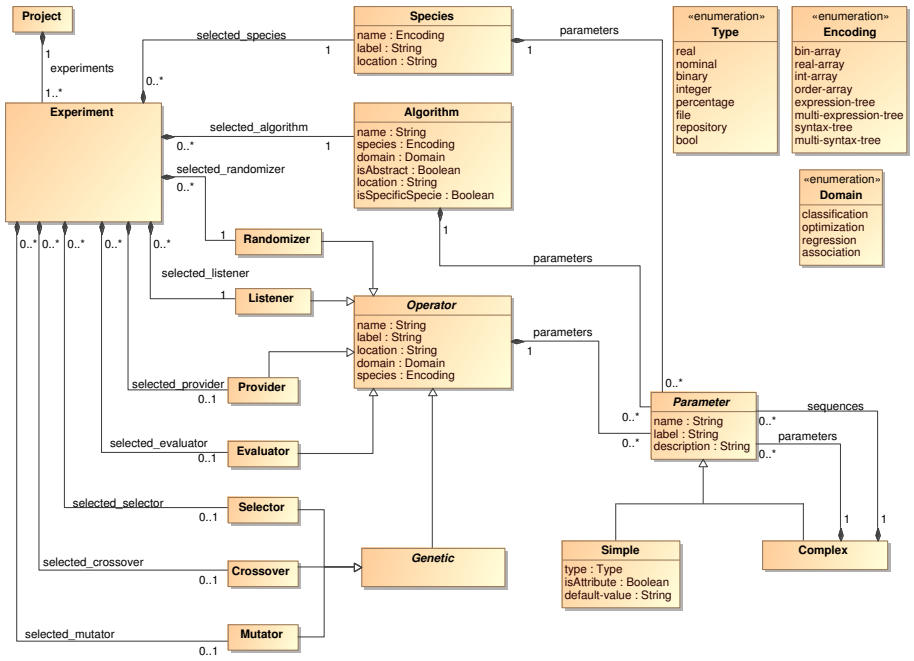


Figura 2. Sintaxis abstracta de LDC

```
<species>
  <label>Bin Array</label>
  <name>bin-array</name>
  <location>net.sf.jclec.binarray.BinArrayIndividualSpecies</location>
  <parameters>
    <simple-parameter>
      <name>genotype-length</name>
      <label>Genotype length</label>
      <type>integer</type>
      <isAttribute>true</isAttribute>
      <default-value>0</default-value>
    </simple-parameter>
    <simple-parameter>
      <name>genotype-schema</name>
      <label>Genotype schema</label>
      <type>binary</type>
      <isAttribute>false</isAttribute>
      <default-value>10010101</default-value>
    </simple-parameter>
  </parameters>
</species>
```

El código LDC anterior define la especie *bin-array* conforme a la sintaxis abstracta definida para dicho lenguaje. De esta manera, se establecen los valores para los campos *name*, *label* y *location* además de los parámetros requeridos por la especie que está siendo especificada. Concretamente, la especie *bin-array* define dos parámetros simples *genotype-length* y *genotype-schema*. Análogamente a lo anterior, ambos parámetros han de fijar los valores para sus propiedades *na-*

me, *label* y *description* y aquéllas derivadas del tipo especializado, en este caso los dos parámetros son de tipo *Simple*. En este sentido, mientras que el primer parámetro es de *tipo entero* y fijado su *valor por defecto* a `0`, el segundo es de *tipo binario* y se fija su *valor por defecto* a `10010101`. El resto de la información contenida en el código anterior se refiere a *(a)* aspectos gráficos, p.ej. cómo los elementos son mostrados en la pantalla: el valor del campo *label* será el texto empleado para etiquetar al elemento, el valor del campo *description* será empleado como texto para la ayuda contextual de la interfaz; y *(b)* información sobre cómo se realiza el parseo de los parámetros en los ficheros de configuración.

4. Caso de Estudio: El problema de la mochila 0–1

Con la finalidad de mostrar la utilidad de *VisualJCLEC* se ha optado por modelar el problema de la mochila, cuya simplicidad permite centrar mejor la atención más en la funcionalidad de la herramienta que en aspectos del problema. El problema de la mochila 0/1 [7] es un problema clásico de optimización combinatoria. Su nombre deriva del problema de maximizar el número de elementos esenciales que pueden ser introducidos en una mochila (de peso máximo) para ser portada en un viaje. De manera breve, el problema se define como sigue: considérense n objetos distintos en una mochila. Cada objeto tiene un peso w_i y un precio p_i asociado para $i = 1, 2, ..., n$. La mochila puede portar un peso que no exceda de una cantidad dada W. Teniendo en cuenta estas consideraciones, el objetivo del problema es llenar la mochila maximizando el precio total de los objetos contenidos.

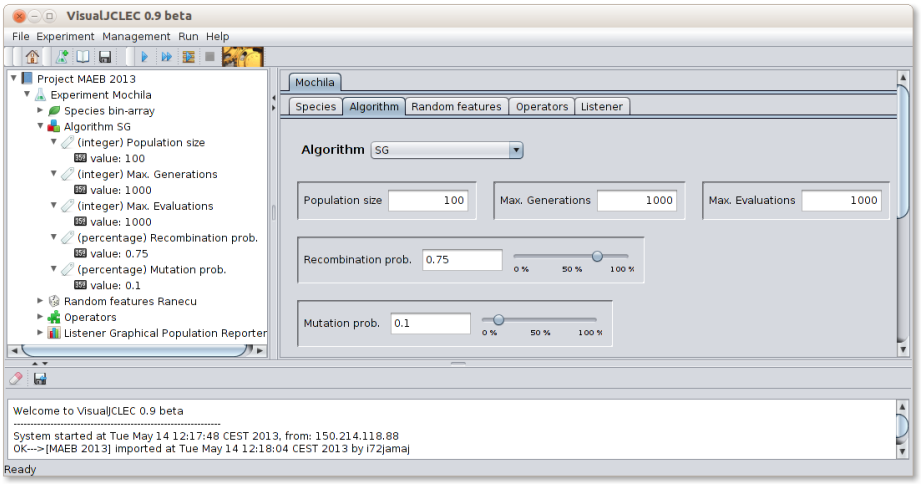


Figura 3. Ventana de configuración del algoritmo

Para este caso de estudio, se ha seguido un punto de vista clásico para resolver este problema NP-difícil. Así, la función de *fitness* es el valor total de la mochila, debiendo tener en cuenta la restricción del peso máximo. Dicho esto, la forma más común de resolver este problema es usando un esquema de codificación binaria. En este esquema, el individuo contiene un genotipo binario con tantos genes como elementos diferentes pueden ser llevados en la mochila. El significado del valor `1` para el gen *i-ésimo* significa que el elemento *i-ésimo* ha sido introducido en la mochila y, por lo tanto, un valor `0` significará lo contrario. Por último, se ha decidido emplear la clase *KnapsackEvaluator*, cuyo código está disponible en [11], para evaluar el *fitness* de los individuos. Una vez realizadas todas las consideraciones anteriores e incorporado el evaluador, se modela el experimento a través de la interfaz gráfica de *VisualJCLEC*. Como se puede apreciar en la Figura 3, se ha optado por un algoritmo generacional simple (*Simple Generational*, SG) para abordar el problema de optimización. Además del algoritmo a utilizar, se han de fijar los valores de los parámetros propios del algoritmo seleccionado. De igual manera son modelados todos los componentes requeridos para la ejecución del experimento en JCLEC: especie, tipo de generador de aleatorios y semilla, operadores genéticos, informadores, etc. En la Figura 4 se observa el formulario creado dinámicamente por *VisualJCLEC*, según el algoritmo seleccionado, para introducir los distintos tipos de operadores y su correspondiente configuración de parámetros asociados a cada uno de ellos.

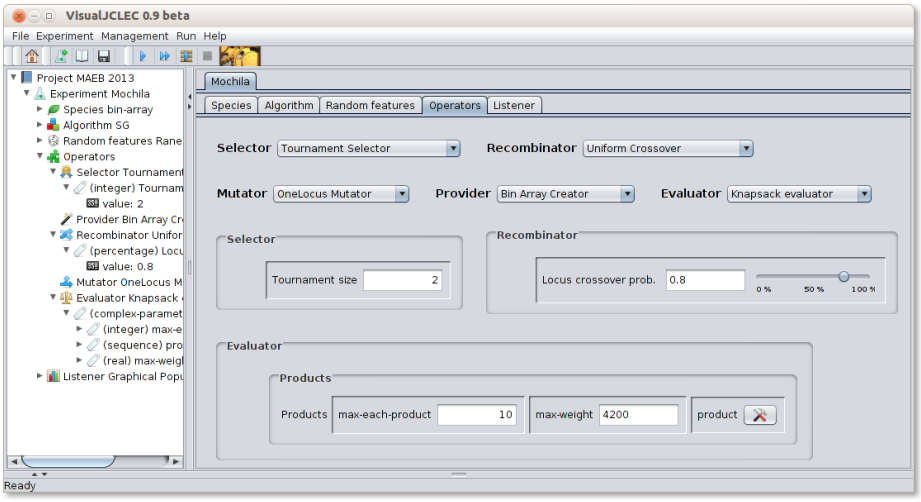


Figura 4. Ventana de configuración de los operadores

Además, *VisualJCLEC* dispone de dos tipos distintos de ejecución de experimentos: secuencial y paralela. En el primer caso, los distintos experimentos de los que consta el proyecto generado se irán ejecutando uno a uno, de modo

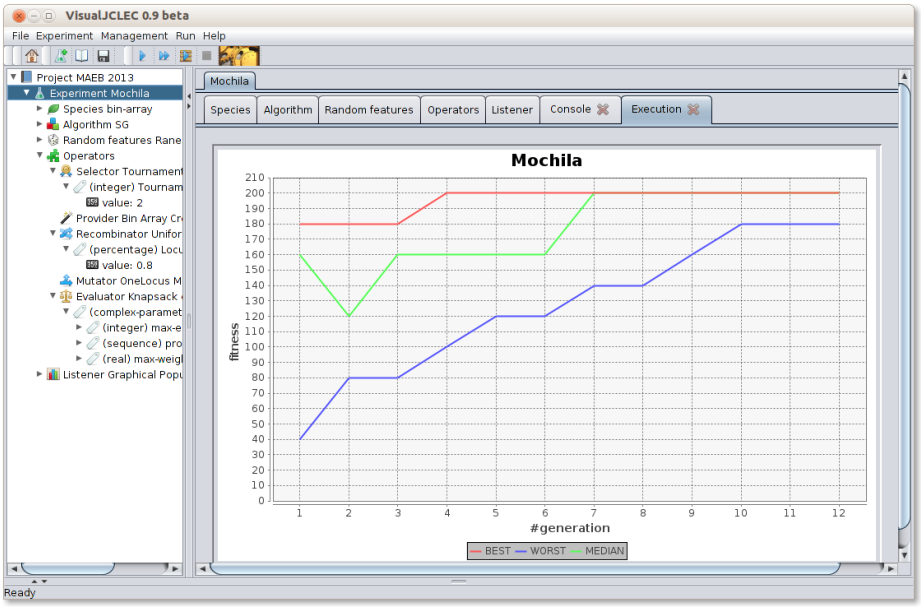


Figura 5. Ventana de gráfica de ejecución

que el fin de la ejecución de un experimento marca el inicio del siguiente. Por otro lado, la ejecución paralela ejecuta todos los experimentos simultáneamente. Al iniciar la tarea de ejecución se generan dos nuevas pestañas en la interfaz (*Console* y *Execution*) en cada experimento ejecutado. En la primera de ellas se muestran los datos de los individuos mejor, peor y medio cada cierto número de generaciones prefijado. En la ventana *Execution* se muestran cada uno de los individuos anteriores gráficamente según su *fitness*, (ver Figura 5).

5. Conclusiones

En el presente trabajo hemos presentado la herramienta *VisualJCLEC*[2], un entorno de trabajo gráfico para CE empleando para ello el *framework* JCLEC. Se han detallado sus principales características: una arquitectura modular segmentada en capas, caracterizada por un alto grado de flexibilidad y escalabilidad junto con la definición de un lenguaje específico para la definición de los componentes. La herramienta presentada permite la edición, gestión y ejecución de experimentos del *framework* JCLEC de forma intuitiva a través de la interfaz gráfica aportada. Finalmente, con la finalidad de presentar y validar la herramienta se ha modelado y resuelto el problema clásico de optimización de la mochila. Como trabajo futuro se plantea extender la herramienta desde dos posibles frentes de mejora. Por un lado, incrementar el número de funcionalidades,

[2] La herramienta está disponible para su descarga en http://jclec.sourceforge.net

tales como preprocesamiento de datos, análisis estadístico, modo de ejecución paralela y distribuida y la capacidad para pausar, modificar el valor de ciertos parámetros y reanudar la ejecución de un experimento. Por otro lado, se propone dotar a la herramienta de la capacidad para desplegar perspectivas que permitan adecuar el entorno de trabajo a dominios específicos, por ejemplo clasificación, asociación o incluso a un problema concreto.

Agradecimientos

Trabajo financiado por el Ministerio de Ciencia e Innovación bajo el proyecto TIN-2011-22408 y los fondos FEDER.

Referencias

1. E. Alba, G. Luque, J. García-Nieto, G. Ordonez, and G. Leguizamon. Mallba: a software library to design efficient optimisation algorithms. *Int. J. Innov. Comput. Appl.*, 1(1):74–85, Apr. 2007.
2. A. Benedetti, M. Farina, and M. Gobbi. Evolutionary multiobjective industrial design: the case of a racing car tire-suspension system. *Evolutionary Computation, IEEE Transactions on*, 10(3):230 – 244, june 2006.
3. S. Cahon, N. Melab, and E.-G. Talbi. ParadisEO: A Framework for the Reusable Design of Parallel and Distributed Metaheuristics. *Journal of Heuristics*, 10(3):357–380, 2004.
4. O. Cordón, E. Herrera-Viedma, and M. Luque. Improving the learning of boolean queries by means of a multiobjective iqbe evolutionary algorithm. *Inf. Process. Manage.*, 42(3):615–632, May 2006.
5. L. Di Gaspero and A. Schaerf. Easylocal++: An object-oriented framework for flexible design of local search algorithms. *Software — Practice & Experience*, 33(8):733–765, July 2003.
6. S. Luke, L. Panait, G. Balan, S. Paus, Z. Skolicki, J. Bassett, R. Hubley, and A. Chircop. ECJ: A Java-based Evolutionary Computation Research System, 2006. http://cs.gmu.edu/∼eclab/projects/ecj/.
7. S. Martello and P. Toth. *Knapsack problems: algorithms and computer implementations*. John Wiley & Sons, Inc., New York, NY, USA, 1990.
8. M. Mucientes, L. Moreno, A. Bugarín, and S. Barro. Evolutionary learning of a fuzzy controller for wall-following behavior in mobile robotics. *Soft Comput.*, 10(10):881–889, May 2006.
9. J. Parejo, A. Ruiz-Cortés, S. Lozano, and P. Fernandez. Metaheuristic optimization frameworks: a survey and benchmarking. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 16:527–561, 2012.
10. C. Romero, S. Ventura, and P. D. Bra. Knowledge discovery with genetic programming for providing feedback to courseware authors. *User Modeling and User-Adapted Interaction*, 14(5):425–464, Jan. 2005.
11. S. Ventura, C. Romero, A. Zafra, J. A. Delgado, and C. Hervás. JCLEC: a Java framework for evolutionary computation. *Soft Computing*, 12(4):381–392, Oct. 2007.

Achieving Replicability: Is there life for our experiments after publication?

Jose Antonio Parejo, Sergio Segura Rueda, and Antonio Ruiz-Cortés*

Department of Computing Languages and Systems. Univeristy of Sevilla

Abstract Metaheuristics are algorithmic schemes that ease the derivation of novel algorithms to solve optimization problems. These algorithms are typically approximated and stochastic, leading to the preeminence of experimentation as the mean of supporting claims in research and applications. However, the huge number of variants and parameters of most metaheuristics, the ambiguity of natural language used in papers, and the lack of widely accepted reporting standards threatens the replicability of those experiments. This problem, that has been identified in the literature by several authors, significantly hinders the construction of a complete and cohesive body of knowledge on the behavior of metaheuristics. This paper proposes a set of minimum information guidelines for reporting metaheuristic experiments, and an experiment description language that supports the meeting of those guidelines. By using this language, metaheuristic optimization experiments are described in a tool-independent and unambiguous way, while maintaining readability and succinctness. Those contributions pave the way for replication using different problem instances and parameters, bringing a new life to metaheuristic experiments after publication.

1 Introduction

An experiment is replicable when its results can be verified and/or clarified through conducting another experiment. This process can be preformed either following the same experimental procedure in similar conditions (named repetitions when the conditions are exactly the same and replications when they are slightly different), or through a different procedure aimed to verify similar hypotheses about the same phenomenon (named conceptual replications).

The quote of [1]: "*The use of precise, repeatable experiments is the hallmark of a mature scientific or engineering discipline*", clashes violently with the statements of some authors in the area of metaheuristic optimization[2]: "*Verifying results found in the literature is in practice almost impossible*", "*running a reportedly good algorithm on your own data is an extremely difficult task*", "the details presented in a typical paper are insufficient to ensure that one would implement the same algorithm".

* This work was partially supported by the EU Commission (FEDER), the Spanish and the Andalusian R&D&I programmes grants SETI (TIN2009-07366), TAPAS (TIN2012-32273), ISABEL (TIC-2533) and THEOS (TIC-5906).

The replicability of experiments using metaheurisitcs is therefore a problem. Since most of the research on metaheuristics relies on empirical validation and experimentation[3], this problem becomes of paramount importance in our area. It is therfore and important problem that significantly hinders the construction of a solid and cohesive body of knowledge on the behavior of metaheuristics.

Experimental replicability is a problem due huge number of variants, parameters and possible customizations of the algorithms, its own stochastic nature, and the lack of an widely accepted scheme of experimental reporting (in contrast to those used in the natural sciences [4]).

Minimum information guidelines have enabled the reuse of existing work and reproducibility of the experimental process in different research areas such as biology and simulation. However, guidelines are not enough. Several set of guidelines has been published by different authors (see for instance [5,6,7,8]), but standard experimental reporting in the area has not improved to the levels of other scientific disciplines [9]. The proposal of this paper for this issue is the creation of: i) experiment description languages (EDLs) that vertebrate the description and reporting of experiments according to the guidelines; and ii) tools that automatically check that the experiment descriptions expressed on those languages are following the guidelines actually.

The remainder of this paper is structured as follows: Section 2 describes our guidelines of minimum information to be reported about metaheuritic optimiztation experiments. Next, 3 describes an experiment description language whose use guarantees the meeting of most of the guidelines defined in previous section. Section 4 presents the tools developed to ease the creation and manipulation of experimental descriptions with the language. Section 5 reports some validation performed on our proposal to ensuer its suitability. Finally, 7 provides some conclusions and points out future work.

2 Minimum Information about Metaheuristic Optimization Experiments

Minimum Information guidelines have enabled the reuse of existing work and reproducibility of the experimental process in different research areas such as biology and simulation. Authors propose a set of guidelines about Minimum Information about Metaheuristic Optimization Experiments (MIaMOE). In so doing, the guidelines defined for simulation experiments in [10] have been adapted and extended to the specific context and characteristics of metaheuristic optimization experiments.

MIaMOE describes the minimum information that should be provided in order to properly the describe metaheuristic optimization experiments. The goal of MIaMOE is to enable experimental replicability. In order to achieve this goal, MIaMOE defines a set of requirements of the reported information. Thus, MIaMOE does not define neither operation procedures, nor specific experimental description formats, but a set of rules for describing experiments. The description of an experiment is MIaMOE-compliant if it meets all the rules defined by

MIaMOE. Specifically, MIaMOE defines the following rule: *For an experiement witn metaheuristic optimization techniques*

1. **Each optimization algorithm used in the experiment must be identified, fully described and accessible.**
   (a) Either the description of each optimization algorithm is complete and unambiguous, or their implementation is provided.
   (b) All parameter values, configuration settings, and governing conditions of the optimization algorithms used in the experiment are unambiguously described. Specifically, it should be described:
      i. The set of parameters, their domain, and the specific value used for experimentation.
      ii. Specific operators and variants of the optimization technique used.
      iii. The optimization process termination criteria
      iv. The random number generation algorithms used and the seeds used for experimentation.
   (c) If an algorithm is not implemented using a common implementation language and runtime, then the implementation should be provided and the implementation language and required run-times used must be clearly specified.
   (d) Any preprocessing step required before the execution of the optimization algorithm must be fully and unambiguously described.
2. **All information used to obtain the results and draw conclusions of the experiment must be provided.**
   (a) All the information used as input by the optimization algorithms must be provided, including problem instances data. If standard benchmarks of problem instances are used, the specific reference where the benchmark was proposed, the specific version (or year) of the benchmark used, and a download address must be provided.
   (b) The variables of each algorithm used to obtain results reported must be identified, including the specific step of algorithms in which results data are collected.
   (c) All the results datasets (raw numerical data and post-processing results) generated by the experiment must be described unambiguously, and a copy of the actual results should be provided.
   (d) If conclusions drawn from the experiment depend on the relation between different results datasets, the specific results datasets to be compared and the equations of the relation must be described unambiguously.
3. **A precise description of the experimentation procedure, its execution context, and other procedures used in the experiment must be provided.**
   (a) All the steps of the experimentation procedure must be clearly described, including:
      i. The total number and order of execution of algorithms on problem instances, or the criteria used to determine such executions and its order (for instance the experimental design).

ii. Information used by algorithms that is produced during the own experimentation process and its processing.
iii. All post-processing steps applied on the raw numerical data used to obtain each result dataset must be described in detail.

(b) The physical and computational equipment used during the experimentation process must be described. Those elements that are required for a replication should be specified explicitly, defining the minimum equipment requirements for replication.

Evaluating if the description of an experiment is MIaMOE-compliant just from a paper is a time-consuming task that must be performed manually. For instance, in a optimization technique comparison experiment, if authors do not provide the source code of the implementation of the techniques and the experimental procedure followed for conducting the experiment, then the description of the experimental procedure, the algorithms used in the paper, and the parametrization used in each experiment must be carefully scrutinized in order to determine MIaMOE-compliance. Moreover, if some optimization techniques compared in the experiment are described using references, the MIaMOE-compliance of the descriptions of those techniques in the referenced papers should be evaluated. Laboratory packages (also named lab-pack), are detailed reports with attached materials used by experimenters to provide all the relevant details about an study. Providing the source code, binaries and the input/output files of the experiments, along with the description of the computation environment in a lab-pack is the most straightforward way to achieve MIaMOE-compliance. However, providing source code and input files does imply being MIaMOE-compliant. For instance, if the algorithms are implemented in JAVA using the standard random number generator provided by the language and the seed, but the specific version of the Java Virtual Machine used for experiment conduction is not specified, then rule 1(b)iv is violated, since the random number generation algorithm is undetermined (it depends on the version of JAVA). Given the previous definitions, a MIaMOE-compliant experiment should be reproducible. However depending on the way researchers follow the guidelines, the investment and effort required for reproducing the experiments can vary dramatically. For instance, we suppose a MIaMOE-compliant a metaheuristics comparison experiment where researchers provide just the pseudo-code of their proposals and references to the publications where the alternative approaches are defined. The reproduction of such an experiment requires to implement each optimization technique based on the descriptions provided in the corresponding papers, which implies a significant amount of effort. Even when researchers provide the full source code, input files and specification of the environmental requirements of their experiments (such as libraries, run-times, specific platform for execution, etc.), the effort required to replicate an environment that meets those requirements and launch the experiment may be non-negligible. For instance, in [11] this effort is described as "considerable agony" even when performed by the own researchers in their lab. On the other hand, using the source code as a description of the experiments has some drawbacks. First, it makes experiment descriptions dependent on the

implementation technologies. Second it penalizes descriptions readability and understandability by non-programmers. Third it lowers the abstraction level at which the experiment is described, forcing to read and understand the specific implementation details of the experiment source in order to gather the information about the experiment. The need of language at an intermediate level of abstraction between natural-language descriptions found in papers and implementations could mitigate such problems is proposed in this paper. Such a language should be both human readable and machine processable. Having an experiment described using such a language in companion with the source code or binaries used in metaheuristic optimization experiments, could enable and ease experiment reproducitiblity, and even automate its replication with proper tools.

3 Metaheuristic Optimization Experiments Description Language (MOEDL)

MOEDL defines a set of elements for describing metaheuristic experiments that are of major relevance for ensuring MIaMOE compliance, such as the objective functions and problem instances used in the experiment, termination criteria used by each technique, the random number generation algorithm and seeds used, etc.

The structure of MOEDL is the result of an extensive analysis of a variety of experiments developed by authors [12,13,14], a careful study of the related literature and a process of successive refinements of the meta-model after applying it to different scenarios. Specifically, we have taken [4] as the main reference for general experiment descriptions and [9,8] for the specific details of metaheuristic optimization experiments. Additionally, we have evaluated other approaches (Section 6) for experiment description and the proprietary formats and classes used by the set of metaheuristic optimization frameworks assessed in [13].

MOEDL supports the succinct description of some common types of experiments in the area. Those experiment types are described below.

Figure 1 depicts the general structure of a MOEDL document. The schema depicted in Fig. 1 is informal and only provides a snapshot of the general stucture of MOEDL documents. In order to accurately describe the abstract syntax of MOEDL, its meta-model is described in [15]. That description is independent of the specific concrete syntax and serialization used for such as xml schemas and documents, plain text files, or graphical notations. For the purpose of this paper, due to space limitations the general structure of such documents in described, without diving deep in the details.

A `MetaheuristicOptimizationExperiment` in MOEDL contains the description of one or more problem types (with their corresponding set of problem instances), and one or more optimization techniques. Each problem instance and optimization technique, has a unique identifier in the context of the document, and their descriptions contains all the details about its parameters and

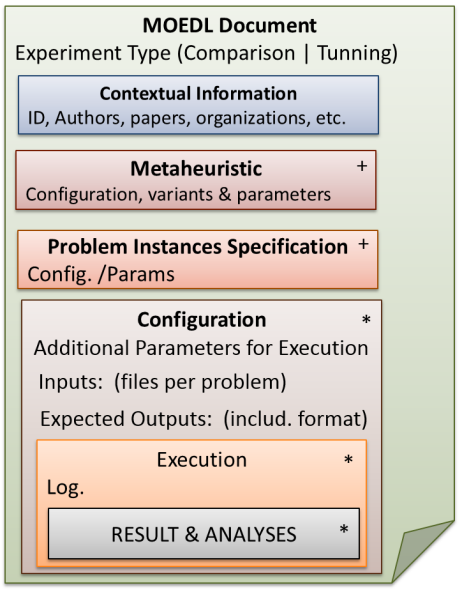


Figure 1. General Structure of MOEDL experimental descriptions

configuration (such as the specific objective function or its concrete parameter values in problem instances, and the specific variant used, the source file where it is implemented, and the parameter values used in the experiment for each metaheuristic). The detailed description of each optimization technique is performed through its `Configuration`, this element supports the description of simple parameters (such as the population size, or the probability of application of the mutation operator in Evolutionary Algorithms) and complex algorithmic variants. However, given a `Technique` and a `Configuration`, there is not guarantee that the technique configuration describes all the relevant parameter of the technique. As a consequence, in order to meet MIaMOE rules 1a and 1b, the provision of a lab-pack containing the source code of the technique and experimental procedure is mandatory. The single exception to this rule is the case when a software framework (such as ECJ or HeuristicLab) is used and the exact replications of the executions of the experiment can be performed automatically using the experiment description. Under those circumstances, the framework is able to enact the optimization technique based on the description provided, and consequently, the source code of the own framework and the experiment description fully describe the optimization techniques applied[1].

Additionally, a MOEDL description can contain a set of global configurations, that are used to specify configuration settings for the experiment as a whole.

[1] If custom variants are described in the framework-specific configuration of the technique and its binaries are provided in the experiment lab-pack. Then the source code of such custom variants should be provided also in the lab-pack

For instance, the termination criterion used for optimization can be specified for the whole experiment or for an specific optimization technique. In order to support the automation of the experiment, and ease replication using different termination criteria, this element defines an extension point in MOEDL, since its description could depend on the software used for implementing the metaheuristic. Either a global termination criterion is specified for the experiment, or each optimization technique should define its own termination criterion. It is worth noting that the use of different termination criteria in this kind of experiments could lead to bias in the comparison and wrong conclusions in the experiments.

The random number generation algorithm, seeds used for the experiment, and termination criterions can also be specified either for the whole experiment or for each specific optimization technique in particular in the configuration. This constraint enforces the fulfillment of the points in MIaMOE rule 1b.

In a techniques comparison experiment, the objective functions values of solutions obtained with several optimization techniques are compared for an specific set of problem instances.

In a Tuning experiment, the objective functions values of solutions obtained with several configurations of one single technique are compared for an specific set of problem instances.

Thus a `TechniqueParametrizationExperiment` contains a single metaheuristic optimization technique and a set of `ComplexParameter` definitions. These definitions specify the space of possible values of the parameter using a `Domain`, since finding its optimal value is the purpose of the experiment. This `Domain` is specified either by extension, i.e. providing the the whole set of possible values (e.g. 100, 200, and 300), or by intension, i.e. specifying a range for its the value and an algorithm or iteration mechanism to generate successive values of the parameter.

4 Supporting Tools & Libraries

An XML-schema encoding the language structure described in the previous section has been created. This schema provides an xml-based syntax for the language, which support the objectives of while maintaining human readability.[2] Moreover, a JAVA library for manipulating this kind of documents has been created. This library is named `JLibMOEDL`, and provides features such as the validation of MOEDL documents against the XML-schema and the semantic constraints exposed above, proving a detailed validation report, the load and saving of experiental descriptons as XML-files, and the representation of all the elements of the experimental description as Java objects. In this sense, `JLibMOEDL` eases the manipulation and automated generation of experimental descriptions with MOEDL.

[2] The XML schemas are available at `http://moses.us.es/schemas/moedl/v1/`.

5 Validation of the proposals

The suitabilty of the proposed language for describing metaheuristic optimization experiments has been validated using a case study. Specifically, two technique comparison experiments, and two parameter tuning experiments supporting the main conclusions of [16] has been described using MOEDL with its xml-based syntax. The language has been expressive enough to accurately describe all those experiments[3].

6 Related Work

Related work of the proposals made in this chapter present in the literature can be divided into two categories: (i) guidelines for reporting and support of the experimental replication, and (ii) languages for describing experiments.

Although recognized as an issue of paramount importance for progress of science and engineering, replication of experiments has been an extremely difficult task in our context [2,11,17].

In the context of experimentation with metaheuristics, specific guidelines has been proposed [18,6,19,7], and the didactic effort of the research community is still going on [17]. However, those proposals focus mainly on aiding the correct conduction of the experiments, on the use of appropriate of experimental designs for the hypothesis and populations addressed [8,20,21], and on the use of the correct statistical analysis tools for such designs [22]. These efforts have been successful, given the current widespread use of statistical tests for validating conclusions and of standard experiment designs in the area. Most of those proposals focus on specifying how researchers must do experimental research, but not how they must report and publish the materials in order to improve experimental replicability. Only a small subset those proposals described guidelines about reporting of experiments, with the aim of enabling experimental replicability (see for instance [6,7]). The main difference between the proposals in this paper and previous work is that our proposal of reporting guidelines (MIaMOE), focuses on *what* should be described (the minimum information required for replication) from an abstract perspective, and it delegates the tasks of specifying *how* to describe such information in the experimental description language (MOEDL), while providing means for checking that all required information is provided through software tooling.

Regarding experiment description languages, we can classify previous proposals in two categories. On the one hand, *Descriptive languages* support the description of the experiment but do not specify how to use such description, delegating the tasks of interpreting the description and using it for replication on researchers. An example of this kind of proposals is EDL [23], that provides a basic xml syntax to organize the description of the experiment. On the other hand, operational languages provide an description of the experiments at such an

[3] The XML files describing the experiments described using MOEDL are available at `http://moses.us.es/?page_id=16`

abstraction level that enables its automated replication, an example of this kind of experimental description language SED-ML [24]. The problem of operational languages is that they need to support very specific structures in order support the required details for replicability, as a consequence, the roposals for operational languages are specific of certains areas, for instance SED-ML is specific for simulation experiments. Another consequence of this need for specificity is that some languages have been created for support the specification of specific experimental artifacts and experimental protocols that of the whole experiment for ceitan areas, such as PEBL [25] that is intended for the creation of experiments in psychology.

Our proposal is operational, in the sense that enables the replication of experiment based on the description of experiments, but also contains some elements whose purpose is merely descriptive, such as the `Context`. Our proposal follows the general approach of SED-ML [24], but it adapts most of its concepts and introduce elements specific for describing experiments with metaheuristics that are MIaMOE-compliant.

7 Conclusions

In this paper, the problem of experimental replicability in the specific context of metaheuristic optimization experiments has been pointed out. Two contributions has been proposed to tacke this problem: (i) a set of guidelines about the information to, and (ii) an experiment description language that supports the meeting of those guidelines. Our proposals provide a balance, where metaheuristic optimization experiments are described in a tool-independent and unambiguous way, while maintaining readability and succinctness. Those contributions pave the way for replication using different problem instances and parameters, bringing a new life to metaheuristic experiments after publication.

References

1. Lewis, J.A., Henry, S.M., Kafura, D.G., Schulman, R.S.: On the relationship between the object-oriented paradigm and software reuse: An empirical investigation. Technical report, Blacksburg, VA, USA (1992)
2. Eiben, A., Jelasity, M.: A critical note on experimental research methodology in ec. Computational Intelligence, Proceedings of the World on Congress on **1** (2002) 582–587
3. Floudas, C.A., Pardalos, P.M.: Encyclopedia of Optimization. 2nd edn. (October 2008) ISBN 978-0-387-74760-6.
4. Gliner, J.A., Morgan, G.A., Leech, N.L.: Research methods in applied settings: an integrated approach to design and analysis (second edition). Volume 2nd. Tylor and Francis Group (2009)
5. Florian, M., Fox, B., Crowder, H., Dembo, R., Mulvey, J.: Reporting computational experience in operations research. Operantions Research **27** (1979) 7–10
6. Barr, R., Golden, B., Kelly, J., Resende, M., Stewart, W.: Designing and reporting on computational experiments with heuristic methods. Journal of Heuristics **1**(1) (1995) 9–32

7. Michalewicz, Z., Fogel, D.B.: How to Solve It: Modern Heuristics. Springer (2004)
8. Bartz-Beielstein, T.: Experimental Research in Evolutionary Computation: The New Experimentalism (Natural Computing Series). 1 edn. Springer (April 2006)
9. Bartz-Beielstein, T., Preuss, M.: Tuning and experimental analysis in evolutionary computation: what we still have wrong. In: GECCO (Companion). (2010) 2625–2646
10. Waltemath, D., Adams, R., Beard, D., Bergmann, F., Bhalla, U., Britten, R., Chelliah, V., Cooling, M., Cooper, J., Crampin, E., et al.: Minimum information about a simulation experiment (miase). PLoS computational biology **7**(4) (2011) e1001122
11. Schwab, M., Karrenbach, N., Claerbout, J.: Making scientific computations reproducible. Computing in Science Engineering **2**(6) (nov/dec 2000) 61 –67
12. Parejo, J.A., Fernández, P., Ruiz-Cortés, A.: Qos-aware services composition using tabu search and hybrid genetic algorithms. In: Talleres de las JISBD. IADIS'08. (2008)
13. Parejo, J.A., Lozano, S., Ruiz-Cortès, A., Fernandez, P.: Metaheuristic optimization frameworks: A survey and benchmarking. Soft Computing (2011)
14. Segura, S., Parejo, J.A., Hierons, R.M., Benavides, D., Ruiz-Cortés, A.: Ethom: An evolutionary algorithm for optimized feature models generation (v. 1.2). Technical report (July 2012)
15. Parejo, J.A.: MOSES: A Software Ecosystem for Metaheuristic Optimization. PhD thesis, University of Sevilla (2013) Early Draft Version available at `https://dl.dropboxusercontent.com/u/1279737/tesis/Prelectura/Dissertation.pdf`.
16. Parejo, J.A., Fernández, P., Ruiz-Cortés, A.: On parameter selection and problem instances generation for qos-aware binding using grasp and path-relinking. Research Report 2011-4, ETSII. Av. Reina Mercedes s/n. 41012. Sevilla. Spain (2011)
17. Bartz-Beielstein, T., Preuß, M., Zaefferer, M.: Statistical analysis of optimization algorithms with r. In Soule, T., Moore, J.H., eds.: GECCO (Companion), ACM (2012) 1259–1286
18. Hooker, J.: Testing heuristics: We have it all wrong. Journal of Heuristics **1**(1) (1995) 33–42
19. Coffin, M., Saltzman, M.: Statistical analysis of computational tests of algorithms and heuristics. INFORMS Journal on Computing **12**(1) (2000) 24–44
20. Ridge, E., Kudenko, D.: Tuning the performance of the mmas heuristic. In Stützle, T., Birattari, M., H. Hoos, H., eds.: Engineering Stochastic Local Search Algorithms. Designing, Implementing and Analyzing Effective Heuristics. Volume 4638 of LNCS. Springer (2007) 46–60
21. Birattari, M., Stützle, T., Paquete, L., Varrentrapp, K.: A racing algorithm for configuring metaheuristics. In: Proc. of the Genetic and Evolutionary Computation Conference. GECCO '02, Morgan Kaufmann (2002) 11–18
22. Derrac, J., García, S., Molina, D., Herrera, F.: A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence alg. Swarm and Evolutionary Computation **1**(1) (2011) 3–18
23. Sinnema, R.: Experiment description language (edl) (2003)
24. Köhn, D., Novère, N.: Sed-ml — an xml format for the implementation of the miase guidelines. In: Proc. of the 6th Inte. Conf. on Computational Methods in Systems Biology. CMSB '08, Springer-Verlag (2008) 176–190
25. Shane, T.: A partial implementation of the bica cognitive decathlon using the psychology experiment building language (pebl). International Journal of Machine Consciousness **2**(02) (2010) 273–288

Problema Dinámico de Rutas de Vehículos con Ventanas de Tiempo

Alondra De Santiago, Belén Melián-Batista[1], Ada Álvarez[2]

[1] ETS de Ingeniería Informática
Universidad de La Laguna, España
alondra.de.santiago.19@ull.edu.es, mbmelian@ull.es
[2] Universidad Autónoma de Nuevo León, México
Facultad de Ingeniería Mecánica y Eléctrica
ada.alvarezs@uanl.mx

Resumen En muchas de las aplicaciones del mundo real que implican la resolución de problemas de rutas de vehículos, la información requerida para dar solución a los mismos no es completamente conocida al comienzo del horizonte de planificación, sino que se revela a lo largo del mismo. En este contexto surgen los problemas dinámicos de rutas de vehículos, en los que pueden presentarse nuevas demandas de servicio. En este trabajo se propone un algoritmo de resolución basado en la metaheurística de búsqueda por entornos variables para resolver el Problema Dinámico de Rutas de Vehículos con Ventanas de Tiempo. Los resultados obtenidos se comparan con la literatura especializada.

Palabras clave:VRP con ventanas de tiempo, problema dinámico, VNS

1. Introducción

Si bien el Problema de Rutas de Vehículos [11] es uno de los problemas más estudiados en la literatura [10], cuando nos referimos a su aplicación en casos reales se debe tener en cuenta un aspecto muy importante, que es la disponibilidad de la información. Puede ocurrir que la información evolucione después de que los vehículos hayan iniciado su recorrido, o bien que exista incertidumbre en los datos, por ejemplo cuando la demanda real de los clientes se encuentre dentro de un rango estimado. Por ello, en los últimos años se ha prestado mayor atención a los problemas donde los datos iniciales pueden no conocerse a priori, denominados problemas dinámicos de optimización [7], [5].

De acuerdo con [2] [7], los problemas de Rutas de Vehículos se pueden clasificar como *estáticos*, si los datos de entrada (demanda, distancia entre clientes, etc.) no dependen explícitamente del tiempo, y *dinámicos*, en otro caso. Por otra parte, si toda la información del problema es conocida, entonces el problema es *determinístico*, en caso contrario, éste será *estocástico*.

En el Problema Dinámico de Rutas de Vehículos con Ventanas de Tiempo (DVRPTW, por sus siglas en inglés) , se tiene al inicio del horizonte de planificación un conjunto de clientes cuya demanda, tiempo de servicio, localización y

ventana de tiempo son conocidas. Sin embargo, conforme transcurre el tiempo, puede que lleguen nuevas solicitudes de clientes y, por tanto, sea necesario re-optimizar las rutas actuales. Es importante que las restricciones de las ventanas de tiempo de todos los clientes se cumplan (*hard-TW*), así como las de capacidad de los vehículos. Por tanto, el DVRPTW consiste en generar un conjunto de rutas donde se satisfagan todas las restricciones y demandas de los clientes con el menor coste posible.

El artículo está organizado de la siguiente manera. En la Sección 2 se describe el Problema Dinámico del Rutas de Vehículos con Ventanas de Tiempo. La metodología de solución se presenta en la Sección 3. Algunos resultados computacionales se muestran en la Sección 4. Finalmente, en la Sección 5 se resumen las conclusiones del trabajo.

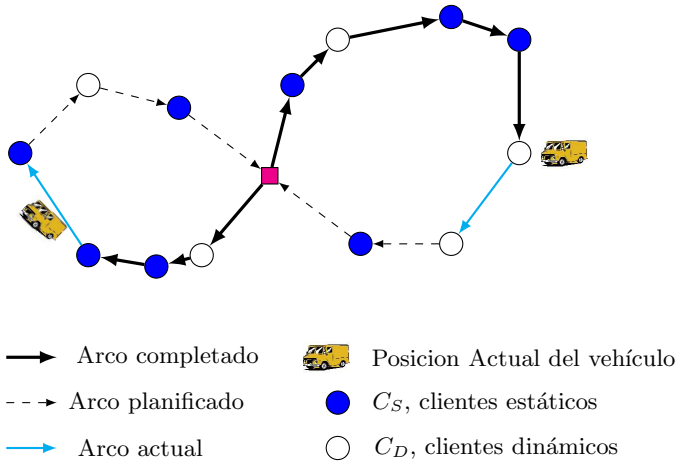
2. Descripción del problema

El problema que se aborda en este trabajo se describe a continuación. Se dispone de un grafo $G = (C, A)$, donde C es el conjunto de nodos y A el conjunto de aristas. Cada nodo representa un cliente o solicitud de servicio. Además, este conjunto incluye al depósito, denotado por el 0. N indica el número máximo de clientes y se tiene un único depósito. Sean d_{ij} y t_{ij} la distancia y el tiempo entre los nodos i y j, respectivamente. Se sabe que los clientes llegan de forma aleatoria durante el horizonte de planificación, que corresponde con la ventana de tiempo del depósito $[a_0, b_0]$. Se dispone de una flota de vehículos homogénea con capacidad Q. Cada vehículo debe partir y regresar al depósito dentro de la ventana de tiempo $[a_0, b_0]$. Cada nodo $i \epsilon C$ tiene asociada una posición geográfica (x_i, y_i), una ventana de tiempo $[a_i, b_i]$, un tiempo de servicio s_i, una demanda d_i y el tiempo de llegada de su solicitud de servicio ct_i. El conjunto de clientes se puede dividir en dos subconjuntos, C_S y C_D, de acuerdo a los requisitos del tiempo, donde C_S es el conjunto de clientes estáticos, es decir, aquellos cuya información se conoce a priori, y C_D es el conjunto de clientes dinámicos, que son aquellos cuyos parámetros no se conocen hasta que se recibe la solicitud de servicio. El operador de cada vehículo no tiene conocimiento previo del tipo de cliente asignado.

En la Figura 1 se observa un ejemplo de un Problema Dinámico de Rutas de Vehículos con Ventanas de Tiempo. Se disponen de dos vehículos para satisfacer la demanda total tanto de los clientes estáticos, C_S, como de los clientes dinámicos, C_D. En esta figura se aprecian tres tipos de arcos, *Arco completado*, *Arco planificado*, y *Arco actual*. El primer tipo de arcos representa el trayecto del vehículo recorrido para visitar a los clientes a los que ya se ha dado el servicio mientras que, en el segundo caso, los arcos representan la ruta asignada a cada vehículo y en el tercer caso, representan los arcos en los que se encuentran los vehículos en el instante de tiempo en que llega una nueva solicitud de servicio.

El objetivo del problema descrito es realizar las rutas necesarias durante el horizonte de planificación de manera que se sirva a todos los clientes, estáticos y dinámicos, a la vez que se minimiza la distancia total recorrida.

Figura 1. Ejemplo de un DVRPTW



Para dar solución a los problemas de rutas dinámicos, en la literatura podemos encontrar dos enfoques: uno basado en *re-optimización periódica* y otro basado en *re-optimización continua* [7]. En el primer caso tenemos que, al inicio del día se aplica un primer proceso de optimización con los clientes cuya información se conoce en ese momento, obteniendo un conjunto de rutas. A continuación, se va aplicando de manera periódica un proceso de optimización para resolver el problema estático correspondiente al estado actual [1]. Un inconveniente de este proceso es que la optimización se debe realizar antes del plan de rutas, lo cual aumenta el retraso para el distribuidor. En el segundo caso [4], enfoque continuo, el proceso de re-optimización se aplica cada vez que una nueva solicitud llega al sistema, por lo que es necesario que la resolución de cada subproblema estático sea realizada en el menor tiempo posible. Además, debido a que la ruta actual está sujeta a cambios en cualquier momento, los distribuidores no conocen su próximo destino hasta que no terminen el servicio de una petición.

Para cuantificar el dinamismo de un problema de rutas de vehículos, de acuerdo con Lund et. al. [6], se define el *grado de dinamismo* δ como el cociente entre el número de clientes dinámicos n_d y el número total de clientes n_{tot}:

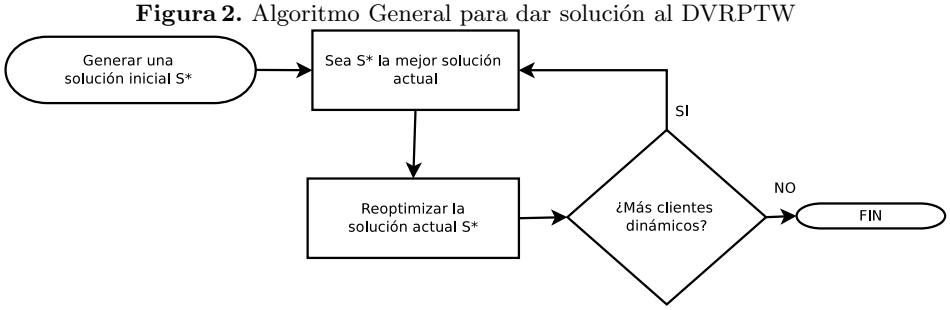
$$\delta = \frac{n_d}{n_{tot}}. \tag{1}$$

3. Metodología de Solución

En esta sección se describe la metodología propuesta para resolver el problema Dinámico de Rutas de Vehículos con Ventanas de Tiempo, la cual se basa en el enfoque de re-optimización continua. El objetivo de este problema es

minimizar la distancia total recorrida. Para ello, es necesario obtener la mejor solución factible en cada instante de tiempo en que llegan uno o varios clientes al sistema. El proceso se repite mientras aparezcan nuevos clientes dinámicos, tal como se ilustra en la Figura 2.

Figura 2. Algoritmo General para dar solución al DVRPTW



Como se observa en la Figura 2, al inicio del horizonte de planificación se genera la mejor solución que satisfaga la demanda de todos los clientes estáticos, S^*. Posteriormente, se deberá aplicar un proceso de re-optimización cuando los datos disponibles cambien, es decir, cada vez que lleguen una o varias solicitudes. Este proceso continúa mientras sigan llegando nuevos clientes. Nótese que es importante considerar el tiempo requerido para la re-optimización de las rutas, pues éste se va sumando a cada una de las rutas que se tienen actualmente, lo cual va incrementando el tiempo en que los clientes son atendidos.

Por tanto, el primer paso será generar la mejor solución factible para todos los clientes estáticos, aplicando el Algoritmo 1 descrito en la subsección 3.1. Este es un algoritmo multi-arranque, que en cada iteración construye una solución y la mejora, entregando como solución final la mejor de todas, S^*.

3.1. Creación de una Solución Inicial

Para construir una solución al problema de rutas de vehículos con ventanas de tiempo para todos los clientes estáticos, se aplica la heurística de Inserción de Solomon I1 (SIH1) [9]. Esta es una heurística de construcción secuencial de rutas. En primer lugar se elige el cliente para comenzar una ruta y, posteriormente, se utilizan reglas de inserción de clientes. Cuando no es posible insertar más clientes en la ruta actual, se empieza una nueva ruta y se repite el procedimiento hasta que no haya más clientes sin ser servidos. Posteriormente, se implementa un método de mejora, descrito en la subsección 3.2. Este proceso se repite hasta obtener una Población Inicial Pop, con $PopSize$ soluciones iniciales. Posteriormente, se elige la solución S^* con la menor distancia total recorrida.

Algoritmo 1: Algoritmo para generar la solución inicial S^*

```
   // Crear la Población Inicial, Pop
1  while (|Pop| < PopSize o hasta alcanzar un número máximo de iteraciones)
   do
2      Sea S una solución factible construida utilizando la Heurística de Inserción
       de Solomon[9];
3      Sea S* ← S;
4      // Aplicar VNS Básico (BVNS, por sus siglas en inglés)
5      while (k <= k_max) do
6          Sea k ← 1;
7          Agitación: Generar una solución S1 aleatoria utilizando el k^th entorno
           de S (S1 ∈ N_k(S));
8          Búsqueda Local: S2 ← LocalSearchDistance(S1) ;
9          if (f(S2) < f(S)) then
10             k ← 1;
11             S ← S2;
           else
12             k ← k + 1;
13     if (f(S) < f(S*)) then
           // Mejorar la Mejor Solución
14         S* ← S;
15 return La Mejor Solución S*;
```

3.2. Algoritmo de Mejora de una Solución

Una vez que se ha creado un solución S aplicando la Heurística de Inserción de Solomon ($SIH1$) en la linea 2 del Algoritmo 1, el siguiente paso será aplicar un método de mejora. Se propone utilizar la metodología basada en Búsqueda por Entornos Variables (VNS, por sus siglas en inglés) [3].

La Búsqueda por Entornos Variables, propuesta por Hansen y Mladenović, se basa principalmente en cambiar sistemáticamente de estructuras de entornos dentro de la búsqueda con el fin de escapar de un óptimo local. Principalente se compone de tres fases: agitación (línea 7), búsqueda local (línea 8) y fase de movimiento (líneas $9-12$). El primer paso consiste en definir el conjunto de estructuras de entornos $\aleph_k(S)$, para $k = 1, ..., max$. En la fase de *agitación*, una solución S_1 se genera de manera aleatoria utilizando el $k-ésimo$ entorno de la solución actual S. Después, una *Búsqueda Local* es aplicada para encontrar la mejor solución S_2 en el entorno de S_1. Si $f(S_2) < f(S)$, se sustituye S por S_2. Este último proceso es denominado fase de movimiento. El esquema anterior se repite siempre que la solución actual se mejore, de lo contrario, se pasa a la siguiente estructura de entorno y se genera una nueva agitación. Este proceso se repite hasta que se cumplan algunos criterios de parada.

Las estructruras de entorno usadas en el proceso de Agitación del VNS se usan en el orden utilizado en [8] y se resumen a continuación:

1. ***GENI***. Dadas las rutas $(0, ..., i-1, i, i+1, ..., 0)$ y $(0, ..., j, j+1, ..., k-1, k, k+1, ..., 0)$, un cliente i de una ruta se inserta en otra ruta entre los clientes j y k, obteniendo así las rutas $(0, ..., i-1, i+1, ..., 0)$ y $(0, ..., j, i, k, j+1, k-1, k+1, ..., 0)$.
2. ***Or-Opt***. Dada una ruta $(0, ..., i-1, i, i+1, i+2, ..., j, j+1, ..., 0)$, supongamos que se quieren reubicar los clientes i e $i+1$ entre los clientes j y $j+1$. Entonces, sin modificar la orientación de la ruta se deben reemplazar los arcos $(i-1, i)$, $(i+1, i+2)$ y $(j, j+1)$ por los arcos $(i-1, i+2)$, (j, i) y $(i+1, j+1)$.
3. ***Cross***. La idea de este movimiento consiste en intercambiar dos arcos de una ruta, por ejemplo, $(i-1, i)$ y $(k, k+1)$ por los arcos $(j-1, j)$ y $(l, l+1)$ de otra ruta.
4. $2-$***opt***. Sea $(0, ..., i, i+1, ..., j, j+1, ..., 0)$ una ruta y supongamos que elegimos el par de arcos $(i, i+1)$ y $(j, j+1)$ y que nos conviene reemplazarlos por los arcos (i, j) y $(i+1, j+1)$, obteniendo la ruta $(0, ..., i, j, ..., i+1, j+1, ..., 0)$. Note que al realizar este movimiento tenemos que invertir la dirección de los clientes entre $i+1$ y j.
5. $(0, 1)$ ***Relocated***. Este movimiento consiste en reubicar a un cliente i entre los clientes j y $j+1$; es decir, dada una ruta $(0, ..., i-1, i, i+1, ..., j, j+1, ..., 0)$, se reemplazan los arcos $(i-1, i)$, $(i, i+1)$ y $(j, j+1)$ por $(i-1, i+1)$, (j, i) y $(i, j+1)$ dentro de una ruta, obteniendo la nueva ruta $(0, ..., i-1, i+1, ..., j, i, j+1, ..., 0)$.
6. $(1, 1)$ ***Interchange***. Sean i y j los clientes a intercambiar de la ruta $(0, ..., i-1, i, i+1, ..., 0)$ y de la ruta $(0, ..., j-1, j, j+1, ..., 0)$, respectivamente. Se reemplazan los arcos $(i-1, i)$, $(i, i+1)$, $(j-1, j)$ y $(j, j+1)$ por $(i-1, j)$, $(j, i+1)$, $(j-1, i)$ y $(i, j+1)$ dentro de cada una de las rutas, obteniendo las dos nuevas rutas $(0, ..., i-1, j, i+1, ..., 0)$ y $(0, ..., j-1, i, j+1, ..., 0)$.

Otro de los componentes para el VNS, es la *Búsqueda Local*. En nuestra implementación se define un entorno aplicando el movimiento la búsqueda que hemos denominado *LocalSearchDistance*, que implementa un movimiento $(1, 1)$ *Relocated*, es decir, un cliente se retira de su posición actual y se inserta en otra ruta, donde la función guía de esta búsqueda es la distancia total recorrida.

Finalmente, tras aplicar el proceso general VNS, se obtiene la mejor solución inicial S^*.

3.3. Proceso de Re-optimización de la solución actual

Partiendo de una solución factible S^*, supongamos que en el instante de tiempo ct_i llega una nueva solicitud de servicio correspondiente al cliente i. El proceso a seguir se explica en lo que sigue.

En primer lugar, se debe conocer la posición actual de cada uno de los vehículos. Si el vehículo está ubicado sobre un arco que une dos clientes, se prioriza que el vehículo atienda el servicio del cliente al que se dirige, denominado cliente actual.

Una vez que se han determinado los clientes a los que se les va a brindar el servicio, el siguiente paso será insertar el nuevo cliente de acuerdo a la cercanía del mismo con cada vehículo. Es decir, primero se intentará insertar el nuevo cliente en la ruta más cercana según la distancia euclideana. Si no fuera factible, entonces se trata de insertar en la siguiente con menor distancia y así sucesivaemtne para cada una de las rutas restantes. Las reglas de inserción de un nuevo cliente consideradas en este trabajo se aplican en el siguiente orden:

1. Insertar el nuevo cliente en la mejor posición factible de la ruta.
2. Generar una nueva ruta para dar servicio al nuevo cliente, considerando que el inicio de servicio de la ruta, que parte del depósito, es ct_i.
3. Si el vehículo se encuentra esperando a dar servicio al cliente actual, tratar de insertar al nuevo cliente antes del cliente actual.
4. Se intentará insertar el nuevo cliente en la primera posición factible de la ruta y, si tras haber agregado la nueva solicitud, se incurriera en infactibilidades para algunos clientes posteriores de la ruta, entonces se generaría una nueva ruta que parte del depósito a partir del cliente donde se genera la primera infactibilidad.

Una vez que se ha agregado el nuevo cliente a una de las rutas actuales o bien a una nueva ruta, el siguiente paso será re-optimizar las rutas considerando sólo aquellos clientes que aún no han sido atendidos. Para ello, se aplica el Algoritmo de Mejora, basado en VNS, anteriormente descrito y se considera el tiempo requerido para este proceso. Posteriormente se agrega a el tiempo de comienzo de cada cliente aún no visitado.

4. Experiencia Computacional

El objetivo de la experiencia computacional preliminar mostrada en esta sección es el análisis del comportamiento de la estrategia propuesta en este trabajo. Los resultados computacionales se obtienen para instancias de la literatura que no corresponden exactamente con el problema real que se pretende abordar. Sin embargo, para estudiar la efectividad de la estrategia, se seleccionan las instancias de la literatura propuestas por Lackner [4], las cuales se basan en las de Solomon [9] de 100 clientes. De ellas se eligen las instancias del tipo $C1$, donde las coordenadas de los clientes se generan por grupos y cuyas ventanas de tiempo son cortas. Respecto al grado de dinamismo, se consideran los valores 0,10, 0,30, 0,50, 0,70 y 0,90, que corresponden a porcentajes del 10 % al 90 % de clientes dinámicos. La experiencia computacional se realizó en un ordenador Intel Core con 3.404 GHz y 4 GB RAM, bajo Ubuntu 11.10.

En necesario mencionar que el problema estándar de la literatura presentado por [4] y descrito en este trabajo constituye la base del problema real considerado. En el problema estándar se permite el intercambio de clientes entre cualesquiera dos rutas durante la re-optimización de las mismas a lo largo del horizonte de planificación. Sin embargo, en el problema real que interesa a los autores, no se permitirá el intercambio de clientes entre cualesquiera dos vehículos, sino

tan sólo entre vehículos de la misma categoría. El problema real sobre el que se aplicará el algoritmo implementado en este trabajo considerará vehículos de distinta categoría; algunos serán vehículos frigoríficos que repondrán máquinas de bebidas frías, mientras que otros contendrán alimentación para máquinas de alimentos de distinto tipo.

Los resultados obtenidos se resumen en la Tabla 1. La primera columna muestra el grado de dinamismo, δ. En las siguientes columnas se muestran los promedios de la distancia total recorrida por los vehículos, $TotD$, del número total de vehículos, NV, del tiempo de CPU en segundos, $Time(s)$, y del número de clientes no atendidos, RS. Estos datos se muestran para el algoritmo propuesto en este trabajo, y también para los mejores resultados de la literatura, resumidos en [4].

Cuadro 1. Resultados para las instancias C1

		Estrategia propuesta				*Hong*(2012)			
	δ	$TotD$	NV	$Time(s)$	RS	$TotD$	NV	$Time(s)$	RS
$C1$	10	868.22	7.8	777.77	4.55	895.77	10.56	622.64	0.22
	30	976.12	9.11	429.62	3.8	962.08	10.56	927.99	0.33
	50	1033.65	10.22	707.4	2.88	1001.18	10.89	985.87	0.22
	70	993.63	11.33	620.37	1.22	1031.68	10.78	768.65	0.22
	90	1040.74	12	615.5	0.66	1039.77	10.78	594.37	0.22

Observando los resultados de la Tabla 1, podemos concluir lo siguiente:

- Comparando la distancia total recorrida y el número de vehículos utilizado para $\delta = 10$, con el algoritmo propuesto se obtiene un menor valor para ambos casos. Sin embargo, este hecho puede deberse a que el número de clientes no atendidos es mayor que el obtenido en el trabajo de Hong (2012).
- Para $\delta = 30$ y $\delta = 50$, tanto el promedio de los vehículos utilizados como el tiempo de cómputo son menores en nuestro caso. Sin embargo, el número de clientes no atendidos es mayor. Además, para ambos valores de grado de dinamismo, la distancia total recorrida es mayor que la obtenida en el trabajo de Hong (2012).
- Comparando la distancia total recorrida y el tiempo de cómputo para $\delta = 70$, obtenemos mejores resultados, pero tanto el promedio de número de vehículos como el número de clientes no atendidos es mayor para este trabajo.
- Finalmente, para $\delta = 90$ con la metodología propuesta en este trabajo se obtienen mayores valores promedios en todas las variables bajo estudio.

Cabe mencionar que la instancia número dos del tipo $C1$ bajo estudio, denotada por $C102$, es la única en la que hay clientes no atendidos, RS, para todos los valores del grado de dinamismo.

Los resultados mostrados son preliminares dado que sólo se tienen para el grupo de instancias $C1$. Nótese que, para realizar comparativas con los resultados obtenidos por Hong (2012), hay que calcular las medias sobre todas las

instancias de cada grupo. A pesar de que la calidad de las soluciones presentadas para las instancias de la literatura no alcanza la obtenida por Hong, note que la estrategia desarrollada en este trabajo se aplicará a un problema diferente al de Hong, lo que constituirá la continuación del presente trabajo. Por lo tanto, el algoritmo propuesto aún debe ser ajustado para alcanzar resultados sin diferencias estadísticas con respecto a los de la literatura para el problema estándar.

5. Conclusiones

En este trabajo se propone una estrategia de solución para resolver el Problema Dinámico de Rutas de Vehículos con Ventanas de Tiempo. Suponemos que la información se va conociendo conforme aparecen nuevas solucitudes de clientes y, por tanto, se aplica un proceso de re-optimización continuo. En este trabajo, el VRPTW dinámico se descompone en una serie de VRPTWs estáticos. El algoritmo que resuelve cada uno de estos problemas se basa en la metaheurística de búsqueda por entornos variables. Los resultados computacionales muestran que el algoritmo propuesto es factible para obtener soluciones de calidad para el VRPTW dinámico. Aunque no alcanzan los niveles de calidad de las soluciones de la literatura para las instancias estándard, la estrategia desarrollada en este trabajo se aplicará, en trabajos futuros, a un problema real basado en el que se presenta en este trabajo.

Agradecimientos Este trabajo ha sido parcialmente financiado por el Ministerio de Economía y Competitividad de España (proyecto TIN2012-32608). Alondra De Santiago agradece a CONACYT por el apoyo de la beca de posgrado.

Referencias

1. Zhi-Long Chen and Hang Xu. Dynamic column generation for dynamic vehicle routing with time windows. *Transportation Science*, 40(1):74–88, February 2006.
2. Gianpaolo Ghiani, Francesca Guerriero, Gilbert Laporte, and Roberto Musmanno. Real-time vehicle routing: Solution concepts, algorithms and parallel computing strategies. *European Journal of Operational Research*, 151(1):1 – 11, 2003.
3. Pierre Hansen, Nenad Mladenović, and José A. Moreno Pérez. Variable neighbourhood search: methods and applications. *Annals of Operations Research*, 175(1):367–407, 2010.
4. Lianxi Hong. An improved lns algorithm for real-time vehicle routing problem with time windows. *Comput. Oper. Res.*, 39(2):151–163, February 2012.
5. Mostepha R. Khouadjia, Briseida Sarasola, Enrique Alba, Laetitia Jourdan, and El-Ghazali Talbi. A comparative study between dynamic adapted {PSO} and {VNS} for the vehicle routing problem with dynamic requests. *Applied Soft Computing*, 12(4):1426 – 1439, 2012.
6. K. Lund, O. Madsen, and J. Rygaard. Vehicle routing problems with varying degrees of dynamism. Technical report, IMM Institute of Mathematical Modelling, 1996.

7. Victor Pillac, Michel Gendreau, Christelle Guéret, and Andrés L. Medaglia. A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1):1 – 11, 2013.
8. PanagiotisP. Repoussis, DimitrisC. Paraskevopoulos, ChristosD. Tarantilis, and George Ioannou. A reactive greedy randomized variable neighborhood tabu search for the vehicle routing problem with time windows. In Francisco Almeida, MaríaJ. Blesa Aguilera, Christian Blum, José Marcos Moreno Vega, Melquíades Pérez Pérez, Andrea Roli, and Michael Sampels, editors, *Hybrid Metaheuristics*, volume 4030 of *Lecture Notes in Computer Science*, pages 124–138. Springer Berlin Heidelberg, 2006.
9. MM Solomon. Algorithms for the Vehicle-Routing and Scheduling Problems with Time Window Constraints. *Operations Research*, 35(2):254–265, 1987.
10. Paolo Toth and Daniele Vigo, editors. *The vehicle routing problem*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2001.
11. T. Vidal, T.G. Crainic, M. Gendreau, and C. Prins. Heuristics for multi-attribute vehicle routing problems: A survey and synthesis. *European Journal of Operational Research*, In press, 2013.

Aprendizaje en problemas de optimización dinámicos: un análisis sobre el algoritmo AHMA

Jenny Fajardo Calderín[1],*Antonio D. Masegosa[2], Alejandro Rosete Suárez[1], David A. Pelta[2]

[1] Departamento de Inteligencia Artificial e Infraestructura y Sistemas
Facultad de Informática, Instituto Superior Politécnico Jose A. Echeverría
[2] Grupo de Investigación en Modelos de Decisión y Optimización
DECSAI, CITIC, Universidad de Granada, España

Resumen Una pregunta que surge en el diseño de esquemas de aprendizaje para Problemas Dinámicos de Optimización consiste en decidir qué hacer con el conocimiento que se ha adquirido una vez que se produce un cambio: olvidarlo o utilizarlo en los cambios posteriores. En este trabajo, intentamos arrojar a la luz sobre este asunto usando el método Adaptive Hill Climbing Memetic Algorithm y el problema de la Mochila como banco de pruebas. Los resultados obtenidos muestran que la respuesta a la pregunta planteada depende de la estructura de la instancia y justifica la necesidad de realizar estudios más completos.

Keywords: problemas de optimización dinámicos, metaheurísticas, AHMA, esquemas de aprendizaje

1. Introducción

El mundo es dinámico, por lo cual todos los días nos enfrentamos a problemas cuya solución actual puede no ser buena en un futuro inmediato debido a los cambios que se producen en nuestro entorno. Los problemas de optimización no son una excepción. Este tipo de problemas rara vez son estáticos, ya que algunos de sus atributos como costo, valor objetivo o restricciones pueden verse alterados con el transcurso del tiempo, dando lugar a la aparición de los así denominado Problemas de Optimización Dinámicos (PODs) [3].

Entre los métodos propuestos para resolver PODs se encuentran: Algoritmos Evolutivos [13], métodos Multi-Swarm [1], Colonias de Hormigas [10,7] y estrategias cooperativas [6,5]. Una tendencia reciente en la resolución de POD es el uso de mecanismos de aprendizaje para adaptar la configuración del algoritmo (parámetros, operadores, etc.) durante la búsqueda [12,6,5,10,9,14]. La principal razón de esta tendencia reside en la dificultad de encontrar una configuración adecuada para el algoritmo, siendo incluso más difícil que en los problemas

* Este trabajo ha sido financiado por el Proyecto TIN2011-27696-C02-01 del Ministerio de Economía y Competitividad de España, P11-TIC-8001 del Gobierno de Andalusía (incluidos fondos FEDER de la Unión Europea).

estáticos. A pesar de esta dificultad, estudios como los anteriores muestran que el aprendizaje aumenta la robustez y el rendimiento de los métodos.

Una pregunta interesante que surge en el diseño de mecanismos de aprendizaje para PODs, es qué hacer con la información aprendida una vez que se detecta un cambio en el entorno: debemos utilizarla para los próximos cambios en el entorno u olvidarla. Aunque encontramos métodos que trabajan un sentido (volver a utilizarla) [12,9] y en otro (olvidar) [5,14], por lo que sabemos, esta cuestión no se ha abordado en la literatura.

En este trabajo, nuestro objetivo es analizar la cuestión anterior, utilizando como marco de experimentación el Adaptive Hill Climbing Memetic Algorithm (*AHMA*) [12], un método del estado del arte para los PODs, que incorpora un mecanismo de aprendizaje para adaptar la distribución de probabilidad con la que se elige entre dos operadores de búsqueda local. El problema seleccionado para el análisis es la versión dinámica del problema de la mochila.

El artículo está estructurado de la siguiente forma, en primer lugar, la Sección 2 presenta AHMA y las variantes que analizamos en este trabajo. Luego, en la Sección 3, se muestra el marco experimental, que describe el índice de referencia, medidas de rendimiento, métodos de comparación y el marco de experimentación. En Sección 4 se analiza los resultados obtenidos y, por último, la Sección 5 ofrece las conclusiones y los trabajos futuros.

2. Adaptive Hill Climbing Memetic Algorithm

El AHMA propuesto por Yang [12] fue diseñado originalmente para resolver PODs combinatorios. AHMA combina un algoritmo genético con una búsqueda local que utiliza dos operadores de acuerdo a una distribución de probabilidad que se ajusta durante la búsqueda. Inicialmente, AHMA genera una población aleatoria de pop_{size} individuos. Luego, en cada generación, el algoritmo selecciona los individuos utilizando el método de la ruleta, se aplica el operador de cruce uniforme con probabilidad p_c, realiza una mutación uniforme a los individuos resultantes con probabilidad p_m, aplica un reemplazo estado-estable donde sobreviven los mejores individuos pop_{size} entre padres e hijos, y se aplica una búsqueda local al mejor individuo de la población (élite). Si la búsqueda local obtiene una solución mejor que el individuo élite éste se sustituye. AHMA también incluye dos métodos para generar diversidad en la población actual: Adaptable Dual Mapping(ADM) y Triggered Random Immigrants (TRI).

Como se mencionó anteriormente, la búsqueda local utiliza dos operadores: Greedy Crossover Hill Climbing (GCHC) y Steepest Mutation Hill Climbing (SMHC). El GCHC aplica un operador de cruce a la solución élite y a otro individuo de la población actual que es seleccionado con el método de la ruleta. Se devuelve el mejor de los dos hijos obtenidos. Este proceso se repite *ls_size* veces. El SMHC selecciona al azar un número de bits de la solución y los invierte. Al igual que en el operador anterior, si la solución resultante mejora a la élite entonces la reemplaza, y el proceso se repite *ls_size* veces. Cada vez que se ejecuta la búsqueda local, AHMA selecciona aleatoriamente uno de estos dos operadores

de acuerdo con una distribución de probabilidad que se adapta a lo largo de la búsqueda, con el objetivo de dar una mayor probabilidad al operador que mejores resultados obtiene. El esquema de aprendizaje que ajusta la distribución de probabilidad se explica a continuación.

2.1. Esquema de aprendizaje

Sean p_{gchc} y p_{smhc} las probabilidades de aplicar GCHC y SHMC, respectivamente, donde $p_{gchc} + p_{smhc} = 1$. Al comienzo de la búsqueda, ambas probabilidades son inicializadas con 0.5, para darles las mismas posibilidades de ser seleccionados. El esquema de aprendizaje ajusta los valores de p_{gchc} y p_{smhc} en función del grado mejora (η) que producen, el cual se calcula de la siguiente forma:

$$\eta = \frac{|f_{imp} - f_{ini}|}{f_{ini}}$$

donde f_{ini} y f_{imp} es de la evaluación de la solución de élite antes y después de aplicar la búsqueda local, respectivamente. En la generación t, los valores de p_{gchc} y p_{smhc} se actualizan de la siguiente forma, donde $\triangle$ es la influencia relativa del grado de mejora en la selección:

$$p_{gchc}(t) = p_{gchc}(t-1) + \triangle \cdot \eta_{gchc}(t)$$

$$p_{smhc}(t) = p_{smhc}(t-1) + \triangle \cdot \eta_{smhc}(t)$$

$$p_{gchc}(t) = \frac{p_{gchc}(t)}{p_{gchc}(t) + p_{smhc}(t)} \qquad p_{smhc}(t) = 1 - p_{gchc}(t)$$

2.2. Variantes de AHMA a estudiar

Con el objetivo de estudiar lo que debemos hacer con la información que aprendió cuando ocurre un cambio en el entorno, en este trabajo se analizan tres variantes del AHMA:

- AHMA Sin Aprendizaje (AHMASA): p_{gchc} y p_{smhc} se fijan a 0.5 durante todo el proceso de búsqueda.
- AHMA Reinicio (AHMAR): p_{gchc} y p_{smhc} se resetean a 0.5 después de cada cambio, con lo que "olvidan" lo aprendido.
- AHMA: p_{gchc} y p_{smhc} se adaptan, como se explica anteriormente, por lo que el aprendizaje se vuelve a utilizar después de cada cambio del entorno.

3. Estudio Experimental

En esta sección se describe el problema de prueba, la medida de rendimiento utilizada, el método de comparación y los detalles de la experimentación realizada.

3.1. Problema de la Mochila Dinámico

El problema de la mochila es un problema de optimización combinatorio NP-Duro [8]. Dado un conjunto de m elementos el problema de la mochila se describe como sigue:

$$\max\ f(x) = \sum_{i=1}^{m} p_i x_i \quad \text{sujeta a} \sum_{i=1}^{m} w_i x_i \leq C \quad x_i \in \{0,1\}\ \ i = 1 \ldots m$$

donde $x = (x_1, ..., x_m)$, $x_i = 1$ si el objeto i se selecciona o $x_i = 0$, lo contrario. Los valores p_i y w_i representan el beneficio y el peso del objeto i, respectivamente, y C es la capacidad de la mochila, que no puede ser superada. Se cree que la mochila es uno de los más fáciles entre los problema NP, pero la dificultad de las instancias, aumenta con la correlación entre pesos y beneficios [11].

Las instancias del problema de la mochila que hemos usado tienen $m = 100$ objetos, mientras que los pesos, beneficios y capacidad se definen de la siguiente forma:

$$Capacidad\ C = 0.6 \times \sum_{i=1}^{m} w_i$$

Trataremos con dos tipos de instancias, dependiendo del nivel de correlación entre beneficio y peso [11]:

Tipo	**Peso**	**Beneficio**
Sin Correlación (SC)	$w_i = U(1, 50)$	$p_i = U(1, 55)$
Correlación Débil (CD)	$w_i = U(1, 50)$	$p_i = 0.5 * w_i + U(1, 5)$

donde $U(a, b)$ es una función que devuelve un entero aleatorio entre a y b procedentes de la distribución uniforme. En este trabajo se genera una instancia por cada tipo.

Con el fin de hacer frente a las soluciones no factibles durante el proceso de optimización, se utiliza el mismo esquema de penalización que [14].

$$f(x) = \begin{cases} \sum_{i=1}^{100} p_i x_i & Si \sum_{i=1}^{100} w_i x_i \leq C \\ 10^{-10} \times (\sum_{i=1}^{100} w_i - \sum_{i=1}^{100} w_i x_i) & \text{en otro caso} \end{cases}$$

Para inducir dinamismo en el problema, utilizamos el generador XOR-POD [14]. Este XOR-POD puede generar entornos dinámicos a partir de cualquier problema binario estático utilizando un operador o-exclusivo (XOR) bit a bit. El problema se cambia cada t generaciones. Para el k-ésimo cambio, se genera incrementalmente una máscara XOR $M(k)$ de la siguiente forma:

$$M(k) = M(k-1) \oplus T(k)$$

donde $\oplus$ es el operador XOR ($a \oplus b = 1 \iff a \neq b$, y $a \oplus b = 0$ en otro caso) y $T(k)$ es una mascara binaria creada aleatoriamente con $\rho \times m$ en el k-ésimo cambio. Para el primer cambio $k = 1$, $M(1) = \{0 \ldots 0\}$. De esta forma, la población en la generación t se evalúa como:

$$f(x,t) = f(x \oplus M(k))$$

donde $k = \lceil t/\tau \rceil$ es el índice de los cambios. Una de las ventajas de este generador XOR radica en que se pueden ajustar fácilmente la frecuencia y la severidad de los cambios. El parámetro τ controla la frecuencia de cambios, mientras que ρ controla la severidad del cambio. Un mayor valor de ρ lleva a cambios más severos, mientras que una menor τ significa cambios más frecuentes.

3.2. Medida de rendimiento

Para evaluar el rendimiento de los algoritmos, hemos utilizado el "offline performance" [2] que se define de la siguiente forma:

$$F_{BG} = \frac{1}{G} \sum_{i=1}^{G} \left(\frac{1}{N} \sum_{j=1}^{N} F_{BG_{ij}} \right)$$

donde G es el número de generaciones (por ejemplo, $G = 10 \times \tau$), N es el total de ejecuciones y $F_{BG_{ij}}$ es la mejor evaluación de la generación i y ejecución j.

3.3. Configuración de la experimentación

Se realizaron 30 ejecuciones, con 100 cambios de la función objetivo, en cada configuración del problema, período, severidad y variante de algoritmo. En los experimentos se consideraron diferentes periodos de cambio ($\tau \in \{10, 25, 50, 75, 100\}$) y severidades ($\rho \in \{0.1, 0.2, 0.5, 0.9\}$). El ajuste de los parámetros para la AHMA se configuró de acuerdo con lo propuesto en Wang [12].

3.4. Métodos de Comparación

Dado que los resultados para cada posible configuración de variante del algoritmo, instancia, severidad y frecuencia de cambio serían difícil de interpretar si se muestra como una tabla con valores numéricos, se utilizó un esquema de ranking (SRCS), propuesto en [4].

Esta técnica utiliza tests no paramétricos (Kruskal-Wallis y Mann-Whitney Wilcoxon con corrección de Holm's para comparaciones múltiples) para evaluar la significación estadística de las diferencias individuales entre cada par de algoritmos en todas las configuraciones del problema. Si la prueba concluye que hay suficiente evidencia estadística de diferencia de rendimiento, el algoritmo con el mayor offline performance suma 1 a su ranking, y el otro suma -1. En caso de empate, ambos reciben un 0. El rango de los valores en el ranking de n algoritmos para cualquier problema es $[-n+1, n-1]$. Cuanto más alto sea el ranking obtenido, mejor se puede considerar un algoritmo en relación con los otros. A cada valor de ranking se asocia un color, utilizando blanco para el valor más alto del ranking $(n-1)$ y el valor más oscuro para el más bajo $(-n+1)$. Los colores para los valores intermedios se ajustan proporcionalmente.

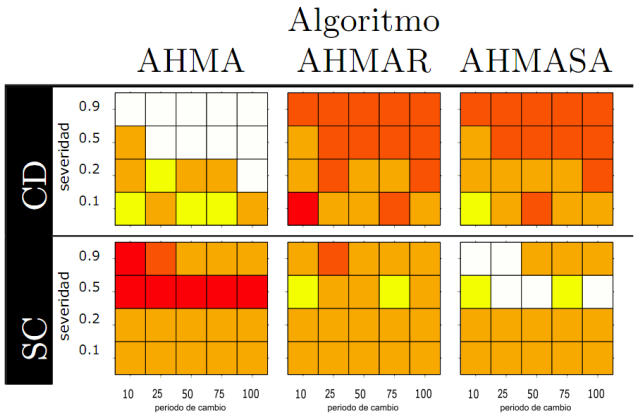


Figura 1: Ranking de los algoritmos AHMA, AHMAR, AHMASA en cada instancia y escenario evaluado. Colores claros representan mejores resultados que colores oscuros.

Dada una instancia del problema, se construye un ranking para cada variación de frecuencia y severidad. Si agrupamos las gráficas de un algoritmo para un problema dado con cada combinación de frecuencia y severidad posible, podemos obtener una matriz de color, donde es fácil observar cómo el algoritmo funciona para ese problema específico. El color blanco en una celda dada indica que el algoritmo es estadísticamente mejor que todos los otros algoritmos para la configuración de un problema específico. Si el color es más oscuro significa que el algoritmo es estadísticamente igual o peor que otros algoritmos para esa configuración del problema.

4. Resultados

4.1. La influencia del aprendizaje

Los resultados sobre el comportamiento de las diferentes variantes del algoritmo aparecen en la Figura 1, obtenida mediante la técnica SRCS [4]. Para analizar los resultados, se debe comparar el color de la misma celda en las matrices de cada método para un determinado problema. Por ejemplo, si queremos ver cual es el mejor método para la instancia mochila SC con severidad 0.9 y frecuencia 10, tenemos que comparar el color de la casilla correspondiente (la de la esquina superior izquierda) en las tres matrices que muestran los resultados de cada método en dicha instancia. En este caso, AHMSA, AHMAR y AHMA tienen los colores blanco, naranja y rojo, respectivamente. Esto significa que para esa configuración de la mochila SC, AHMASA es el mejor método, seguido por AHMAR, mientras que AHMA es el que tiene un peor rendimiento.

Primero vamos a analizar los resultados desde el punto de vista de aprendizaje vs. no aprendizaje.

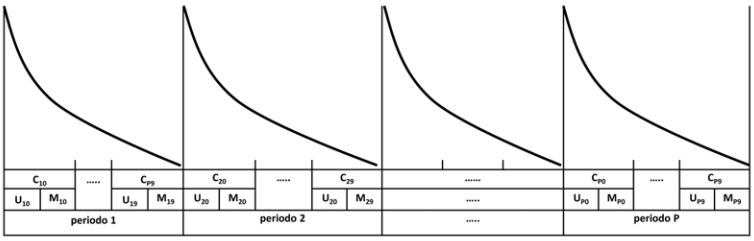


Figura 2: Esquema de recogida de datos para el análisis del comportamiento de un operador. En cada punto de control C_{pk} de cada período, se recogen dos medidas U_{pk} y M_{pk}.

El algoritmo original AHMA obtiene, en general, mejores resultados que AHMASA para la instancia CD, sobre todo para valores altos de severidad. Sin embargo, AHMASA es mejor que AHMA en la instancia SC para valores de severidad altos. Al considerar AHMAR vs. AHMASA, podemos observar que ambos métodos tienen un comportamiento muy similar en los dos casos de prueba (y en todas las configuraciones). Una vez más, sobre la instancia SC y teniendo en cuenta los valores más altos de severidad, AHMASA es mejor que la alternativa que incorpora aprendizaje (AHMAR).

Finalmente vamos a comparar las dos variantes que emplean aprendizaje. Recordemos que la diferencia entre ellos es qué hacen con el aprendizaje adquirido después de un cambio. En un esquema (AHMAR), todo se olvida, mientras que en el otro, no se toma ninguna acción (AHMA) y el aprendizaje se acumula de un cambio a otro. Es evidente que olvidar no es una buena manera de proceder cuando se considera la instancia CD (AHMA es mejor que AHMAR). Sin embargo, en la instancia SC, AHMAR es mejor que AHMA, especialmente para niveles de severidad altos.

Desde un punto de vista global, es evidente que el aprendizaje obtiene mejores resultados cuando el problema tiene cierta estructura. En la instancia SC, no existe correlación entre pesos y beneficios de los objetos, lo que provoca que los algoritmos que usan aprendizaje no sean capaces de obtener información útil del efecto de los buscadores locales.

4.2. Análisis de la evolución del aprendizaje

En esta sección vamos a tratar de proporcionar información del comportamiento de cada operador y su influencia en el mecanismo de aprendizaje. Para los experimentos, se utilizaron las dos instancias comentadas anteriormente, pero hemos elegido un escenario específico con frecuencia de cambio $\tau = 50$ y severidad $\rho = 0.5$. El objetivo es analizar la frecuencia de uso y la eficacia de cada operador de búsqueda local en la evolución a lo largo del tiempo.

En la Figura 2 se muestra un diagrama del cálculo de ambas medidas. Cada período en el que el problema puede ser considerado como estático se dividió en

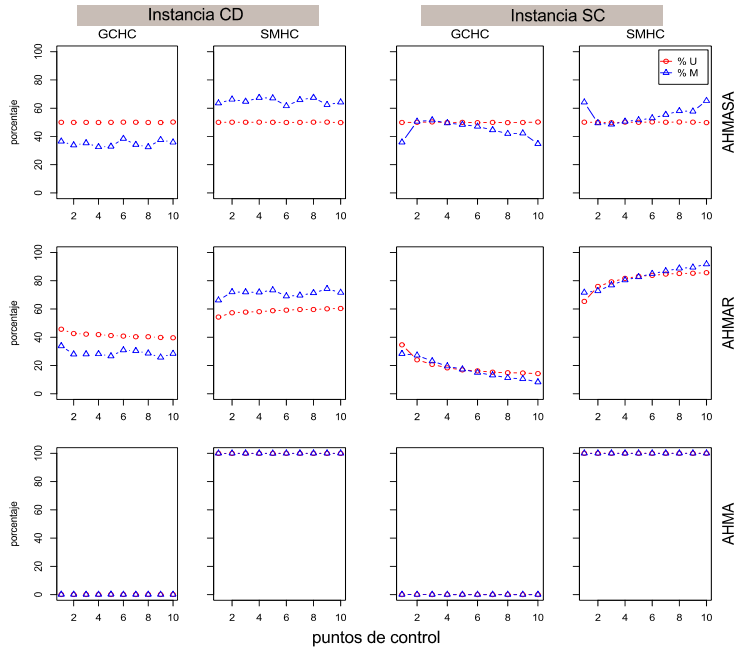


Figura 3: Comportamiento de los operadores SMHC y GCHC. Las series con círculos indican el porcentaje promedio de uso del operador mientras que las series con triángulos, el porcentaje promedio de mejoras obtenidas.

10 puntos de control distribuidos uniformemente. Consideramos C_{pk} como el punto de control $k-$ésimo en el período p. En cada punto de control C_{pk} se tomaron dos medidas para cada operador (en la Fig. 2 se muestra un solo operador) donde U_{pk} y M_{pk} representan el porcentaje de veces que ha sido seleccionado y el porcentaje de las mejoras de la solución que produjo hasta dicho punto de control. Las medidas se promediaron para cada punto de control k, para todos los periodos (P) y ejecuciones (N), de la siguiente forma:

$$\%U^k = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{P}\sum_{p=1}^{P} U_{pk}\right) \quad \%M^k = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{P}\sum_{p=1}^{P} M_{pk}\right)$$

En las Figura 3 se muestran los resultados para las instancias CD y SC. Cada fila corresponde a un algoritmo, mientras que las columnas describen las medidas para GCHC y SMHC en ese orden. La serie con círculos representa el porcentaje de veces que se ha utilizado y la serie con puntas triangulares el porcentaje de mejoras de la mejor solución que produjo.

Como se puede observar, la primera columna corresponde al esquema de "no aprendizaje" (AHMASA), es decir, la probabilidad de selección es constante a lo largo de la ejecución y por lo tanto el porcentaje de uso es de 50 % para cada operador. Esta línea sirve para ilustrar cómo cada operador contribuye a

la mejora y, a ser posible, esta curva debería ser capturada por el mecanismo de aprendizaje.

En el caso de la instancia SC, el funcionamiento de los algoritmos fue (de mejor a peor) AHMA > AHMAR = AHMASA. Los gráficos de AHMASA muestran que la mayoría de las mejoras ($60 - 70\,\%$) son producidas por el operador SMHC. Cuando se utiliza AHMAR, las probabilidades de selección se reinician después de un cambio, y podemos observar que el algoritmo es capaz de aumentar ligeramente el uso del operador SMHC, teniendo así más mejoras. El AHMA original, promueve rápidamente el uso del operador de SMHC porque es el que proporciona la mayor parte de las mejoras, y por lo tanto, obtiene crédito muy rápidamente. En resumen, el método utiliza simplemente el mejor método para el problema en cuestión todo el tiempo.

El caso de la instancia SC, es diferente desde varios puntos de vista. En primer lugar, el rendimiento de los métodos es AHMASA > AHMAR > AHMA, indicando así que el aprendizaje no es beneficioso en absoluto. En AHMASA podemos ver tres comportamientos temporales diferentes. Al inicio SHMC ofrece más mejoras que GCHC. Entre los puntos de control 2 al 5, los operadores se comportan de forma similar. En la última mitad del período, el porcentaje de mejoras proporcionado por SMHC claramente aumenta. A diferencia de los casos anteriores, AHMAR proporciona mejores resultados que AHMA. El algoritmo AHMAR promueve el uso de SMHC, que muestra una relación clara entre el uso y las mejoras proporcionadas por cada operador.

AHMA se comporta exactamente de la misma forma que en los casos anteriores y no puede hacer uso de las mejoras que GCHC puede proporcionar. En este ejemplo, ni AHMAR ni AHMA capturan las fluctuaciones en el rendimiento de los dos operadores, y, como consecuencia, no pueden adaptarse correctamente a la instancia por lo que conducen peores resultados.

5. Conclusiones y Trabajos Futuros

En este trabajo se ha analizado un cuestión importante que se plantea en el diseño de esquemas de aprendizaje para PODs: nos olvidamos del aprendizaje adquirido una vez que se produce un cambio o se utiliza para el nuevo entorno. Con este fin, hemos utilizado AHMA, un algoritmo de estado del arte para PODs binarios que incorpora un mecanismo de aprendizaje. En concreto, se analizaron tres variantes del algoritmo original: AHMASA, AHMAR y AHMA. La primera variante no realiza el aprendizaje, la segunda se olvida del aprendizaje después de los cambios y en la tercera se acumula el aprendizaje durante todo el proceso de búsqueda.

Los experimentos se llevaron a cabo en dos instancias de una versión dinámica del problema de la mochila. Veinte configuraciones (cuatro severidades $\times$ cinco frecuencias de cambio) fueron consideradas para cada uno de los casos. Los resultados obtenidos mostraron que AHMAR tiene resultados similares o incluso peores que AHMASA. Por otra parte se vio que AHMA tiene un comportamiento generalmente mejor que las otras dos variantes cuando el espacio de

búsqueda tiene cierta estructura. Por el contrario, cuando el espacio de búsqueda no está estructurado, AHMAR y AHMASA obtuvieron resultados similares o mejores que AHMA.

En términos generales, los resultados muestran que la respuesta a la pregunta que nos planteamos en este trabajo depende más de la estructura de la instancia que de otros factores. Dado que el estudio se ha centrado sólo en un problema, un método y un sistema de aprendizaje relativamente simple, está claro que se hace necesario un análisis más profundo para poder generalizar estas conclusiones.

Referencias

1. T. Blackwell and J. Branke. Multiswarms, exclusion, and anti-convergence in dynamic environments. *IEEE Transactions on Evolutionary Computation*, 10(4), 2006.
2. J. Branke and H. Schmeck. Designing evolutionary algorithms for dynamic optimization problems. In *Advances in Evolutionary Computing*, Natural Computing Series, pages 239–262. 2003.
3. C. Cruz, J. R. González, and D. A. Pelta. Optimization in dynamic environments: a survey on problems, methods and measures. *Soft Computing*, 15(7):1427–1448, 2011.
4. I. G. del Amo, D. A. Pelta, J. R. Gonzalez, and A. D. Masegosa. An algorithm comparison for dynamic optimization problems. *Applied Soft Computing*, 12(10):3176 – 3192, 2012.
5. J. R. Gonzalez, C. Cruz, I. G. del Amo, and D. A. Pelta. An adaptive multiagent strategy for solving combinatorial dynamic optimization problems. In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2011)*, volume 387 of *Studies in Computational Intelligence*, pages 41–55, 2012.
6. J. R. Gonzalez, A. D. Masegosa, and I. G. del Amo. A cooperative strategy for solving dynamic optimization problems. *Memetic Computing*, 3:3–14, 2011.
7. M. Guntsch, M. Middendorf, and H. Schmeck. An ant colony optimization approach to dynamic tsp. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 860–867, 2001.
8. H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack problems.* Springer Verlag, 2004.
9. L. Liu, D. Wang, and W. Ip. A permutation-based dual genetic algorithm for dynamic optimization problems. *Soft Computing*, 13(7):725–738, 2009.
10. M. Mavrovouniotis and S. Yang. A memetic ant colony optimization algorithm for the dynamic travelling salesman problem. *Soft Computing*, 15(7):1405–1425, 2011.
11. D. Pisinger. Where are the hard knapsack problems? *Computers and Operations Research*, 32(9):2271 – 2284, 2005.
12. H. Wang, D. Wang, and S. Yang. A memetic algorithm with adaptive hill climbing strategy for dynamic optimization problems. *Soft Computing*, 13(8-9):763–780, 2009.
13. S. Yang, Y. Ong, and Y. Jin. *Evolutionary Computation in Dynamic and Uncertain Environments*, volume 51 of *Studies in Computational Intelligence.* 2007.
14. S. Yang and X. Yao. Experimental study on population-based incremental learning algorithms for dynamic optimization problems. *Soft Computing*, 9(11):815–834, 2005.

Algoritmo Genético Empotrado en Dispositivos Móviles para Control de Conectividad en Redes Inalámbricas Heterogéneas

María D. Jaraíz-Simon, Juan A. Gómez-Pulido, Miguel A. Vega-Rodríguez

Departamento de Tecnologías de Computadores y Comunicaciones
Universidad de Extremadura, España
{mariajaraiz,jangomez,mavega}@unex.es

Abstract. Un aspecto muy importante de las comunicaciones modernas trata del acceso a redes inalámbricas heterogéneas por parte de dispositivos móviles, acceso que es decidido según la calidad del servicio y las preferencias del usuario para conmutar a la mejor red disponible. En este contexto dinámico, es esencial una adecuada gestión de los recursos de las distintas redes, lo cual constituye un problema de optimización que ha de ser resuelto. No obstante, los aspectos de movilidad y reducido tamaño de los dispositivos obligan a los algoritmos que resuelven el problema a ser ligeros y rápidos, capaces de proporcionar soluciones cercanas al tiempo real para procesadores de bajas prestaciones. Estas restricciones nos han impulsado a desarrollar algoritmos inteligentes que eviten computaciones lentas y masivas. En este artículo proponemos un algoritmo genético capaz de ser ejecutado en procesadores empotrados de bajo consumo, el cual mejora el rendimiento en comparación con el ofrecido por otros algoritmos para este mismo problema de optimización.

Keywords: Algoritmos Genéticos; Redes Inalámbricas; Vertical Handoff; Dispositivos Móviles; Calidad del Servicio.

1 Introducción

En este trabajo consideramos entornos dinámicos donde un terminal móvil puede acceder a distintas redes inalámbricas heterogéneas (UMTS, WiMax, WLAN) que actúan como puntos de acceso. Cada una de estas redes se caracteriza por los valores de sus parámetros de Calidad de Servicio (QoS). Cuando un terminal (tablet, smartphone) descubre nuevas redes, puede desconectarse de la actual para conectarse a otra según su QoS. Este es el proceso *Vertical Handoff* (VH) [1], el cual tiene tres fases: descubrimiento, decisión y ejecución. En este artículo abordamos únicamente la fase de decisión, que es fundamentalmente algorítmica.

Para seleccionar la mejor red necesitamos una métrica que proporcione el grado de bondad de cada una de ellas en un instante dado. Esta medida se calcula a partir de un conjunto de pesos asignados a los parámetros QoS, asignación que puede estar influída por las preferencias del usuario [2]. Dicha medida es dada por la función de fitness (F), calculada en la fase de decisión VH.

En un modelo simplificado con un único servicio de red, F se formula aplicando (1) [3], donde n es la red disponible, i representa un determinado parámetro QoS, w_i es su peso asociado y p_i^n es su función de coste (ln es el logaritmo neperiano). Se considera $p_i'^{(n)} = p_i^{(n)}$ si a mayor p, mayor fitness (la calidad empeora, como en el retardo o coste económico), o $p_i'^{(n)} = \frac{1}{p_i^{(n)}}$ si a mayor p, menor fitness (la calidad mejora, como en el ancho de banda). Satisfacer la restricción expresada en (1) condiciona mucho la programación de los algoritmos de optimización.

$$F^{(n)} = \sum_i w_i ln(p_i'^{(n)}) : \sum_i w_i = 1 \tag{1}$$

Hay diversos retos en la fase de decisión VH. Por un lado, si el terminal se mueve deprisa, los algoritmos deben ser rápidos para dar una solución cercana al tiempo real. Por otro lado, los algoritmos implican mucho cálculo aritmético (repeticiones del cálculo del sumatorio para todas las combinaciones de pesos), por lo que el esfuerzo computacional aumenta con la precisión requerida a las soluciones, con el número de parámetros QoS (NQ) o con el número de redes disponibles. Un alto esfuerzo computacional impide tiempos de respuesta pequeños, aún más si tenemos en cuenta los procesadores de bajo rendimiento de los terminales móviles. Nuestra investigación aborda estos retos.

2 Definición del Problema

Un conjunto de pesos produce un fitness determinado para cada red. Los fitness de las redes cambian si consideramos otro conjunto de pesos. El objetivo es obtener el fitness mínimo, que corresponderá con la red seleccionada para la fase de decisión VH. Por tanto, cuantas más combinaciones de pesos se evalúen, más posibilidades habrá de obtener un fitness mínimo. Esto es importante porque muchas aplicaciones de comunicaciones avanzadas necesitan un alto QoS [4]. La cantidad de combinaciones de pesos posibles puede ser enorme como para descartar búsquedas triviales, dado el esfuerzo computacional requerido, de aquí la necesidad de buscar algoritmos inteligentes que eviten la generación de todas las posibles combinaciones y proporcionen soluciones cercanas al tiempo real.

Los valores QoS caracterizan la red, mientras que una combinación de pesos que satisface la restricción (1) mide su calidad mediante F. Cada peso puede valer entre 0 y 1, y sólo las combinaciones que satisfacen dicha restricción son válidas o, en adelante, soluciones del problema de optimización (hay muchas más combinaciones que soluciones). Cada solución es evaluada para las redes disponibles, obteniéndose distintos fitness. Así, el problema de optimización consiste en encontrar la solución (conjunto de pesos) que, aplicada a las redes disponibles, devuelva el fitness mínimo, que corresponderá a la red óptima.

El espacio de soluciones puede ser enorme (según el número de parámetros QoS y la precisión requerida), por lo que es necesario utilizar algoritmos inteligentes. Existen algunos sencillos como *Analytic Hierarchy Process* (AHP) [5], *Simple Additive Weighting (SAW)* [6] y *Technique for Order Preference by Similarity to Ideal Solution* (TOPSIS) [7], pero que no alcanzan gran precisión.

Para tratar este problema de optimización consideramos dos intancias experimentales y tres perfiles de preferencias de usuario, a partir de datos proporcionados en [5, 7], para así validar nuestras heurísticas de optimización y comparar sus rendimientos. Las dos instancias, DS1 y DS2, se muestran en la Tabla 1:

- DS1 [5]. Consta de tres redes WLAN y una UMTS para terminales móviles que transmiten archivos de datos. El interés reside en el alto número de parámetros QoS, que piden un alto esfuerzo computacional a los algoritmos.
- DS2 [7]. Considera dos servicios, conversación y streaming, para un escenario de seis redes heterogéneas caracterizadas por cinco parámetros QoS.

Table 1: Instancias DS1 y DS2, con los parámetros QoS: B = ancho de banda (kbps), E = BER (dB), D = retardo (ms), S = nivel de seguridad, C = coste económico (eur/MB), L = latencia (ms), J = jitter (ms), R = error de emisión, A = media de retransmisiones por paquete, P = tasa de pérdida de paquete.

Net	Type	B	E	D	S	C	L	J	R	A	P
DS1											
0	UMTS	1.700	0,001	19	8	0,9	9	6	0,5	0,4	0,07
1	WLAN	2.500	10E-5	30	7	0,1	30	10	0,2	0,2	0,05
2	WLAN	2.000	10E-5	45	6,5	0,2	28	10	0,25	0,3	0,04
3	WLAN	2.500	10E-6	50	6	0,5	30	10	0,2	0,2	0,04
DS2											
0	Wi-Fi	5.100	0,01	70	2	0,2	-	-	-	-	-
1	Wi-Fi	5.100	0,01	65	1	0,2	-	-	-	-	-
2	WiMax	256	0,01	85	3	0,3	-	-	-	-	-
3	Wi-Fi	5.100	0,01	75	3	0,2	-	-	-	-	-
4	Wi-Fi	5.100	0,01	55	3	0,2	-	-	-	-	-
5	UMTS	384	0,03	80	5	0,2	-	-	-	-	-

En cuanto a las preferencias del usuario, establecemos tres perfiles:

- P1 (general). Perfil genérico, el usuario no especifica ninguna restricción, por lo que cualquier parámetro QoS puede tener un peso entre 0 y 1.
- P2 (conversación). Los parámetros más importantes son el retardo y el coste económico, restringiendo los pesos del retardo a valores entre $0,5$ y $0,7$.
- P3 (streaming). Perfil para aplicaciones multimedia. Las restricciones son para el ancho de banda (entre $0,5$ y $0,7$) y el retardo (entre $0,1$ y $0,3$).

Nuestras propuestas heurísticas parten del algoritmo SEFI, diseñado para la búsqueda directa de soluciones [8]. SEFI encuentra todas las posibles soluciones dada una precisión de búsqueda (h), y determina el esfuerzo computacional y el tamaño del espacio de soluciones. Por ejemplo, para NQ=5, $h = 0,01$, DS1 y P1, hay 100.000.000 combinaciones y 8.000.000 soluciones. SEFI explora el espacio de soluciones de forma uniforme según h: si h decrece, el número de combinaciones (y por tanto, de soluciones) aumenta. Esta búsqueda uniforme evita dejar áreas del espacio de soluciones inexploradas.

3 Propuestas de Algoritmos Inteligentes Rápidos

Hemos diseñado dos algoritmos de optimización para evitar generar y evaluar todas las posibles soluciones, como hace SEFI, y así satisfacer la restricción de reducir el tiempo de computación por debajo de un segundo. Además, los hemos diseñado ligeros de código para aplicaciones empotradas. El primero es una aproximación adaptativa y el segundo tiene una naturaleza evolutiva.

3.1 SEFISA: Propuesta de Algoritmo Adaptativo

La clave para encontrar la red óptima es aumentar NQ, que implica reducir h para no aumentar el tiempo de computación. Este equilibrio fundamenta diseñar una técnica adaptativa, SEFISA, inspirada en el algoritmo Simulated Annealing (SA) [9]. Este algoritmo busca un óptimo dentro de un espacio de soluciones; una vez encontrado, la siguiente generación reduce el espacio y lo centra en el anterior óptimo, comenzando la búsqueda de nuevo con más precisión. Las sucesivas reducciones se calculan mediante un factor de reducción (red).

Hemos utilizado el aspecto adaptativo de SA para definir SEFISA, cuyo funcionamiento se describe en el Algoritmo 1 y en la Figura 1. Inicialmente (generación #0), SEFISA ejecuta SEFI donde los espacios de búsqueda de tamaños D_i para los pesos w_i se restringen a los límites u_{min_i} y u_{max_i}, impuestos por el perfil de usuario. En esta primera generación establecemos la constante div, o factor de división, que determina en cuántas muestras se dividirá el espacio de búsqueda más pequeño D_{min}. La precisión h para SEFI (sólo para esta generación) se calcula dividiendo D_{min} (w_1 en la Figura 1) por div, de forma que algunos D_i tendrán mayor o igual número de muestras que div para generar combinaciones. Así, h es distinto en cada generación y depende de D_{min} y div.

Una vez encontrado el fitness mínimo en la generación inicial, el correspondiente conjunto de pesos (solución óptima) se utiliza para centrar en ellos los nuevos y reducidos espacios de búsqueda D'_i de la siguiente generación (generación #1 en la Figura 1), donde los D_i se reducen mediante el factor de reducción, generalmente igual a 2. Los nuevos D'_i se utilizan para calcular nuevos límites de búsqueda v_{min_i} y v_{max_i}, teniendo en cuenta que, para la primera generación, los espacios de búsqueda fueron determinados por u_{min_i} y u_{max_i} (2).

$$D'_i = \frac{D_i}{red} : (D_i = v_{max_i} - v_{min_i}) \wedge (D_{i,0} = u_{max_i} - u_{min_i}) \tag{2}$$

El cálculo de los nuevos límites a partir de D'_i ha de tener en cuenta algunas situaciones especiales. Las más habituales aparecen cuando los límites impuestos por u_{min_i} y u_{max_i} se exceden. Supongamos una generación #J donde $D_i^{(J)}$, definido por el intervalo $\{V_{min_i}^{(J)}, V_{max_i}^{(J)}\}$, está dentro del intervalo $\{U_{min_i}, U_{max_i}\}$ (que no depende de la generación). El espacio de búsqueda se reduce a la mitad en la siguiente generación #J+1, $D_i^{(J+1)} = \frac{D_i^{(J)}}{2}$, luego $V_{min_i}^{(J+1)} = wb_i^{(J)} - \frac{D_i^{(J+1)}}{2}$ y $V_{max_i}^{(J+1)} = wb_i^{(J)} + \frac{D_i^{(J+1)}}{2}$, donde $wb_i^{(J)}$ es el peso i-ésimo de la mejor solución encontrada en la generación #J. Para evitar que estos nuevos límites estén

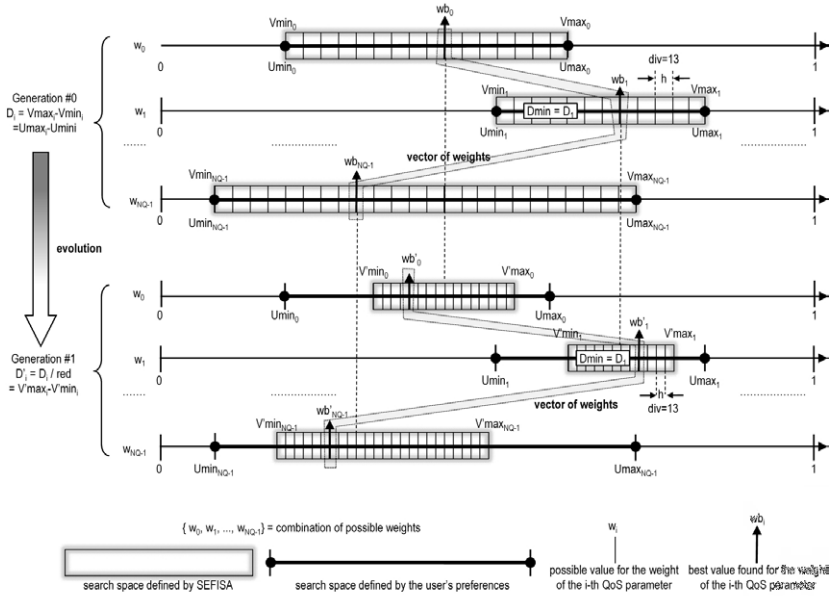


Fig. 1: Esquema de funcionamiento de SEFISA.

Algorithm 1 Pseudo-código de SEFISA.

Selecciona instancia y perfil, y determina límites $U_{min_i}, U_{max_i} \Rightarrow D_i \Rightarrow D_{min}$
Selecciona *red* y *div* $\Rightarrow h$
$IdGeneration = 0$
while criterio de parada no alcanzado **do**
  Ejecuta SEFI (h) $\Rightarrow$ obtiene óptimo wb_i
  $D_i = D_i/2$ y centrado en $wb_i \Rightarrow$ determina V_{min_i} y V_{max_i}
  if límites excedidos u otras causas **then**
    Ejecuta acciones correctoras sobre los espacios de búsqueda
  end if
  Determina $D_{min} \Rightarrow h$
  $IdGeneration{+}{+}$
end while

antes o después de los mínimos y máximos posibles U_{min_i} y U_{max_i}, hacemos: si $V_{min_i}^{(J+1)} \leq U_{min_i} \Rightarrow V_{min_i}^{(J+1)} = U_{min_i}$, y si $V_{max_i}^{(J+1)} \geq U_{max_i} \Rightarrow V_{max_i}^{(J+1)} = U_{max_i}$.

Una vez determinados los nuevos espacios de búsqueda, SEFI genera todas las combinaciones NC y determina las soluciones NS ($NS \subseteq NC$) que satisfacen la restricción ahora formulada en (3).

$$\sum_{i=0}^{i=NQ-1} w_i = 1 : (v_{min_i} \leq w_i \leq v_{max_i}) \wedge (0 \leq u_{min_i} \leq v_{min_i} \leq v_{max_i} \leq u_{max_i} - 1) \tag{3}$$

Finalmente, el nuevo óptimo encontrado se utiliza para determinar las posiciones y tamaños de los espacios de búsqueda de la siguiente generación, y así sucesivamente. Una vez alcanzado un criterio de parada en las generaciones, la red escogida será aquella a la que corresponda el fitness mínimo encontrado.

3.2 GAVH: Propuesta de un Algoritmo Genético

Hemos diseñado un algoritmo basado en los Algoritmos Genéticos (GAs) [10], llamado GAVH, que mejora los resultados de SEFISA y supera sus limitaciones, cumpliendo los planteamientos de pequeño tamaño y bajo tiempo de respuesta.

GAVH maneja poblaciones donde los individuos son soluciones del problema. La Figura 2 muestra la aproximación de GAVH al problema. La población es inicialmente generada con un número par de individuos. Cada individuo se identifica mediante $w_indiv_{[ip][iq]}$, donde ip e iq son los índices del individuo en la población y del parámetro QoS. De hecho, cada individuo tiene tantos valores de fitness como redes se encuentren. Así, el fitness de un individuo es $F[ip][in]$, donde in es el índice de la red inalámbrica. Finalmente, consideramos el mismo tamaño de población a lo largo de las generaciones, según la constante *popsize*.

Para generar la población inicial, GAVH fija valores de: NQ, NN (número de redes), h y *popsize*. SEFI genera la población inicial, porque así se garantiza una selección uniforme de individuos que cubren todo el espacio de soluciones.

El Algoritmo 2 muestra la operación de GAVH, que comienza tras definir la población inicial. Hay tres fases secuenciales con los operadores genéticos aplicándose de forma cíclica a lo largo de las generaciones (NG):

Al igual que en SEFISA, una vez alcanzado el criterio de parada, la red escogida será aquella a la que corresponda el fitness mínimo hallado.

- Selección. Los individuos se clasifican según sus fitness, seleccionándose la mitad (*popsize*/2) con mejores fitness como *padres A* (que sobrevivirán a esta generación) y el resto como *padres B*.
- Cruce. Este operador (Figura 3) empareja un padre A (wA) con un padre B (wB), en un único par cuyos individuos no son utilizados en otros pares. Después, los genes del hijo wS de cada par se generan con el promedio de los genes correspondientes de los padres, según (4). No obstante, este promedio podría no aplicarse al hijo a partir de cierta posición de gen, para asegurar la restricción (1); en ese caso, los demás genes tomarían otro valor (5).
- Mutación. El cruce es un operador exacto que da siempre el mismo resultado. Introducir mutación proporciona diversidad a la población gracias a su naturaleza aleatoria, y requerirá hacer varias ejecuciones del mismo experimento. Consideramos dos políticas de mutación posibles:

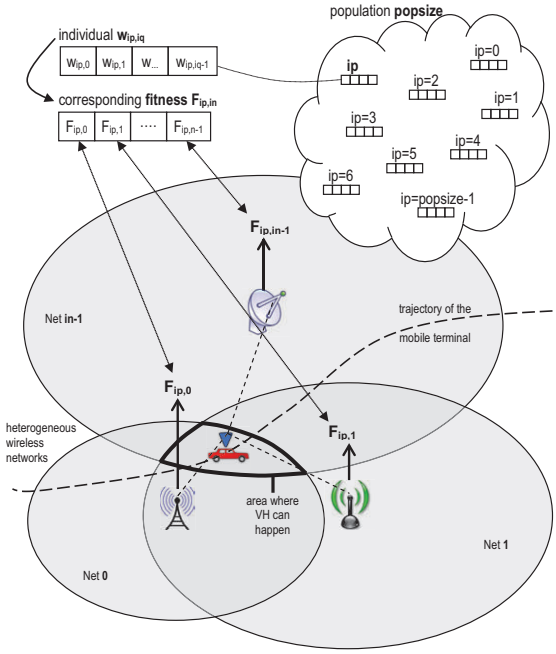


Fig. 2: Un individuo se asocia con tantos valores de fitness como redes haya.

Algorithm 2 Pseudo-código de GAVH.

Inicialización: *dataset*, $profile$, NN, NQ, h, *popsize*
SEFI genera la población inicial, y se establecen *pmut*, *smut* y *qmut*
while número de ejecuciones del experimento no alcanzado **do**
  Selección de la población inicial
  while criterio de parada no alcanzado **do**
    Fase #1: { Selección de los padres wA y wB de acuerdo con sus valores F }
    Fase #2: { Cruce de padres para generar los hijos wS y descartar padres wB }
    Fase #3: { Mutación de algunos individuos }
    Obtención de la solución óptima de la generación
    $IdGeneration$++
  end while
  Obtención de la solución óptima global
end while
Obtención de la solución óptima global promedio

- Mutación por desplazamiento. Los genes del individuo w se desplazan *qmut* posiciones a la derecha (Figura 4a). Si $qmut = 0$, no hay mutación. Este valor se fija al comienzo, y su valor máximo es NQ-1.
- Mutación por intercambio. Dos genes del individuo se intercambian según dos posiciones predefinidas de *qmut* (Figura 4b), aplicando distinta aleatoriedad en los individuos, en las generaciones o en las ejecuciones.

Los individuos mutados se escogen aleatoriamente en cada generación, según un porcentaje *pmut*. Hemos podido comprobar que es mejor tener valores pequeños de *qmut*, entre 5% y 10%, para tener buenos resultados.

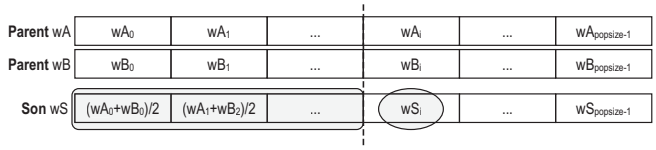
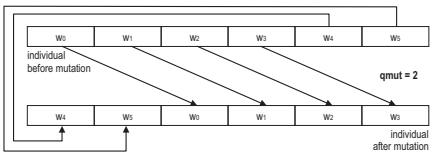


Fig. 3: Los padres wA y wB generan el hijo wS, donde wS_i es el i-ésimo gen del individuo. Este gen se crea de acuerdo con las reglas expresadas en (4) y (5).

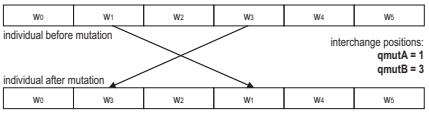
$$(\frac{wA_i + wB_i}{2} + \sum_{j=0}^{j=i-1} \frac{wA_i + wB_i}{2}) < 1 \rightarrow wS_i = \frac{wA_i + wB_i}{2} \tag{4}$$

$$(\frac{wA_i + wB_i}{2} + \sum_{j=0}^{j=i-1} \frac{wA_i + wB_i}{2}) \geq 1 \rightarrow \tag{5}$$

$$wS_k = R : R = \frac{1 - \sum_{j=0}^{j=i-1} \frac{wA_i + wB_i}{2}}{popsize - 1}, \forall k = \{i, ..., popsize - 1\}$$



(a) Desplazamiento



(b) Intercambio

Fig. 4: Tipos de mutación.

Hemos programado GAVH en ANSI C sin librerías externas para su máxima portabilidad a cualquier plataforma de procesamiento de dispositivos móviles.

GAVH usa las mismas instancias experimentales de SEFI y SEFISA, para hacer una comparación realista. De esta forma, se han efectuado muchos experimentos con distintas configuraciones del algoritmo (*popsize*, NG, *pmut*, *smut* y *qmut*) y del problema (instancia, perfil, NN y NQ). Los experimentos han permitido ajustar estos parámetros para obtener una configuración óptima. Cada experimento (configuración distinta del algoritmo y del problema) constó de 50 ejecuciones para analizar estadísticamente los resultados: fitness mínimo, máximo y medio (solución del experimento), varianza y desviación estándar.

El ajuste de los parámetros afecta a la mutación (porcentaje, estrategia y posición), tamaño de población y número de generaciones. La configuración común del problema consistió en el perfil $P1$, instancia $DS1$, cuatro redes y cuatro parámetros QoS. El ajuste involucró 21.000 ejecuciones: 4 tandas experimentales (combinaciones de h={0,01; 0,005} y $pmut$={5; 20}) x 15 tipos de experimentos (combinaciones de NG={50; 500; 2.000} y $popsize$={50; 100; 500; 1.000; 2.500}) x 7 experimentos ($smut$=0 y combinaciones de $qmut$={1; 2; 3} para $smut$=1 y $qmut$={0; 1; 2} para $smut$=2) x 50 ejecuciones. El análisis de los resultados constató que: hay un mejor comportamiento para bajos porcentajes de mutación y menores valores de h (que implican mejores precisiones para la generación de soluciones); tamaños bajos de población dan peores resultados; la mejor estrategia de mutación es por intercambio en la tercera posición; hay un ligero estancamiento de la solución óptima para más de 50 generaciones; etc. Se ha tenido en cuenta la restricción de un tiempo de computación inferior a un segundo y que la dispersión de los resultados de las distintas ejecuciones dada por las desviaciones estándar fuera baja. De esta forma, los valores ajustados fueron: NG=50, $popsize = 1.000$, $h = 0,005$, $pmut = 5$, $smut = 2$ y $qmut = 2$.

4 Comparativa de Rendimientos

Hemos comparado los rendimientos entre AHP [5] y nuestras heurísticas (SEFI, SEFISA y GAVH). El algoritmo AHP se ha aplicado con las mismas instancias experimentales para este mismo problema de optimización [7], por lo que se pueden reproducir los experimentos y comparar los resultados de forma realista. La Tabla 2 muestra las soluciones óptimas encontradas por los algoritmos para el mismo marco experimental. Cada valor de fitness proporcionado por AHP ha sido calculado utilizando (1) para la combinación de pesos dada en [7].

Table 2: Soluciones óptimas (combinaciones de pesos y sus correspondientes fitness para los perfiles $P2/P3$) calculadas por AHP, SEFI, SEFISA y GAVH para el mismo marco experimental: $DS2$, NN=6, NQ=5, considerando distintas precisiones.

Weight	QoS	AHP	SEFI h=,01	SEFI h=,005	SEFISA div=3	SEFISA div=10	GAVH h=,01	GAVH h=,005
w0	B	,065/,545	,47/,6	,485/,6	,34/,7	,44/,7	,54/,69	,68/,685
w1	E	,065/,035	,01/,28	,005/,29	,09/,1	,02/,16	,36/,13	,23/,19
w2	D	,614/,178	,05/,1	,05/,1	,05/,1	,05/,1	,02/,01	,015/,005
w3	S	,128/,121	,01/,01	,005/,005	,03/,05	,02/,02	,07/,06	,07/,02
w4	C	,128/,121	,1/,001	,005/,005	,04/,05	,02/,02	,01/,11	,005/,1
	Fitness	1,3/-4,4	-2,1/-6,0	-2,2/-2,1	-1,4/-6,2	-1,9/-6,4	-5,8/-6,5	-5,8/-6,5

Se observa en la Tabla 2 que las soluciones dadas por GAVH son mejores siempre, incluso con bajas precisiones. También GAVH mejoró el tiempo de

computación en todos los casos, con un promedio de 0,85 segundos por ejecución del algoritmo para una precisión de $h = 0,005$.

5 Conclusiones y Trabajo Futuro

En este trabajo proponemos un algoritmo genético rápido y ligero para utilizarse con terminales móviles en escenarios de conectividad basada en QoS sobre redes inalámbricas heterogéneas, donde la solución al problema de optimización ha de ser dada casi en tiempo real. Las soluciones obtenidas mejoran las obtenidas por otras heurísticas, tanto en precisión como en tiempo de computación.

El algoritmo se ha probado en un procesador de bajo consumo (Microblaze a 125MHz y consumo de 96.98 mW), y planteamos como trabajo futuro su integración en el software de conectividad inalámbrica de algún dispositivo móvil.

6 Agradecimientos

Este trabajo ha sido financiado por el Ministerio de Economía y Competitividad (proyecto TIN2012-30685) y el Gobierno de Extremadura (ayuda GR10025).

References

1. Kassar M, Kervella B, and Pujolle G. 2008. *An overview of vertical handover decision strategies in heterogeneous wireless networks.* Computer Comm 31, pp:2607-2620.
2. Song Q and Jamalipour A. 2008. *A quality of service negotiation-based vertical handoff decision scheme in heterogeneous wireless systems.* European Journal of Operational Research 191(3), pp:1059-1074.
3. McNair and Zhu F. 2004. *Vertical Handoffs in Fourth-Generation Multinetwork Environments.* IEEE Wireless Communications 11, pp:8-15.
4. Zhu F and McNair J. 2004. *Optimizations for vertical handoff decision algorithms.* Wireless Communications and Networking Conf. (WCNC 2004), IEEE, pp: 867-872.
5. Song Q and Jamalipour A. 2005. *A Network Selection Mechanism for Next Generation Networks.* IEEE Int. Conf. on Comms. (ICC 2005). IEEE, pp:1418-1422.
6. Stevens-Navarro E, Lin Y, and Wong V. 2008. *An MDP-Based Vertical Handoff Decision Algorithm for Heterogeneous Wireless Networks.* IEEE Transactions on Vehicular Technology 57(2), pp:1243-1254.
7. Lassoued I, Bonnin J, Hamouda Z, and Belghith A. 2008. *A Methodology for Evaluating Vertical Handoff Decision Mechanisms.* 7th Int Conf on Networking, pp:377.
8. Jaraiz M, Gomez J, Vega M, and Sanchez J. 2012. *Fast Decision Algorithms in Low-Power Embedded Processors for QoS Based Connectivity of Mobile Sensors in Heterogeneous Wireless Sensor Networks.* Sensors 12(2), pp:1612-1624.
9. Kirkpatrick S, Gelatt D, and Vecchi M. 1983. *Optimization by Simulated Annealing.* Science 220, pp:671-680.
10. Goldberg D. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, Reading.

Sobre el rol de la auto-adaptación en ambientes dinámicos

Pavel Novoa-Hernández[1], Carlos Cruz Corona[2], and David A. Pelta[2]

[1] Dept Matemáticas, Universidad de Holguin, Holguin 80100, Cuba
pnovoa@facinf.uho.edu.cu
[2] Dept. Ciencias de la Computación e I.A., Universidad de Granada, España
{carloscruz,dpelta}@decsai.ugr.es

Resumen En este trabajo se analiza el rol de la auto-adaptación bajo ambientes dinámicos en dos niveles: auto-adaptación de elementos del algoritmo, y ajuste de características en los individuos de la población. Para este fin, se consideran varios algoritmos multipoblacionales, basados en el método Evolución Diferencial. A partir de los resultados es posible concluir que la auto-adaptación tiene un efecto positivo en ambientes dinámicos cuando es aplicada en ambos niveles.

1. Introducción

Existen muchos escenarios reales que pueden ser modelados como problemas dinámicos de optimización (PDOs). Estos problemas se caracterizan por la naturaleza dinámica de los elementos del modelo (ej. la función objetivo, espacio de búsqueda, restricciones, etc.). En particular, en la presente investigación se consideran aquellos PDOs con la función objetivo variable en el tiempo. Formalmente, sea $\Omega^{(t)} \subseteq \mathbb{R}$ el espacio de búsqueda en el tiempo $t \in \mathbb{N}_0$, un PDO se define como la búsqueda del conjunto de soluciones óptimas $\mathcal{X}^{(t)}$ tal que:

$$\mathcal{X}^{(t)} = \left\{ \mathbf{x}_{opt}^{(t)} \ : \ f^{(t)}\left(\mathbf{x}_{opt}^{(t)}\right) \leq f^{(t)}\left(\mathbf{x}^{(t)}\right), \ \forall \mathbf{x} \in \Omega^{(t)} \right\} \tag{1}$$

donde $f^{(t)} : \Omega^{(t)} \to \mathbb{R}$ es la función objetivo en el tiempo t.

La resolución de PDOs mediante técnicas metaheurísticas ha ganado un gran interés en los últimos años,[8,17]. El éxito de estos métodos se debe en gran medida a dos razones: sus habilidades para lidiar con escenarios complejos, y segundo, las técnicas de adaptación que han sido incorporados en estos métodos para enfrentar el dinamismo de los problemas. En este último caso los enfoques existentes pueden clasificarse, de acuerdo a [12], en cuatro categorías principales: *diversidad durante la ejecución*, *diversidad después del cambio*, enfoques basados en *memorias*, y enfoques *multipoblacionales*.

Recientemente, tal como se plantea en [16]: "*otro enfoque es hacer uso de mecanismos auto-adaptativos de los EAs y otras metaheurísticas para enfrentar los cambios*". La aplicación directa de estos mecanismos, los cuales han sido originalmente diseñados para facilitar la convergencia en problemas de optimización estacionarios, no basta para resolver la dinámica de algunos PDOs (ej.

aquellos con alta severidad en los cambios). Una forma intuitiva de resolver esta limitación es aplicar auto-adaptación a algoritmos que ya cuentan con mecanismos de adaptación dinámica (ej. *diversidad durante la ejecución*), como fue mostrado con éxito en [18]. Hipotéticamente, dados que estos enfoques de adaptación están diseñados para tratar con la dinámica de los PDOs, la aplicación a su vez de modelos auto-adaptativos en este contexto debería conllevar a obtener algoritmos más robustos en estos ambientes dinámicos.

Por ello, el objetivo general de este trabajo es analizar el rol de la auto-adaptación aplicada tanto a nivel de elementos del algoritmo, como a nivel de ajuste de características en los individuos. En este sentido, esta contribución se ha estructurado de la siguiente manera: primero, en la Sección 2 se revisan brevemente los mayores progresos relacionados con la aplicación de la auto-adaptación en ambientes dinámicos. En la Sección 3 se analizan los resultados de los experimentos computacionales de varios algoritmos basados en Evolución Diferencial. Finalmente, la Sección 4 está dedicada a las conclusiones de la investigación.

2. Auto-adaptación en ambientes dinámicos

La auto-adaptation es, desde el punto de vista de la computación evolutiva, una técnica eficiente de control de parámetros. Esencialmente, según [15] "*un algoritmo auto-adaptativo controla la función de trasmisión entre los padres e hijos de la población por sí mismo y sin un control externo*".

Sin embargo, incluir parámetros estratégicos en los genotipos de los individuos no es la única forma de garantizar auto-adaptación en un algoritmo. De acuerdo a la investigación presentada por [3], es posible observar que este comportamiento también está presente de manera implícita en otras partes del algoritmo, tales como los operadores de cruzamiento de algunos Algoritmos Genéticos de codificación real (RCGAs).

En el contexto de la optimización estacionaria, la auto-adaptación ha sido ampliamente estudiada y aplicada en diversos escenarios (véase [15] y [13]). Esto no es así para la optimización en ambientes dinámicos, donde actualmente existen pocos trabajos. Es posible encontrar investigaciones pioneras [1,2], las cuales estuvieron orientadas a estudiar si la auto-adaptación, tal y como ha sido concebida para la optimización estacionaria, podría resultar eficiente en ambientes dinámicos. A partir de estas investigaciones y de otras más recientes ([7]) se puede concluir que existen escenarios en los que la auto-adaptación realmente mejora el rendimiento de los algoritmos, mientras que en otros simplemente no es suficiente (ej. en problemas muy complejos). Por ese motivo, otros investigadores han considerado la inclusión de esquemas adicionales de adaptación como el empleo de varias poblaciones. En este caso se encuentran los algoritmos sofisticados propuestos por [18,20]. Mientras que [6,20] emplean mecanismos auto-adaptativos aplicados a algoritmos clásicos , otros [11,18] utilizan modelos auto-adaptativos aplicados a algoritmos ya adaptados a PODs (por ejp. diversidad durante la ejecución). En cualquier caso, se han hecho pocos estudios de la aplicación de la auto-adaptación tanto al algoritmo en sí, como al algoritmo ya adaptado al

dinamismo mediante algunas de las técnicas explicadas anteriormente. En ese sentido, los experimentos que se describirán en la próxima Sección tienen por objetivo analizar este tema.

3. Experimentos computacionales

Los algoritmos seleccionados en este estudio están basados en el método *Evolución Diferencial* (DE), y son los siguientes: el DE estándar (sin auto-adaptación) tomado de [21] , JDE (con auto-adaptación en los parámetros) tomado de [5], y SspDE (con auto-aptación en los parámetros y las estrategias de mutación) tomado de [19]. Obsérvese que estos algoritmos son adecuados para nuestro estudio dado que poseen modelos auto-adaptativos diferentes (como se puede ver en el eje y de la Fig. 1).

La auto-adaptación en el caso de JDE ocurre en los parámetros tasa de cruzamiento (CR) y en el factor de escala (F). Básicamente, cada individuo de la población codifica en su genotipo una realización particular de estos parámetros, los cuales se inicializan de manera aleatoria al comienzo de cada ejecución en los rangos [0.0, 1.0] y [0.1,1.0], respectivamente. Luego, en cada iteración del algoritmo, las realizaciones de CR y F en los individuos serán objeto de mutación antes de ser empleadas para la creación de lo que se conoce como el *vector donante*. Estas mutaciones se realizan con tasas de $\tau_1 = \tau_2 = 0.1$.

En SspDE no solo se controlan los parámetros CR y F, sino también la estrategia de mutación para la creación del *vector donante*. Sin embargo, los individuos no contienen una sola realización de cada parámetro, sino una lista de longitud fija LP. Estas listas contienen las realizaciones que serán aplicadas durante las LP iteraciones siguientes. Luego que el algoritmo alcanza LP iteraciones, se lleva a cabo la auto-adaptación de las listas.

Por otro lado, en relación con la adición de algún mecanismo de adaptación o auto-adaptación se consideran los siguientes: el enfoque multipoblacional como adaptación para actualizar las subpoblaciones (tomado de [4] y se indica con la letra "m"); y la generación de individuos (aleatorios) de tipo *quantum* para mejorar la *diversidad durante la ejecución*, tanto como mecanismo de adaptación (tomado de [4] y se indica con la letra "Q"), como de auto-adaptación (tomado de [18] y se indica con las letras "SQ"). Véase el eje x de la Fig. 1).

Para estudiar los algoritmos considerados, se seleccionaron instancias de los estándares de comparación *Rotación* y *Composición* del generador GDBG [14], los cuales fueron empleados en la competición sobre optimización dinámica del congreso CEC'2009. Ambos generadores agrupan 49 instancias diferentes de problemas, los que se derivan de la combinación de 7 funciones y 7 tipos de cambio. Cada función posee 10 picos, con la excepción de la función $F1(50)$ que emplea 50 picos. Estas funciones se referirán de la manera siguiente: Rotación de picos ($F1$), Rotación de picos con 50 picos ($F1(50)$), Composición *Sphere* ($F2$), Composición *Rastrigin* ($F3$), Composición *Griewank* ($F4$), Composición *Ackley* ($F5$), y Composición híbrida ($F6$). En relación a los cambios, GDBG propone los siguientes: Paso corto ($T1$), Paso largo ($T2$), Aleatorio ($T3$), Caótico ($T4$),

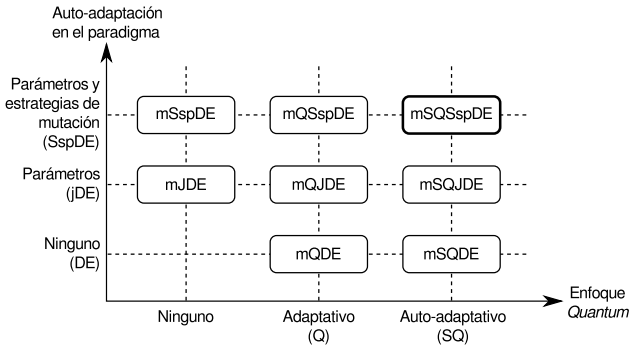


Figura 1. Algoritmos considerados en el estudio, derivados a partir de la combinación de los métodos (DE, JDE, y SspDE) y el tipo de enfoque quantum (ninguno, adaptativo y auto-adaptativo).

Recurrente ($T5$), Recurrente con ruido ($T6$), y Dimensional ($T7$). Este último tipo de cambio, varía las dimensiones del espacio de búsqueda en el rango $[5, 15]$. En general, por cada instancia de problema: Ω = [-5.0, 5.0]^{10}. El número de cambios en cada simulación fue de 60, siendo la frecuencia de estos cada 10000 evaluaciones. En consecuencia, cada ejecución terminó cuando el algoritmo alcanzó 600000 evaluaciones de la función objetivo. Como medida de rendimiento se seleccionó el *error de la mejor solución antes del cambio*, la cual fue también sugerida en la competición CEC'2009:

$$e_{bac} = \frac{1}{\mathsf{NC}} \sum_{c=1}^{NC} \left(f_{opt}^{(c)} - f_{best}^{(c)} \right) \tag{2}$$

donde NC representa el número de cambios en el problema, mientras que $f_{opt}^{(c)}$ y $f_{best}^{(c)}$ son los valores de la función objetivo en el óptimo del problema y la mejor solución del algoritmo antes del cambio c, respectivamente.

3.1. Análisis de los resultados

Los resultados de los algoritmos considerados se analizan estadísticamente siguiendo la metodología propuesta en [10] en el contexto de la optimización estacionaria, y que propone la aplicación de pruebas no paramétricas. Esta metodología puede ser aplicada en este caso dado que se basa en los valores agregados de varios algoritmos en diversos problemas. Específicamente, se ha divido el estudio en tres grupos de acuerdo al tipo de función empleada por las instancias de problema. En el primer grupo aparecen las instancias con funciones unimodales (F1, F1(50) y F2), mientras que en el segundo las funciones multimodales (F3–F6). Finalmente, en un tercer grupo, se considerarán los resultados en todas las funciones.

Siguiendo la metodología, primero se aplica la prueba de Friedman para detectar diferencias a nivel de grupos, esto es, entre todos los algoritmos. Los resultados muestran que en los tres grupos existen diferencias significativas de manera general entre los algoritmos (p-valor< 0.05). Para detectar en qué pares de algoritmos existen tales diferencias, se procedió con una prueba post-hoc de Holm. En este caso, la Figura 2 muestra la relación entre los algoritmos en los tres grupos de experimentos según los p-valores de la prueba de Holm. Los arcos entre los algoritmos indican que los algoritmos involucrados no son significativamente diferentes entre sí. Los gráficos también muestran, a través de barras de colores, el ranking promedio de los algoritmos según la prueba de Friedman. Los números a la izquierda muestran el orden según el rendimiento relativo obtenido, donde un valor bajo corresponde a un buen rendimiento.

A partir de la Fig. 2 se puede observar que en las instancias de problemas con funciones unimodales 2-a), los algoritmos de mejores resultados son los métodos que combinan auto-adaptación en los elementos del algoritmo, con mecanismos de adaptación / auto-adaptación de las características de los individuos. En este casos son: mSQSspDE (método SspDE que tiene auto-adaptación en los parámetros y las estrategias de mutación, con enfoque multipoblacional y auto-adaptación quantum añadidos), mSQJDE (método JDE que tiene auto-adaptación en los parámetros, con enfoque multipoblacional y auto-adaptación quantum añadidos) y mSQDE (método DE estándar, con enfoque multipoblacional y auto-adaptación quantum añadidos). Los algoritmos de peores resultados son los métodos básicos con auto-adaptación y sólo con enfoque multipoblacional añadido(mJDE y mSspDE)

En las instancias de problemas con funciones multimodales (Fig. 2-b) los algoritmos de mejores resultados vuelven a ser los métodos que combinan auto-adaptación en los elementos del algoritmo, con mecanismos de adaptación / auto-adaptación de las características de los individuos: mSQDE, mSQJDE y mSQSspDE. Se hace notar que el método mSspDE esta vez alcanza una mejor posición , incluso comparable con la obtenida por mSQSspDE, principalmente debido a su gran rendimiento en una de las instancias de problema (con función $F3$).

Teniendo en cuenta a todas las instancias de problema, se puede ver que en general los métodos mSQDE, mSQJDE y mSQSspDE logran los mejores resultados. Los algoritmos de peores resultados son mJDE y mSspDE.

Como se puede apreciar, estos resultados son en cierta medida, consistentes con los obtenidos en investigaciones anteriores [1,2,23]. Note por ejemplo que en algunos casos (ej. instancias de problema con $F3$) la auto-adaptación es adecuada, pero en otros casos no es suficiente para tratar con la dinámica del problema. Esto también apoya nuestra percepción de que la auto-adaptación puede ser más efectiva cuando tiene en cuenta la dinámica del problema, esto es, cuando es aplicada explícitamente a algoritmos con algún mecanismo de adaptación. Nótese que el efecto de tal tipo de auto-adaptación mejora algoritmos simples como el DE estándar. De hecho, mSQDE (el algoritmo derivado de la combinación del

DE estándar y el enfoque *quantum* auto-adaptativo) posee de forma general una mejor posición (Fig. 2-c).

Otro aspecto destacable es el buen rendimiento que provoca el modelo de auto-adaptación quantum en todos los casos. En ese sentido, basta comparar los algoritmos mSQJDE y mSQSspDE *versus* mJDE, mSspDE (con auto-adaptación en el método básico), y mSQDE (con auto-adaptación en el enfoque *quantum*). Comparados con los primeros, note por ejemplo que mSQJDE y mSQSspDE son significativamente mejores que mJDE y mSspDE en problemas con funciones unimodales (Fig.2-a), mientras que en instancias multimodales, esta superioridad disminuye levemente en relación a mSspDE (Fig. 2-b). Sin embargo, desde el punto de vista general (Fig. 2-c) puede verse que estas diferencias son significativas. Comparados con mSQDE, se puede apreciar que en todos los casos, no existen diferencias significativas entre los algoritmos.

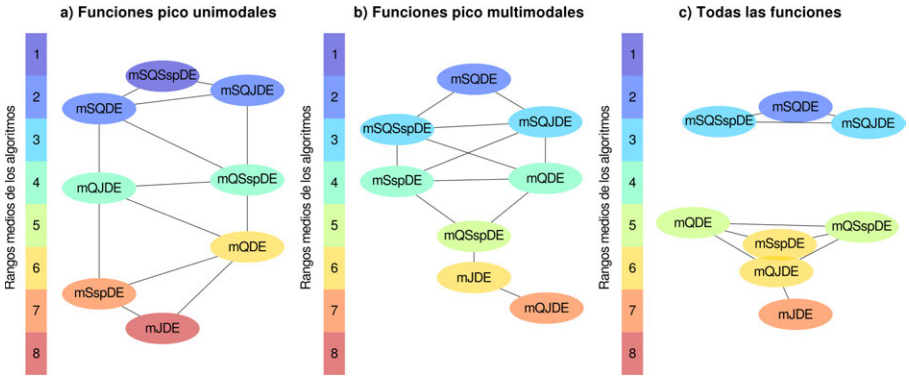


Figura 2. Rangos medios de los algoritmos según la prueba de Friedman. Los arcos directos indican que no existen diferencias significativas entre los algoritmos involucrados (de acuerdo a los $p-$valores ajustados de la prueba post-hoc de Holm.

Como consecuencia de esto último, se puede concluir que la auto-adaptación tiene mayor efecto cuando es aplicada al enfoque de adaptación *quantum*. Asimismo, los métodos que combinan auto-adaptación en los elementos del algoritmo, con mecanismos de adaptación/auto-adaptación de las características de los individuos, como mSQJDE y mSQSspDE no ofrecen mejoras significativas a la variante mSQDE. Al parecer la combinaciones de varios modelos auto-adaptativos independientes produce un exceso de diversidad en la población durante la ejecución, afectando así la convergencia del algoritmo.

Finalmente, consideremos las variantes auto-adaptativas del enfoque *quantum*, representados por los algoritmos mSQDE, mSQJDE, y mSQSspDE. En ese sentido, se puede ver que estas variantes poseen en general un rendimiento superior que sus contrapartes adaptativas mQDE, mQJDE y mQSspDE. Es-

pecíficamente, las diferencias son claras en instancias de problema con función $F2$, $F4$, $F5$, y $F6$. En el caso de las otras funciones, las variantes adaptativas son ligeramente mejores que las auto-adaptativas. En sentido general (Fig. 2-c) las diferencias entre ambos grupos son significativas a favor de las variantes auto-adaptativas del enfoque *quantum*.

Análisis de los distintos esquemas de auto-adaptación. Con el objetivo de explicar con más detalles el comportamiento de los algoritmos en función de los esquemas auto-adaptativos presentes en ellos, en este apartado se hace un análisis de la adaptabilidad de los algoritmos. Para ello se han seleccionado dos instancias de problemas basadas en las funciones F1 (Rotation peaks) y F3 (Composition Rastrigin), las cuales emplean el tipo de cambio T2 (Paso largo). La intención es analizar problemas con diversas características. Además, los algoritmos seleccionados son: mSQDE (auto-adaptación quantum y enfoque multipoblacional añadidos al algoritmo DE estándar), mJDE y mSspDE (con auto-adaptación tal como fueron diseñados por sus autores, y con enfoque multipoblacional añadido). Resulta de interés entonces, conocer en qué medida son efectivos los modelos auto-adaptativos estudiados. En este caso conviene definir qué objetivo tienen estos modelos en el contexto de la optimización en ambientes dinámicos. Se puede ver que el papel de los modelos tratados en esta investigación es influir inteligentemente en el balance de explotación y exploración del algoritmo.

En el contexto de los algoritmos poblacionales, lo anterior se traduce en un mantenimiento adecuado de la diversidad de los individuos de la población. Idealmente, la diversidad de la población debería estar en correspondencia con el estado de la búsqueda. Por ejemplo, debe haber un nivel bajo de diversidad si el óptimo del problema está cerca y viceversa. De manera que se puede decir que los modelos auto-adaptativos influyen en la diversidad de la población con el objetivo de disminuir la distancia al óptimo del problema. Basados en este último razonamiento, en este trabajo proponemos la siguiente medida para cuantificar el nivel de *adaptabilidad* de los algoritmos:

$$adaptabilidad = relDistOpt - relDivPob \tag{3}$$

donde $relDistOpt$ y $relDivPob$ son a su vez, magnitudes normalizadas correspondientes a la distancia al óptimo del problema, y la diversidad de la población, respectivamente (véase [9] y [22]). Ambas magnitudes varían en el intervalo $[0, 1]$ por tanto, $adaptabilidad \in [-1; 1]$. En este sentido, si la *adaptabilidad* se acerca a 1 indica que el algoritmo es incapaz de generar la suficiente diversidad para alcanzar al óptimo. Por el contrario, un valor cercano a -1, indica que el algoritmo genera mucho más diversidad que la necesaria. De manera que el valor ideal sería 0, el cual significa una generación de diversidad acorde a la distancia del óptimo del problema. Teniendo en cuenta estos aspectos, considérese la Figura 3 donde se muestra la adaptabilidad de los algoritmos mSQDE, mJDE y mSspDE. En particular las gráficas 3-a) y c) muestran la adaptabilidad en función de las generaciones del algoritmo, mientras que las gráficas 3-b) y d) exhiben el

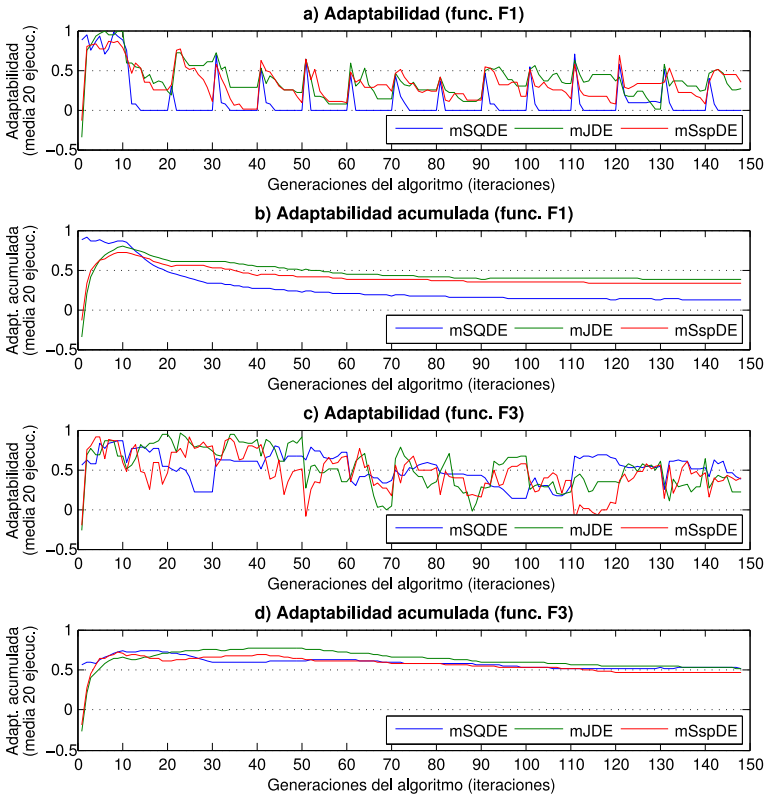


Figura 3. Evolución en el tiempo de la *adaptabilidad* de los algoritmos mSQDE, mJDE, y mSspDE, en instancias de problema con función F1 y F3, y tipo de cambio T2.

promedio acumulado de estos valores. Estas últimas gráficas han sido empleadas para visualizar la tendencia general de esta medida.

Como se aprecia, el comportamiento de los algoritmos es consistente con el análisis estadístico previo. Por ejemplo, el algoritmo mSQDE presenta una mejor adaptabilidad que el resto en el escenario con función F1 (véase la gráfica 3-b). No obstante, a juzgar por la gráfica 3-a), los tres algoritmos presentan comportamientos auto-adaptativos en general, ya intentan generar diversidad después de cada cambio. Los cambios en el ambiente se encuentran en las gráficas cada 10 generaciones del algoritmo. Sin embargo, de los tres, solo mSQDE logra una adaptación más rápida y efectiva que el resto. Note como el valor acumulado de la adaptabilidad de este algoritmo tiende a 0 más rápido que los otros conforme aumenta el tiempo. En el caso del escenario con función F3, tal comportamiento auto-adaptativo deja de ser tan visible como en el caso anterior (gráfica 3-c). Indudablemente, la complejidad de la función *Composition Rastrigin* afecta de

forma negativa la adaptabilidad de todos los algoritmos de manera similar, de ahí que los tres exhiban tendencias similares (gráfica 3-d). Pero, en todos los casos con tendencias que nunca se aproximan a 0 (al menos para los 15 cambios representados en la gráfica 3-d).

4. Conclusiones

En este trabajo se ha investigado, de manera empírica, el rol de la auto-adaptación en ambientes dinámicos. En este sentido, hemos considerado 8 algoritmos multipoblacionales, basados en el paradigma Evolución Diferencial. Los algoritmos se obtuvieron a partir de la presencia o no de modelos auto-adaptativos, tanto a nivel de paradigma como de éstos con mecanismos de adaptación dinámica. Los resultados obtenidos a partir de experimentos realizados sobre 49 instancias de problemas con diferentes funciones pico y tipos de cambio, se resumen de la siguiente manera: a) la auto-adaptación a nivel de paradigma no es suficiente para lidiar con la dinámica del problema; b) la auto-adaptación tiene un efecto positivo cuando es aplicada a un algoritmo con adaptación dinámica; y c) los modelos auto-adaptativos del enfoque de adaptación (quantum) resultaron significativamente superiores que sus variantes adaptativas.

Agradecimientos Este trabajo está parcialmente financiado por los projectos TIN2011-27696-C02-01 del Ministerio de Economía y Competitividad, y P11-TIC-8001 de la Junta de Andalucía.

Referencias

1. Angeline, P.J., Fogel., D.B., Fogel, L.J.: A comparison of self-adaptation methods for finite state machines in dynamic environments. In: L. J. Fogel, P. J. Angeline, .T.B. (ed.) Evolutionary Programming V: Proceedings of the Fifth Annual Conf. on Evolutionary Programming. pp. 441–449. Cambridge, MA: MIT Press. (1996)
2. Bäck, T.: On the behavior of evolutionary algorithms in dynamic environments. In: IEEE Int. Conf. on Evolutionary Computation. pp. 446–451. IEEE Press (1998)
3. Beyer, H.G., Deb, K.: On self-adaptive features in real-parameter evolutionary algorithms. Evolutionary Computation, IEEE Transactions on 5(3), 250 –270 (jun 2001)
4. Blackwell, T., Branke, J.: Multiswarms, exclusion, and anti-convergence in dynamic environments. IEEE Transactions on Evolutionary Computation 10(4), 459–472 (2006)
5. Brest, J., Greiner, S., Boskovic, B. Mernik, M., Zumer, V.: Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. Transactions on Evolutionary Computation 10(6), 646–657 (2006)
6. Brest, J., Zamuda, A., Boskovic, B., Maucec, M.S., Zumer, V.: Dynamic optimization using self-adaptive differential evolution. In: CEC'09: Proceedings of the Eleventh conference on Congress on Evolutionary Computation. pp. 415–422. IEEE Press, Piscataway, NJ, USA (2009)

7. Chun-Kit, A., Ho-Fung, L.: An empirical comparison of cma-es in dynamic environments. In: Coello Coello, C.e.a. (ed.) PPSN 2012, Part I, LNCS 7491. pp. 529–538. Springer-Verlag Berlin Heidelberg (2012)
8. Cruz, C., González, J.R., Pelta, D.: Optimization in dynamic environments: a survey on problems, methods and measures. Soft Computing 15(7), 1427–1448 (2011)
9. Deb, K., Beyer, H.: Self-adaptive genetic algorithms with simulated binary crossover. Evolutionary computation 9(2), 197–221 (2001)
10. García, S., Molina, D., Lozano, M., Herrera, F.: A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the cec 2005 special session on real parameter optimization. J Heuristics 15, 617–644 (2009)
11. Halder, U., Maity, D., Dasgupta, P., Das, S.: Self-adaptive cluster-based differential evolution with an external archive for dynamic optimization problems. In: Panigrahi, B., et al. (eds.) SEMCCO 2011, LNCS, vol. Part I, pp. 19–26. Springer-Verlag Berlin Heidelberg (2011)
12. Jin, Y., Branke, J.: Evolutionary optimization in uncertain environments-a survey. Evolutionary Computation, IEEE Transactions on 9(3), 303–317 (2005)
13. Kramer, O.: Evolutionary self-adaptation: A survey of operators and strategy parameters. Evolutionary Intelligence 3(2), 51–65 (2010)
14. Li, C., Yang, S.: A generalized approach to construct benchmark problems for dynamic optimization. In: Simulated Evolution and Learning. Lecture Notes in Computer Science, vol. 5361, p. 391–400. Springer Berlin / Heidelberg (2008)
15. Meyer-Nieberg, S., Beyer, H.G.: Self-adaptation in evolutionary algorithms. In: Lobo, F., Lima, C., Michalewicz, Z. (eds.) Parameter Setting in Evolutionary Algorithms, Studies in Computational Intelligence, vol. 54, pp. 19–46. Springer Berlin / Heidelberg (2007)
16. Nguyena, T.T., Yang, S., Branke, J.: Evolutionary dynamic optimization: A survey of the state of the art. Swarm and Evolutionary Computation 6, 1–24 (2012)
17. Novoa-Hernández, P., Corona, C., Pelta, D.: Efficient multi-swarm pso algorithms for dynamic environments. Memetic Computing 3(3), 163–174 (2011)
18. Novoa-Hernández, P., Cruz Corona, C., Pelta, D.A.: Self-adaptive, multipopulation differential evolution in dynamic environments. Soft Computing - A Fusion of Foundations, Methodologies and Applications pp. 1–22 (2013), (In press)
19. Pan, Q.K., Suganthan, P., Wang, L., Gao, L., Mallipeddi, R.: A differential evolution algorithm with self-adapting strategy and control parameters. Computers & Operations Research 38, 394–408 (2011)
20. du Plessis, M.C., Engelbrecht, A.P.: Self-adaptive competitive differential evolution for dynamic environments. In: IEEE Symposium on Differential Evolution - SDE'2011 (2011)
21. Storn, R., Price, K.: Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11, 341–359 (1997)
22. Weicker, K.: Performance measures for dynamic environments. In: Guervós, J.J.M., Adamidis, P., Beyer, H.G., nas, J.L.F.V., Schwefel, H.P. (eds.) Parallel problem solving from nature. vol. VII, pp. 64 – 73 (2002)
23. Weicker, K., Weicker, N.: On evolution strategy optimization in dynamic environments. In: Congress on Evolutionary Computation. vol. 3, pp. 2039–2046 (1999)

Automatic Design of Electronic Amplifiers using Grammatical Evolution

Federico Castejón and Enrique J. Carmona

Dpto. de Inteligencia Artificial, ETSI Informática, Universidad Nacional de Educación a Distancia (UNED), Juan del Rosal 16, 28040, Madrid, Spain
fcastejon@gmail.com
ecarmona@dia.uned.es

Abstract. An algorithm for automatic synthesis of one stage analog amplifiers based on Grammatical Evolution (GE) and Koza's developmental expressions is presented. GE is an evolutionary algorithm which can generate code in any programming language and uses variable length binary strings. The binary genome determines which production rules in a BNF grammar definition are used in a genotype to phenotype mapping process to a program. Koza introduced developmental functions in Genetic Programming for the synthesis of analog circuits. Our GE-based algorithm uses a grammar comprising only ten of these functions. The fitness function has different terms to meet the different design specifications of the amplifier. Finally, this methodology was applied to a case study and the best circuit obtained proved to be a good approximation to the target amplifier. Also a comparison to a human designed amplifier is shown.

Keywords: Automatic Design of Electronic Circuits, Evolutionary Algorithm, Grammatical Evolution, Developmental expressions, NGSpice.

1 Introduction

Since late 70s, digital circuits have been replacing analog circuits in general, but some functions still have to remain analog mostly because transducers are analog as well. Unlike digital circuits, analog circuits still require to be designed by experts due to the lack of general analog design tools. Much effort has been done in Electronic Design Automation (EDA) to get aiding tools in the design of analog or mixed signal integrated circuits, like for example standard cell libraries or rule based expert systems [2].

Automatic circuit synthesis comprises two tasks: topology selection and circuit sizing. These tasks can be accomplished together or separately [5]. In this field, evolutionary algorithms have been initially used for circuit sizing among other optimizing algorithms like simulated annealing. On the other hand, topology has been selected by human designers, or aided by formerly mentioned EDA tools.

The field of Evolutionary Electronics appeared in 1998 [11], and its goal is the synthesis of electronic circuits using evolutionary algorithms. Inspired in natural selection, evolutionary algorithms can obtain solutions to complex problems using trial and error. Thus, using a tentative population of potential solutions, they are selected according to their fitness or closeness to the desired solution. Then the best solutions are crossed over and mutated. Successive generations lead to survival of the fittest. Evolutionary algorithms applied to synthesis tasks do not use design rules nor expert knowledge for the design [3], but can lead to unconventional solutions which challenge human designers intuition [1]. On the other hand, obtained circuits can be difficult to analyze by human designers or lead to rejection.

Apart from the design of integrated and digital circuits, evolutionary algorithms have been used with good results for automatic synthesis of analog circuits [9,10]. In particular, Koza and collaborators' work in this area, using genetic programming, is especially relevant [4].

We present an algorithm for synthesizing one stage analog amplifier, based on Grammatical Evolution (GE) and Koza's developmental expressions. GE is an evolutionary algorithm which can generate code in any programming language and uses variable length binary strings. The binary genome determines which production rules in a BNF grammar definition are used in a genotype to phenotype mapping process to a program. In fact, as far as the authors know, there is no work in the literature that applies this evolutionary paradigm to automatic design of analog circuits. A ten member subset of Koza's developmental functions has been also used, with a fixed transistor embryo. Evaluation of each circuit (individual) is done with NGSpice [7] which is based on Spice3 [6].

The rest of the paper is structured as follows: Section 2 describes our method in detail. Section 3 is devoted to the application of our method to a case study: the automatic synthesis of a BJT Common Emitter Amplifier, with a comparison to a human designed amplifier. Finally, conclusions and future work are given in Section 4.

2 Method Description

In this section, we present our proposal to the automatic design of electronic amplifiers. This includes a description of the grammar, the function set and the fitness function used. We also devote a short introduction to Grammatical Evolution, evolutionary paradigm in which our method is based.

2.1 Grammatical Evolution

Grammatical Evolution (GE) was introduced by Michael O'Neill and Conor Ryan [8], as an alternative to Genetic Programming. GE is an evolutionary algorithm which can generate code in any programming language and uses variable length binary strings, unlike Genetic Programming which (GP) is based on parse trees for representing genomes. Crossover and mutation operators work

over these binary strings and the decoding of each individual is based on a BNF grammar of the language. A BNF grammar is defined by a tuple of four elements: $\{S, T, N, R\}$ where S is the start symbol, T is the set of terminal symbols, N is the set of non-terminals and R is the set of rules of production.

GP needs special crossover and mutation operators to work over parse trees and to preserve their consistence as syntactically correct programs. This consideration is not needed in GE, which can use generic and standard operators, while the decoding process guarantees the correctness of the resulting program.

In GE, the unit of information is called codon, drawing inspiration from Biology. Normally, each byte is considered a codon, so it can have 256 possible values. The decoding of any string consists on traversing it, using each codon for each choice of the production rules of the language grammar. The chosen production rule is the one numbered with the codon modulus the number of choices that correspond to the rule that is being considered, as shown in equation 1. A decoding example is provided in subsection 2.2.

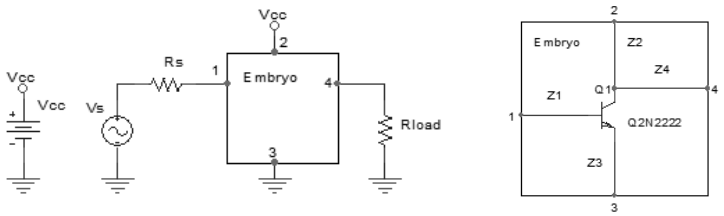
$$productionRule = codon \, MOD \, numberOfRules \tag{1}$$

2.2 Koza's Developmental Functions

John R. Koza and his collaborators introduced a developmental process approach to represent and synthesize analog circuits using GP [4]. The process is based on a circuit embryo which is divided into a test fixture and the embryo itself. The test fixture is the part of the circuit which does not change during the developmental process and comprises, for example, the power supply, the signal source, the source resistor and the load resistor. The embryo itself comprises modifiable wires, which are modified by the developmental functions of the specific developmental expression. Developmental functions operate over a modifiable wire or component, transforming it into a new modifiable component or a mesh formed by new modifiable wires and/or components. Developmental expressions, made of developmental functions, are represented by GP parse trees, and codify the developmental process which leads to a final circuit.

Our approach combines GE with Koza's developmental expressions. First of all embryo and test fixture are shown, followed by our proposed developmental expressions grammar for decoding chromosomes into developmental expressions. A decoding example is provided and finally our proposed fitness function is shown.

Embryo and Test Fixture As will be shown in the next section, our goal is obtaining an one-stage common emitter amplifier, so we have modified the original embryo proposed by Koza, to contain one fixed transistor which cannot be modified by the developmental process. Only the connections to the transistor can be modified. The used embryo is set into a test fixture (fig. 1a) with a signal source, its associated source resistor, and a load resistor. It also includes a fixed transistor with four modifiable wires, Z_1, Z_2, Z_3 and Z_4 (fig. 1b).



(a) (b)

Fig. 1: Test fixture (a) and embryo (b)

Developmental Expressions Grammar Table 1 shows the grammar used for the developmental expressions in Extended Backus-Naur Form (EBNF). The start symbol is *devExpression* which produces a final developmental expression. The symbol *RPB* stands for result-producing branch which comprises the process of transformations for a determined modifiable wire or component.

```
S =              devExpression
T =              {"LIST","END", "CUT", "NOP", "PARALLEL", "SERIES", "THREE_GROUND",
                 "PAIR_CONNECT", "THREE_VCC", "R", "C",'e', '0', '1',...,'9',
                 "-12","-11",...,"-3"}
N =              {RPB, FUN0, FUN1, FUN2, FUN3, CC, resistorValue, capacitorValue,
                 digit,nonZeroDigit,capacitorExp}
R = formed by the following rules of production
devExpression =  "LIST", '(', RPB, ')', { '(', RPB, ')' };
RPB =            FUN0 | FUN1, '(', RPB, ')' | FUN2, '(', RPB, ')', '(', RPB, ')' |
                 FUN3, '(', RPB, ')', '(', RPB,')', '(', RPB, ')' | CC, '(', RPB, ')';
FUN0 =           "END" | "CUT";
FUN1 =           "NOP";
FUN2 =           "PARALLEL";
FUN3 =           "SERIES" | "THREE_GROUND" | "PAIR_CONNECT" | "THREE_VCC";
CC =             ( "R", resistorValue, | "C", capacitorValue );
resistorValue =  nonZeroDigit, '.', digit, 'e', digit;
capacitorValue = nonZeroDigit, '.', digit, 'e', capacitorExp;
digit =          '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9';
nonZeroDigit =   '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9';
capacitorExp =   "-12" | "-11" | "-10" | "-9" | "-8" | "-7" | "-6" | "-5" | "-4" |
                 "-3";
```

Table 1: Developmental expressions grammar

Decoding example The following example chromosome, [34, 5, 19, 30, 43, 15, 64, 68, 15, 15, 24, 25, 37, 45, 36, 44, 42, 81, 100, 104, 75, 8, 14, 12, 4, 17, 13, 90, 88], is decoded as expression: *LIST (C 2.0e-9 (END))(THREE_ VCC(END)(CUT)*

(END))(R 1.0e4 (END))(R 5.7e3 (END)). The first steps of the decoding process are as follows:

1. The decoding starts with S symbol, that is, with *devExpression*.
2. Since our embryo comprises four modifiable wires, our expression need to comprise four *RPB's*, so production rule for *devExpression* expands as *LIST (RPB)(RPB)(RPB)(RPB)*
3. Production rule for first *RPB* has five choices and first codon value is 34, so $34\,MOD\,5$ gives rule *#4* which gives: *LIST (CC (RPB))(RPB)(RPB)(RPB)*
4. Expansion of rule for *CC* with codon value 5 ($5\,MOD\,2$) gives rule *#1* which is associated to *C* function, so the expression is now: *LIST (C capacitorValue (RPB))(RPB)(RPB)(RPB)*
5. Expansion of capacitorValue needs three codons: 19, 30 and 43, giving value $2.0e-9$. *LIST (C 2.0e-9 (RPB))(RPB)(RPB)(RPB)*
6. Next is rule *#0* for *RPB* which gives, *FUN0*. Now the expression is: *LIST (C 2.0e-9 (FUN0))(RPB)(RPB)(RPB)*
7. Next is rule *#0* for *FUN0* which gives *END*. Now the expression is: *LIST (C 2.0e-9 (END))(RPB)(RPB)(RPB)*
8. Next is rule *#0* for *RPB* which gives, *FUN0*. Now the expression is: *LIST (C capacitorValue (FUN0))(RPB)(RPB)(RPB)*
9. And so on.

Developmental function set As shown before, the grammar of our GE based implementation of Koza's developmental expressions uses a subset of only ten functions of all the functions defined by Koza. Table 2 shows the functions used.

Fitness function We have considered just five goal circuit parameters as specification for the amplifier: gain, low cut-off frequency, input and output impedances and dynamic margin. In order to reduce processing time, we have limited the measures of gain to three frequencies, and the impedances and dynamic margin are measured just to one frequency. Equation 2 shows the fitness function in this implementation. Where each term is the difference between the measure from the circuit and the design specification normalized by the latter value. So, the closer the measures are to the specified values, the closer F is to zero, making it a minimization problem.

$$F = \left[\sum_{i=1}^{3} \Delta_{G_{dB}}(f_i)\right] + \Delta_{G_{dB}}(f_L) + \triangle_{Z_{in}}(f_{int1}) + \triangle_{Z_{out}}(f_{int1}) + \Delta_{DM}(f_{int2}) \quad (2)$$

The gain $G(f)$ is measured in decibels (dB) at three frequencies f_1, f_2 and f_3. The gain is also measured at the specified low cutoff frequency f_L. Input and output impedances, Z_{in} and Z_{out} are measured at an intermediate frequency, f_{int1}, and DM is measured at an intermediate frequency f_{int2}. Dynamic margin requires a transient analysis, which is measured with a sinusoidal input signal

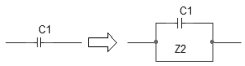
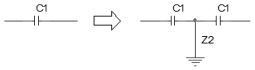
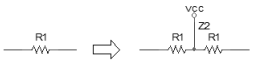
Developmental function	Arity	Description
END	0	End of a modifiable wire
CUT	0	Cut and ends a modifiable wire
NOP	1	No operation, the modifiable wire continues being modifiable
R	2	
C	2	
PARALLEL	2	
SERIES	3	
PAIR_CONNECT	3	
THREE_GROUND	3	
THREE_VCC	3	

Table 2: Developmental functions

having an amplitude high enough to saturate the output. We estimate the dynamic margin from the maximum and minimum saturated output values from one cycle, and we also use the output voltage when there is no input, from an operation point analysis. Equation 3 shows the expression for estimating the dynamic margin, where V_{max} is the maximum output voltage, V_{min} is the minimum output voltage, and V_c is the output voltage value when there is no input.

$$DM = 2\min\left(V_{max} - V_c, V_c - V_{min}\right) \tag{3}$$

3 Case Study: BJT Common Emitter Amplifier

In this section, we present the design specifications of the target amplifier and the configuration parameters for the algorithm. Finally, we present the results of the experiments conducted and the best circuit obtained.

3.1 Design Specifications and Configuration Parameters

Table 3a shows the design specifications of the target amplifier and table 3b shows the GE-based algorithm parameters used. The values of the components V_{cc}, R_s and R_{load} are known in advance, that is, they form part of the test fixture (see fig. 1a) and are not considered in the evolutionary process. There is no

established fitness goal as a termination condition. Instead we allow running up the algorithm to the maximum number of generations. The algorithm parameters for the experiment conducted were obtained from preliminary runs. In these preliminary runs we also found that the algorithm was capable of giving good results without a duplication operator, which was introduced in GE [8]. So no duplication operator was used in the experiment.

$Gain$	f_L	Z_{in}	Z_{out}	DM	V_{cc}	R_s	R_{LOAD}
12dB	10Hz	20KΩ	6KΩ	18V	20V	600Ω	100KΩ

(a)

Goal	Amplifier circuit
Embryo	Common emitter embryo
Representation	Variable length strings of codons (bytes)
Initialization	Random byte strings of 1-10 length
Length	1-10
Fitness function	see equation (2), where $f_1 = 1kHz$, $f_2 = 10kHz$, $f_3 = 100kHz$, $f_L = 10Hz$, and$f_{int1} = f_2$, $f_{int2} = f_1$
Grammar	see table 1
Function set	see table 2
Population	1000
Crossover	1 point, codon string, limited to 250 codons
Crossover probability	0.5
Mutation	Bitwise
Mutation probability	0.001
Parent selection	3 member-Tournament selection
Replacement selection	Generational
Elitism	2
Wrapping	4
Termination condition	250 generations
Execution number	51

(b)

Table 3: Design specifications (a) and algorithm parameters (b)

3.2 Results and Evaluation

Table 4a shows the algorithm performance measures including Success Rate (SR) and Mean Best Fitness (MBF), after 51 executions.

The criterion for success is fitness lower than 0.6. Higher values provide very low dynamic margin values, which are considered unacceptable. Performance results show a high MBF value and also a high standard deviation, nevertheless, since the problem is a design one, only one high quality solution is needed, so

this behavior is not an issue. In fact, an algorithm which provides few high quality solutions and bad solutions the rest of the time is preferred over an algorithm which provides just over average solutions every time. Table 4b shows the results of the three successful circuits, obtained by the evolutionary algorithm (first three rows). Also included the performance of a circuit designed by hand by an electronic engineer (fourth row). The results shown include measures of gain, drop at low cut-off frequency, input and output impedances and dynamic margin.

$Hits$	SR	$maxBF$	$minBF$	$MBF \pm SD$	$Mean\ exec.\ time$
3	5.88%	45.880	0.438	7.402±14.102	6.3'

(a)

$Circuit$	$G_{dB}(f_i)$			$G_{dB}^{drop}(10Hz)$	$Z_{in}(10kHz)$	$Z_{out}(10kHz)$	$DM(1kHz)$	$Fitness$
	$(1kHz)$	$(10kHz)$	$(100kHz)$	(dB)	$(k\Omega)$	$(k\Omega)$	(V)	
1	*12.02*	*12.00*	*10.60*	*-3.00*	*19.01*	*6.39*	*14.32*	*0.438*
2	*12.05*	*12.05*	*11.97*	*-3.01*	*9.36*	*5.92*	*17.98*	*0.559*
3	*11.94*	*11.93*	*11.08*	*-3.05*	*22.78*	*6.37*	*12.75*	*0.596*
human	*11.34*	*11.34*	*11.34*	*-1.55*	*25.29*	*5.99*	*14.79*	*0.698*

(b)

```
LIST (THREE_VCC(END)(END)(C 3.2e-3 (C 2.4e-11 (PARALLEL(END)(C 3.5e-8 (C 1.8e-9 (R 7.5e5
(R 2.2e4 (END)))))))))(R 6.4e7 (NOP(NOP(NOP(C 9.2e-9 (C 8.8e-11 (PARALLEL(END)(R 6.8e8
(NOP(C 5.8e-3 (C 7.3e-11 (PARALLEL(END)(R 6.4e3 (END)))))))))))))))(END)(C 1.5e-7 (END))
```

(c)

Table 4: Summary of results: Algorithm's performance global measures (a), results obtained by the three best evolutionary amplifiers and by a human designer (b), developmental expression of the best evolutionary circuit (c).

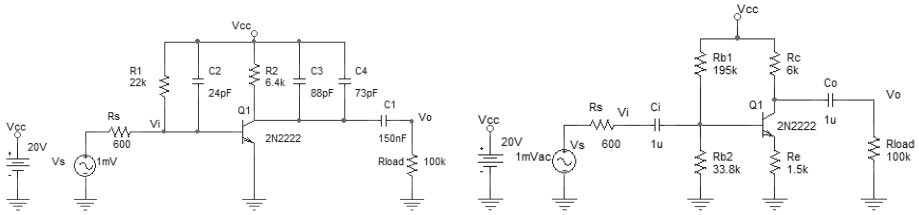
Circuit #1 is meets all of the goal criteria but the dynamic margin. Circuits #2 and #3 show a trade off between dynamic margin and input impedance.

Through an inspection of the parameters of the shown circuits, it seems that is somehow easy for the algorithm to provide good values for gain, low cut-off frequency and output impedance, while showing difficulties for meeting the goal criteria for dynamic margin and input impedance, showing a trade off between them in two cases. This behavior is also found in non successful circuits. Table 4c shows the best circuit developmental expression.

Human designed circuit shows an approximately met specifications, due to the design process itself in which some approximations have to be done, and the transistor, a non-linear component, has to be linearized by using an equivalent circuit. Calculated fitness value is slightly worse than the fitness value of the third circuit.

3.3 Best circuit analysis

Figure 2 shows the best evolutionary amplifier obtained (fig. 2a), and the amplifier obtained by the human designer (fig. 2b). No simplifications have been done to the best circuit. So, there are some components, like resistors in series or parallel, which could be simplified, or in some cases neglected due to their values. In this way, capacitors C_2, C_3 and C_4, due to their low capacity values (pF), only have an impact in high cut-off frequency (not considered in our analysis). They are probably allowed in the circuit due to lack of a explicit high cut-off frequency term in the fitness function, other than the gain criterion at $100kHz$. These capacitors could be neglected, while meeting all specifications criteria. It can also be seen that the circuit polarization network is poor, because this configuration is known to be very sensitive to variation in transistor's current gain. Better polarization networks are based on a voltage divider and also use an emitter resistor. We have also found that the used estimation of dynamic margin is misled by spurious peaks in transient analysis. Actual dynamic margin was measured to be $7V$ which is lower than the estimated. Another minor dysfunction is the absence of an input capacitor, so there is a bias current into the signal source.



(a) (b)

Fig. 2: Amplifiers obtained: best evolutionary circuit (a), human design (b)

4 Conclusions and Future Work

Grammatical evolution has been proved as a valid method for evolving one stage amplifiers using Koza's developmental expressions as the object language. A new grammar comprising ten developmental functions from Koza's set was used. The goal specification for the amplifier included five parameters: gain measured at three frequencies, low cut-off frequency, input and output impedances and dynamic margin. The embryo included a fixed transistor, in common emitter configuration, which cannot be modified during the developmental process. It also included four modifiable wires. The conducted experiment showed that the algorithm has a low success rate, but this is not an issue in a design problem,

because we can afford a high number of runs until getting a small set of good results. Three successful circuits have been obtained and their characteristics have been shown. The best circuit met all of the goal criteria but the dynamic margin. Comparison to a human designed amplifier shows promising results.

Future work could include a weighting of the fitness function terms because some deviations from the goal in one parameter may be more acceptable than in the others. The best obtained circuit showed a polarization network without feedback, which is known to be very sensitive to variation in transistor's current gain. Analysis of variation of this parameter should be introduced and taken into account in fitness function. A better estimation method is also needed for dynamic margin, since the used estimation is misled by spurious peaks.

Acknowledgement

This work was supported by funds of the Advanced Artificial Intelligence Master Program of the Universidad Nacional de Educación a Distancia (UNED), Madrid, Spain.

References

1. A. E. Eiben and J. E. Smith. *Introduction to Evolutionary Computing (Natural Computing Series)*. Springer, October 2008.
2. G.G.E. Gielen and R.A. Rutenbar. Computer-aided design of analog and mixed-signal integrated circuits. *Proceedings of the IEEE*, 88(12):1825 –1854, dec 2000.
3. J.B. Grimbleby. Automatic analogue circuit synthesis using genetic algorithms. *Circuits, Devices and Systems, IEE Proceedings* -, 147(6):319 –323, dec 2000.
4. J.R. Koza, D. Andre, F.H. Bennett, and M.A. Keane. *Genetic Programming III: Darwinian Invention & Problem Solving*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition, 1999.
5. E. Martens and G. Gielen. Classification of analog synthesis tools based on their architecture selection mechanisms. *Integration, the VLSI Journal*, 41(2):238 – 252, 2008.
6. L. W. Nagel and D.O. Pederson. *SPICE: Simulation program with integrated circuit emphasis*. Electronics Research Laboratory, College of Engineering, University of California, 1973.
7. P. Nenzi and H. Vogt. Ngspice users manual version 23, 2011.
8. M. O'Neill and C. Ryan. Grammatical evolution. *IEEE Transactions on Evolutionary Computation*, 5:349–358, 2001.
9. E. Tlelo-Cuautle and M. Duarte-Villaseñor. Evolutionary electronics: Automatic synthesis of analog circuits by genetic algorithms. In Ang Yang, Yin Shan, and Lam Bui, editors, *Success in Evolutionary Computation*, volume 92 of *Studies in Computational Intelligence*, pages 165–187. Springer Berlin / Heidelberg, 2008.
10. E. Tlelo-Cuautle, I. Guerra-Gómez, M. Duarte-Villaseñor, L.G. de La Fraga, G. Flores-Becerra, G. Reyes-Salgado, C.A. Reyes-García, and G. Rodriguez-Gómez. Applications of evolutionary algorithms in the design automation of analog integrated circuits. *Journal of Applied Sciences*, 10:1859–1872, December 2010.
11. R.S. Zebulum, M.A. Pacheco, and M. Vellasco. Comparison of different evolutionary methodologies applied to electronic filter design. In *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, pages 434 –439, may 1998.

Auto-Ajuste del Foco de Atención para la Mejora de un Sistema de Detección de Punto Láser

Francisco Chávez[1], Eddie Clemente[2] León Dozal[3], Francisco Fernández de Vega[1], and Gustavo Olague[3]

[1] Universidad de Extremadura, C/. Santa Teresa de Jornet, 38, 06800, Mérida, España
fchavez@unex.es,fcofdez@unex.es,
[2] CICESE, Tecnológico de Estudios Superiores de Ecateped, Avda. Tecnológico S/N, Valle de Anáhuac, Ecatepec de Morelos, México,
eddie.clemente@gmail.com,
[3] CICESE, Carretera Ensenada-Tijuana, No. 3918, zona Playitas, C.P. 22860. Ensenada, B.C., México
leon.dozal@gmail.com,gustavo.olague@gmail.com

Resumen En este trabajo se presenta un nuevo enfoque basado en la técnica de Foco de Atención para la mejora de un sistema de control de entorno por medio de un puntero láser. Los punteros láser han sido utilizados como elementos de indicación en pantallas de proyección, pero nosotros hemos adaptado ese uso para sistemas de control de entornos. Este tipo de sistemas requiere una precisa detección del punto láser para enviar órdenes de control correctas a los diferentes dispositivos. El sistema debe ser capaz de eliminar los falsos positivos, situaciones en las que se envían órdenes incorrectas a los dispositivos, evitando así la activación/desactivación de forma autónoma. La idea es aplicar un proceso de auto-ajuste a la técnica de Foco de Atención, mediante un algoritmo basado en Programación Genética, para centrar la atención del sistema de reconocimiento visual en un área reducida de una imagen, donde se encuentre el punto láser buscado. Los resultados muestran que el nuevo enfoque mejora significativamente el nivel de precisión cuando el punto láser está presente -los usuarios envían órdenes-, mientras que se mantienen los valores extremadamente bajos de falsos positivos proporcionadas por técnicas anteriores.

Keywords: Auto-Ajuste, Foco de Atención, Puntero Láser, Sistema de Control de Entorno

1. Introducción

El mundo tecnológico que nos rodea ha hecho posible que estemos rodeados de cientos de dispositivos electrónicos, formando ya parte de nuestra vida cotidiana. La tecnología es muy útil y cada vez está más presente en los hogares,

convirtiendo a éstos en lo que se conoce como *hogares inteligentes*. Este tipo de hogares son particularmente útiles para personas mayores y discapacitadas, ya que les permiten llevar una vida más sencilla y adaptable a sus necesidades. En este contexto, los sistemas de interacción son el componente principal. Un sistema de interacción se puede definir como: *conjunto de objetos interrelacionados que ejercen claramente una interacción entre ellos*. Este concepto se puede aplicar a los hogares inteligentes [1,11] obteniendo lo que denominamos un *Sistema de Control de Entorno*.

Entre los diferentes métodos disponibles, varios autores han considerado el uso de punteros láser como sistema de interacción [9,10]. Nosotros hemos estudiado y aplicado el uso de punteros láser comerciales como sistemas de interacción, en combinación con vídeo cámaras estándar que monitorizan un entorno. La idea es proporcionar a los usuarios la capacidad de enviar órdenes mediante la proyección de puntos láser sobre los dispositivos, utilizando como nexo de unión, un sistema de control de domótico [2,3,4,5]. En los trabajos anteriormente mencionados, mediante la aplicación de técnicas de visión artificial, el sistema es capaz de detectar un punto láser. Por otro lado, en [2,4,5] se analizó la aplicación de técnicas de Soft Computing, como los Sistemas Basados en Reglas Difusas (SBRD) que, en combinación con técnicas de Template Matching (TM), ofrecen buenos resultados en la detección de un punto láser. En estos trabajos se confirmó la idoneidad de los SBRD, pero siguen mostrándose los problemas que surgen en la aplicación de técnicas clásicas de TM en combinación con Umbralización Dinámica (UD). Ambos sistemas TM+UD y TM+SBRD necesitan de un proceso previo de filtrado de imágenes, el cual selecciona una sección de la imagen como candidata a ser o no un punto láser. Esta selección se realiza por medio de TM, posteriormente ambos enfoques tratan de decidir si se trata o no de la imagen de un punto láser, utilizando UD o SBRD respectivamente.

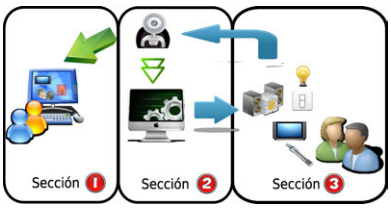
Se ha detectado que el proceso previo realizado por TM aporta en ocasiones imágenes incorrectas, por lo que en este trabajo se presenta un nuevo enfoque basado *Brain Programming* [13,14], el cual trata de emular y auto-ajustar el método basado en el Foco de Atención -*Focus of Attention (en inglés)*- (FOA). Este nuevo enfoque permite reducir el espacio de búsqueda utilizado por TM, gracias a que centra su foco de atención en una sección de la imagen donde se encuentra el punto láser. Este trabajo presenta los resultados obtenidos con la incorporación de FOA a los métodos previos de detección de un punto láser, mejorando notablemente la tasa de aciertos del sistema, manteniendo extremadamente bajo el número de falsos positivos y aumentando de forma muy destacada el número de aciertos en imágenes con punto láser.

El resto del trabajo se organiza de la siguiente forma: la sección 2 describe nuestros trabajos previos, mientras que la sección 3 describe la técnica utilizada -Foco de Atención-. La sección 4 muestra los resultados obtenidos y el conjunto de experimentos llevados a cabo con la nueva técnica presentada. Finalmente la sección 5 resume las conclusiones de este trabajo.

2. Trabajos previos

A lo largo de esta sección se detalla el sistema de interacción con los dispositivos del entorno por medio de un puntero láser. En combinación con diferentes vídeo cámaras y un algoritmo de procesamiento de imágenes, el sistema analiza las imágenes del entorno, detecta si se encuentra presente un punto láser en una de esas imágenes, y envía una orden de apagado/encendido a un sistema de control domótico. (ver figura 1)

Figura 1. Sistema de control de entorno.



- 1^a *Sección*: Herramientas de control que permiten al usuario indicar dónde se encuentran los dispositivos controlables por el sistema en el entorno. Por medio de esta herramienta el usuario define las zonas de interés de búsqueda del punto láser, coincidentes con los dispositivos, lo que denominamos *zonas activas*.
- 2^a *Sección*: En esta sección el sistema utiliza diferentes algoritmos para analizar las imágenes enviadas por la vídeo cámara y poder localizar correctamente si en una de las zonas activas se encuentra o no un punto láser.
- 3^a *Sección*: Utilizando un sistema de control domótico se envían las órdenes a los dispositivos.

La principal contribución presentada en este trabajo está relacionada con la 2^a *Sección*. Por medio de los diferentes algoritmos presentados, utilizando para ello diferentes técnicas de análisis de imágenes en combinación con algoritmos como FOA, se pretende mejorar sustancialmente la parte del sistema dedicada a la correcta detección del punto láser. A continuación se detallan los algoritmos utilizados en trabajos previos [2,3,4,5]:

1. Algoritmos basados en técnicas clásicas de análisis de imágenes para extracción de características:
   - Umbralización Dinámica (UD)
   - Template Matching (TM)
2. Algoritmos basados en SBRD para toma de decisiones:
   - Template Matching + Umbralización Dinámica (TM+UD)
   - Template Matching + SBRD diseñado por un experto y ajustado por un algoritmo genético.(TM+$SBRD_{ajustado-AG}$).
   - Template Matching + SBRD basado en aprendizaje evolutivo(TM + $SBRD_{aprendizaje-AG}$)

TM y UD fueron los primeros algoritmos utilizados para extracción de características de un punto láser. La figura 2 muestra lo que denominamos *template*, que es la imagen real de un punto láser, la cual se utiliza para comparar y detectar si en una de las imágenes enviadas por la vídeo cámara se encuentra o no el punto láser.

Figura 2. Template láser

Posteriormente a TM+UD se sutituyó por TM+SBRD para poder extraer características relevantes de la imagen y poder tomar una decisión de si es o no un punto láser. El proceso se basa en detectar la zona de máxima correlación entre la imagen analizada y un template similar al mostrado en la figura 2, utilizando para ello el algoritmo TM. Posteriormente, una vez extraída la sección candidata de la imagen, se extraen ciertas características que son analizadas por un SBRD, el cual ofrece una decisión indicando si es o no un punto láser dicha sección de imagen (Ver [2,4,5]).

3. Metodología

Tras un estudio pormenorizado de las técnicas anteriormente descritas en los trabajos [2,3,4,5], podemos observar que la técnica de TM proporciona imágenes candidatas erróneas. Esto hace que los diferentes sistemas utilizados como toma de decisión -UD o SBRD-, no actúen de forma correcta. Por esta razón, en este trabajo consideramos la posibilidad de mejorar los resultados obtenidos con TM+UD. Para ello, hemos introducido un nuevo algoritmo que emplea técnicas de programación genética (PG) basado en FOA, el cual centrará la atención y reducirá la sección de la imagen donde se encuentre el punto láser facilitando el trabajo de TM. FOA es un algoritmo auto-adaptativo que permite obtener una sección de la imagen analizada en la que se encuentre el punto láser buscado. Al ser esta sección considerablemente más pequeña que la imagen anteriormente analizada, el algoritmo TM debe obtener imágenes candidatas correctamente.

3.1. Algoritmo FOA

Hoy en día, desafiantes problemas de visión por computador están recibiendo cada vez más interés por parte de la comunidad de computación evolutiva [12]. Se dice que los seres humanos son incapaces de crear una representación mental completa de todo su campo visual, por lo que, con el fin de procesar la información visual, la persona debe centrarse en una región del entorno y no en el entorno completo. La idea de FOA es encontrar una parte de la escena que contiene información importante para la solución de una tarea dada [6,8].

Un fenómeno básico que define la atención visual es la selectividad, que se entiende como la capacidad de filtrar información no deseada [7]. Por lo tanto,

FOA es una abstracción del proceso natural que se puede aplicar para encontrar la ubicación del punto de láser en la imagen y contribuir durante la etapa de TM obteniendo mejores imágenes candidatas.

FOA se basa en la teoría de funciones de integración [16], que establece que la atención visual se divide en dos grandes etapas: extracción e integración de características.

Extracción En esta etapa se recibe la imagen de la vídeo cámara y se lleva a cabo la extracción de las características visuales. Esto se realiza a través de, lo que llamamos, operadores visuales evolucionados -*evolved visual operators (en inglés)*-(EVO). Hay tres operadores EVO diferentes, cada uno especializado en la detección de una sola característica que son: la orientación (EVO_O), color (EVO_C), y forma (EVO_S). Hay un cuarto operador que no es evolucionado, la intensidad. Estos operadores se aplican sobre la imagen de forma paralela, dando lugar a un mapa visual -*visual map (en inglés)*-(VM) por característica, véase la figura 3. En resumen, se propone el uso de un algoritmo basado en PG para obtener estos operadores visuales artificiales. Los operadores visuales utilizados dentro de FOA se generan mediante evolución artificial basada en las funciones y terminales listados en la tabla 1. Las definición de las funciones y terminales utilizados puede consultarse en los trabajos [6,8,13,14].

Tabla 1. Funciones y terminales para los operadores EVO

Operador	Funciones	Terminales
EVO_O	$+, -, \times, \div, \lvert+\rvert, \lvert-\rvert, \sqrt{I_{T_O}}, I^2_{T_O}, log_2(I_{T_O}),$ $G_{\sigma=1}, G_{\sigma=2}, \lvert I_{T_O}\rvert, \frac{I_{T_O}}{2}, D_x, D_y$	I_r, I_g, I_b, I_c, I_m, I_y, I_k, I_h, I_s, I_v, $G_{\sigma=1}(I_r)$, $G_{\sigma=2}(I_r)$, $D_x(I_r)$, $D_y(I_r)$, $D_{xx}(I_r)$, $D_{yy}(I_r)$, $D_{xy}(I_r)$, ...
EVO_C	$+, -, \times, \div, \lvert+\rvert, \lvert-\rvert, \sqrt{I_{T_C}}, I^2_{T_C}, log_2(I_{T_C}),$ $Exp(I_{T_C}, Complement(I_{T_C})$	I_r, I_g, I_b, I_c, I_m, I_y, I_k, I_h, I_s, I_v, RG_{oppn}, YB_{oppn}
EVO_S	$+, -, \times, \div, round(I_{T_S}), \lfloor I_{T_S}\rfloor, \lceil I_{T_S}\rceil,$ $dilation_{diamond}(I_{T_S}), dilation_{square}(I_{T_S}),$ $dilation_{disk}(I_{T_S}), erosion_{diamond}(I_{T_S}),$ $erosion_{square}(I_{T_S}), erosion_{disk}(I_{T_S}),$ $skeleton(I_{T_S}), boundary(I_{T_S}), hit-miss_{diamond}(I_{T_S}), hit-miss_{square}(I_{T_S}),$ $hit-miss_{disk}(I_{T_S}), top-hat(I_{T_S}),$ $bottom-hat(I_{T_S}), open(I_{T_S}), close(I_{T_S})$	I_r, I_g, I_b, I_c, I_m, I_y, I_k, I_h, I_s, I_v
EFI	$+, -, \times, \div, \lvert+\rvert, \lvert-\rvert, \sqrt{I_{T_{fi}}}, I^2_{T_{fi}},$ $Exp(I_{fi}), G_{\sigma=1}, G_{\sigma=2}, \lvert I_{T_{fi}}\rvert, round(I_{T_{fi}}),$ $\lfloor I_{T_{fi}}\rfloor, \lceil I_{T_{fi}}\rceil, D_x, D_y$	CM_I, CM_O, CM_C, CM_S, $D_x(CM_I)$, $D_y(CM_I)$, $D_{xx}(CM_I)$, $D_{yy}(CM_I)$, $D_{xy}(CM_I)$, ...

Una vez obtenidos los VMs, el siguiente paso es generar los mapas de visibilidad -*conspicuity maps (en inglés)*-(CMs). Los CMs se obtienen como en el modelo de Walther y Koch [17].

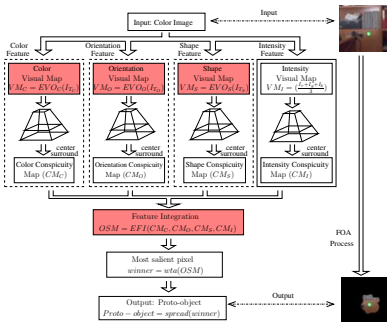
Integración El siguiente paso es la fusión de los CMs intermedios en un único mapa de información relevante denominado mapa de prominencia -*saliency map (en inglés)*-(SM). Este mapa define el lugar de las regiones de imagen más

prominentes; dadas las características de intensidad, orientación, color y forma. En otras palabras, el objetivo de esta etapa es decidir dónde debe dirigirse la atención dentro de la imagen. En este trabajo esa tarea en encontrar el punto láser.

A través de una función que llamamos -*Integration Feature Evolved (en inglés)*-(EFI) se consideran las diferentes combinaciones de los CMs para obtener el resultado esperado.

Posteriormente, con la integración de características se realiza un mapa de prominencia optimizado -*optimized saliency map (en inglés)*-(OSM). Este OSM corresponde con la región más prominente de la imagen dada. Esta región se conoce con el nombre de *proto-objeto* P_t. La figura 3 muestra el proceso completo.

Figura 3. Esquema de FOA



3.2. Definición del auto-ajuste de FOA

A continuación, se describe la codificación para el proceso de auto-ajuste de FOA. Este proceso descriptivo del auto-ajuste de FOA se divide en dos etapas, la primera es la descripción del genotipo, y la segunda etapa es la definición de la función de evaluación.

En la primera fase, FOA se auto-ajusta para aprender a concentrarse en una región prominente de la imagen, con el objetivo de localizar un punto láser. El genotipo utilizado por el algoritmo de PG se define utilizando cuatro los operadores. Los vistos anteriormente: orientación (EVO_O), color (EVO_C), y forma (EVO_S) y un cuarto, la integración de las características extraídas con el objetivo de obtener el CM usando el operador EFI.

La función de evaluación utilizada por el algoritmo de auto-ajuste de FOA se basa en *F-measure*, para medir la precisión entre la región seleccionada por FOA y la ubicación real del punto láser [15]. Esta función se basa en dos parámetros ρ y ϑ, donde:

- ρ: número de píxeles de ambas regiones dividido por el número de píxeles del proto-objeto.
- ϑ: número de píxeles que corresponden con la intersección de ambas regiones dividido por el número de píxeles que conforman el punto láser.

F-measure se define como:

$$F_{measure} = 2 * \frac{\rho \times \vartheta}{\rho + \vartheta} \tag{1}$$

4. Experimentos y Resultados

Como se describe en la sección 2, las técnicas clásicas arrojan un número de errores debido a los problemas ocasionados por TM, la idea principal de este trabajo es incluir FOA para ayudar al algoritmo TM y reducir ese número de errores.

Antes de presentar los resultados obtenidos, debemos tener en cuenta que FOA utiliza un algoritmo basado en PG para su proceso de auto-ajuste. Por ello, es necesario aportar un conjunto de ejemplos lo suficientemente amplio para obtener el mejor resultado. Este conjunto está formado por 480 imágenes obtenidas en un entorno real de funcionamiento con las siguientes características:

- Tipo de superficie donde se proyecta el láser.
- Distancia ente el puntero y la superficie donde se ha proyectado la luz láser.
- Tipo de luz del entorno.

Este conjunto de 480 imágenes se divide en 5 grupos de forma aleatoria, con un 20 % cada uno de ellos. Utilizando 4 de los grupos para entrenamiento y 1 para pruebas, y permutando 5 veces, obtenemos 5 subconjuntos de 384 de entrenamiento y 96 de pruebas. Se aplica el método conocido como *5-fold cross-validation* realizando un total de 6 experimentos diferentes con cada subconjunto, obteniendo un total de 30 ejecuciones, lo que nos permite validar los datos obtenidos.

El mejor individuo encontrado a lo largo de las ejecuciones es el siguiente:

$$\begin{aligned} EVO_O &= D_Y(C) \\ EVO_C &= \frac{Complement(C)^2}{Exp(M)} \\ EVO_S &= R \\ EFI &= \lfloor CM_C \rfloor^2 \end{aligned} \tag{2}$$

Podemos observar el la función EFI únicamente utiliza el CM referente al color y descarta otro tipo de características que si eran usadas en trabajos anteriores.

La tabla 2 muestra los datos del proceso de auto-ajuste de FOA a lo largo de las 30 ejecuciones para localizar el mejor individuo. Puede observarse que en la parte de pruebas del sistema, los individuos obtenidos alcanzan un acierto promedio del 83.92 %.

4.1. Nuevo algoritmo: FOA+TM+DU

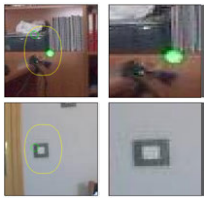
La figura 4 (arriba) muestra un ejemplo de como el mejor individuo obtenido con FOA es capaz de centrar su atención en la región donde se encuentra un

Tabla 2. Resultados del auto-ajuste de FOA

FOA	Entrenamiento	Prueba
$\bar{x}$	84.36 %	83.92 %
σ	0,0545	0,0708

punto láser. Esta región será extraída y utilizada por TM para obtener la sección de imagen candidata de ser punto láser, que al ser menor, hace que TM tienda a generar mejores resultados. Por el contrario, en la mayoría de los casos no se encuentra el punto láser en un entorno real. Pero FOA está preparado para centrarse en una región de la imagen. Es el caso del ejemplo mostrado en la figura 4 (abajo). En este caso TM también ofrecerá una sección de imagen candidata.

Figura 4. Imagen con punto láser (arriba). Imagen sin punto láser (abajo)



En ambos casos debemos contar con un sistema para tomar la decisión de si la imagen candidata extraída con TM es o no un punto láser, en este trabajo se utiliza para ello UD.

La tabla 3 presenta la comparación entre TM+UD y FOA+TM+UD. Utilizando en ambos casos diferentes valores de precisión (VP) en el algoritmo TM. Este valor es utilizado para obtener un vector de secciones de imagen candidatas cuando la correlación entre dicha sección y el template utilizado sea mayor a VP. Se selecciona como imagen candidata final la de mayor correlación con el template. Si no existe ninguna sección con correlación mayor a VP, la imagen resultante será una imagen negra.

Tabla 3. Resultados TM+UD frente a FOA+TM+UD

System	General	Images with Laser	Images without Laser	VP
TM+UD	83,43 %	67,70 %	98,42 %	VP > 0.3
FOA+TM+UD	86,46 %	78,88 %	93,69 %	
TM+UD	83,43 %	67,70 %	98,42 %	VP > 0.4
FOA+TM+UD	86,46 %	78,26 %	94,28 %	
TM+UD	**8**3,43 %	**6**7,70 %	98,42 %	VP > 0.5
FOA+TM+UD	**8**6,77 %	**7**5,78 %	97,24 %	
TM+UD	80,40 %	60,04 %	99,80 %	VP > 0.6
FOA+TM+UD	82,53 %	64,18 %	100,00 %	
TM+UD	72,83 %	44,31 %	100,00 %	VP > 0.7
FOA+TM+UD	73,13 %	44,93 %	100,00 %	

Podemos observar que FOA+TM+UD presenta siempre los mejores resultados. Otra conclusión es que se mantienen los valores de acierto en las imágenes sin punto láser, pero se mejoran notablemente en las imágenes con punto láser. Se deduce que FOA es un perfecto complemento para la detección de puntos láser.

Por otro lado, en trabajos previos (ver [2,4,5]) se ha sustituido UD por un SBRD como elemento para toma de decisiones. La tabla 4 presenta los resultados de la nueva aproximación presentada en este trabajo comparándola con el posterior algoritmo TM+SBRD, obteniendo esta nueva aproximación mejores resultados. Esto nos permite afirmar que una vez que se añada FOA al algoritmo TM+SBRD se incrementará el nivel de acierto.

Tabla 4. Results TM+UD, FOA+TU+UD, TM+$SBRD_{ajustado-AG}$

System	General	Images with Laser	Images without Laser
TM+UD	83,43 %	67,70 %	98,42 %
TM+$SBRD_{ajustado-AG}$	85,99 %	72,75 %	98,74 %
FOA+TM+UD	86,77 %	75,78 %	97,24 %

5. Conclusiones

Este trabajo presenta el proceso de auto-ajuste de un algoritmo basado en el Foco de Atención utilizando técnicas de PG, para trabajar junto a técnicas clásicas de visión por computador en un problema específico: detectar un punto láser.

Gracias a la técnica basada en el Foco de Atención, es posible detectar pequeñas secciones de una imagen donde se encuentre un punto láser, que posteriormente serán analizadas utilizando técnicas clásicas como Template Matching y Umbralización Dinámica para localizar el punto láser. Los resultados presentados avalan la efectividad de la nueva técnica, observando que la técnica inicial obtiene un 67,7 % de acierto frente al 75,78 % de la nueva técnica, incrementando más de un 8 %.

Para finalizar, como trabajo futuro se combinará el nuevo algoritmo basado en el Foco de Atención junto con técnicas de Soft Computing como los Sistemas Basados en Reglas Difusas, para mejorar los resultados presentados en este trabajo.

6. Agradecimientos

Este trabajo está soportado por el Misterio de Ciencia e Innovación de España, bajo el proyecto TIN2011-28627-C04-03, Gobierno de Extremadura, Consejería de Economía-Comercio e Innovación and FEDER, proyecto GRU10029. Junto con CONACyT -proyecto 155045- y TESE -proyecto DIMI-MCIM-004/08- México. El primer autor soportado por la beca 164884 de CONACyT, México.

Referencias

1. M. Chan, D. Estève, C. Escriba, and E. Campo.: A review of smart homes-present state and future challenges. Comput. Methods Prog. Biomed., 91(1):55–81 (2008).
2. F. Chávez, F. Fernández, R. Alcalá, J. Alcalá-Fdez, and F. Herrera.: Evolutionary learning of a laser pointer detection fuzzy system for an environment control system. 2011 IEEE International Conference on Fuzzy Systems (FUZZ), 256–263 (2011).
3. F. Chávez, F. Fernández, R. Alcalá, J. Alcalá-Fdez, G. Olague, and F. Herrera.: Hybrid laser pointer detection algorithm based on template matching and fuzzy rule-based systems for domotic control in real home environments. Applied Intelligence,10.1007/s10489-010-0268-6, 1–17, (2010).
4. F. Chávez, F. Fernández, J. Alcalá-Fdez, R. Alcalá, F. Herrera, and G. Olague.: Genetic tuning of a laser pointer environment control device system for handicapped people with fuzzy systems. IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2010), 1–8 (2010).
5. F. Chávez, F. Fernández, M. Gacto, and R. Alcalá.: Automatic laser pointer detection algorithm for environment control device systems based on template matching and genetic tuning of fuzzy rule-based systems. International Journal of Computational Intelligence Systems, 5(2):368–386 (2012).
6. E. Clemente, G. Olague, L. Dozal, and M. Mancilla. : Object Recognition with an Optimized Ventral Stream Model Using Genetic Programming. EvoApplications, 315–325 (2012).
7. R. Desimone and J. Duncan.: Neural mechanisms of selective visual attention. Annual Reviews, 18, 193–222 (1995).
8. L. Dozal, G. Olague, E. Clemente, and M. Sánchez. : Evolving Visual Attention Programs through EVO Features. EvoApplications, 326–335 (2012).
9. C. Kemp, C. Anderson, H. Nguyen, A. Trevor, and Z. Xu.: A point-and-click interface for the real world: laser designation of objects for mobile manipulation. 3rd ACM/IEEE international conference on Human robot interaction (HRI'08), 241–248 (2008).
10. N. W. Kim, S. J. Lee, B.-G. Lee, and J.-J. Lee.: Vision based laser pointer interaction for flexible screens. Human-Computer Interaction. Interaction Platforms and Techniques, 4551 of Lecture Notes in Computer Science, 845–853 (2007).
11. H. Nakashima, H. Aghajan, and J. Augusto.: Handbook of Ambient Intelligence and Smart Environments, Springer (2010).
12. G. Olague. Evolutionary Computer Vision - The First Footprints -. : Springer (to appear).
13. G. Olague, E. Clemente, L. Dozal, and D. E. Hernández. : Evolving an Artificial Visual Cortex for Object Recognition with Brain Programming. Studies in Computational Intelligence, Springer (to appear).
14. G. Olague, L. Dozal, E. Clemente, and A. Ocampo. : Optimizing an Artificial Dorsal Stream on Purpose for Visual Attention. Studies in Computational Intelligence, Springer (to appear).
15. C. B. Pérez, and G. Olague. : Evolutionary learning of local descriptor operators for object recognition. GECCO, 1051–1058 (2009).
16. A. M. Treisman and G. Gelade.: A feature-integration theory of attention. Cognitive psychology, 12(1):97–136 (1980).
17. D. Walther and C. Koch.: Modeling attention to salient proto-objects. Neural Networks, 19(9):1395–1407 (2006).

Coevolución para el Diseño Automático de Máquinas de Estados Finitos*

J. Manuel Colmenar[1], Alfredo Cuesta-Infante[1], José L. Risco-Martín[2], J. Ignacio Hidalgo[2]

[1] C.E.S. Felipe II, U. Complutense de Madrid, 28300 Aranjuez, España
{jmcolmenar, alfredo.cuesta}@ajz.ucm.es
[2] Dept. de Arquitectura de Computadores y Automática
U. Complutense de Madrid, 28040 Madrid, España
{jlrisco, hidalgo}@dacya.ucm.es

Resumen En este artículo presentamos el sistema GOURMET (*Grammatical based Optimization Under fitness Refinement by Means of Evolutionary Techniques*). Se trata de una metodología evolutiva para la inferencia de máquinas de estado finitas basada en Gramáticas Evolutivas (Grammatical Evolution). Nuestra aproximación presenta dos claras ventajas. En primer lugar, el flujo de diseño acepta como entrada descripciones de alto nivel, haciendo más fácil y asequible el proceso a los diseñadores de sistemas. En segundo lugar, al contrario de lo que sucede con otras aproximaciones, el diseñador no necesita definir un conjunto de valores de entrada para entrenar al sistema, ya que estos valores son generados de manera automática mediante un algoritmo genético en tiempo de ejecución. Nuestros experimentos obtuvieron la máquina de estados correcta para el problema del reconocedor de secuencias en un 99,5 % de las ocasiones y en un 100 % para el problema de diseño de una máquina de vending.

Keywords: Grammatical evolution, Genetic programming, Genetic algorithms, Design/synthesis, Mathematical modeling/curve fitting

1. Introducción

Las máquinas de estados finitos (*finite state machines*, FSMs) proporcionan un mecanismo de alto nivel para la descripción de sistemas complejos, de manera que, a partir de una FSM, muchas herramientas de diseño son capaces de generar el sistema hardware subyacente. Sin embargo, acometer el diseño de una FSM desde cero requiere experiencia y conocimientos previos. En consecuencia, la creación de herramientas que ayuden en la creación automática de FSMs puede resultar muy útil, por ejemplo, para el prototipado rápido en diseño de hardware. Bajo este objetivo, y dado que la obtención de la FSM irreducible a

* Trabajo financiado por los proyectos TIN2008-00508 y MEC Consolider Ingenio CSD00C-07-20811 del Ministerio de Ciencia y Tecnología.

partir de un conjunto de entradas es un problema NP-completo [2], las técnicas metaheurísticas encajan perfectamente en esta clase de problemas de optimización. Varios trabajos de la literatura abordaron la inferencia de FSMs. En [5] se propuso un algoritmo eficiente consistente en la inferencia de un autómata finito determinista a partir de un conjunto de entradas, aplicando posteriormente un paso de minimización del número de estados. En este caso, el conjunto de entradas (conjunto de entrenamiento) provenía de la observación de sistemas existentes, y se porporcionaba como dato de entrada al inicio del algoritmo. En [4] se propone la generación de FSMs utilizando programación genética y estrategias evolutivas. Los resultados obtenidos no son demasiado buenos según los propios autores. Más recientemente, en [8] se presenta un algoritmo genético para la inferencia de FSMs, junto con un detallado estado del arte en el área bajo estudio. Sin embargo, el algoritmo requiere un conjunto de entrenamiento más una serie de reglas y fórmulas para describir las restricciones del sistema.

En consecuencia, hay dos líneas principales en la literatura. Por un lado, están aquéllas en las que se requiere un conjunto de entrenamiento de valores de entrada y sus salidas correspondientes. Sin embargo el problema de cómo seleccionar las entradas y la forma de determinar el tamaño del conjunto de entrenamiento por lo general no están cubiertos. Además, en la mayoría de las definiciones de los problemas existen algunas entradas que son más importantes que otras como, por ejemplo, los reconocedores de secuencias [3]. Por otro lado, es muy típico requerir fórmulas complejas, reglas o restricciones para describir el problema en cuestión. Creemos que una descripción de alto nivel del problema podría ser más cómodo y útil.

En este trabajo se propone un flujo de optimización que, por medio de la evolución gramatical (GE), en colaboración con un algoritmo genético (AG), se obtiene la descripción FSM que resuelve un problema de descrito a alto nivel verificando además un conjunto de valores de entrada generado automáticamente. Hemos llamado a este esquema de colaboración GOURMET (*Grammatical based Optimization Under fitness Refinement by Means of Evolutionary Techniques*). GOURMET genera automáticamente el conjunto de entrenamiento de los entradas que intervienen en el cálculo de la función de aptitud o fitness a través de un AG que se encarga de obtener el conjunto de entradas representativas. Este AG trabaja en colaboración con un bucle GE que busca la FSM concreta que resuelve un problema que se describe con un lenguaje de alto nivel. Hemos aplicado GOURMET a dos tipos diferentes de problemas: el detector de secuencia de 3 bits y el problema de la máquina expendedora. Los resultados mostraron que con el uso de una gramática sencilla y una breve descripción de alto nivel del problema, se encuentra con éxito y en un tiempo razonable, una FSM solución para casi el 100 % de las ejecuciones.

El resto del artículo está organizado como sigue. En la sección 2 se da una visión general del sistema. En la sección 3 se describe el espacio de búsqueda y la codificación. La sección 4 explica el proceso evolutivo con la colaboración del AG y la GE. Los resultados experimentales se recogen en la sección 5 y las conclusiones en la 6.

2. GOURMET, visión general

Cualquiera de nosotros es capaz de calificar o medir algún producto o resultado, pero no es capaz de producirlos. Incluso los expertos a veces carecen de la habilidad pura y, a veces la falta de conocimiento. Muchos ejemplos ilustran esta situación. Por ejemplo, un piloto de motocicletas puede ser capaz de determinar cuán bueno es el rendimiento de una moto sin necesidad de tener ningún conocimiento de mecánica o ingeniería. Por supuesto que esta información le ayudaría a ser más preciso, pero no es una característica obligatoria, como lo demuestran algunos campeones del mundo. Del mismo modo, un gourmet podría no ser capaz de cocinar, ni siquiera una tortilla, pero sus desarrollados sentidos le permiten apreciar todos los sabores en un plato.

De esta manera, cuando a un cocinero se le solicita cocinar un plato nuevo por primera vez, podría utilizar las habilidades del gourmet para mejorar su resultado hasta adivinar la receta. El gourmet podrá degustar el plato y asignarle una calificación. La evaluación propuesta por el gourmet será una realimentación que permite al chef mejorar el plato. Además, cuanto mayor sea el número de platos probados, mayor será la experiencia adquirida por el gourmet. El gourmet puede incluso recordar los platos del pasado y cambiar su calificación para ser más preciso y justo con el chef.

Desde el punto de vista algorítmico, el chef representa un algoritmo de búsqueda sobre la mejor solución, mientras que el gourmet representa el objetivo o función de aptitud para ser optimizado. De esta manera, la función de aptitud también mejorará a medida que el número de iteraciones del algoritmo de búsqueda crece.

Hemos transformado la analogía en un esquema de colaboración, como se muestra en la Figura 1, donde se definen tres módulos:

- GE: Evolución gramatical. Se corresponde con el chef, y genera las FSMs candidatas a ser evaluadas por el gourmet.
- FIT: Fitness. Pertenece a la alta cocina, y produce el resultado esperado para una entrada dada de acuerdo a una especificación de alto nivel del problema. Recibe como parámetros de entrada la FSM para su evaluación y un conjunto de entradas a procesar. Devuelve el fitness, calculado como el número de diferencias entre los resultados esperados y los dados por la FSM.
- AG: Algoritmo genético. También pertenece al gourmet y genera el conjunto de entradas que se evalúa para cada FSM candidata.

El módulo GE realiza la búsqueda de una solución FSM y, una vez encontrada, el módulo AG se utiliza para encontrar la secuencia de entrada para la que el FSM genera la peor salida. Este valor de entrada se incorporará al módulo FIT, y tomará parte en la evaluación de la aptitud de las FSM posteriores. A continuación se describe un ejemplo que corresponde a la búsqueda de una FSM que permite el reconocimiento de la secuencia 01 a partir de una entrada en serie de diez bits (con solapamiento). Inicialmente, el módulo FIT no tiene ninguna información previa, por lo que comienza con una entrada generada al

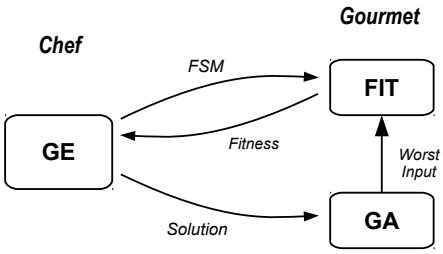


Figura 1. Principales módulos de la optimización con la analogía de chef y gourmet.

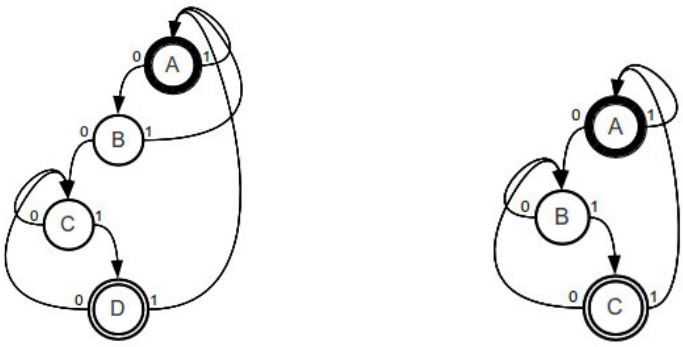
azar. Para este simple reconocedor, supongamos que genera 1001001010 como cadena de entrada, cuya cadena de salida correcta es 0001001010. Con el fin de encontrar la mejor solución, el módulo GE explora el espacio de búsqueda FSM mediante la evaluación de cada solución candidata con el módulo de FIT, que en este punto sólo tiene una entrada para la prueba. Después de cierta exploración, GE genera una solución que produce la salida correcta dada la entrada inicial (0 diferencias). Vamos a llamar a esta solución sol_1 . La Figura 2(a) muestra la primera FSM obtenida para esta muestra. Cada nodo del grafo tiene dos entradas posibles: 0, representado en la rama izquierda, y 1, representado en la rama derecha. Probablemente, esta primera solución no será la FSM *golden*, es decir, la FSM que produce la salida correcta para cualquier entrada dada. A continuación, el módulo de AG se ejecuta tomando sol_1 y trata de encontrar la peor entrada posible. Este valor será la secuencia de entrada para la que el número de bits de salida incorrectos es máxima. En el caso de que estemos frente a una solución final, el AG no encontrará ninguna cadena de entrada peor, pues ya se habrá conseguido una correcta. En el segundo paso, para nuestra sol_1 , el AG devuelve como cadena de entrada peor 0101010101. Su cadena de salida correcta es 0101010101, mientras que la cadena de salida generada por sol_1 es 0000000000. A continuación, se añade la nueva entrada al módulo FIT, que ahora cuenta con dos valores de entrada.

La utilización del módulo AG tiene dos beneficios diferentes. Por un lado, se comprueba si las soluciones candidatas propuestas por GE son *golden*, en cuyo caso, el AG no encontraría entrada alguna que produjese fallos. Por otro lado, el AG contribuye a la mejora de la función de aptitud añadiendo entradas que serán comprobadas en las siguientes iteraciones de GE.

Para el ejemplo comentado, al terminar las iteraciones definidas para el módulo GE, se obtuvo la FSM que se muestra en la Figura 2(b).

3. Espacio de Búsqueda y Codificación

Desde un punto de vista general, el trabajo presentado define un marco que es independiente del problema concreto que se quiere abordar. De esta manera, se definirá el espacio de búsqueda en términos de FSM y se explicará la codificación.



(a) Primera FSM. (b) FSM final.

Figura 2. FSMs para el detector de secuencia 01.

Siguiendo un esquema típico de Máquina de Moore, hemos definido una FSM como un grafo dirigido con las siguientes propiedades:

- Sólo hay un estado inicial, representada en las figuras por bordes en negrita.
- Cada estado produce una sola salida.
- A partir de un determinado estado existen tantas transiciones diferentes como valores de entrada podría recibir.
- Aquellas transiciones que conducen al estado inicial se denominan "Reset".

En un primer enfoque, teniendo en cuenta las salidas binarias, si un estado produce 1 como valor de salida, se denota el estado con un borde de doble línea. Si se produce un estado 0 como valor de la salida, se utiliza una línea simple. Como se ve en la Figura 2(b), el estado inicial de esa FSM es A, mientras que el único estado que genera 1 como salida es de C. Dado que el conjunto de posibles FSM para un problema dado tiene un tamaño infinito, es necesario limitar el espacio de búsqueda. En nuestra propuesta se aplican Gramáticas Evolutivas (GE) para explorar el espacio de búsqueda, por lo tanto, se utiliza una gramática para limitar el espacio de búsqueda.

Las Gramáticas Evolutivas (GE) [7] son una forma de programación genética basada en gramáticas. La Figura 3 muestra la gramática que hemos utilizado para en el espacio de búsqueda de FSMs. Como se puede ver, el símbolo no terminal `<FSM>` representa el estado inicial, que es siempre un estado, `<State>`. Cada `<State>` corresponde a un identificador, `<StateId>` asociado a un valor de salida (0 o 1, ver regla **IV**). Desde cualquier `<State>`, salen dos transiciones que se corresponden con `<Trans>`. La primera corresponde con el valor de entrada 0, mientras que la segunda corresponde con el valor de entrada 1.

Por lo tanto la gramática de la Figura 3 permite una FSM con un máximo de 8 estados que se corresponden con las etiquetas (`<StateId>`) A a D combinados con ambos valores de salida. El espacio de búsqueda queda limitado de esta forma a todas las FSMs que se pueden obtener con todas las posibles combinaciones de transiciones entre, como másimo, estos 8 estados.

```
N = {<FSM>,<State>,<StateId>,<Output>,<Trans>}
T = {Reset,0,1,A,B,C,D}
S = <FSM>

I    <FSM>       ::= <State>
II   <State>     ::= <StateId><Output><Trans><Trans>
III  <Trans>     ::= <State> | Reset
IV   <Output>    ::= 0 | 1
V    <StateId>   ::= A | B | C | D
```

Figura 3. Gramática que describe una FSM de hasta 8 estados: de A a D con salida 0 más A a D con salida 1.

4. Proceso Evolutivo

Como se ha explicado en la sección 1, proponemos un esquema evolutivo llamado GOURMET donde dos elementos principales colaboran: una gramática evolutiva (GE) y un algoritmo genético (AG). Por un lado, la gramática evolutiva intenta encontrar la máquina de estados finitos con el mejor fitness y, por otro lado, evolucionamos la función de fitness mediante un algoritmo genético. El módulo de GE recibe una población inicial de máquinas generadas de forma totalmente aleatoria. Este módulo detiene su ejecución cuando se encuentra una solución, esto es, cuando se encuentra una máquina de estados que no contiene errores, o cuando se llega al máximo número de generaciones. Hay que tener en cuenta que la gramática evolutiva utiliza el módulo de evaluación de fitness (FIT) para probar cada una de las máquinas candidatas.

El módulo de GE termina su ejecución cuando se encuentra una solución, es decir, al encontrar una FSM sin errores o cuando GE alcanza su máximo número de generaciones. Una vez que termina la GE, se comprueba el número de iteraciones, que se corresponde con el número de veces que se ejecuta el módulo de GE. Si no se alcanza este límite, el módulo de algoritmo genético (AG) realiza una búsqueda sobre la mejor solución proveniente de GE. Además, si GE encuentra más de una solución, el AG se ejecuta una vez para cada una de ellas. Por el contrario, Si la GE anterior no ha encontrado una solución, el AG se ejecuta para los mejores individuos.

El módulo GE dentro de GOURMET se divide en dos etapas consecutivas. El primer paso es una GE clásica donde se realiza la exploración sobre el espacio de búsqueda determinado por la gramática. Los parámetros para esta primera evolución se muestran en la Tabla 1, columna GE-1. Los valores de estos parámetros se eligieron tras un estudio experimental. El segundo paso del módulo de GE consiste en una búsqueda adicional sobre la población que viene de la primera GE con el objetivo de aumentar el número de soluciones o, al menos, aumentar el número de individuos similares a la solución obtenida. Por lo tanto, una búsqueda corta (100 generaciones) se ejecuta con una probabilidad alta de mutación, mientras que la probabilidad de cruce es baja. La Tabla 1, columna GE-2, muestra los valores de los parámetros para la segunda fase de la GE. Como resultado,

se realiza una búsqueda local que permite la mejora de las soluciones obtenidas, que se utilizarán más adelante en el módulo del AG.

Cuadro 1. Resumen de los parámetros utilizados en las dos fases de GE y en el AG.

Parámetro	GE-1	GE-2	AG
Tamaño de la población	500	500	200
Generaciones	6000	100	1000
Probabilidad de cruce	0.85	0.2	0.8
Probabilidad de mutación	0.2	0.95	0.02
Tamaño de cromosoma	100	100	100
Max. número wraps	0	0	-
Fin si encuentra solución	true	false	false

Tanto las fases de GE como el AG utilizan cruce basado en un solo punto, mutación entera uniforme y selección por torneo. También se implementa elitismo manteniendo los 10 mejores individuos.

La función de aptitud en GOURMET es una medida de la distancia, en número de valores diferentes, entre la salida generada por la FSM en evaluación y la salida esperada (correcta). Por lo tanto, se evalúa cada individuo teniendo en cuenta el conjunto actual de valores de entrada. Entonces, para cada entrada, la salida propuesta por el individuo se compara con la salida correcta esperada. El número de diferencias entre las dos salidas se acumula, obteniendo así un valor de fitness. Los individuos que no tienen errores recibirán el mejor valor de aptitud (el más bajo), que es de 0. La aptitud se obtiene así:

$$Fitness = \sum_{i=1}^{N} \sum_{j=1}^{L} \mathit{diff}\,(i,j)$$

donde N es el número de cadenas de entrada actual, L es la longitud de la secuencia de entrada, y *diff* (i,j) devuelve 1 si el bit de salida j es incorrecto para la entrada i, y 0 en caso contrario.

La función de aptitud depende, por tanto, de las cadenas de entrada de las que se disponga en cada momento. Estas cadenas son seleccionadas por el AG tras la búsqueda de la peor entrada para cada una de las soluciones candidatas en cada iteración de GE. En consecuencia, se produce una coevolución entre GE y AG. La Tabla 1 muestra también los parámetros seleccionados para el AG.

Tras cada iteración, las mejores soluciones (élites) obtenidas en la ejecución de ambas fases GE se reinsertan en la población aleatoria que se crea para la siguiente iteración sobre GE. Esto permite conservar la información genética de los mejores individuos a la vez que permite a las soluciones correctas pervivir a lo largo de las iteraciones previstas para GE.

Hemos implementado GOURMET en Java. El módulo de GE fue codificado utilizando GEVA [6], una conocida herramienta de GE basada en Java, mientras que el resto del código fue desarrollado por nuestro grupo de investigación. Los algoritmos, programas y pruebas están a disposición del público en [1].

```
N = {<FSM>,<InitState>,<State>,<StateId>,<Output>,<Trans>}
T = {Reset,0,1,A,B,C,D,E}
S = <FSM>

I    <FSM>       ::= <InitState>
II   <InitState> ::= <StateId>0<Trans><Trans>
III  <State>     ::= <StateId><Output><Trans><Trans>
IV   <Trans>     ::= <State> | Reset
V   <Output>    ::= 0 | 0 | 0 | 1
VI  <StateId>   ::= A | B | C | D | E
```

Figura 4. Gramática para el reconocedor de secuencia de 3 bits.

5. Resultados experimentales

GOURMET se ha probado en dos problemas diferentes: los detectores de secuencia y el problema de la máquina expendedora. En ambos usamos cadenas de entrada de 100 bits de longitud, que corresponde a una enorme espacio de búsqueda para el conjunto de entrada (2^{100} combinaciones).

Aplicamos GOURMET con el objetivo de encontrar las ocho FSM que implementan todos posibles reconocedores de secuencia de 3 bits. Para ello utilizamos la gramática de la Figura 4, presentando un estado inicial con un valor de salida fijo de 0. Además hemos reducido la probabilidad de tener 1 como salida modificando la regla **V** para `<output>`.

Para cada uno de la ocho patrones diferentes del reconocedor de 3 bits, lanzamos GOURMET 30 veces, iterando 50 veces el bucle principal de la GE. Después de cada ejecución, se obtuvo una FSM como resultado. Se encontró que las FSMs resultantes podrían ser de tres tipos diferentes: soluciones *golden*, es decir, FSM no reducible que resuelve correctamente el problema; soluciones reducibles, es decir, soluciones equivalentes a la correcta, pero con exceso de estados (reducible); y FSMs que no resultaban ser solución puesto que obtuvieron valores de fitness superiores a 0.

La Tabla 2 resume los resultados en seis columnas. En primer lugar, se indica el patrón de 3 bits, seguido, respectivamente, por el porcentaje de soluciones *golden* y reducibles encontradas. La cuarta columna muestra el número de ejecuciones que no encontraron ninguna solución. Las columnas quinta y sexta indican el tiempo medio de ejecución de un ciclo de optimización, y el número total de de entradas procesadas teniendo en cuenta todas las iteraciones para cada patrón.

Como se ve en los resultados, nuestro algoritmo obtiene soluciones correctas para todas las ejecuciones, excepto para una del patrón de 011 que no encontró ninguna solución. Tenga en cuenta que sólo estamos interesados en soluciones sin fallos. Por lo tanto, FSMs parcialmente correctas no son consideradas como soluciones. Por otra parte, todas los FSM obtenidas fueron validadas, y se ha verificado que eran exactamente la misma que la solución correcta minimizado FSM para cada patrón, o cualquier máquina reducible equivalente.

A partir de los resultados que se muestran en la Tabla 2 se puede observar que es más difícil de encontrar la FSM solución en algunos de los patrones.

Cuadro 2. Resultados para los reconocedores de secuencia de 3 bits.

Patrón	Golden	Reducible	NO sol.	Avg. T. (secs)	# Inputs
000	23.33 %	76.66 %	0	10735.78	3336
001	65.52 %	34.48 %	0	6465.56	1756
010	93.10 %	6.89 %	0	8860.21	1456
011	93.10 %	3.45 %	1	17995.69	1529
100	92.86 %	7.14 %	0	6685.07	1568
101	96.55 %	3.45 %	0	6012.92	1428
110	63.33 %	36.66 %	0	6298.69	1833
111	23.33 %	76.66 %	0	9862.46	3211

En concreto, en aquellos en los que todos los bits son iguales: 000 y 111. Para ambos, el porcentaje de soluciones golden es bajo y el tiempo de ejecución es alto. El tiempo de ejecución está directamente relacionado con el rendimiento de la búsqueda. Si una iteración sobre GE encuentra un FSM correcta, ya sea golden o reducible, el AG no obtendrá una cadena de entrada que provoque fallos. Por lo tanto, dado que las elites se reinsertan en la población de la próxima iteración de GE, la solución será un inmigrante. Entonces, GE se ejecutará sólo una generación, ya que está configurado para detenerse cuando se encuentra una solución, reduciendo el tiempo total de ejecución. Por lo tanto, cuanto más corto el tiempo de ejecución, más pronto se encontró la solución.

Además, definimos un valor umbral para las iteraciones de GE donde no se encontró una solución. En estos casos, el AG se ejecuta sobre estas FSMs con el fin de obtener sus peores entradas. De esta manera, se contribuye a la mejora del conjunto de entradas para FIT, de modo que no se desperdicia el trabajo realizado por la iteración sobre GE. Sin embargo, en esos casos, no hay inmigrantes enviados a la siguiente iteración de GE. Este hecho es el que hace que las ejecuciones para los patrones 000 y 111 sean más lentas puesto que cada GE ejecuta todas sus iteraciones.

El problema de la máquina expendedora representa un escenario diferente. Podemos describirlo con las siguientes reglas: (1) La máquina dispensa un producto después de haber recibido una cierta cantidad de dinero. (2) Después de la distribución del producto (salida 1), la máquina vuelve al estado inicial. (3) Se aceptan dos tipos de monedas: 5 ¢, y 10 ¢, y no devuelve cambio. A pesar de la simplicidad de la descripción de alto nivel de este problema, la búsqueda de una FSM reducida que resuelva el problema es más difícil que en el problema reconocedor de secuencia. Hemos abordado el problema de la máquina expendedora para un importe de 15 ¢. La configuración experimental y la gramática fueron los mismos que en el problema de reconocimiento de patrones. La optimización se ha realizado correctamente porque todas las 30 ejecuciones obtuvieron la FSM *golden*. El tiempo de ejecución promedio fue de 25.330,45 segundos, y 1.274 entradas fueron procesadas. En consecuencia, todos los problemas fueron resueltos con éxito por GOURMET obteniendo la FSM no reducible.

6. Conclusiones y trabajo futuro

Las FSM son descripciones útiles de bajo nivel que solucionan muchos tipos de problemas basados en estado. Encontrar una implementación correcta de FSM es un problema que se complica cuando el número de estados y transiciones crece. En este trabajo, proponemos un flujo evolutivo donde se obtienen, de manera automática, implementaciones de FSM a través de gramáticas evolutivas. Utilizando un algoritmo genético, el flujo genera automáticamente el conjunto de entrenamiento de las entradas que se probarán por la función de fitness, descubriendo las entradas con la mayor tasa de fracaso. Además, no se requieren reglas o fórmulas complejas, sino que sólo se requiere un código Java de alto nivel que resuelva el problema abordado. Hemos llamado a este flujo evolutivo GOURMET (*Grammatical based Optimization Under fitness Refinement by Means of Evolutionary Techniques*). Nuestros experimentos mostraron muy buenos resultados para el problema del reconocedor de secuencia de 3 bits, donde se obtuvo la FSM solución no reducible en el 68,38 % de las ejecuciones, una solución reducible equivalente en el 31,20 % de los casos, y simplemente no encontraron una solución en una sola ejecución para la secuencia de 011. Además, el problema de la máquina expendedora para 15 ¢ también fue resuelto con éxito, obteniendo la FSM reducida en el 100 % de las ejecuciones.

La fase de evaluación es el cuello de botella en GOURMET, por lo que el aumento de paralelismo reducirá el tiempo de ejecución, lo que permitirá intensificar la exploración para poder hacer frente a futuros problemas más complejos.

Referencias

1. JECO: Java Evolutionary Computation library, 2013. `http://code.google.com/p/paba/`.
2. E. M. Gold. Complexity of automaton identification from given data. *Information and Control*, 37(3):302 – 320, 1978.
3. S. Goren and F. Ferguson. Checking sequence generation for asynchronous sequential elements. In *International Test Conference 1999. Proceedings (IEEE Cat. No.99CH37034)*, pages 406–413. Int. Test. Conference.
4. C. Johnson. Genetic programming with fitness based on model checking. In M. Ebner, M. O'Neill, A. Ekárt, L. Vanneschi, and A. Esparcia-Alcázar, editors, *Genetic Programming*, volume 4445 of *Lecture Notes in Computer Science*, pages 114–124. Springer Berlin / Heidelberg, 2007.
5. A. L. Oliveira and J. P. Silva. Efficient algorithms for the inference of minimum size DFAs. *Machine Learning*, 44:93–119, 2001.
6. M. O'Neill, E. Hemberg, C. Gilligan, E. Bartley, J. McDermott, and A. Brabazon. GEVA - grammatical evolution in Java. *SIGEVOlution*, 3(2):17–22, 2008.
7. M. O'Neill and C. Ryan. Grammatical evolution. *IEEE Trans. Evolutionary Computation*, 5(4):349–358, 2001.
8. F. Tsarev and K. Egorov. Finite state machine induction using genetic algorithm based on testing and model checking. In *Proceedings of the 13th annual conference companion on Genetic and evolutionary computation*, GECCO '11, pages 759–762, New York, NY, USA, 2011. ACM.

Sumador evolutivo de 2 bits tolerante a fallos

Juan Lanchares, Oscar Garnica, J. Ignacio Hidalgo, and Jose L. Risco-Martín

Grupo de Arquitecturas Paralelas y Algoritmos Bioinspirados,
Universidad Complutense de Madrid, España
http://bioinspired.dacya.ucm.es

Resumen En este artículo se presenta un SoC (del inglés System on Chip) evolutivo que converge a un sumador de dos bits tolerante a fallos. El SoC se compone de un procesador Microblaze sobre el que se ejecuta un algoritmo genético que realiza la evolución, una FPGA (del inglés Field Programmable Gate Array) virtual implementada sobre la FPGA real cuya misión es realizar la evaluación de los circuitos durante la evolución e implementar el circuito final obtenido, y un módulo de evaluación encargado de generar los vectores de entrenamiento y los valores esperados para calcular la función de coste de cada circuito. Se han realizado tres pruebas diferentes. En la primera de ellas se permite que el circuito evolucione pudiendo seleccionar 8 funciones diferentes en cada Bloque Lógico Configurable o CLB (de sus siglas en inglés Configurable Logic Block) virtual. En la segunda, sólo se van a utilizar en los CLBs las funciones necesarias para implementar un sumador tradicional, lo que sería equivalente a permitir exclusivamente la evolución de las conexiones. Por último se han simulado fallos en la FPGA virtual, en concreto se ha provocado el fallo del 16 % de los CLB virtuales. Como resultados se ha obtenido que permitiendo únicamente la evolución de las conexiones se alcanza la convergencia mucho más rápidamente y que si se producen fallos el sistema es capaz de evitar las zonas dañadas de la placa alcanzando la convergencia al circuito esperado.

Keywords: Hardware Evolutivo, Tolerancia a Fallos, Estrategias Evolutivas

1. Introducción

El hardware evolutivo (EHW de sus siglas en inglés, Evolvable Hardware) es un hardware capaz de modificar su estructura interna autónomamente, es decir, sin intervención del diseñador, para alcanzar el circuito objetivo. La modificación de la estructura se realiza en tiempo de ejecución y puede deberse a dos factores: las condiciones de trabajo del entorno han cambiado (variaciones de temperatura, picos de potencia, radiaciones, cambios de localización, ajuste fino de parámetros de algoritmo, etc.) o se ha producido un fallo en alguno de los componentes del sistema. En ambos casos la modificación se produce o bien porque el hardware ha dejado de funcionar correctamente o bien porque la calidad de los resultados generados por el hardware ha disminuido. Es importante resaltar que las modificaciones se realizan sin que previamente el diseñador tenga

conocimiento de cómo van a evolucionar las condiciones internas y externas de trabajo [6] [3] [2].

El EHW se inspira en el principio darwiniano de evolución de forma que un conjunto de posibles soluciones (circuitos) compiten entre sí para poder llegar a la fase de reproducción en la que las características de dos de ellos se combinan para obtener un circuito probablemente mejor. Este nuevo circuito formará parte de una nueva población más apta en promedio que la anterior [8][4] [1].

Una de las principales ventajas del EHW se deriva del hecho de que sólo tiene en cuenta el valor de la función de coste para autodiseñarse, y esto le dota de una gran adaptabilidad en tiempo de ejecución. Esta característica puede ayudar a solucionar problemas de tolerancias a fallos y de adaptación a entornos cambiantes, problemas que aparecen a menudo en aplicaciones críticas como por ejemplo las misiones espaciales o problemas en medicina.

Entre los ejemplos de utilidad resaltan la tolerancia a fallos cuando el sistema está sometido a radiaciones extremas o a temperaturas que puedan dañar los dispositivos lógico-programables, y el diseño de estructuras in situ en cuyo caso el sistema se puede diseñar directamente en el entorno en el que va a trabajar. Esta tecnología se ha utilizado para diseñar una antena incorporada en la misión Marte Odyssey, lanzada en el 2001 y que en la actualidad continúa en funcionamiento y enviando imágenes [5].

En este artículo se presenta un SoC evolutivo que converge a un sumador de dos bits tolerante a fallos. Los resultados experimentales demuestran que permitiendo únicamente la evolución de las conexiones se alcanza la convergencia mucho más rápidamente y que si se producen fallos el sistema es capaz de evitar las zonas dañadas de la placa alcanzando la convergencia al circuito esperado.

El resto del trabajo está organizado de la siguiente manera. En la sección 2 se hace una breve introducción al Hardware Evolutivo. En la sección 3 se explica la propuesta y en la 4 se dan los detalles del sistema hardware, es decir del System On Chip que implementa el EHW. En la sección 5 se comentan los resultados experimentales y en la sección 6 se exponen las conclusiones obtenidas en este trabajo y las posibles líneas de trabajo futuro.

2. Hardware Evolutivo

Como ya hemos mencionado, el hardware evolutivo se basa en los principios darwinianos de la evolución. Por lo tanto, trabaja con un conjunto de posibles soluciones (representaciones de circuitos) o poblaciones de individuos, de manera que los circuitos compiten entre sí para poder llegar a la fase de reproducción en la que las características de dos de ellos se combinan para obtener un circuito probablemente mejor que formará parte de una nueva población más apta en promedio que la anterior.

Para conseguir esto no se trabaja directamente con circuitos sino con representaciones de los mismos. Un circuito se representa mediante la cadena de ceros y unos necesaria para implementarlo en una FPGA. A este código se le

llama cromosoma. Un algoritmo evolutivo se encarga de llevar a cabo la evolución de la población de cromosomas. La evolución se inicia evaluando cada uno de los cromosomas de la población. La evaluación se utiliza para saber lo que se aproxima un circuito al objetivo. Para realizarla el cromosoma se utiliza para programar la FPGA (ver Figura 2. Una vez programada se envía al circuito un vector de test. Las salidas proporcionadas por el circuito se comparan con las salidas esperadas. Contabilizando el número de coincidencias entre las salidas esperadas y las salidas generadas se tiene una medida de lo que se aproxima el circuito que está siendo evaluado al circuito objetivo [2].

Después de evaluados todos los cromosomas se genera la siguiente población utilizando operadores de cruce y mutación. El operador de cruce consiste en seleccionar dos cromosomas e intercambiar su información entre sí mientras que el operador de mutación consiste en seleccionar un cromosoma y aplicar cambios aleatorios a los bits que componen el cromosoma. Ambos operadores se aplican con una determinada probabilidad. Este proceso de evolución se repite hasta que se obtiene un circuito que cumple las especificaciones deseadas. Finalmente, el circuito obtenido se implementa sobre una FPGA. Este proceso puede verse modificado para implementar otro tipo de algoritmo evolutivo, como por ejemplo el que se utiliza en este artículo, una Estrategia Evolutiva (EE). La EE aplica solamente el operador de mutación para generar nuevos individuos.

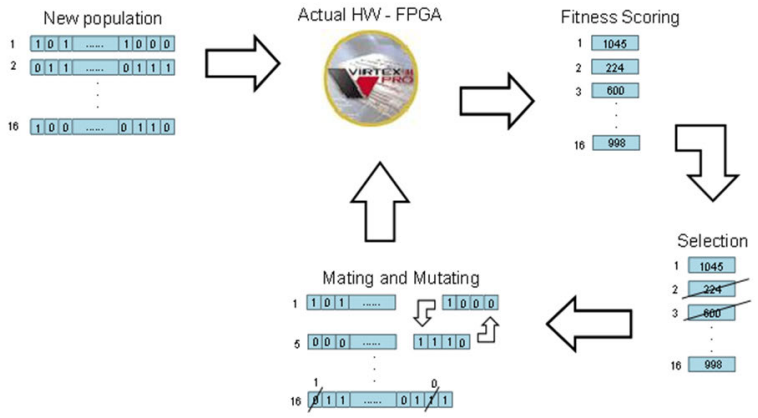


Figura 1. Sistema de Hardware Evolutivo.

Como se puede deducir de lo expuesto, este tipo de hardware se diferencia del clásico en dos aspectos fundamentales: la adaptación autónoma en tiempo de ejecución y el modo en que se implementa el circuito. Centrándonos en la primera diferencia, el hardware clásico no puede modificar su estructura sin una intervención -directa o indirecta- del diseñador, y sin haber tenido conocimien-

to y haber previsto determinados cambios en el entorno. En el caso del EHW esta adaptación se realiza en tiempo de ejecución sin intervención externa y sin necesidad de evaluar previamente los posibles cambios que se produzcan en el entorno ya que es precisamente el proceso evolutivo descrito quien lo realiza. En cuanto al modo de implementación del circuito, en el hardware clásico hay que fijar qué se quiere hacer mediante un conjunto de especificaciones y el cómo se hace viene prefijado por un flujo de diseño conocido. No es así en el caso del EHW, donde sólo hay que indicar qué se quiere hacer mediante la función de coste, pero no es necesario aportar ningún conocimiento de cómo se va a realizar la implementación puesto que ésta la realiza el algoritmo evolutivo [7].

El mayor problema del EHW es la escalabilidad. Hasta el momento se ha conseguido implementar circuitos de extrema sencillez si se les compara con el grado de complejidad existente en la actualidad. Esto se debe a varios factores que se explican a continuación. El primero de ellos es la propia escalabilidad de la representación. Los circuitos complejos necesitan muchas puertas y conexiones para implementarse. Para poder representar estos circuitos se necesitan cromosomas extremadamente grandes que a su vez dan lugar a espacios de soluciones enormes. Por ejemplo, para implementar un circuito de unas cien puertas lógicas se podría necesitar, dependiendo de la representación, un cromosoma de hasta dos mil bits.

Otro problema asociado a la escalabilidad es la forma en la que se debe representar la función de coste. Aparentemente la forma más sencilla de fijar el objetivo de convergencia es usar las tablas de verdad que describen el comportamiento del sistema, pero esto método queda invalidado en el momento que el circuito alcanza cierta complejidad. En estos casos se deben utilizar métodos alternativos siendo uno de los más extendidos utilizar unos vectores de entrenamiento más unos vectores de test que nos permiten en cierta medida cubrir todos los posibles comportamientos del circuito.

En este sentido es en el que nosotros hacemos nuestra propuesta. La idea no es evolucionar todo un circuito desde cero en el que la obtención de la función de coste es costosísima, y el espacio de soluciones enorme lo que hace la convergencia prácticamente imposible. Nuestra propuesta aplicada a la tolerancia a fallos es partir de un circuito que se sabe que funciona, cuyos módulos funcionales son conocidos de antemano , de manera que se realiza un proceso de ubicación y rutado evolutivo, lo que disminuiría enormemente la complejidad del cromosoma, de los operadores genéticos y por lo tanto se reduciría enormemente el espacio de soluciones , consiguiendo una convergencia en mejores tiempos. En definitiva la idea es incluir información del problema a la evolución para facilitarla.

3. La Propuesta

Cuando surgió la idea del hardware evolutivo el principal objetivo de la evolución no sólo era converger a un circuito que cumpliera los funcionalidad fijada, sino que de alguna manera este circuito mejorara la implementación que se pudiera hacer mediante métodos clásicos de diseño, bien fuera porque en la adapta-

ción era capaz de hacerse tolerante a fallos, bien porque el diseño final mejoraba en área, consumo o rendimiento al clásico. Para conseguir este objetivo la idea general era dar la máxima flexibilidad posible al diseño y esto se conseguía en parte fijando un amplio repertorio de funciones que se podían implementar en los CLBs de la FPGA. Esto da lugar a cromosomas de una elevada complejidad. Por ejemplo si un CLB puede implementar 16 funciones diferentes, el cromosoma necesita 4 bits para seleccionar cada una de las posibles funciones del CLB. Por lo tanto, si el circuito se compone de 1000 CLBs, el cromosoma necesita 4000 genes sólo para representar las funciones. A esto habría que añadirle los genes necesarios para representar las conexiones entre los CLBs. como resultado el tamaño final del cromosoma hace prácticamente inviable la convergencia del circuito en unos tiempos aceptables, sobre todo si estamos hablando de sistemas autónomos on-line.

Para solucionar el problema, algunas propuestas usan bloques lógicos más complejos, es decir, en lugar de utilizar puertas lógicas básicas, se utilizan bloques funcionales más complejos, de manera que se reduce la complejidad del cromosomas y por lo tanto del espacio de búsqueda. El problema de esta aproximación es que conceptualmente sigue la misma línea que la anterior. La diversidad de bloques que se pueden seleccionar es muy amplia con lo que la complejidad y amplitud del espacio de búsqueda sigue siendo demasiado grande. El problema de ambas aproximaciones es que pretenden realizar la evolución desde cero, sin incluir ningún tipo de información al sistema evolutivo del circuito final a obtener a parte de la proporcionada por la función de coste. Esta aproximación puede ser correcta cuando la evolución se aplica al diseño evolutivo, cuyo objetivo es encontrar un circuito que mejore en algún aspecto los obtenidos mediante un diseño clásico y en el que se tiene a disposición del diseñador toda la infraestructura de un laboratorio. Pero no es una aproximación válida cuando lo que se pretende es implementar un sistema evolutivo autónomo en el que ni se dispone de las infraestructuras de un laboratorio, ni el tiempo de evolución puede ser indeterminado puesto que la adaptación a estímulos externos o externos se tiene que llevar a cabo en tiempos aceptables.

En los sistemas evolutivos autónomos la idea es que el circuito evolucione o bien para adaptarse a cambios en las condiciones de trabajo externas, o bien a condiciones de trabajo internas (fallos hardware). En estos casos se parte de un diseño en concreto cuya funcionalidad se conoce de antemano, y que está diseñada y probada, por lo tanto no es necesario que el circuito evolucione desde cero, es suficiente que el circuito evolucione para poder adaptarse a las necesidades.

Esto lleva a una importante conclusión: en el caso de los sistemas evolutivos autónomos, el sistema debería incluir información sobre el circuito a evolucionar, lo que se puede traducir en que las funciones a utilizar deberían ser exclusivamente aquellas que de antemano sabemos va a utilizar el circuito. De esta manera se reduce enormemente el espacio de soluciones dando lugar a una convergencia más rápida. Esta propuesta debería estudiarse en más profundidad para el caso de la adaptación a variaciones externas puesto que en estos casos la adaptación

puede no afectar tanto a la funcionalidad del circuito como a otros factores, como ralentización del circuito por cambios bruscos de temperatura, parece totalmente adecuada cuando se utiliza con el objetivo de hacer los circuitos tolerantes a fallos. Suponiendo que un conjunto de CLBs y de conexiones de una FPGA falla el objetivo sería recolocar los bloques en otros CLBs activos y reconexionar el circuito hasta alcanzar la funcionalidad deseada evitando las zonas dañadas. En definitiva ya no se buscan nuevas funciones sino que se recolocan las funciones que sabemos implementan la funcionalidad del circuito.

El trabajo que presentamos en este artículo es una primera aproximación a la propuesta, cuyo objetivo es observar las diferencia en la convergencia entre la aproximación sin conocimiento del problema y con conocimiento del problema. El circuito que vamos a utilizar es un sumador de dos bits, circuito sencillo y en el que evidentemente no vamos a poder comprobar la diferencia de escalabilidad entre las dos aproximaciones. Pero, como decimos, este trabajo es sólo una primera aproximación y esperamos en un futuro poder enfrentarnos a problemas de más envergadura.

4. El sistema

La plataforma general de EHW se puede ver como un sistema empotrado formado por los siguientes componentes: un microprocesador sobre el que se ejecutará el algoritmo evolutivo, la FPGA virtual encargada de implementar los circuitos cuyo código se envía desde el microprocesador, y el IP evaluador encargado de evaluar la validez de los circuitos cargados en la FPGAV, el sistema de memoria, buses de sistema y periféricos. A continuación se explican en detalle cada uno de los módulos. La Figura 2 representa un esquema general del sistema de hardware evolutivo.

4.1. El microprocesador

El objetivo del procesador es ejecutar el algoritmo genético que lleva a cabo la evolución. Desde el procesador se envía a la FPGA virtual el cromosoma que configura el circuito a evaluar. Una vez evaluado, el circuito se envía al procesador el valor de la función de coste obtenida para que el algoritmo pueda seguir su evolución.

4.2. La FPGA virtual o FPGAv

Este módulo es el encargado de implementar los circuitos cuyos cromosomas se envían desde el procesador. Esta FPGAV es una matriz de nodos programables conectados mediante una red de interconexión que se implementa sobre la FPGA real. Esta FPGAV debe estar conectada con el IP evaluador de manera que pueda recibir el vector de test (función de coste) y enviar las salidas generadas, y con el procesador para poder recibir el cromosoma que describe el circuito que se quiere implementar.

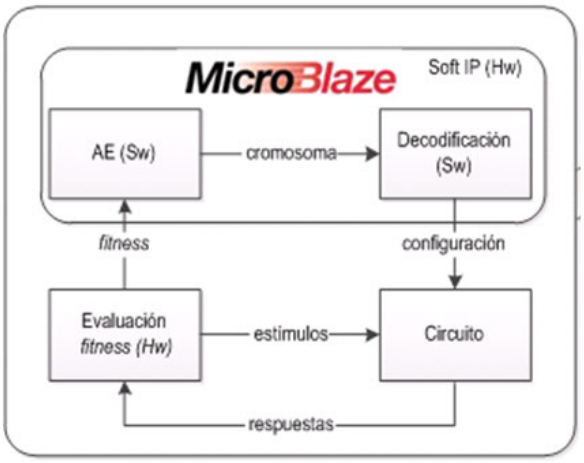


Figura 2. Sistema de Hardware Evolutivo.

4.3. IP evaluador

El IP evaluador es el encargado de realizar la evaluación de los circuitos cuyo cromosoma se envía desde el procesador del sistema. Tendrá que ser capaz de generar un vector de test y un vector de salidas esperadas. Para realizar la evaluación se envía el vector de test a la FPGAv y se comparan las salidas generadas con las salidas esperadas contabilizando los aciertos obtenidos.. Este módulo también debe incluir toda la lógica necesaria para intercambiar información con el procesador. El IP evaluador es de vital importancia porque a la postre es el que va a guiar la evolución de los circuitos.

5. Resultados experimentales

Con el objetivo de validar la propuesta se ha estudiado la convergencia de un sumador de 2 bit en tres casos diferentes. En el primero, el algoritmo evolutivo puede seleccionar 8 funciones diferentes para cada CLB del circuito programable, en el segundo sólo se pueden seleccionar las 3 funciones lógicas necesarias para implementar el sumador, y en el tercer experimento se estudia la convergencia de este último circuito simulando fallos en el circuito programable. En todos los casos se ha utilizado una Estrategia Evolutiva (λ+1) para diferentes valores de λ. Este tipo de algoritmo evolutivo genera λ soluciones nuevas mediante mutación, a partir de la mejor solución (+1) de la generación anterior. El padre no se conserva de una generación a otra

Para implementar el hardware evolutivo se ha utilizado el sistema de desarrollo XUP Virtex-II Pro. Debido a que es un sistema anticuado han surgido ciertos problemas que ha habido que solucionar. Por ejemplo la FPGA virtual no entraba en tiempos por lo que se tuvo que segmentar en tres etapas. Por

Parámetro 1	Valor
Elitismo	1
Tamaño de la población	λ
Longitud del cromosoma	216
Probabilidad de cruce	0 %
Probabilidad de mutación de funciones	100 %
Probabilidad de mutación de conexiones	100 %
Límite de generaciones	20.000

Cuadro 1. Parámetros de las Estrategias Evolutivas utilizadas

otro lado la cantidad de memoria BRAM del sistema no es muy elevada lo que ha obligado a utilizar poblaciones de individuos muy pequeñas comparadas con el espacio total de soluciones lo que en cierta medida ralentiza la convergencia del circuito. Los circuitos obtenidos se han simulado con *Modelsim* para asegurarnos que efectivamente el circuito obtenido implementa el circuito buscado. A continuación se explica con más detalle cada uno de los experimentos

5.1. Convergencia sin información del circuito

En esta primera prueba el sistema sólo va a recibir información sobre QUÉ es lo que queremos diseñar y en ningún caso se le da información de CÓMO queremos diseñarlo. Es decir, la convergencia se va a guiar exclusivamente por la función de coste del sumador plasmada mediante las tablas de verdad del circuito. como no se da ninguna información sobre como implementar el circuito vamos a permitir que el número de funciones que puede seleccionar el algoritmo genético sea lo suficientemente amplia y flexible. con este objetivo se han fijado las 8 funciones lógicas que aparecen en la tabla 2. como se puede observar en estas funciones aparecen 3 conjuntos completos lo que nos asegura que se va a poder implementar cualquier circuito lógico. También se añade una función "dejar pasar" cuyo objetivo es utilizar el CLB como una conexión dado que el número de conexiones de la FPGA virtual es muy reducido.

función	código cromosoma		
AND	000	NOR	100
OR	001	XNOR	101
XOR	010	NOT	110
NAND	011	DEJAR-PASAR	111

Cuadro 2. Funciones lógicas disponibles en un CLB virtual

5.2. Convergencia con información del circuito

En este experimento añadimos conocimiento del problema al sistema. Sabiendo que un sumador se puede implementar usando exclusivamente puertas

lógicas XOR, AND y, NOT sólo vamos a permitir seleccionar esas puertas en la estrategia evolutiva. Al hacer esto estamos limitando la complejidad y el tamaño del espacio de búsqueda y por lo tanto facilitando la convergencia. La limitación de funciones a usar en el diseño no se realiza modificando la FPGA virtual sino que se realiza modificando las funciones del algoritmo genético que generan la población inicial aleatoria y la mutación de funciones. En ambos casos sólo se permiten las funciones xor, and, not y dejar pasar.

5.3. Estudio de la tolerancia a fallos

En el tercer estudio que realizamos comprobamos que el circuito puede evolucionar esquivando zonas dañadas de la fgpa para alcanzar un circuito final que cumple los requisitos. Los fallos los vamos a simular mediante software, forzando al algoritmoevolutivo a comportarse de una determinada manera. En concreto vamos a simular los fallos de 4 CLBs de la FPGA virtual. Para ello, suponemos que esas celdas no pueden generar ninguna función útil y por lo tanto ningún CLB de la FPGA puede conectar su entrada a la salida de los CLBs dañados. Esto traducido al algoritmo significa que determinadas conexiones de la FPGA virtual no van a poder seleccionarse en las diferentes fases del algoritmo. Para estudiar en más en profundidad el comportamiento bajo fallos hemos realizado tres pruebas diferentes:

- Desde el inicio de la evolución se suponen los CLBs dañados
- Los CLBs se dañan en mitad de la evolución
- Los CLBs se dañan una vez que se ha alcanzado un circuito válido

5.4. Discusión

La tabla 3 expone los resultados experimentales para el caso de convergencia con información del circuito, pero los tiempos de convergencia y los datos relativos a individuos son similares en todos los casos. No creemos de mayor utilidad un análisis más exhaustivo de los datos, puesto que nuestro objetivo es probar que la metodología funciona. La tabla recoge los valores de ejecución de la estrategia evolutiva para distintos valores de λ. Los resultados recomiendan un valor de entre 15 y 20 para λ. Los resultados experimentales demuestran que el sistema evoluciona y consigue converger a pesar de estar trabajando sobre un hardware defectuoso, en el cual hay CLBs dañados, por los que el sistema no puede acceder en su camino hacia la evolución. El sistema detecta si hay fallo al principio, y comienza la evolución disponiendo de menos CLBs. Obviamente, el número de generaciones necesarias para obtener la convergencia, es superior (en torno al 20 %) con respecto al caso de un hardware libre de fallos, ya que dispone de menos CLBs teniendo así mayores restricciones.

6. Conclusiones

En este artículo se ha presentado un sumador evolutivo de 2 bits tolerante a fallos. El sistema haciendo uso del hardware evolutivo, es capaz de obtener

λ	25 ind.	20 ind.	15 ind.	12 ind.
Tiempo por individuo (s)	1,55	1,30	1,29	1,28
Tiempo en 500 generaciones (s)	38,66	26,09	19,33	15,33
Generaciones (avg)	4416	4939	6779	13106
Tiempo medio (s)	341	257	262	401

Cuadro 3. Resultados experimentales

un circuito sumador implementado sobre una FPGA. Para realizar la evolución se ha implementado una Estrategia Evolutiva $(\lambda + 1)$, que converge en tiempos aceptables. Para comprobar la tolerancia a fallos, se han simulado errores bajo tres escenarios diferentes, en todos ellos el sistema es capaz de encontrar una nueva implementación que evite las zonas dañadas. Se trata de un trabajo preeliminar por lo que las posibles lineas de trabajo y sus aplicaciones son muy amplias, desde el diseño de nuevos circuitos de mayor complejidad a la mejora del algoritmo evolutivo y ajuste de sus parámetros.

Referencias

1. H. Garis. Evolvable hardware genetic programming of a darwin machine. In R. Albrecht, C. Reeves, and N. Steele, editors, *Artificial Neural Nets and Genetic Algorithms*, pages 441–449. Springer Vienna, 1993.
2. G. W. Greenwood and A. M. Tyrrell. *Introduction to Evolvable Hardware: A Practical Guide for Designing Self-Adaptive Systems*. Wiley, John and Sons, 2006.
3. T. Higuchi, Y. Liu, and X. Yao, editors. *Evolvable Hardware*. Genetic and Evolutionary Computation. Springer US, 2006.
4. T. Higuchi, T. Niwa, T. Tanaka, H. Iba, H. de fs, and T. Furuya. Evolving hardware with genetic learning: a first step towards building a darwin machine. In *Proceedings of the second international conference on simulation of adaptive behavior*, pages 417–424. MIT Press, 1993.
5. J. Lohn, G. Hornby, and D. Linden. An evolved antenna for deployment on nasa,Äôs space technology 5 mission. In U.-M. O,ÄôReilly, T. Yu, R. Riolo, and B. Worzel, editors, *Genetic Programming Theory and Practice II*, volume 8 of *Genetic Programming*, pages 301–315. Springer US, 2005.
6. L. Sekanina. Evolutionary design space exploration for median circuits. In G. R. Raidl, S. Cagnoni, J. Branke, D. W. Corne, R. Drechsler, Y. Jin, C. R. Johnson, P. Machado, E. Marchiori, F. Rothlauf, G. D. Smith, and G. Squillero, editors, *Applications of Evolutionary Computing, EvoWorkshops2004: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoMUSART, EvoSTOC*, volume 3005 of *LNCS*, pages 240–249, Coimbra, Portugal, 5-7 Apr. 2004. Springer Verlag.
7. L. Sekanina, T. Martinek, and Z. Gajda. Extrinsic and intrinsic evolution of multifunctional combinational. In G. G. Yen, L. Wang, P. Bonissone, and S. M. Lucas, editors, *Proceedings of the 2006 IEEE Congress on Evolutionary Computation*, pages 9676–9683, Vancouver, 6-21 July 2006. IEEE Press.
8. A. Thompson. *Hardware Evolution: Automatic Design of Electronic Circuits in Reconfigurable Hardware by Artificial Evolution*. Springer US, 1998.

Un Nuevo Enfoque para el Posicionamiento de Routers Sobre Redes de Sensores Inalámbricos

Jose M. Lanza-Gutierrez[1], Juan A. Gomez-Pulido[1] y
Miguel A. Vega-Rodriguez[1]

Departamento de Tecnología de Computadores y Comunicaciones, Universidad de Extremadura, Escuela Politécnica, Campus Universitario s/n, 10003 Cáceres (España)
{jmlanza, jangomez, mavega}@unex.es

Resumen Hoy en día, las redes de sensores inalámbricos son ampliamente utilizadas. Es por ello, por lo que surgen trabajos tratando de solventar los inconvenientes que éstas presentan. En este artículo, estudiamos la optimización de redes de sensores tradicionales previamente establecidas, mediante la inclusión de routers y teniendo en cuenta tres importantes factores: el consumo energético, la cobertura ofrecida y la robustez de la red. Este problema de optimización np-completo ha sido resuelto mediante dos algoritmos genéticos multiobjetivo (NSGA-II y SPEA2). La calidad de los resultados ha sido analizada mediante dos conocidos indicadores multiobjetivo, el hipervolumen y el ratio de cobertura. Los resultados obtenidos han sido analizados estadísticamente, concluyendo que el SPEA2 proporciona un mejor comportamiento medio. Además, hemos comprobado como la incorporación de estos nodos repetidores constituye una buena vía para optimizar estas redes.

Keywords: algoritmos evolutivos, cobertura, computación evolutiva, consumo energético, redes de sensores, robustez, NSGA-II, SPEA2

1. Introducción

La tecnología de las redes de sensores inalámbricos (WSN, del inglés *wireless sensor networks*) es una de las áreas emergentes mas importantes hoy en día dentro de las comunicaciones inalámbricas. Es por ello, por lo que podemos encontrar numerosos ejemplos de su utilización en aplicaciones como la agricultura de precisión, el control de acceso y la monitorización de procesos industriales [1].

Tradicionalmente, este tipo de redes se compone de sensores que monitorizan variables físicas como la temperatura o la humedad; y de un nodo colector encargado de recoger toda esta información. Algunos aspectos como la utilización de dispositivos económicos y sus bajos costes de despliegue, promueven la utilización de esta tecnología. Sin embargo, también presentan importantes problemas que afectan a aspectos tales como la calidad de servicio o los costes energéticos.

Usualmente, los sensores son alimentados mediante baterías. En consecuencia, las WSNs son especialmente sensibles al desgaste energético sufrido por estos dispositivos. Cada vez que un sensor realiza una captura, debe enviar esta información hacia el nodo colector, lo que conlleva un desgaste. A su vez, es habitual

que haya varios saltos entre ambos dispositivos, lo que implica que los nodos intermedios tengan que retransmitir esta información. Estos costes energéticos adicionales provocan la existencia de cuellos de botella, lo que afecta negativamente al servicio proporcionado por la red. Recientemente y con el propósito de aliviar la carga de trabajo de los sensores, fue incorporado un nuevo tipo de dispositivo especializado en tareas de retransmisión: los routers [1].

Como podemos imaginar, el diseño eficiente de una WSN tradicional depende de multitud de factores, más aun si consideramos la inclusión de routers. Es por este motivo, por el que el diseño eficiente de WSNs ha sido definido como un problema de optimización np-completo a lo largo de la literatura, por lo que podemos encontrar un gran número de trabajos centrados en esta temática. Por ejemplo, optimizando WSNs tradicionales podemos encontrar utilizando heurísticas a Liu et al. [2] (optimizan el consumo energético, la cobertura y la conectividad, dividiendo la red en conjuntos disjuntos de sensores, decidiendo cuáles deben estar activos en cada momento); y utilizando computación evolutiva [3] (EC, del inglés *evolutionary computation*) a Konstantinidis et al. [4] (optimizan la cobertura y el tiempo de vida de la red asignando diversos niveles de potencia a los sensores), y a Martins et al. [5] (maximizan el tiempo de vida de la red asignando también diversas potencias a los sensores). Habitualmente, las WSNs tradicionales requieren de la utilización un gran número de sensores redundantes para incrementar sustancialmente el tiempo de vida éstas. La inclusión de routers permite aliviar este inconveniente. De esta forma, podemos encontrar autores que estudian esta segunda vía, usando heurísticas a Han et al. [6] (maximizando la tolerancia a fallos) y a Xu et al. [7] (estudiando métodos de posicionamiento aleatorio de routers para optimizar el tiempo de vida de la red), y usando EC a Pérez et al. [8] (proponiendo un algoritmo para minimizar el coste energético y el número de routers utilizados).

En este artículo, optimizamos WSNs previamente establecidas mediante la inclusión de routers, teniendo en cuenta tres importantes factores nunca antes considerados conjuntamente: el consumo energético, la cobertura y la robustez de la red. Puesto que es un problema np-completo, hemos utilizado técnicas computacionales no convencionales, específicamente dos algoritmos genéticos multiobjetivos (NSGA-II [9] y SPEA2 [10]) ampliamente utilizados en la EC).

El resto de este trabajo se estructura de la siguiente manera: en la sección 2 definimos el problema del posicionamiento eficiente de routers en WSNs. A continuación, detallamos la metodología utilizada al resolver este problema. En la sección 4, presentamos la evaluación de los resultados obtenidos. Finalmente, en la sección 5 se encuentran las conclusiones y el trabajo futuro.

2. Definición del problema: posicionamiento eficiente de routers sobre redes de sensores inalámbricos

Una WSN, como la considerada en este artículo, está compuesta por una serie de dispositivos inalámbricos estáticos: N routers, M sensores y un nodo colector. Todos ellos situados sobre un escenario de dimensiones $D_x \times D_y$, donde

aquellos dispositivos que se encuentren entre si a una distancia inferior a R_c podrán comunicarse directamente.

Periódica y simultáneamente, todos los sensores realizan una captura acerca de su entorno en un radio de sensibilidad R_s, la cual es enviada inmediatamente al nodo colector. El protocolo de enrutamiento utilizado es el proporcionado por el algoritmo de Dijkstra para el camino mínimo entre dispositivos. Suponemos además una perfecta sincronización entre los dispositivos.

Como ya hemos comentado, es habitual que los sensores estén alimentados por baterías, lo que implica que sus tiempos de vida se encuentren sujetos al desgaste energético al que se vean sometidos. Cuando un sensor envía una captura al nodo colector, es usual que este camino implique a diversos dispositivos. Si este paquete llega a un sensor, éste tendrá que reenviarlo de nuevo hacia el nodo colector siguiendo la ruta marcada, lo que implica un coste energético adicional. Con el propósito de aliviar este desgaste, se incorporan unos dispositivos denominados routers, caracterizamos por poseer unas mayores capacidades energéticas y estar especializados en tareas de comunicación. Estos dispositivos se encargan únicamente de retransmitir todos los paquetes recibidos hacia el nodo colector, y al igual que éste, consta de un sistema de alimentación ininterrumpida.

En este artículo, utilizamos un conocido modelo energético [11] para simular el consumo energético sufrido por los sensores. De este modo, la potencia de transmisión que un sensor i necesita para alcanzar otro dispositivo j (router, sensor o nodo colector) es definido como

$$\begin{gathered} P_i = \beta \cdot d_{i,j}^{\alpha} \mid d_{i,j} < R_c \\ para\ i = 1, 2, \ldots, M\ y\ j = 1, 2, \ldots, M + N + 1, \end{gathered} \tag{1}$$

donde $d_{i,j}$ es la distancia entre i y j, α es el exponente de pérdida de paquetes y β es el parámetro de calidad de la transmisión. Inicialmente, todos los sensores comienzan con una misma carga en sus baterías (EIC), la cual decrecerá a medida que los sensores envíen paquetes. De esta forma, la energía residual de un sensor i en tiempo t es dada por

$$\begin{gathered} E_i(t) = E_i(t-1) - [(r_i(t) + 1) \cdot P_i \cdot amp \cdot K] \\ for\ i = 1, 2, \ldots, M\ and\ t > 0, \end{gathered} \tag{2}$$

donde amp es el consumo energético por bit del amplificador de emisión, K es el tamaño del paquete de información, $r_i(t)$ es el número de paquetes que el sensor i debe enviar en tiempo t debido a las retransmisiones, y el término $+1$ se debe a que en el instante t, el sensor i realiza una captura que también debe transmitir. Al igual que en otros trabajos, el consumo energético depende únicamente de las tareas de envío, considerándose despreciable el consumo debido a las tareas de recepción, procesamiento y captación.

En trabajos anteriores, se estudió como optimizar una red de este tipo atendiendo a dos importantes criterios: el consumo energético y la cobertura ofrecida

[12]. En este artículo, proponemos optimizar un tercer factor, la robustez, lo que proporciona un mayor realismo a esta definición del problema.

Con anterioridad a definir estos factores, es necesario comprender el concepto de tiempo de vida de la red (LF, del inglés *lifetime*). Éste es el número de unidades de tiempo durante el cual la red es capaz de proporcionar información útil acerca de su entorno. Habitualmente, un umbral de cobertura (CV, del inglés *coverage threshold*) es utilizado para tal fin, de modo que la cobertura ofrecida por los sensores debe ser superior a este valor.

De este modo, el problema del posicionamiento eficiente de routers en redes de sensores inalámbricos (RNPP, del inglés *relay node placement problem*), puede definirse como un problema de optimización multiobjetivo np-completo donde, dadas las posiciones de M sensores y un nodo colector, el número de routers que se desean posicionar y los parámetros $D_x \times D_y$, R_c, R_s, IEC, K, CV, Err (el error del canal), α, β y amp; el objetivo es optimizar los siguientes factores:

Consumo energético medio (AEC, del inglés *average energy consumption*). Factor a minimizar que proporciona el consumo energético medio de los sensores a lo largo del tiempo de vida de la red. Este valor viene dado por (3), donde sa(t) es el número de sensores que poseen carga en sus baterías en tiempo t.

$$f_1 = \frac{\sum_{t=1}^{LF} \sum_{i=1}^{M} \left(\frac{E_i(t-1) - E_i(t)}{sa(t)} \right)}{LF} \tag{3}$$

Cobertura media (AC, del inglés *average coverage*). Factor a maximizar que proporciona la cobertura media ofrecida por los sensores, a lo largo del tiempo de vida de la red. En la literatura, podemos encontrar principalmente dos metodologías a la hora de calcular este valor. Por un lado, algunos autores afirman que la cobertura ofrecida por un sensor es una circunferencia de radio R_s, por lo que la cobertura global, se calcula como la unión de todas las áreas. Por otro lado, otros autores utilizan una matriz binaria de puntos de demanda ubicada sobre la superficie, de modo que todos los puntos que se encuentren a una distancia inferior a R_s de un sensor activo, tomarán valor 1; finalmente los puntos activos son contabilizados. En este artículo, hemos optado por utilizar esta última metodología, puesto que aunque es sensiblemente menos precisa, es también menos compleja de computar [13]. De este modo, la cobertura media viene dada por (4), donde R es una matriz que contiene un total de $(D_x+1)\cdot(D_y+1)$ puntos de demanda distribuidos en forma de rejilla sobre la superficie, teniendo cada una de las celdas una dimensión constante de una unidad de medida; y R^t_{xy} el valor del punto de demanda situado en la posición (x,y) de la matriz R en tiempo t.

$$f_2 = \frac{\sum_{t=1}^{LF} \sum_{x=0}^{D_x} \sum_{y=0}^{D_y} \left(\frac{R^t_{xy}}{D_x + 1 \cdot D_y + 1} \right)}{LF} \tag{4}$$

Robustez de la red (NR, del inglés *network reliability*). Es la tolerancia a fallos de la red. Este factor a maximizar representa la probabilidad de que un sensor

envíe su captura hacia el nodo colector de forma exitosa [14]. La robustez de un sensor es dada por (5), donde P es el número de caminos disjuntos (obtenidos mediante el algoritmo de Suurballe [14]) entre el nodo colector y el sensor i, h_l es el número de saltos en el l-ésimo camino disjunto, y Err es el error del canal. De esta forma, la robustez media de la red es definida como (6).

$$Re(i) = 1 - \prod_{l=1}^{P}(1 - (1 - Err)^{h_l}) \quad for\ i = 1, 2, \ldots, M \tag{5}$$

$$f_3 = \sum_{i=1}^{M}\left(\frac{Re(i)}{M}\right) \tag{6}$$

3. Algoritmos utilizados

La resolución de problemas np-completos mediante el uso de técnicas convencionales es una tarea realmente compleja. Es por este motivo, por el que en la literatura podemos encontrar diversas metodologías, siendo la computación evolutiva la más ampliamente utilizada, puesto que proporciona soluciones aproximadas de calidad en tiempos de cómputo reducidos.

Dentro de la EC encontramos un importante conjunto de técnicas: los algoritmos evolutivos (EAs, del inglés *evolutionary algorithms*). Estos son algoritmos basados en población que utilizan mecanismos procedentes de la teoría de la evolución de Darwin, como la mutación y la reproducción. En este artículo hemos utilizado dos conocidos algoritmos (NSGA-II y SPEA2) pertenecientes a un subtipo de los EAs: los algoritmos genéticos (GAs, del inglés *genetic algorithms*), caracterizados por codificar sus individuos en forma de cromosomas.

La codificación utilizada en este trabajo ha sido la misma para ambos algoritmos, de este modo un cromosoma es un vector de N routers, cada uno de ellos con sus correspondientes coordenadas x e y. Como ya se ha comentado, ambos conocidos algoritmos han sido utilizados en un trabajo previo, por lo que los detalles de implementación aparecen reflejados en [12].

4. Discusión de los resultados obtenidos

El propósito de esta sección es comparar los resultados obtenidos mediante las metaheurísticas previamente citadas, utilizando para ello un conjunto de casos de prueba. Puesto que no encontramos ninguno que se ajustara a esta definición del problema, optamos por generarlo poniéndolo a disposición de la comunidad científica en [15]. Estas instancias están compuestas por tres escenarios de distinto tamaño, donde un conjunto de sensores y un nodo colector son posicionados. Cada una de estas instancias representa una WSN tradicional, en la que se han insertado el número mínimo de sensores para cubrir toda su superficie, y que se desea optimizar mediante la inclusión de un número determinado

Tabla 1. Instancias utilizadas en este articulo. Incorporan los puntos de referencia utilizados para el cálculo del hipervolumen.

Instancia	$D_x \times D_y$	M	*HO-AEC*	*HO-AC*	*HO-NR*	*Ref AEC*		*Ref AC*		*Ref NR*	
						ideal	*nadir*	*ideal*	*nadir*	*ideal*	*nadir*
100x100_15_30	100x100	15	0,1091	89,24 %	95,67 %	0,02	0,1	100 %	60 %	100 %	50 %
200x200_15_30	200x200	57	0,2791	87,10 %	93,23 %	0.10	0.30	100 %	60 %	100 %	50 %
300x300_15_30	300x300	128	0,4225	76,44 %	85,28 %	0.04	0.50	100 %	60 %	100 %	50 %

Tabla 2. Desviación estandar e hipervolumen mediano para cada combinación de instancia, número de routers utilizados, criterio de parada y algoritmo.

NSGA-II ($\overline{Hyp}$% σ)					
Instancia(*routers*)	Número de evaluaciones				
	50 000	100 000	200 000	300 000	400 000
100x100_15_30(2)	41,01 % 0,0030	41,25 % 0,0024	41,47 % 0,0002	41,48 % 0,0001	41,48 % 0,0000
100x100_15_30(3)	53,54 % 0,0050	54,15 % 0,0018	54,46 % 0,0019	54,56 % 0,0011	54,63 % 0,0005
200x200_15_30(2)	32,49 % 0,0100	33,22 % 0,0042	33,53 % 0,0025	33,64 % 0,0018	33,74 % 0,0021
200x200_15_30(4)	41,46 % 0,0180	43,21 % 0,0167	45,07 % 0,0109	45,57 % 0,0134	45,96 % 0,0116
200x200_15_30(6)	48,75 % 0,0345	53,12 % 0,0193	55,65 % 0,0161	57,00 % 0,0168	57,68 % 0,0156
200x200_15_30(9)	57,14 % 0,0254	61,82 % 0,0223	65,57 % 0,0211	67,45 % 0,0194	68,31 % 0,0174
300x300_15_30(6)	28,35 % 0,0074	29,44 % 0,0068	30,42 % 0,0061	30,81 % 0,0060	31,05 % 0,0057
300x300_15_30(12)	29,84 % 0,0068	31,53 % 0,0100	32,86 % 0,0098	33,81 % 0,0107	34,37 % 0,0112
300x300_15_30(18)	31,26 % 0,0061	32,92 % 0,0088	34,30 % 0,0107	34,99 % 0,0097	35,41 % 0,0099
300x300_15_30(24)	33,40 % 0,0060	34,99 % 0,0137	36,51 % 0,0157	37,22 % 0,0133	37,86 % 0,0132
SPEA2 ($\overline{Hyp}$% σ)					
Instancia(*routers*)	Número de evaluaciones				
	50 000	100 000	200 000	300 000	400 000
100x100_15_30(2)	41,07 % 0,0021	41,24 % 0,0016	41,31 % 0,0015	41,46 % 0,0002	41,46 % 0,0002
100x100_15_30(3)	53,76 % 0,0038	54,27 % 0,0029	54,56 % 0,0011	54,61 % 0,0007	54,64 % 0,0007
200x200_15_30(2)	32,56 % 0,0054	32,88 % 0,0053	33,21 % 0,0032	33,38 % 0,0031	33,47 % 0,0026
200x200_15_30(4)	42,41 % 0,0150	44,03 % 0,0148	45,03 % 0,0153	45,54 % 0,0141	45,72 % 0,0130
200x200_15_30(6)	53,35 % 0,0180	55,98 % 0,0179	57,53 % 0,0072	58,57 % 0,0124	59,09 % 0,0084
200x200_15_30(9)	61,49 % 0,0179	65,42 % 0,0200	67,85 % 0,0184	68,99 % 0,0165	69,70 % 0,0132
300x300_15_30(6)	29,45 % 0,0062	30,55 % 0,0071	31,19 % 0,0072	31,54 % 0,0068	31,78 % 0,0055
300x300_15_30(12)	31,58 % 0,0071	33,19 % 0,0106	34,62 % 0,0116	35,41 % 0,0113	36,00 % 0,0115
300x300_15_30(18)	33,44 % 0,0089	35,22 % 0,0086	36,73 % 0,0092	37,68 % 0,0080	38,34 % 0,0093
300x300_15_30(24)	35,43 % 0,0077	37,04 % 0,0094	38,63 % 0,0076	39,45 % 0,0082	40,20 % 0,0093

de routers. En la tabla 1, se encuentran las principales características de estas instancias, donde el nombre de cada una de ellas sigue el patrón $D_x \times D_y_R_s_R_c$.

Tal y como indicamos en la sección 2, el RNPP viene definido por una serie de parámetros. Además de los mostrados en la Tabla 1 ($D_x \times D_y$ y M), hemos utilizado los siguientes: $R_c = 30$ y $R_s = 15$ tomados de [5], $K = 128KB$, $CV = 70\,\%$, $Err = 10\,\%$ y los parámetros energéticos $IEC = 5J$, $\alpha = 2$, $\beta = 1$ y $amp = 100pJ/bit/m^2$ que han sido tomados de [4].

Estas instancias serán optimizadas mediante la inclusión de routers, pero teniendo en cuenta que la incorporación de estos nuevos dispositivos repetidores incrementará el coste de la red. Por este motivo, hemos optado por no incluir más de un 20 % de estos elementos. Tal y como se observa en la Tabla 2, hemos definido un total de 10 casos de prueba, donde el número de routers incorporados a cada instancia es mostrado entre paréntesis.

Tabla 3. P-valores procedentes del test de Wilcoxon−Mann−Whitney al comparar los hipervolúmenes obtenidos. Los p-valores mayores de $0,05$ aparecen sombreados.

	SPEA2 vs NSGA-II				
$WSN(routers)$	50 000	100 000	200 000	300 000	400 000
100x100_15_30(2)	0,2505	0,9060	1,0000	1,0000	1,0000
100x100_15_30(3)	0,0486	0,0182	0,0188	0,0370	0,1032
200x200_15_30(2)	0,3920	0,9938	0,9999	0,9995	0,9999
200x200_15_30(4)	0,0273	0,0410	0,5530	0,6136	0,7949
200x200_15_30(6)	0,0000	0,0000	0,0000	0,0001	0,0001
200x200_15_30(9)	0,0000	0,0000	0,0000	0,0008	0,0005
300x300_15_30(6)	0,0000	0,0000	0,0000	0,0000	0,0000
300x300_15_30(12)	0,0000	0,0000	0,0000	0,0000	0,0000
300x300_15_30(18)	0,0000	0,0000	0,0000	0,0000	0,0000
300x300_15_30(24)	0,0000	0,0000	0,0000	0,0000	0,0000

Tabla 4. Para cada instancia, número de routers y criterio de parada, se muestra el algoritmo que proporciona un comportamiento significativamente superior, si alguno lo proporciona.

Mejor algoritmo: indicador hipervolumen				
50 000	100 000	200 000	300 000	400 000
—	—	NSGA-II	NSGA-II	NSGA-II
SPEA2	SPEA2	SPEA2	SPEA2	—
—	NSGA-II	NSGA-II	NSGA-II	NSGA-II
SPEA2	SPEA2	—	—	—
SPEA2	SPEA2	SPEA2	SPEA2	SPEA2
SPEA2	SPEA2	SPEA2	SPEA2	SPEA2
SPEA2	SPEA2	SPEA2	SPEA2	SPEA2
SPEA2	SPEA2	SPEA2	SPEA2	SPEA2
SPEA2	SPEA2	SPEA2	SPEA2	SPEA2
SPEA2	SPEA2	SPEA2	SPEA2	SPEA2

Ambos algoritmos han sido configurados adecuadamente antes de comenzar la experimentación. Para ello y partiendo de un tamaño de población habitual de 100 individuos, cada uno de los parámetros es ajustado mediante un barrido paramétrico, siguiendo una metodología ampliamente utilizada y descrita en [12]. De este modo, el SPEA2 adquiere unas probabilidades de mutación y cruce del 70 % y el 60 % respectivamente; y el NSGA-II unos valores del 80 % para ambos parámetros. Estos valores de mutación pueden parecer un poco elevados, sin embargo la estrategia avariciosa utilizada provoca que estos valores sean más elevados de lo habitual, tal y como puede comprobarse en [12].

En la Tabla 2, se encuentran los hipervolúmenes promedios ($\overline{Hyp}$) para cada instancia, número de routers utilizados, criterio de parada y algoritmo, así como su desviación estándar (σ). Estos valores han sido obtenidos utilizando un total de 31 ejecuciones independientes para cada una de estas combinaciones, con el propósito de obtener validez estadística (al menos es necesario utilizar un tamaño muestral de 30 unidades). Los valores de referencia utilizados para calcular estos hipervolúmenes son los mostrados en la Tabla 1 (*Ref AEC*, *Ref AC* y *Ref NR*), siendo todos ellos obtenidos de forma experimental.

El primer paso a la hora de analizar estos resultados es estudiar el tipo de valor promedio que debe utilizarse: media o mediana. Para ello hemos utilizado los tests de Shapiro−Wilk y Kolmogórov−Smirnov−Lilliefors con las siguientes hipótesis: H_0 los datos siguen una distribución normal, y H_1 no la siguen. Obteniendo un p-valor inferior a $0,05$ para todos los casos, por lo que podemos asumir que los datos no siguen una distribución normal, y por lo tanto debemos utilizar la mediana como valor promedio.

Si observamos estos hipervolúmenes, podemos comprobar como en ocasiones, alguno de estos algoritmos parece proporcionar un comportamiento superior, sin embargo no sabemos si estas diferencias son significativas. Para ello, puesto que los datos no siguen una distribución normal y las muestras son independientes, hemos utilizado el test de Wilcoxon−Mann−Whitney con las siguientes hipótesis: H_0 el SPEA2 proporciona peores resultados que el NSGA-II, y H_1 no los

proporciona. Los p-valores obtenidos se encuentran en la Tabla 3, donde aquellos valores mayores de $0,05$ aparecen resaltados.

Si observamos detenidamente estos resultados, podemos comprobar como el SPEA2 proporciona un mejor comportamiento en instancias más complejas, mientras que el NSGA-II proporciona un mejor comportamiento en las más sencillas. Además, podemos observar como en algunas ocasiones no hay diferencias entre ellos, como ocurre en las instancias de complejidad media. Estas conclusiones aparecen reflejadas en la tabla 4.

Tabla 5. Relación de cobertura entre ambos algoritmos para cada combinación de instancia, número de routers y criterio de parada.

	C (NSGA-II,SPEA2)					**C (SPEA2,NSGA-II)**				
Instancia(routers)	50 000	100 000	200 000	300 000	400 000	50 000	100 000	200 000	300 000	400 000
100x100_15_30(2)	45,60 %	23,71 %	66,56 %	84,40 %	96,03 %	69,34 %	76,59 %	63,49 %	87,61 %	82,04 %
100x100_15_30(3)	43,81 %	25,91 %	36,56 %	46,87 %	46,59 %	20,08 %	32,42 %	25,95 %	29,76 %	57,29 %
200x200_15_30(2)	40,00 %	40,44 %	57,37 %	44,77 %	63,60 %	43,48 %	62,09 %	35,09 %	34,43 %	39,16 %
200x200_15_30(4)	25,44 %	68,93 %	9,92 %	68,79 %	42,18 %	44,44 %	7,24 %	78,57 %	29,51 %	58,66 %
200x200_15_30(6)	0,00 %	4,49 %	0,00 %	38,19 %	42,52 %	96,61 %	87,59 %	87,09 %	55,98 %	61,11 %
200x200_15_30(9)	0,00 %	19,15 %	1,11 %	9,21 %	0,00 %	93,75 %	66,67 %	86,11 %	41,18 %	99,25 %
300x300_15_30(6)	7,04 %	54,78 %	8,29 %	3,14 %	15,09 %	66,10 %	20,00 %	87,70 %	80,68 %	50,98 %
300x300_15_30(12)	0,00 %	3,80 %	46,61 %	1,08 %	7,97 %	95,24 %	84,78 %	22,03 %	95,76 %	60,00 %
300x300_15_30(18)	10,00 %	3,77 %	6,56 %	6,49 %	16,67 %	68,09 %	87,27 %	68,89 %	68,33 %	86,30 %
300x300_15_30(24)	21,95 %	22,00 %	5,26 %	23,81 %	16,67 %	65,00 %	57,89 %	79,55 %	46,30 %	90,32 %
Partial average	**19.38 %**	**26.70 %**	**23.83 %**	**32.68 %**	**34.73 %**	**66.21 %**	**58.25 %**	**63.45 %**	**56.95 %**	**68.51 %**
Average			**27.46 %**					**62.68 %**		

Además del hipervolumen, hemos utilizado otra conocida medida multiobjetivo: la relación de cobertura. Hemos estudiado la relación de dominancia entre los frentes medianos de la distribución de 31 muestras para cada algoritmo, caso de prueba y criterio de parada. En la tabla 5, podemos comprobar como el SPEA2 proporciona una dominancia media del 62,88 %, mientras que el NSGA-II proporciona un 27,46 %. Si la observamos detenidamente, podemos llegar a las mismas conclusiones que con el hipervolumen, es decir el SPEA2 proporciona un mejor comportamiento es instancias complejas, mientras que el NSGA-II es mejor es las más sencillas, siendo el SPEA2 mejor en promedio.

Hasta este punto, hemos analizado la calidad de los resultados desde el punto de vista multiobjetivo. Ahora, estudiaremos la influencia causada en las funciones objetivo por la inclusión de estos routers. Para ello, hemos tomado los valores extremos de los frentes medianos previamente obtenidos, comparando estos valores con los obtenidos para estas mismas instancias sin la incorporación de routers. Estos valores de referencia se encuentran en la Tabla 1, y son los llamados $HO - AEC$, $HO - AC$ y $HO - NR$. De esta forma, en la tabla 6, encontramos los valores extremos obtenidos para cada instancia, al utilizar el algoritmo que proporciona el mejor comportamiento según los análisis previos; así como un indicador de calidad que muestra el porcentaje con respecto a los valores de referencia en el que se ha incrementado (valor positivo) o decrementado (valor negativo) su valor. En esta tabla, podemos comprobar como efectivamente, la

inclusión de routers en WSNs estáticas como las estudiadas en este trabajo, es una buena alternativa para optimizarlas. Por ejemplo, se puede observar como el AEC se ha visto reducido hasta un 59,49 %, así como el AC y el NR que se han visto incrementados hasta un 15,22 % y un 8,79 % respectivamente.

Tabla 6. Valores extremos de los frentes medianos obtenidos, para un criterio de parada de 400 000 evaluaciones. Valores procedentes del NSGA-II tienen un sombreado más oscuro que los del SPEA2.

	AEC		*AC*		*NR*	
Instancias(*routers*)	mejor	peor	mejor	peor	mejor	peor
100x100_15_30(2)	0,0570 -47,80 %	0,0692 -36,60 %	91,56 % 2,60 %	87,64 % -1,79 %	98,59 % 3,05 %	96,42 % 0,78 %
100x100_15_30(3)	0,0442 -59,49 %	0,0596 -45,37 %	91,55 % 2,59 %	84,51 % -5,30 %	99,15 % 3,64 %	98,11 % 2,55 %
200x200_15_30(2)	0,1980 -29,05 %	0,2377 -14,85 %	88,77 % 1,92 %	85,46 % -1,88 %	94,90 % 1,79 %	93,38 % 0,16 %
200x200_15_30(4)	0,1717 -38,48 %	0,2276 -18,45 %	89,12 % 2,32 %	85,86 % -1,43 %	95,52 % 2,46 %	94,13 % 0,97 %
200x200_15_30(6)	0,1448 -48,12 %	0,2015 -27,81 %	90,15 % 3,50 %	86,43 % -0,76 %	96,12 % 3,09 %	94,56 % 1,43 %
200x200_15_30(9)	0,1192 -57,28 %	0,2222 -20,39 %	89,46 % 2,71 %	83,60 % -4,02 %	96,43 % 3,43 %	94,88 % 1,77 %
300x300_15_30(9)	0,2522 -40,31 %	0,3451 -18,32 %	87,71 % 14,75 %	78,01 % 2,05 %	91,02 % 6,73 %	85,91 % 0,74 %
300x300_15_30(12)	0,2415 -42,84 %	0,3437 -18,66 %	88,08 % 15,22 %	78,96 % 3,30 %	91,62 % 7,44 %	89,54 % 5,00 %
300x300_15_30(18)	0,2348 -44,43 %	0,3580 -15,27 %	87,83 % 14,90 %	75,97 % -0,62 %	92,49 % 8,45 %	86,02 % 0,87 %
300x300_15_30(24)	0,2161 -48,85 %	0,3527 -16,52 %	87,95 % 15,06 %	75,91 % -0,70 %	92,77 % 8,79 %	86,29 % 1,18 %

5. Observaciones finales

En este artículo, hemos resuelto el posicionamiento eficiente de routers en redes de sensores inalámbricos previamente establecidas, mediante la incorporación de routers. Considerando conjuntamente tres importantes factores: el consumo energético, la cobertura proporcionada y la robustez de la red.

Hemos puesto a disposición de la comunidad un conjunto de casos de prueba propio. Este aspecto facilita que este trabajo pueda ser replicado y continuado. Con el mismo propósito, hemos utilizado dos conocidos indicadores multiobjetivo: el hipervolumen y la relación de cobertura.

Los resultados obtenidos fueron analizados estadísticamente. Concluyendo que el SPEA2 proporciona un mejor comportamiento que el NSGA-II en instancias complejas, mientras que el NSGA-II es mejor en las más sencillas. Además, hemos estudiado la influencia causada en las funciones objetivo debido al uso de routers, concluyendo que ésta es una buena vía para optimizar las WSNs.

Como líneas de trabajo futuras, sería interesante plasmar estas conclusiones en redes reales, con el fin de validar nuestras hipótesis. Además, se podría incorporar a este estudio otras metaheurísticas. Nuestro objetivo es el de encontrar un algoritmo que proporcione buenos resultados en términos generales, y que por tanto pueda ser utilizado ampliamente en el despliegue de WSNs eficientes.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el Ministerio de Economía y Competitividad y el FEDER (Fondo Europeo de Desarrollo Regional), bajo el

proyecto TIN2012-30685 (proyecto BIO, y por el Gobierno de Extremadura, a través de la ayuda GR10025 al grupo TIC015.

Referencias

1. Mukherjee, J.Y.B., Ghosal, D.: Wireless sensor network survey. Computer Networks **52** (2008) 2292 – 2330
2. Liu, L., Hu, B., Li, L.: Energy conservation algorithms for maintaining coverage and connectivity in wireless sensor networks. IET Communications **4** (2010) 786–800
3. Zitzler, E.: Evolutionary algorithms for multiobjective optimization: Methods and applications (1999)
4. Konstantinidis, A., Yang, K.: Multi-objective energy-efficient dense deployment in wireless sensor networks using a hybrid problem-specific moea/d. Applied Soft Computing **11** (2011) 4117–4134
5. Martins, F., Carrano, E., Wanner, E., Takahashi, R., Mateus, G.: A hybrid multiobjective evolutionary approach for improving the performance of wireless sensor networks. IEEE Sensors Journal **11** (2011) 545–554
6. Han, X., Cao, X., Lloyd, E.L., Shen, C.C.: Fault-tolerant relay node placement in heterogeneous wireless sensor networks. IEEE Transactions on Mobile Computing **9** (2010) 643–656
7. Xu, K., Hassanein, H., Takahara, G., Wang, Q.: Relay node deployment strategies in heterogeneous wireless sensor networks. IEEE Transactions on Mobile Computing **9** (2010) 145–159
8. Perez, A., Labrador, M., Wightman, P.: A multiobjective approach to the relay placement problem in wsns. In: Proceedings of IEEE WCNC. Volume 1. (2011) 475–480
9. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast elitist multi-objective genetic algorithm: Nsga-ii. IEEE Transactions on Evolutionary Computation **6** (2000) 182–197
10. Zitzler, E., Laumanns, M., Thiele, L.: Spea2: Improving the strength pareto evolutionary algorithm. Technical report, Computer Engineering and Networks Laboratory (TIK), ETH Zurich (2001)
11. Konstantinidis, A., Yang, K., Zhang, Q.: An evolutionary algorithm to a multiobjective deployment and power assignment problem in wireless sensor networks. In: Proceedings of IEEE GLOBECOM. (2008) 1–6
12. Lanza-Gutierrez, J.M., Gomez-Pulido, J.A., Vega-Rodriguez, M.A., Sanchez-Perez, J.M.: Relay node positioning in wireless sensor networks by means of evolutionary techniques. In: Autonomous and Intelligent Systems. Volume 7326 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2012) 18–25
13. Wang, B.: Coverage problems in sensor networks: A survey. ACM Comput. Surv. **43** (2011) 32:1–32:53
14. Deb, B., Bhatnagar, S., Nath, B.: Reinform: Reliable information forwarding using multiple paths in sensor networks. In: Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks. (2003) 406–415
15. Lanza-Gutierrez, J.M., Gomez-Pulido, J.A., Vega-Rodriguez, M.A., Sanchez-Perez, J.M.: Instance sets for optimization in wireless sensor networks. http://arco.unex.es/wsnopt (2011)

Extracción de reglas de asociación frecuentes en bases de datos relacionales

J. M. Luna, J. R. Romero, C. Romero, y S. Ventura

Departamento de Informática y Análisis Numérico,
Universidad de Córdoba, Campus de Rabanales, Edificio Albert Einstein,
14071 Cordoba, España. Tel.:+34957212218. Fax:+34957218630
{jmluna,jrromero,cromero,sventura}@uco.es

Abstract. La mayoría de las propuestas existentes en el campo de la minería de asociación tienen como finalidad el descubrimiento de reglas de asociación en datos almacenados en una única tabla. Sin embargo, con el creciente interés en el almacenamiento de información, cada vez es más usual el almacenamiento de información en bases de datos relacionales. Las propuestas existentes en este campo requieren un alto coste computacional y de memoria debido al incremento del número de instancias y atributos. Además, debido a su distribución en diferentes tablas, las reglas de asociación descubiertas por algunos de los algoritmos pueden presentar valores incorrectos en las métricas de evaluación. En este trabajo se presenta un nuevo algoritmo de programación genética gramatical para la extracción de reglas de asociación en entornos relacionales. La principal característica de este algoritmo es la representación de las reglas en modo árbol, lo que permite mantener la estructura relacional de la base de datos original. Los resultados experimentales demuestran la eficiencia de este algoritmo.

Palabras clave: Minería de datos, programación genética, reglas de asociación, base de datos relacional

1 Introducción

La minería de datos trata de descubrir información no trivial que, por lo general, se encuentra oculta en los datos. La tarea de minería de datos puede descomponerse en dos tareas principales, una primera predictiva, cuya finalidad es la construcción de un clasificador lo más exacto y preciso posible [1], y una segunda descriptiva, donde tiene un especial peso la técnica de minería de asociación [2].

Las técnicas de descubrimiento de reglas de asociación buscan relaciones frecuentes y de interés entre elementos de un conjunto de datos [3]. Esta técnica fue introducida en 1994 por *Agrawal* [4], el cual estableció que, dado un conjunto de elementos $\mathcal{E} = \{e_1, e_2, e_3, ..., e_n\}$, y un conjunto de instancias $\mathcal{I} = \{i_1, i_2, i_3, ..., i_n\}$ en un conjunto de datos, una regla de asociación se define como una implicación de la forma $X \Rightarrow Y$, donde $X \subset \mathcal{E}$, $Y \subset \mathcal{E}$, y $X \cap Y = \emptyset$.

En la actualidad, existe un creciente interés en el almacenamiento de información, y dicho almacenamiento en grandes bases de datos relacionales resulta

esencial en muchos ámbitos [5]. Mientras que la extracción de reglas de asociación sobre datos almacenados en una única tabla ha sido objeto de estudio a lo largo de los años, existen muy pocas propuestas que aborden la minería de asociación sobre datos relacionales. Estos tipos de datos poseen una estructura mucho más compleja y almacenan una mayor cantidad de información, por lo que los algoritmos clásicos de extracción de reglas de asociación no pueden ser aplicados de manera directa.

Muchos autores han realizado una transformación de los datos de manera que todas las tablas del sistema relacional son transformadas en una única tabla [6]. No obstante, esta transformación conlleva un incremento en el número de atributos e instancias, pues todas las tablas deben ser tenidas en cuenta. Además, diferentes instancias de una tabla pueden ser duplicadas en la tabla resultante en función del tipo de relación mantenida, por lo que las reglas de asociación descubiertas pueden no ser reales si este aspecto no se ha tenido en cuenta.

En este trabajo se presenta una versión inicial de un modelo para la extracción eficiente de reglas de asociación sobre bases de datos relacionales. Debido a que una base de datos relacional puede ser considerada como un conjunto de tablas unidas en modo de árbol [7], presentamos un modelo de programación genética gramatical donde las soluciones son codificadas en forma de árbol, preservando la estructura original de la base de datos. El modelo propuesto hace uso de una gramática de contexto libre que permite restringir la codificación de las soluciones presentadas como árboles de derivación. Cada nodo del árbol representa una tabla del modelo relacional, y cada rama del arbol es una relación existente entre dos tablas. De este modo, no es necesario realizar la unión de todas las tablas en una única tabla, y el proceso de extracción de reglas es mucho más rápido y sencillo. Además, el uso de una propuesta evolutiva permite reducir el tiempo de cómputo y la memoria requerida con respecto a propuestas de búsqueda exhaustiva.

Con el objetivo de demostrar la eficiencia de nuestra propuesta, analizamos los resultados obtenidos al aplicar el modelo sobre una base de datos relacional de más de 220.000 instancias. Los resultados demuestran que nuestra propuesta es de gran interés, obteniendo resultados prometedores en la extracción de reglas de asociación sobre datos relaciones sin ningún paso previo de transformación o preprocesado.

El presente trabajo se estructura de la siguiente manera: la Sección 2 describe el modelo propuesto y sus características principales; la Sección 3 presenta la base de datos utilizada y los resultados obtenidos; por último, en la Sección 4 se detallan las conclusiones a las que se ha llegado con este trabajo.

2 Modelo de programación genética propuesto

El modelo que se describe en este trabajo representa las reglas de asociación en forma de árboles de derivación mediante el uso de una gramática de contexto libre. La principal ventaja de este algoritmo es su aplicación sobre bases de datos relacionales sin necesidad de realizar ninguna transformación de los datos o del

esquema de datos original. Además, el uso de una gramática para representar las reglas de asociación permite restringir el espacio de búsqueda y establece una gran flexibilidad en la forma de las reglas.

2.1 Criterio de codificación

En cualquier algoritmo de programación genética gramatical, las soluciones son codificadas por medio de un genotipo y un fenotipo. El genotipo es representado como un árbol de derivación cuya estructura es generada aleatoriamente conforme a la gramática de contexto libre definida. Por el contrario, el fenotipo representa la regla de asociación representada por el genotipo.

Una gramática de contexto libre se define formalmente como la tupla (Σ_N, Σ_T, P, S), donde Σ_N representa el alfabeto de símbolos no terminales, Σ_T el alfabeto de símbolos terminales, P el conjunto de reglas de producción permitidas, y S el símbolo inicial de la gramática. Además, se establece que Σ_N y Σ_T son conjuntos disjuntos.

Cada individuo es codificado mediante un proceso de derivación. Este proceso se realiza a partir del símbolo inicial de la gramática, realizando una serie de pasos mediante la aplicación de reglas del conjunto P. Con el fin de evitar árboles demasiado profundos, el número máximo de reglas de producción utilizadas en el proceso de derivación puede ser predefinido por el usuario experto. De esta forma, se permite indicar el tamaño de regla de asociación máximo deseado.

En el modelo propuesto, la gramática de contexto libre utilizada (Figura 1) permite mantener la estructura relacional de la base de datos sobre la que se esté trabajando. Cualquier individuo generado a partir de esta gramática tiene su origen en una tabla inicial del modelo relacional y que es seleccionada de manera aleatoria. De este modo, y partiendo de dicha tabla inicial, se generan relaciones entre tablas. Nuevas tablas pueden ser consideradas en una regla específica si, y únicamente si, existe una relación real dentro del modelo relacional sobre el que se trabaja. Finalmente, hay que destacar que cualquier condición tomará un valor, tanto contínuo como discreto, dentro de la tabla que la engloba. Estos valores

$G = (\Sigma_N, \Sigma_T, P, S)$ donde:

S = ReglaAsociación
Σ_N = {ReglaAsociación, Antecedente, Consecuente }
Σ_T = {'Y', 'Condición', 'TablaInicial', 'Tabla' }
P = {ReglaAsociación = 'TablaInicial', Antecedente, Consecuente ;
Antecedente = 'Condición' | 'Y', 'Condición', Antecedente |
'Y', 'Condición', 'Tabla', Antecedente ;
Consecuente = 'Condición' | 'Y', 'Condición', Consecuente |
'Y', 'Condición', 'Tabla', Consecuente ; }

Figura 1. Gramática de contexto libre utilizada en la propuesta

estarán establecidos mediante operadores lógicos comúnmente utilizados en el lenguaje SQL, como son =, ! =, IN, NOT IN, BETWEEN y NOT BETWEEN.

Los operadores lógicos =, ! =, IN y NOT IN, están diseñados para trabajar sobre datos nominales. La principal diferencia entre el operador = y el operador IN, es que el primero trata con un único valor, mientras que el segundo puede tratar con más de un valor nominal. Por el contrario, los operadores BETWEEN y NOT BETWEEN están diseñados para trabajar sobre datos numéricos, estableciendo un rango de valores mínimo y máximo.

Para una mejor compresión del tipo de árbol de derivación generado a partir de la gramática de contexto libre utilizada, la Figura 2 muestra un ejemplo de árbol de derivación. Dicho árbol de derivación representa una regla de asociación con un antecedente y consecuente establecidos sobre una tabla inicial. Esta tabla inicial posee una condición establecida en el antecedente y otro en el consecuente. Además, el antecedente establece la inclusión de una nueva tabla, que está relacionada con la tabla inicial, y sobre la que se establecen dos condiciones, las dos condiciones más profundas dentro del árbol de derivación.

En el estudio experimental (Sección 3) se muestra en detalle el fenotipo obtenido a partir de árboles de derivación. Hay que indicar que las líneas discontínuas representan los símbolos terminales, mientras que las líneas contínuas

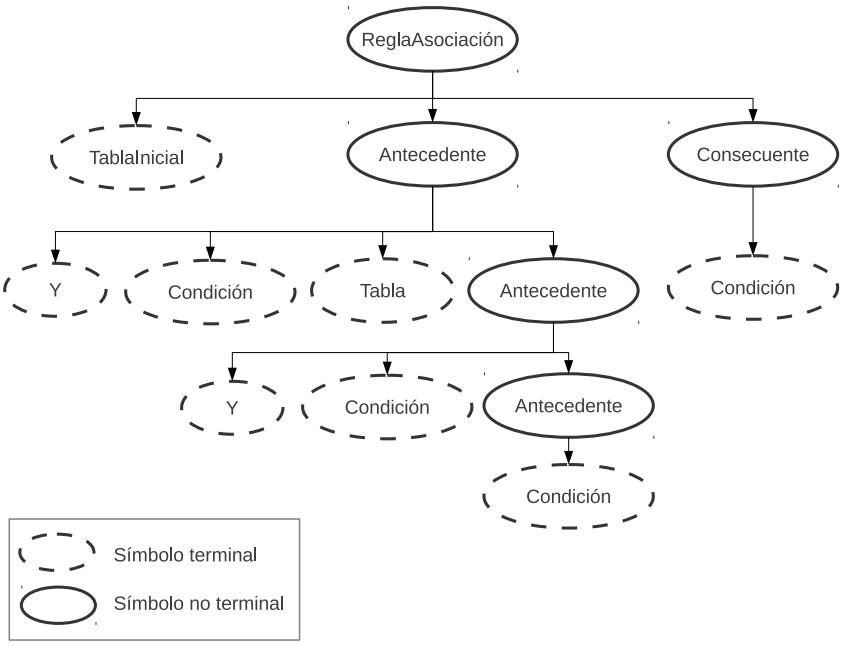


Figura 2. Ejemplo de genotipo de una solución obtenida a partir de la gramática de contexto libre definida en la Figura 1

los no terminales. Los símbolos terminales son los que darán significado a la regla, estableciendo un recorrido pre-orden del árbol de derivación para obtener el significado correcto de la regla representada.

2.2 Evaluación de los árboles de derivación

El proceso de evaluación de cada individuo y, por tanto, de cada regla de asociación generada a partir de la gramática, representa una tarea de gran relevancia para la determinación de la calidad de las reglas obtenidas. Diversos investigadores han descrito medidas objetivas para la evaluación de reglas de asociación [8, 9]. Dos de las medidas más importantes y ampliamente utilizadas en este campo son el soporte y la confianza. El soporte se define como la proporción del número de transacciones que satisfacen tanto el antecedente como el consecuente de la regla. Por el contrario, la confianza o exactitud de la regla establece la proporción del número de transacciones que incluyen el antecedente y el consecuente entre todas las transacciones que componen el antecedente. Estas dos medidas están relacionadas, pues la confianza de una regla nunca puede ser menor que el soporte de la misma.

En el modelo propuesto en este trabajo, utilizamos como función de *fitness* el soporte de la regla (ver Ecuación 1), buscando reglas de asociación lo más frecuentes posibles y, por tanto, lo más exactas posibles. Con el fin de desechar reglas no válidas, por ejemplo, aquellas que poseen un mismo atributo en el antecedente y consecuente, un valor de 0 es asignado a su función de fitness.

$$fitness = \begin{cases} 0 & \text{si } X \cap Y \neq \emptyset \\ soporte(regla) & \text{en otro caso} \end{cases} \quad (1)$$

Hay que destacar que el cálculo de estas medidas no es una tarea trivial como ocurre en la mayoría de las propuestas para la extracción de reglas de asociación. Al tratar con datos relacionales, la evaluación de las reglas tiene que hacerse mediante consultas SQL, las cuales son generadas de manera automática en base al tipo de regla representada por el árbol de derivación.

2.3 Operador genético

Como cualquier algoritmo evolutivo, el objetivo del modelo propuesto es la búsqueda de soluciones lo más óptimas posibles en base a una función de *fitness* o aptitud. Así, cada generación del proceso evolutivo tiene como objetivo la obtención de nuevas soluciones en las que dicha función de *fitness* es mejorada. Con tal fin, proponemos un operador genético que permita la obtención de soluciones con un valor de soporte más elevado que las originales. En minería de asociación, el valor de soporte de una regla depende de la frecuencia de sus condiciones, por lo que obteniendo condiciones de mayor frecuencia se aumenta la probabilidad de incrementar el soporte de la regla resultante.

El operador genético propuesto en este trabajo selecciona la condición de menor soporte de una regla y la modifica. Existen dos posibilidades de modificación, siendo aleatoria la selección de una u otra. El primer tipo de modificación

consiste en el reemplazo de la condición seleccionada, por lo que ésta es completamente cambiada. El otro tipo de modificación permitida consiste en el reemplazo del operador lógico utilizado, por lo que el atributo y valores seleccionados permanecen intactos. De este modo, si el operador lógico de dicha condición era el operador =, entonces prodía ser utilizado el operador ! = en su lugar. Es decir, el operador lógico es reemplazado por su operador opuesto.

2.4 Algoritmo propuesto

El modelo propuesto en este trabajo sigue el esquema de cualquier algoritmo evolutivo, tal y como se muestra en la Figura 3. Dicho modelo comienza con la creación aleatoria de un conjunto de individuos o población inicial a partir de la gramática de contexto libre propuesta en la Sección 2.1. Este primer proceso de generación de individuos está diseñado para que siempre se comience por un conjunto válido de individuos. De esta forma, si un individuo generado no cubre ninguna instancia de la base de datos, es decir, su soporte es 0, entonces otro nuevo individuo es generado y este proceso se repite hasta que el individuo obtenido cubra, al menos, una instancia.

Una vez que el conjunto de soluciones iniciales es obtenido, comienza un proceso evolutivo llevado a cabo a lo largo de un número predeterminado de generaciones, donde el operador genético descrito en la Sección 2.3 es aplicado en cada generación en base a una probabilidad. Este operador genético genera un nuevo conjunto de soluciones que formarán parte de la población, uniéndose

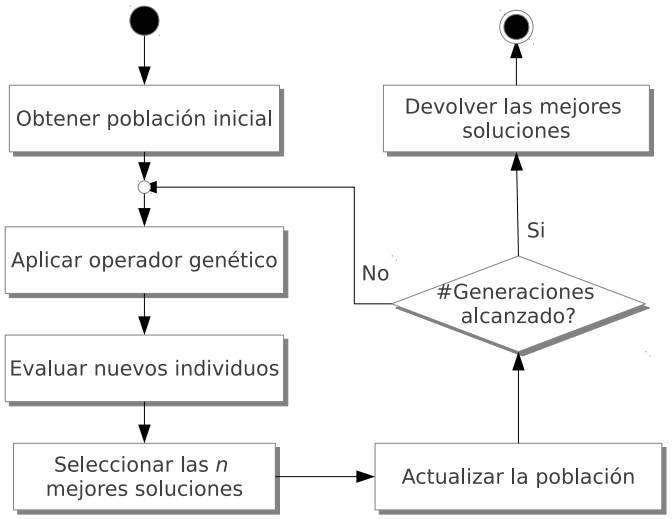


Figura 3. Diagrama de flujo del modelo propuesto para la extracción de reglas de asociación en bases de datos relacionales

a las soluciones existentes en la generación previa. A partir de este conjunto resultante, se seleccionan las n mejores soluciones distintas, es decir, aquellas que proporcionen los mejores valores en la función de *fitness* y que representan diferentes soluciones. Hay que destacar que, con el fin de evitar un crecimiento incontrolado de las soluciones, el tamaño de la población general del proceso evolutivo permanece inalterable, por lo que las peores soluciones se van desechando.

Este proceso evolutivo continúa hasta que se alcance el número máximo de generaciones establecido. En ese instante, el algoritmo evolutivo propuesto devuelve el conjunto de n soluciones, aquellas con mejor función de aptitud.

3 Estudio experimental

En este estudio experimental se demuestra la eficiencia del algoritmo propuesto, el cual ha sido implementado mediante el uso de la librería JCLEC [10] para la computación evolutiva. Esta librería es de libre disposición y se encuentra disponible en la web `http://jclec.sf.net`.

Para llevar a cabo el análisis experimental, se ha utilizado una base de datos relacional cuyas tablas y atributos se muestran en la Tabla 1. Mediante el uso de esta base de datos, se intenta demostrar que el modelo propuesto en este trabajo permite la extracción de reglas de asociación entre diferentes tipos de relaciones entre tablas, es decir, relaciones del tipo 1:1, 1:N y N:M. La base de datos utilizada incluye diferentes relaciones, diferente número de atributos y de instancias, donde el número total de instancias es de cerca de 220.000 registros.

Tabla 1. Base de datos relacional utilizada en el estudio experimental

Relación	**Atributos**	**# Instancias**
Pais	NombrePais, habitantes, capital	14
Centro Comercial	tienda, nombre, empleados, **NombrePais**	14
Cliente	pasaporte, ciudad, edad, **tienda**	20,000
Compra	identificador, fecha, precio, **pasaporte**	50,000
Objeto	codigo, precio	100
C-O (Compra-Objeto)	**codigo**, **identificador**	149,739
Prenda	**codigo**, nombre, temporada	50
Comida	**codigo**, nombre, peso	50

Los experimentos han sido realizados estableciendo, como parámetros del modelo evolutivo, un tamaño de población de 50 individuos, un número máximo de generaciones de 100, una probabilidad de 0.9 del operador genético, y un tamaño de población élite o número máximo de soluciones buscadas de 20.

Algunas de las reglas de asociación obtenidas al ejecutar el algoritmo sobre la base de datos relacional se muestran en la Tabla 2. Como se comprueba, el algoritmo propuesto es capaz de obtener reglas de asociación que incluyen una única tabla tal y como lo haría un algoritmo clásico de asociación. Por ejemplo,

Tabla 2. Ejemplo de reglas de asociación obtenidas al ejecutar el algoritmo

#	Regla	Soporte	Confianza
1	**SI** cliente.edad NOT BETWEEN 33, 34 **ENTONCES** cliente.ciudad != Salvador	0.904	0.982
2	**SI** pais.habitantes BETWEEN 4000000, 143000000 **ENTONCES** pais.capital != Brasilia	0.857	1.000
3	**SI** comida.peso NOT IN (4Kg.) Y objeto.precio BETWEEN 4.600, 47.024 Y objeto.precio NOT BETWEEN 43.879, 44.069 **ENTONCES** comida.nombre != piña	0.840	1.000

las reglas 1 y 2 sólo tienen en cuenta la tabla *cliente* y *pais*, respectivamente. De este modo, si nos centramos en la regla número uno, vemos que si la edad de un cliente no está entre 33 y 34 años, entonces el cliente no es de El Salvador, y esta regla se satisface en el 90% de los casos con una exactitud del 98%.

Además, como era de esperar, el modelo propuesto puede obtener reglas de asociación que engloban más de una tabla sin perder la estructura relacional original. Para comprobar este hecho, nos centraremos en la regla número tres, en la que se parte de una tabla inicial, como es la tabla *comida*, y se tienen en cuenta otras tablas relacionadas directamente con dicha tabla, como es el caso de la tabla *objeto*. El significado de esta regla de asociación descubierta es que el 84% de las comidas que tiene un peso diferente de 4kg y cuyo precio está entre 4.6$ y 47.024$, excluyendo los valores entre 43.879$ y 44.069$, no son piña. Esta regla se da con una exactitud máxima, es decir, siempre que se cumple el antecedente, se cumple también el consecuente.

Finalmente, en este estudio mostramos el árbol de derivación (Figura 4) obtenido para una de las reglas de asociación descubiertas por el algoritmo. El fenotipo de dicha regla de asociación corresponde a la regla número 3 de la tabla de reglas.

4 Conclusiones

En este trabajo se ha presentado un nuevo modelo de extracción de reglas de asociación en bases de datos relacionales. Para ello, el modelo propuesto utiliza una gramática de contexto libre que representa las soluciones en la forma de árboles de derivación.

La principal ventaja del modelo presentado es que, debido a la codificación de las soluciones en modo de árbol, permite mantener la estructura relacional de la base de datos original. De esta forma, no es necesaria ninguna transformación de los datos para convertirlos en una estructura de datos fácilmente aplicable por otras propuestas. Además, debido a que ninguna transformación de los datos es

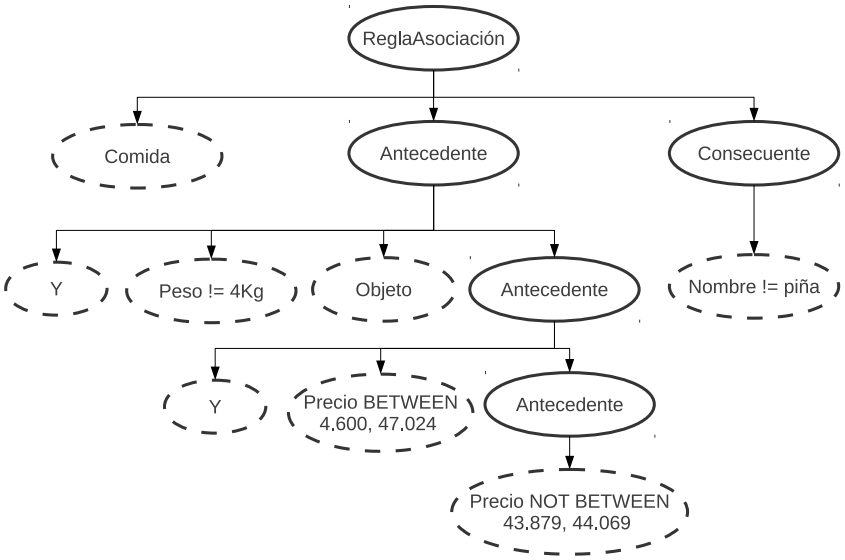


Figura 4. Árbol de derivación que representa la regla número 3 de la Tabla 2

requerida, éstos no sufren ninguna alteración en medidas de calidad de las reglas, como es el caso de la frecuencia de aparición o soporte.

Es importante reseñar que las bases de datos relacionales, por lo general, están formadas por un gran número de registros, por lo que la aplicación de un modelo de búsqueda exhaustiva no es recomendable. Así, el uso de un modelo evolutivo como el propuesto está totalmente justificado. A esto hay que sumar la enorme ventaja que ofrece el uso de una gramática de contexto libre en este campo, pues permite restringir el espacio de búsqueda y codificar las soluciones de una manera flexible y sin perder la estructura de los datos originales.

Los resultados presentados en el estudio experimental revelan que el modelo propuesto es capaz de descubrir reglas de asociación en una base de datos muy diversa, donde diferentes tipos de relaciones entre tablas han sido consideradas.

Agradecimientos

El presente trabajo ha sido financiado por la Junta de Andalucía y el Ministerio de Ciencia e Innovación bajo los proyectos P08-TIC-3720 y TIN-2011-22408, y los fondos FEDER.

Referencias

1. Olmo, J.L., Romero, J.R., Ventura, S.: Using ant programming guided by grammar for building rule-based classifiers. IEEE Transactions on Systems, Man, and

Cybernetics, Part B: Cybernetics, **41**(6) (2011) 1585–1599
2. Luna, J.M., Romero, J.R., Ventura, S.: Design and behavior study of a grammar-guided genetic programming algorithm for mining association rules. Knowledge and Information Systems **32**(1) (2012) 53–76
3. Han, J., Pei, J., Yin, Y., Mao, R.: Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach. Data Mining and Knowledge Discovery **8** (2004) 53–87
4. Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules in Large Databases. In Bocca, J.B., Jarke, M., Zaniolo, C., eds.: VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases, Santiago de Chile, Chile, San Francisco: Morgan Kaufmann (September 1994) 487–499
5. Konan, A., Gndem, T., Kaya, M.: Assignment query and its implementation in moving object databases. International Journal of Information Technology and Decision Making **9**(3) (2010) 349–372
6. Ng, E., Fu, A., Wang, K.: Mining Association Rules from Stars. In: Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM 2002), Maebashi City, Japan. (December 2002)
7. Jiménez, A., Berzal, F., Cubero, J.: Using Trees to Mine Multirelational Databases. Data Mining and Knowledge Discovery (2011) 1–39
8. F. Berzal and I. Blanco and D. Sánchez and M.A. Vila: Measuring the Accuracy and Interest of Association Rules: A new Framework. Intelligent Data Analysis **6**(3) (2002) 221–235
9. Romero, C., Luna, J.M., Romero, J.R., Ventura, S.: RM-Tool: A Framework for Discovering and Evaluating Association Rules. Advances in Engineering Software **42**(8) (2011) 566 – 576
10. Ventura, S., Romero, C., Zafra, A., Delgado, J.A., Hervás, C.: JCLEC: A java framework for evolutionary computation. Soft Computing **12**(4) (2008) 381–392

Aplicación de Algoritmos Meméticos Híbridos de Doble Fase en la Optimización de Tableros de Puentes de Vigas Artesa de Hormigones No Convencionales

José Vicente Martí Albiñana[1], Víctor Yepes Piqueras[2], Tatiana García Segura[3],

Instituto de Ciencia y Tecnología del Hormigón (ICITECH), Departamento de Ingeniería de la Construcción y Proyectos de Ingeniería Civil, Escuela Técnica Superior de Ingenieros de Caminos, Canales y Puertos, Universitat Politècnica de València.
[1]jvmartia@upv.es; [2]vyepesp@upv.es; [3]tagarse@cam.upv.es.

Resumen. Este artículo trata la optimización económica de puentes de vigas artesa de hormigones no convencionales. Muestra la eficacia de los algoritmos meméticos (MA) combinando algoritmos genéticos (GA) con una búsqueda local (VNSL) en una primera fase de diversificación y el recocido simulado (SA) en una segunda fase de intensificación. Se aplica la IAP(98) y la EHE-08 para evaluar la estructura. Los cálculos se programan en lenguaje fortran. Los algoritmos se aplican a un puente de 30 m de luz y 12 m de ancho. El diseño consta de 41 variables discretas. Se utilizan hormigones distintos para viga y losa, con diferentes dosificaciones de fibras de acero en viga. Se procesan seis heurísticas, obteniéndose más de 100 soluciones optimizadas con una duración de 60 minutos por diseño. La calidad de las soluciones y los tiempos de proceso computacional permiten aplicar esta heurística de un modo práctico a casos reales.

Palabras clave: Optimización, algoritmos meméticos híbridos, vigas artesa, hormigones no convencionales, puentes

1. Introducción

La inteligencia artificial, desde su aparición hacia mediados de los años 50, se ha introducido en distintos campos tales como la programación automatizada, la resolución de problemas restringidos, la investigación operacional, la planificación, las redes neuronales, etc. En el diseño de estructuras se selecciona un conjunto de variables conforme a unas restricciones estructurales para las cuales la aplicación de la inteligencia artificial permite obtener buenos resultados. Las variables del diseño incluyen los distintos materiales, la geometría de las secciones y los refuerzos de acero. En el presente, el diseño económico de estructuras de hormigón, está muy condicionado por la experiencia de los ingenieros calculistas. El diseño tradicional de estructuras de hormigón tiene como objetivo la obtención de estructuras seguras, funcionales y durables a lo largo de su vida útil [1]. Tradicionalmente los diseños se han basado en la previa experiencia del ingeniero proyectista, que diseña y predimensiona las distintas partes y secciones de la estructura, para a continuación, mediante la utilización de programas informáticos comprobar la bonanza del diseño inicial, y ajustar

éste en los puntos críticos que presentan. Si las dimensiones o la resistencia de los materiales son insuficientes, la estructura se redefine. Tal proceso conduce a diseños seguros, pero en cambio, la economía de las estructuras de hormigón queda muy condicionada a la experiencia del diseñador estructural.

Los métodos de optimización se ocupan de minimizar una función objetivo, que generalmente es el coste o el peso de la estructura. Su aplicación a las estructuras de hormigón ha sido muy reducida comparada con las desarrolladas para las estructuras de acero [2]. Esto es debido a la mayor simplicidad en la modelización de las estructuras de acero, en las que una pieza metálica queda definida por su longitud y tipo de perfil estandarizado. Los métodos heurísticos, de reciente desarrollo van ligados a la evolución de los procedimientos de la inteligencia artificial, y pueden utilizarse allí donde el elevado número de variables en juego impiden la resolución en un tiempo de cálculo razonable de los problemas mediante la programación matemática. Este grupo incluye un amplio número de algoritmos de búsqueda, tales como algoritmos genéticos y meméticos, recocido simulado, aceptación por umbrales, redes neuronales, búsqueda tabú, colonias de hormigas, etc. Dichos métodos se han aplicado con éxito en áreas diferentes de la ingeniería estructural [3]. Se basan en algoritmos simples, pero que requieren un gran esfuerzo computacional, puesto que incluyen una gran cantidad de iteraciones en las cuales se evalúa la función objetivo y se comprueban las restricciones estructurales. Entre los primeros trabajos de la optimización heurística aplicados a las estructuras de acero, que deben ser mencionados, se encuentran los de Jenkins y los de Rajeev y Krishnamoorthy en 1992 [4.5], que aplicaron algoritmos genéticos. Las primeras aplicaciones sobre estructuras de hormigón armado (HA), incluyen en 1997 el trabajo de Coello et al [6], que utilizaron algoritmos genéticos para la optimización económica de las vigas. Por otro lado, nuestro grupo de investigación ha presentado trabajos recientes de diseño automático y optimización de estructuras de hormigón armado con algoritmos genéticos [7] y con otras técnicas heurísticas [8-10], así como trabajos de optimización con hormigón pretensado [11,12].

Las estructuras objeto de estudio son los puentes de vigas artesa prefabricadas de hormigón pretensado, que se emplean como pasos superiores sobre otras vías de comunicación. Tiene como particularidad la utilización de hormigones no convencionales en la fabricación de las vigas: hormigones reforzados con fibras de acero (HPF). Las luces de los puentes habitualmente oscilan entre los 20 y 40 m. Estos puentes se componen de vigas prefabricadas de hormigón pretensado (HP) en forma de U, con losa superior colaborante construida in situ (ver fig.1 y fig. 2). El tablero del puente objeto de optimización está compuesto por dos vigas paralelas de 30 m de luz, que integran en la parte superior una losa de HA para el tráfico de vehículos de 12 m de anchura (fig. 2). Para su optimización, se calculan los costes para cada una de las soluciones generadas, en función de las variables geométricas, los materiales y los refuerzos activo y pasivo. A continuación, se evalúan todos los estados límites relevantes, y posteriormente se aplica un algoritmo memético híbrido capaz de encontrar soluciones optimizadas en coste. El artículo, tras realizar el planteamiento del problema de optimización, define el algoritmo citado, presenta los resultados obtenidos y recoge las principales conclusiones.

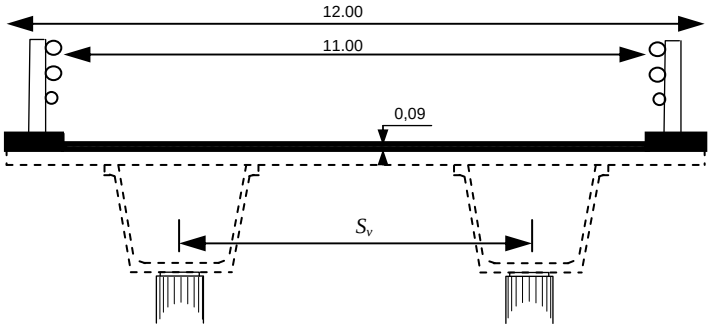


Fig. 1. Esquema sección transversal del tablero del puente

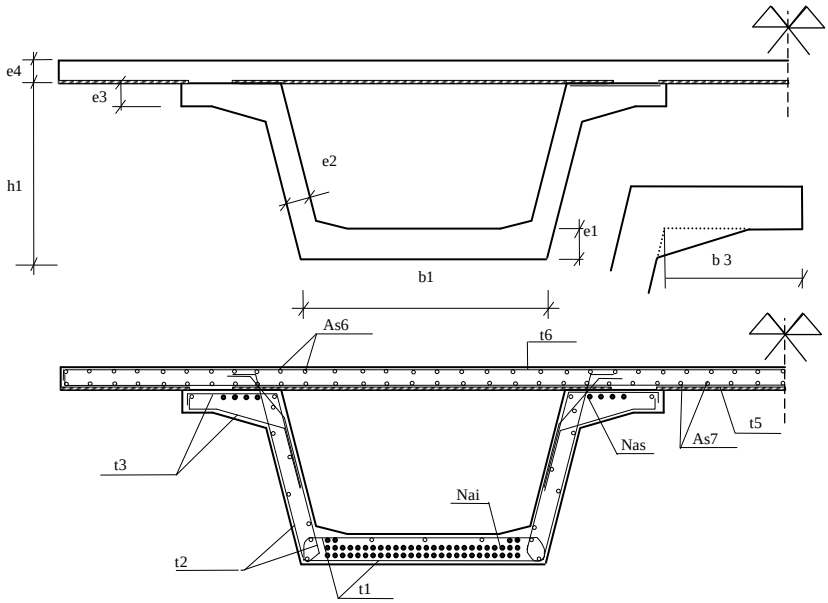


Fig. 2. Variables geométricas y armaduras activa y pasiva

2. Definición del problema y heurística aplicada

El problema propuesto reside en la optimización económica de la estructura de hormigón. Su resolución busca minimizar la función objetivo F de la expresión (1), satisfaciendo también las restricciones de las expresiones (2).

$$F(x_1, x_2, ..., x_n) = \sum_{i=1}^{r} p_i \cdot m_i(x_1, x_2, ..., x_n) \quad (1)$$

$$g_j(x_1, x_2, ...x_n) \leq 0 \quad (2)$$

$$x_i \in (d_{i1}, d_{i2}, ..., d_{iq_i}) \quad (3)$$

Los valores x_1, x_2,…, x_n son las variables de diseño del problema, que pueden tomar uno de los valores discretos indicados en la expresión (3). La función objetivo en la expresión (1) es la suma de los precios unitarios multiplicados por las mediciones de las unidades de obra (hormigón, acero, encofrado, transporte, colocación, etc.). La expresión (2) indica las restricciones geométricas y de constructibilidad, así como todos los estados últimos y de servicio que la estructura ha de cumplir. Las soluciones que satisfacen las restricciones en la expresión (2) se denominan factibles, y las que no, soluciones no factibles. Se distinguen dos procesos: la construcción de la viga pretensada en planta, y la de la losa de hormigón en la obra. Los principales precios unitarios considerados se dan en la tabla 1.

Tabla 1. Precios básicos de la función coste de las estructuras

Descripción	Precio unitario en viga (€)	Precio unitario en losa (€)
Kilogramo del acero pasivo (B-500-S)	2.81	1.50
Kilogramo del acero activo (Y-1860-S7)	3.62	NA
m de molde en viga	80.37	NA
m^2 de encofrado en losa	NA	32.00
m^3 de hormigón HA-25	NA	70.00
m^3 de hormigón HA-30	NA	75.00
m^3 de hormigón HA-35	NA	80.00
m^3 de hormigón HA-40	NA	85.00
m^3 de hormigón HP-35	130.81	NA
m3 de hormigón HP-40	142.74	NA
m3 de hormigón HP-45	152.10	NA
m3 de hormigón HP-50	163.59	NA
Kilogramo de fibras de acero	1.10	NA

El método de búsqueda aplicado en este trabajo es un algoritmo memético híbrido de doble fase. Proporciona la combinación de la naturaleza de búsqueda en paralelo realizada por los algoritmos evolutivos y la búsqueda local que mejora cada una de las soluciones que forman la población. El calificativo de meméticos lo aplicó por primera vez Moscato [13] en 1989 a estos algoritmos híbridos inspirándose en el término inglés *meme*, acuñado por Dawkins [14] para referirse al análogo del gen en el contexto de la evolución cultural. La idea de emplear la hibridación entre metaheurísticas basadas tanto en algoritmos poblacionales como en otros de búsqueda local permite mejorar la efectividad de los híbridos al combinar los esfuerzos de diversificación e intensificación [15]. Krasnogor y Smith [16] proporcionan una guía y ejemplos de aplicación para este tipo de algoritmos. Hay que indicar que la función coste puede verse penalizada para el caso en que las soluciones incumplan alguna de las restricciones impuestas.

En cuanto al algoritmo de búsqueda local empleado dentro del algoritmo memético, se presentan en este trabajo dos fases: una primera de *diversificación*, que permite obtener soluciones de calidad; y una segunda fase, de *intensificación*, que toma las soluciones anteriores como punto inicial de calidad en una nueva búsqueda.

En la primera fase se utiliza una variedad de la técnica de búsqueda de entornos a gran escala denominada *Very Large-Scale Neighborhood Search* (VLSN). En particular, y siguiendo la clasificación propuesta por Ahuja et al. [17], la variante empleada pertenece a los métodos de búsqueda por entornos variable, *Variable-Depth Neighborhood Search* (VDNS). Aunque una de las primeras aplicaciones de esta estrategia puede encontrarse en la resolución de problemas de rutas [18], por lo que hemos podido averiguar, la aplicación de este tipo de búsqueda local en la optimización de estructuras es muy escasa. VDNS se basa en realizar una búsqueda local con un movimiento hasta alcanzar un óptimo local; a continuación se cambia el movimiento a otro más amplio que permita escaparse de dicho óptimo y se sigue así hasta completar un número predefinido de movimientos, cada uno de ellos más amplio que el inmediatamente anterior. En el algoritmo propuesto en el presente artículo, el primer movimiento se define por el cambio aleatorio de una sola variable, eligiendo siempre la nueva solución si mejora a la anterior. El segundo movimiento mueve dos variables simultáneamente, y así sucesivamente. Debe definirse, en este caso, un número de movimientos sin mejora para pasar de un movimiento al siguiente. Esta heurística ha sido aplicada por Martí et al. [19], comprobando que las mejoras obtenidas se centran principalmente en las primeras generaciones, reduciéndose su efectividad a partir de la generación 25 en los casos estudiados.

Para completar la diversificación, se entra en una segunda fase de intensificación con una búsqueda local a partir de las soluciones mejoradas. En el caso de problemas no-lineales se aplican técnicas dependientes del gradiente y que finalizan en un óptimo local para cada una de las soluciones. Se aplica el recocido simulado -simulated annealing (SA)-, que fue propuesto originalmente por Kirkpatrick el al. [20] para el diseño de circuitos electrónicos. El algoritmo comienza con una solución obtenida de la primera fase que es cambiada por un movimiento al azar pequeño de los valores de las variables. Para aceptar la solución se evalúa en términos de coste y de factibilidad estructural. Se aceptan soluciones de mayor coste cuando un número aleatorio entre 0 y 1 es más pequeño que la expresión exp (-ΔE/T), donde ΔE es el incremento del coste y T es la temperatura actual, y se adopta como la nueva solución de trabajo. La temperatura inicial se reduce geométricamente (T=kT) por medio de un coeficiente de enfriamiento k. En cada paso de temperatura se ejecutan un número determinado de iteraciones llamado cadena de Markov. El algoritmo se detiene cuando la temperatura queda reducida a un porcentaje pequeño de la temperatura inicial y, simultáneamente, no hay mejoras en un número consecutivo de cadenas de Markov. Este método, es capaz de sobrepasar óptimos locales en temperaturas de rango alto-medio para converger gradualmente cuando la temperatura se reduce a cero. El método del SA contiene en sí mismo una fase de diversificación y otra de intensificación, y ajustando la Temperatura inicial se reduce la primera de ellas.

3. Ejemplo de tablero de puentes de vigas artesa de hormigones no convencionales.

El ejemplo estudiado se corresponde con los puentes formados por tableros constituidos por dos vigas artesa prefabricadas pretensadas de hormigón autocompac-

tante reforzado con fibras de acero y una losa fabricada in situ de hormigón convencional. El análisis incluye 41 variables del diseño. Las Fig. 1 y 2 muestran las variables geométricas principales consideradas en este análisis. Las variables geométricas incluyen ocho variables: el canto de la viga (h_1), el espesor de la losa (e_4), el ancho del ala inferior de la viga (b_1) y su espesor (e_1), el ancho y espesor de las alas superiores de la viga (b_3 y e_3), el espesor de las almas (e_2) y la separación entre vigas (S_v) . Con respecto a la resistencia del hormigón, son tres las variables que definen el hormigón, la resistencia a compresión de la losa, de la viga y la resistencia a tracción de la viga. El pretensado con armadura activa se encuentra definido por 4 variables: el número de torones en las alas superiores, el número de torones totales dispuestos entre las primeras, segundas y terceras capas del ala inferior, el número de secciones con fundas dispuestas en la segunda capa y el número de secciones con fundas en la tercera capa. Finalmente, son 23 variables las que definen la armadura pasiva estándar dispuesta para la viga y la losa superior (Fig. 2). Todas las variables en este análisis son discretas, pudiendo tomar cada una de ellas un determinado rango de valores que forman un espacio aproximado de 3.15×10^{52} soluciones. En cuanto a los parámetros, éstas son magnitudes que se toman como datos fijos y que, por tanto, no afectan a la optimización de la estructura. Los principales parámetros se han dividido en geométricos, de carga, de coste, de armado y de exposición. Entre ellos destacan el ancho del tablero, la inclinación de las almas, la luz de las vigas, la esbeltez mínima de la viga, las cargas muertas, la distancia de transporte, la dificultad de montaje de las vigas y el tipo de aceros. Las condiciones de durabilidad son las relativas a la instrucción de hormigón EHE [21]. Se han considerado las restricciones estructurales estándares de la normativa española IAP-98 [22], eso incluye comprobaciones en los estados límite últimos y de servicio para flexión, cortante y torsor, así como fisuración y las deformaciones para la envolvente de tensiones debida al peso propio y a las cargas del tráfico. Las tensiones y las reacciones se obtienen como resultado de un programa de desarrollo propio que combina las matrices de rigidez asociadas a los distintos estados evolutivos de la fabricación de las vigas en planta, el hormigonado de la losa sobre las vigas en obra y el tablero completo en servicio. Aplica el método de la rigidez mediante un emparrillado acoplando 20 barras que equivalen a 21 secciones para modelizar la viga y otro emparrillado formado por dos vigas paralelas discretizadas como antes, pero con conexiones entre los nudos de las barras anteriores. Las flechas verticales se limitan a 1/1000 de la luz para la combinación de cargas cuasi-permanente. Se considera la fatiga del hormigón y del acero.

Tabla 2. Resumen variables de la mejor solución del puente

h1 (m)	e4 (m)	b1 (m)	b3 (m)	e1 (m)	e2 (m)	e3 (m)	Hv (KN)	Hl (KN)	p1 (n)	p2 (n)	p3 (n)	p4 (n)	S_v (m)	fR3k (KN)
1,66	0,24	1,97	0,26	0,15	0,10	0,17	45	40	36	9	0	2	5,85	6,5

Tabla 3. Armadura pasiva en la sección de apoyo para la mejor solución

As6	As7	t_1	t_2	t_3	t_4	t_5
122Ø16	49Ø6	Ø8/200	Ø12/200	Ø6/200	Ø20/250	Ø20/300

Tabla 4. Cuantías de acero y hormigón para la mejor solución

Acero pasivo viga (kg)	Acero pasivo losa (kg)	Total acero pasivo (kg/m^2)	Hormigón viga (m^3/m^2)	Hormigón losa (m^3/m^2)
3165	11312	38,92	0,114	0,240

Los algoritmos y el módulo de comprobación estructural se programaron en Intel® Visual Fortran Compiler Integration for Microsoft Visual Studio 2008, sobre un ordenador portátil con un procesador INTEL® CoreTM i7-2720QM CPU@2.20 GHz. Se procesaron seis ejecuciones para una luz de vano de 30 metros, obteniéndose según los casos, 50, 100 ó 150 soluciones optimizadas. En la tabla 5 se muestran los parámetros heurísticos más importantes de los seis procesos y los costes optimizados junto con el tiempo de proceso. En las tablas 2, 3 y 4 se muestran los resultados obtenidos para la solución más económica de las variables geométricas y armadura activa, de la armadura pasiva en la sección de apoyo y de las cuantías de hormigón y acero.

La heurística procesada comienza con la generación aleatoria de 50, 100 ó 150 individuos. Cada una de estas soluciones se mejora mediante una búsqueda VDNS, hasta alcanzar un óptimo local. Para ello, comienza moviendo sólo una variable, y cuando lleva cinco movimientos continuos sin mejora *(no_mej*), se incrementa en uno el número variables (*var*) simultáneas que se mueven, hasta un máximo de cuatro. Con esta nueva población mejorada de 50 individuos, se procede a la aplicación de un algoritmo genético. Los algoritmos genéticos hacen evolucionar una población de individuos que se someten a acciones aleatorias (mutaciones y cruzamientos) de forma que se seleccionan los individuos que se consideran mejor adaptados [23]. No se ha aplicado *elitismo* con el objeto de obtener finalmente soluciones más variadas, sacrificando la obtención de una solución de mejor calidad dentro de la fase de diversificación. La fig. 3 muestra la evolución típica del coste medio de la población de soluciones y el coste de la solución de menor coste respecto al tiempo de cálculo para la heurística 6. Al principio, existe cierta dificultad en la obtención de las primeras 150 soluciones, 8045 segundos. Ello se debe a la complejidad que encuentra el algoritmo en alcanzar una solución factible en la que encaje el pretensado -alrededor de 54 segundos-. A la nueva generación obtenida, se le vuelve a aplicar el VDNS, repitiéndose el proceso hasta un máximo de 50 generaciones. Los parámetros aplicados al algoritmo SA tiene en cuenta la calidad de las soluciones tomadas como punto de partida, disponiendo como valor de la *temperatura inicial* 0.005 ó 0.010 veces el coste de la solución. El resto de parámetros son los siguientes: *cadenas de Markov* de 2000 iteraciones, *coeficiente reductor de la temperatura* de 0.85, y *criterio de parada* del 2% de la *temperatura inicial* y 4 *cadenas sin mejora*.

La utilización de penalizaciones en el algoritmo genético proporciona algunas soluciones iniciales para el SA no factibles. Debido a la complejidad de encajar el pretensado en las vigas, algunos procesos no convergen, reduciéndose el número de soluciones optimizadas. El coste mínimo de las seis heurísticas ha sido de 88318 euros. La Fig. 3 incluye la evolución del coste aceptado de la heurística SA para la solución más económica de las 150 iniciales. Se ha delimitado las fases de diversificación y de intensificación.

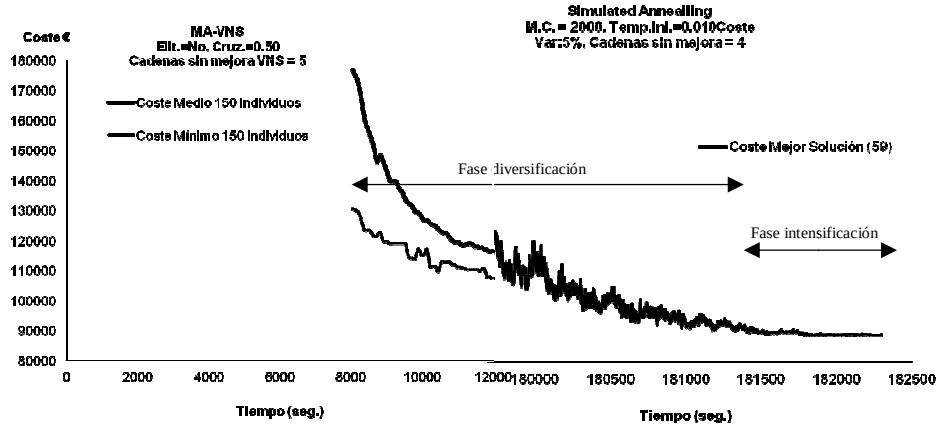


Fig. 3. Evolución del coste respecto al tiempo en el algoritmo memético híbrido de fase doble

Tabla 5. Cuadro de las heurísticas estudiadas aplicando algoritmos memético híbridos

	Genético:		VNLS:		SA:			
	Generaciones = 50		Var.= 4		C.Markov= 2000			
	Elitismo = No		No_mej. = 5		Coef. enfr.= 0,85 Terminación: 0,2			
	Cruzamiento = 0,50				Temp.Ini.			
	Penalización = Si				Terminación: 4 C.Markov			
Heur.	Indiv. (Nº)	Coste med. (€)	Coste mín. (€)	Tiempo (sg)	Temp.Ini. (%coste)	Coste mín. (€)	Tiempo med. SA (sg)	Tiempo (sg)
1	50	111082	110574	7835	0,5	89326	2288	122217
2	50	111082	110574	7835	1,0	89470	2920	153824
3	100	107614	107002	14432	0,5	88363	2181	232549
4	100	107614	107002	14432	1,0	88363	2729	287283
5	150	102664	98906	21563	0,5	88565	1828	295735
6	150	102664	98906	21563	1,0	88318	2283	363957

La Tabla 5 muestra los costes mínimos obtenidos en el proceso de las heurísticas, valores que son muy similares entre sí, y que son algo inferior a los costes de las soluciones tradicionales para hormigones convencionales, lo que permite tener en consideración este material para su aplicación en la práctica estructural. Los resultados optimizados dentro de cada heurística, proporcionan soluciones muy similares con desviaciones muy bajas, algunas de las soluciones se repiten. En el caso de la heurística 6, de las 150 soluciones posibles, 126 han convergido y de ellas, un total de 97 son distintas, lo que representa un 64.7% del total. En la Fig. 4 se representan las medias acumuladas y la desviación estándar de las soluciones obtenidas. El tiempo medio de proceso de cada solución optimizada es de 62 minutos. Resulta interesante comparar las mejores soluciones alcanzadas aplicando técnicas heurísticas respecto a

estructuras realmente construidas y calculadas mediante procedimientos habituales. Se han comprobado para casos similares ahorros apreciables, en torno al 7-8% [19].

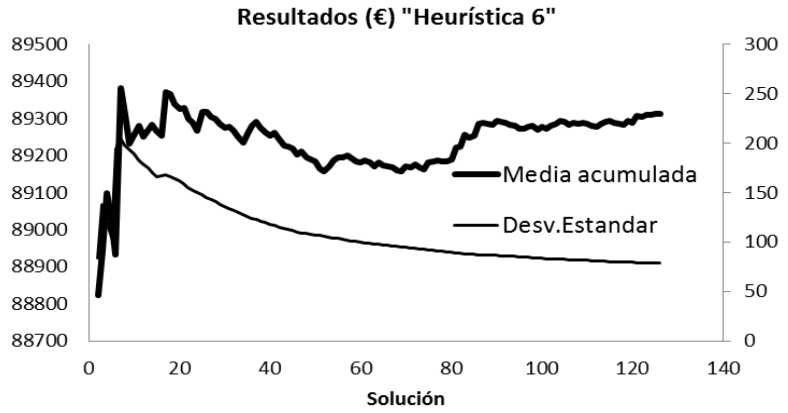


Fig. 4. Media acumulada y desviación estándar (euros) para las soluciones obtenidas

4. Conclusiones

El algoritmo memético híbrido de fase doble MA_VNSL_SA se ha aplicado con éxito en el análisis de optimización de puentes de vigas artesa de hormigones no convencionales. Se ha probado como un algoritmo de búsqueda eficiente, obteniéndose múltiples soluciones con escasas desviaciones en la mayoría de los distintos procesos, alcanzando soluciones buenas en un tiempo razonable.

El coste de las soluciones optimizadas permite considerar al hormigón no convencional como material competitivo respecto al hormigón tradicional para su utilización en la ejecución de puentes. El ejemplo presentado muestra el alto potencial de los algoritmos meméticos, pudiéndose aplicar con garantías a casos de estructuras reales.

Agradecimientos

Los autores agradecen el aporte financiero realizado para este trabajo por el Ministerio de Ciencia e Innovación (Proyecto de Investigación BIA2011-23602) y por la Universitat Politècnica de València (Proyecto de Investigación PAID-06-12).

Referencias

[1] British Standards, BS 8110-1: 1997 Structural use of concrete. Part 1: Code of practice for design and construction, British Standards, London, 1997.
[2] Cohn M.Z. and Dinovitzer A.S., Application of structural optimization. ASCE Journal of Structural Engineering, 120(2), pp 617-649, 1994.
[3] Yepes V. and Medina J.R. Economic heuristic optimization for the heterogeneous fleet. ASCE J. of Transp. Engineering,132(4), pp 303-311, 2006.
[4] Jenkins W.M., Plane frame optimum design environment based on genetic algorithm, ASCE Journal of Structural Engineering, 118(11), pp. 3103-3112, 1992.

[5] Rajeev S. and Krisnamoorthy C.S., Discrete optimization of structures using genetic algorithms, ASCE J. of Str. Engineering, 118(5), pp. 1233-1250, 1992
[6] Coello C.A., Christiansen A.D. and Santos F., A simple genetic algorithm for the design of reinforced concrete beams, Eng.with Comput., 13, pp. 185-196, 1997.
[7] Martinez, F.J.; González-Vidosa, F.; Hospitaler, A.; Yepes, V., Heuristic optimization of RC bridge piers with rectangular hollow sections, Computers & Structures 88 (2010) 375-386.
[8] Paya, I.; Yepes, V.;, F. González-Vidosa, A. Hospitaler, On the Weibull cost estimation of building frames designed by simulated annealing, Meccanica 45 (2010) 693-704.
[9] Carbonell, A., Gonzalez-Vidosa, F., Yepes, V.: Design of reinforced concrete road vaults by heuristic optimization. Adv. Eng. Software 42, 151-159 (2011)
[10] Yepes, V.; González-Vidosa, F.; Alcala, J.; Villalba, P.; CO2-Optimization design of reinforced concrete retaining walls based on a VNS-Threshold acceptance strategy, Journal of Computing in Civil Engineering ASCE 26 (2012) 378-386.
[11] Martí, J.V.; González-Vidosa,F. Design of prestressed concrete precast pedestrian bridges by heuristic optimization, Advances in Engineering Software 41 (2010) 916-922.
[12] Martí, J.V., González-Vidosa F, Yepes, V., Alcalá, J.: Design of prestressed concrete precast road bridges with hybrid simulated annealing. Eng. Struct. 48, 342–352 (2013)
[13] Moscato, P. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms, Technical Report Caltech Concurrent Computation Program Report 826, Caltech, Pasadena, California, USA, 1989.
[14] Dawkins, R.The selfish gene, Clarendon Press, Oxford, UK, 1976.
[15] Blum, C.; Puchinger, J.; Raidl, G.R.; Roli, A. Hybrid metaheuristics in combinatorial optimization: A survey, Applied Soft Computing 11 (2011) 4135-4151.
[16] Krasnogor,N.; Smith, J.; A tutorial for competent memetic algorithms: model, taxonomy, and design issues, IEEE Trans. on Evolut. Comput. 9 (2005) 474-488.
[17] Ahuja,R.K.; Ergun, Ö.; Orlin, J.B.; Punnen, A.P. A survey of very large-scale neighborhood search techniques, Discr.Applied Mathematics 123 (2002) 75-102.
[18] Lin, S., Kernighan, B. An effective heuristic algorithm for the traveling salesman problem, Operations Research 21 (1973) 498-516.
[19] Martí, JV.; Yepes, V.; González-Vidosa, F; Luz, A. “Diseño automático de tableros óptimos de puentes de carretera de vigas artesa prefabicadas mediante algorítmo meméticos híbidos”. Artículo aceptado por la revista RIMNI con referencia RIMMNI-D-12-00008R2 (2013).
[20] Kirkpatrick S., Gelatt C.D. and Vecchi M.P., Optimization by simulated annealing, Science, 220(4598), pp. 671-680, 1983.
[21] M. Fomento, EHE. Instrucción de Hormigón Estructural. Madrid, 1998.
[22] M. Fomento, IAP-98. Instrucción sobre las acciones a considerar en el proyecto de puentes de carretera, M.Fomento: Madrid, 1998.
[23] Holland, J.H., Adaptation in natural and artificial systems, University of Michigan Press, Ann Arbor, 1975.

Una comparación del método Tabu y GRASP para el Problema de Clasificación con Capacidades

A. Martínez-Gavara[1], V. Campos[1], M. Gallego[2], R. Martí[1]

[1]Departamento de Estadística e Investigación Operativa, Universidad de Valencia,
[2]Departamento de Ciencias de la Computación, Universidad Rey Juan Carlos de Madrid.
gavara@uv.es, vicente.campos@uv.es, micael.gallego@gmail.com, Rafael.Marti@uv.es

Resumen El problema de clasificación con capacidades (The Capacitated Clustering Problem, CCP) consiste en la formación de un número determinado de grupos de un conjunto de elementos de tal manera que la suma de los pesos de los elementos de cada grupo se encuentra dentro de unos límites de capacidad, y la suma de los beneficios entre los pares de elementos de un mismo grupo sea máximo. Este problema lo podemos encontrar, por ejemplo, en la planificación de las instalaciones de procesamiento y distribución de correo. Recientemente, se ha propuesto un método GRASP con VNS (ver Deng y Bard (2011)). En este trabajo se propone una búsqueda tabú y GRASP basados en diferentes entornos para este problema NP-duro. El problema de agrupamiento de máxima diversidad (The Maximally Diverse Grouping Problem, MDGP) es un caso particular del CCP en donde todos los elementos tienen un peso de una unidad (ver Gallego et al. (2013)). Este problema ha sido estudiado recientemente en el contexto académico cuando se forman grupos de estudiantes. Hemos adaptado su mejor método, una búsqueda tabú con oscilación estratégica (TS_SO) a nuestro problema, CCP. Realizamos un estudio empírico con 50 ejemplos, que muestra la superioridad de la adaptación del TS_SO, con respecto al GRASP con VNS.

1. Introducción

Dado un grafo $G = (V, E)$, donde V es el conjunto de n nodos y E es el conjunto de aristas, sea $w_i \geq 0$ el peso del nodo $i \in V$ y sea $c_{i\,j}$ el beneficio de la arista $(i, j) \in E$. El Problema de Clasificación con Capacidades (The Capacitated Clustering Problem, CCP), consiste en la partición de V en p grupos de forma que la suma de los pesos de los elementos en cada grupo esté dentro de unas capacidades enteras mínimas y máximas, L y U, y la suma de los beneficios entre los pares de elementos en los grupos sea máximo.

El CCP se puede formular como un problema cuadrático entero con variables binarias x_{ik} que toman el valor 1 si el elemento i se encuentra en el grupo k y 0 en otro caso.

$$
\begin{array}{lll}
\text{(CCP) Maximizar} & \sum_{k=1}^{p} \sum_{i=1}^{n-1} \sum_{j>i}^{n} c_{ij}\, x_{ik}\, x_{jk} & \\
\text{s.a.:} & \sum_{k=1}^{p} x_{ik} = 1 & i = 1, \ldots, n \\
& L \leq \sum_{i=1}^{n} w_i x_{ik} \leq U & k = 1, \ldots, p \\
& x_{ik} \in \{0, 1\} & i = 1, \ldots, n \quad k = 1, \ldots, p
\end{array}
$$

La función objetivo obtiene el beneficio de todos los pares de elementos que se encuentran en el mismo grupo. El primer conjunto de restricciones fuerza la asignación de cada elemento a un grupo. Por otro lado, el segundo conjunto de restricciones nos limita que la suma de los pesos de los nodos dentro de cada grupo esté entre L y U.

En [1] los autores resumen de forma detallada las formulaciones y heurísticos previos que se pueden encontrar en la literatura del CCP, que no reproduciremos aquí. Los autores proponen un procedimiento de búsqueda adaptativo aleatorizado (Greedy Randomized Adaptive Search Procedures, GRASP) con búsqueda por entornos variables (Variable Neighborhood Search, VNS) que, de acuerdo con su estudio computacional, mejora las aproximaciones previas. Tal y como mencionan en el artículo, una interesante aplicación del problema la podemos encuentrar en la organización del procesamiento y distribución del correo dentro del servicio postal de EEUU. Sin embargo, cabe destacar en el estudio computacional, que los autores sólo consideran los pesos en los nodos w_i iguales a 1. En este artículo, extendemos el estudio a cualquier tipo de instancias del CCP para comparar los diferentes diseños.

Revisando la literatura reciente, nos encontramos con un problema muy relacionado con el que nos concierne. Se trata del problema de agrupamiento de máxima diversidad (The Maximally Diverse Grouping Problem, MDGP), el cual consiste en clasificar un conjunto de elementos en p grupos de manera que la diversidad entre los elementos en cada grupo sea máxima. Dicha diversidad entre los elementos de un grupo se calcula como la suma de las distancias entre cada par de elementos. El objetivo del problema es maximizar la diversidad total, es decir, la suma de la diversidad de todos los grupos, cuando el número de elementos de cada grupo se encuentra dentro de un intervalo dado. Por tanto, el MDGP es un caso especial del CCP en el cual los pesos $w_i = 1$ en todos los nodos, y la distancia entre cualquier par de nodos (i, j) es el beneficio $c_{i\,j}$. El MDGP también es conocido como el problema de la $k-$partición en Feo et al. en [2]. Gallego et al. en [4] proponen un tabu search con oscilación estratégica, TS_SO, que mejora los enfoques previos para el MDGP.

En la siguiente sección resumimos el método empleado por Deng y Bard para el CCP. En la sección 3 detallamos la extensión del método TS_SO para el CCP. En la sección 4 describiremos un método basado en GRASP. Finalmente, en la sección 5 presentamos los experimentos computacionales, estarán basados en 50 instancias generales del CCP con $n = 82, 240$ y 480 nodos.

2. Método CCP Previo

Deng y Bard proponen en [1] un GRASP para el CCP, el cual, como es habitual en este tipo de métodos, alterna dos fases, la construcción y la mejora. En la construcción, los p grupos son inicializados utilizando el algoritmo de la arista más pesada (HWE), y completado con un procedimiento aleatorizado. Los autores proponen otro método constructivo, sin embargo, tal y como muestran sus experimentos numéricos, el HWE da mejores resultados, por ello nos centraremos en él. El algoritmo HWE identifica los p nodos con mayor peso y los asigna a cada uno de los p grupos. A continuación, se identifican las aristas más pesadas con dichos nodos y se incorporan los nodos incidentes a ellas al grupo correspondiente.

Tras la aplicación del algoritmo HWE, todos los grupos contienen dos nodos. Para completar los grupos, se crea una lista de candidatos CL para continuar con la construcción siguiendo la metodología GRASP. La particularidad la encontramos en el hecho que dicha lista está creada tanto con nodos como por aristas, que se pueden incluir en un grupo mientras no se exceda la capacidad máxima permitida, U. Esta no es una estrategia convencional para el GRASP, tal y como presentan Festa y Resende en [3]. En la sección 5 comparamos este diseño con los diseños convencionales en los cuales la CL contiene nodos o aristas, pero no ambos.

En la segunda fase del GRASP, los autores proponen tres entornos diferentes para mejorar la solución construida. Y basándose en ellos, proponen dos métodos distintos. El primero, llamado CNS, encadena los entornos de forma cíclica o secuencial, utilizando cada vez un entorno independientemente de que se consiga mejorar o no, y finalizando cuando no hay mejora posible en ningún entorno. El segundo, llamado VND, cambia de un entorno al siguiente solamente cuando el actual no consigue ninguna solución de mejora. Este último termina cuando el tercer entorno no consigue mejorar. Finalmente, los autores implementan un esquema VND aleatorizado (RVND) que consiste en elegir de forma aleatoria el próximo entorno a aplicar. Utilizaremos para nuestra experimentación el método RVND, a este método previo lo llamaremos Prev_GRASP.

3. Adaptación de MDGP

En [4] los autores proponen un método de búsqueda tabú con oscilación estratégica, TS_SO, para MDGP, el cual, es actualmente el mejor método conocido para resolver este problema. Como describimos en la primera sección, el MDGP es un caso particular del CCP, por lo cual lo hemos adaptado fácilmente al problema más general, CCP. Describiremos de forma breve el método original, el cual está basado en tres pasos: construcción de la solución inicial, búsqueda local, y oscilación estratégica.

La construcción de la solución inicial (greedy construction, GC) para el MDGP comienza seleccionando al azar p nodos y asignando uno a cada grupo. Tras este paso inicial, el procedimiento realiza $n - p$ iteraciones para asignar

los elementos restantes no asignados a los grupos. Con el fin de generar una solución factible, las iteraciones se dividen en dos fases. En la primera fase, los elementos se asignan a grupos que actualmente contienen menos elementos que el número mínimo deseado de elementos. En la segunda fase, los elementos restantes se asignan a grupos con un número de elementos que es menor que el número máximo deseado de elementos.

En nuestra adaptación de GC para el CCP, la primera fase añade un elemento seleccionado aleatoriamente i al grupo k con $W_k < L$ de manera que $I(i,k)$ sea máximo, donde $I(i,k)$ es el incremento de la función objetivo (beneficio) si el nodo i se añade al grupo k. Cuando todos los grupos tienen un peso total mayor o igual que L, la segunda fase se aplica de la misma manera, añadiendo elementos seleccionados al azar a grupos de forma que se maximice $I(i,k)$, sin exceder el límite superior de la capacidad U. Una vez que obtenemos una solución inicial, se aplica una búsqueda local (fase 2 en el método TS_SO), la cual, intenta mejorar las soluciones obtenidas con inserciones e intercambios de nodos. Este método se adapta fácilmente al CCP simplemente comprobando la factibilidad de la solución en términos del peso de los elementos en cada grupo, en lugar del número de elementos.

Hasta ahora, hemos supuesto que la viabilidad se mantiene a través de la búsqueda. Sin embargo, en la tercera fase del algoritmo, la oscilación estratégica (SO), explora soluciones para que las restricciones de cardinalidad del grupo puedan ser violadas. En particular, el método aplica la búsqueda local del paso anterior pero permitiendo que el límite de las capacidades inferior y superior puedan superarse en una cierta cantidad so que oscila entre 0 y $so_{max} = 4$. Con lo cual, cuando $so > 0$, en la búsqueda local se permiten soluciones no factibles. Para crear el patrón de oscilación, el valor de so vuelve al valor uno después de cada aplicación del método con una solución mejorada, y se incrementa en una unidad en caso contrario, hasta que éste alcanza so_{max}.

En nuestra adaptación de SO a CCP, tenemos en cuenta que la suma de los pesos de los elementos de un grupo k (W_k) debe estar dentro de los límites, $L - so$ y $U + so$.

Este método no garantiza que la solución final sea factible. Si no es así, hay que aplicar un mecanismo de reparación que consiste en la eliminación de elementos del grupo k, infactible, que supera el límite superior, $W_k > U$ y añadir elementos a las agrupaciones s que no llegan al límite inferior, $W_s < L$. Los elementos se seleccionan al azar y el proceso continúa hasta que se cumplan las restricciones de peso de todos los grupos. Llamamos a todo el método adecuado para CCP como AdTS_SO, y vamos a compararlo con Prev_GRASP y los nuevos métodos que se describen a continuación en nuestra experiencia computacional.

4. Método GRASP simplificado para CCP

En nuestra opinión, el método GRASP de Deng y Bard, [1] obtiene soluciones de alta calidad, pero tiene demasiados elementos. Como cuestión de hecho, se hibrida un GRASP con VNS, en el cual se emplean tres entornos distin-

tos, los cuales se combinan de una manera aleatoria. Por otra parte, como se mencionó anteriormente, la lista de candidatos en el proceso de construcción considera tanto nodos como aristas. En esta sección se propone una versión simplificada de su método con una aplicación GRASP estándar, en el que sólo los nodos son candidatos en el proceso de construcción y se consideran dos simples entornos los cuales se combinan en un diseño de búsqueda por entornos variables descendente (VND) determinista.

El nuevo método GRASP que hemos diseñado, comienza inicializando los p grupos con p nodos selecionados al azar, no los más pesados como HWE. A continuación, se exploran todos los grupos asignando nodos hasta que todos ellos satisfagan la restricción de límite inferior (es decir, hasta que la suma de los pesos de los elementos ya asignados al grupo sea mayor o igual que L). Para ello, el conjunto de candidatos al grupo k, CL_k, se ha formado con los nodos sin asignar que pueden ser admitidos en el grupo k. El método evalúa utilizando $I(i,k)$ la asignación de nodo i al grupo k. Entonces, RCL_k, la lista de candidatos restringida, se forma con los candidatos i con una evaluación $I(i,k)$ dentro de un porcentaje $\alpha \in]0,1]$ del valor máximo I_{max} en CL_k:

$$RCL_k = \{i \in CL_k : \alpha\, I(i,k) \geq I_{max}\} \quad \text{donde } I_{max} = \max_{i \in CL_k} I(i,k)$$

Así pues, nuestro método constructivo selecciona al azar un elemento en RCL_k, y realiza la asignación correspondiente. El método se aplica secuencialmente empezando con el grupo $k = 1$, y asignando elementos hasta que se verifica el límite inferior ($L \leq W_1$ donde $W_1 = \sum_{j \in V_1} w_j$). Entonces, el método pasa al grupo 2 y procede del mismo modo, hasta que la suma de todos los elementos asignados a cada grupo sea mayor o igual que L.

En los siguientes pasos, el conjunto de candidatos CL está formado por los nodos i sin asignar tal que, existe un grupo k que la suma de los pesos de los elementos ya asignados a ese grupo k más el peso del nodo i, es menor o igual que U, es decir, no supera el límite superior:

$$CL = \{i : \exists k \,/\, W_k + w_i \leq U,\ 1 \leq k \leq p\} \quad \text{donde } W_k = \sum_{j \in V_k} w_j$$

El método evalúa $I(i,k)$ para todo i en la CL, y construye RCL con los nodos tales que su evaluación $I(i,k)$ esté dentro de un porcentaje $\alpha \in]0,1]$ del valor máximo I_{max} en CL, de forma similar a la realizada anteriormente, y lleva a cabo una de ellas seleccionada al azar. El proceso se detiene cuando todos los nodos se han asignado a los grupos. A diferencia de la primera parte del método, los candidatos i pueden estar evaluados a varios grupos paralelamente.

Nuestro método de mejora en el GRASP consiste en una búsqueda por entornos variables descendente determinista (VND) basada en dos entornos, N_0 y N_1. El entorno N_0 implementa un movimiento de inserción clásica, en la que un nodo i se traslada de un grupo k a otro $\hat{k}$, siempre que ambos grupos siguen siendo posibles después del movimiento (es decir, si $L \leq W_k - w_i$ y $W_{\hat{k}} + w_i \leq U$). Del mismo modo, N_1 implementa un movimiento de cambio, en el que un nodo i se

traslada de un grupo k a otro $\hat{k}$, y al mismo tiempo un nodo j se mueve desde el grupo $\hat{k}$ al k. La medida sólo se realiza si la solución final es factible. Ambos entornos interactúan dentro de un VNS. El método cambia de un entorno al otro solamente cuando en el actual no consigue ninguna solución de mejora. El método termina cuando no consigue mejorar en ninguno de los dos entornos. Vamos a comparar este método GRASP simplificado, llamado simplemente GRASP, con los métodos anteriores en la Sección siguiente.

5. Resultados numéricos

En esta sección se describen los experimentos computacionales que llevamos a cabo para probar la eficacia y la eficiencia de los tres procedimientos mencionados anteriormente:

- Prev_GRASP, GRASP diseñado por Deng y Bard en [1].
- AdTS_SO, nuestra adaptación al CCP del TS_SO propuesto originalmente para el MDGP de Gallego et al. en [4].
- GRASP, el nuevo método utilizado, sección 4.

Implementamos Prev_GRASP y AdTS_SO en Java SE 6. GRASP para el CCP ha sido implementado en C. Todos los experimentos se llevaron a cabo en un procesador Intel Core 2 Quad CPU Q 8300 con 6 GB de RAM.

Se han empleado 50 instancias en nuestra experimentación. El conjunto está dividido en dos subconjuntos, el conjunto de datos que llamaremos RanReal (RR), contiene 40 instancias, donde los beneficios $c_{i\,j}$ son generados utilizando una distribución uniforme $U(0, 1000)$ propuestas por Gallego et al. para el MDGP. Las hemos adaptado al CCP mediante la generación de pesos en los nodos utilizando una $U(0, 10)$. Por tanto, obtenemos 20 instancias con $n = 240$, $p = 12$, $L = 75$ y $U = 125$, y otras 20 con $n = 480$, $p = 20$, $L = 100$ y $U = 150$. Finalmente, tenemos el conjunto DB, que usa los beneficios del ejemplo de Deng y Bard, sin embargo, hemos generado 10 conjuntos de pesos en $U(0, 10)$. En concreto, tenemos 10 instancias más con $n = 82$, $p = 8$, $L = 25$, y $U = 75$.

Para la experimentación preliminar, hemos seleccionado 15 casos representativos, con características diferentes. Consideramos 6 de cada uno de los dos conjuntos RanReal y 3 en el conjunto DB. En nuestra experimentación intentamos examinar el parámetro $\alpha \in]0, 1]$, generando 100 soluciones distintas para diferentes valores de α, 0,2, 0,4, 0,6 y 0,8, y observando el número de mejores soluciones obtenidas en los distintos valores, así como el porcentaje medio de la desviación a la mejor solución encontrada, nos lleva a concluir que el parámetro α debe tener el valor de 0,6.

Por otro lado, en el caso de los métodos GRASP, también es importante conocer el origen de las mejores soluciones después de aplicar el procedimiento de mejora. Por ello, estamos interesados en medir la correlación entre el valor de la función objetivo de las soluciones construidas y el valor de las soluciones mejoradas, para comprobar si las buenas soluciones finales vienen o no de las mejores soluciones construidas. Si la correlación es alta, esta respuesta podría llevarnos

a poner en marcha mecanismos de filtro para ahorrar el tiempo de cálculo invertido en la mejora de soluciones construidas que provienen de soluciones que no superan un umbral de calidad y que por tanto darían soluciones mejoradas de escasa calidad. Para ello, hemos generado 100 soluciones para cada problema en nuestra serie de 15 casos representativos (conjunto de pruebas) y calculado el coeficiente de correlación entre ambos valores objetivos en cada uno de ellos. El valor resultante de $r = 0,01$ indica que la correlación es extremadamente baja y por lo tanto buenas soluciones finales puede venir de soluciones construidas de cualquier valor. La Fig. 1 confirma este hecho, ya que representa ambos valores (el valor de la solución construida en el eje x y la mejorada en el eje y) en 100 soluciones de un ejemplo representativo, en este caso RR240, y cuyo comportamiento es similar en otras instancias. Se observa que los puntos se encuentran dispersos en el plano sin ningún patrón.

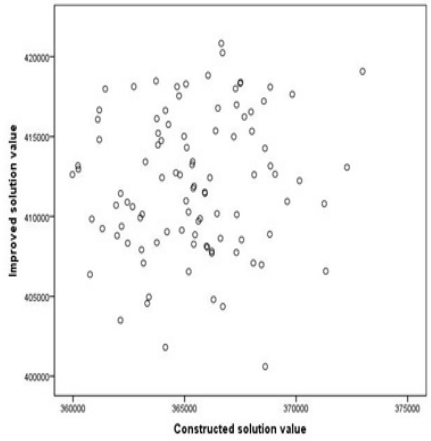


Figura 1. Valor de la solución contruida y valor de la solución mejorada.

En nuestro segundo test preliminar se compara GRASP (con $\alpha = 0{,}6$) con Prev_GRASP de Deng y Bard, en el conjunto con 15 instancias. En particular, limitamos los dos métodos a 60 segundos en cada instancia. Cómo se puede observar en la tabla 1, ambos métodos tienen un rendimiento similar en promedio, aunque el nuevo GRASP alcanza valores un poco mejores que el anterior. Además, alcanza 9 de las 15 mejores soluciones y exhibe una desviación de 2,84 %, en contraste con las 6 mejores soluciones y 10,99 % obtenidos con Prev_GRASP. Es interesante señalar que Prev_GRASP mejora sobre GRASP en los casos RanReal tamaño medio ($n = 240$), ya que se obtiene una desviación media de 0,00 %, mientras que GRASP obtiene 7,21 %. La situación cambia cuando se aplican los métodos para los casos más grandes en este juego. GRASP supera Prev_GRASP en las instancias de RanReal ($n = 480$) en el que las antiguas exhibe una desviación de 0,00 % a partir de las mejores soluciones conocidas, mientras que la

segunda muestra un valor de 27,38 %. En la tabla 1 se muestra para cada método el valor promedio de la función objetivo (Valor), el porcentaje de la desviación media a la mejor solución obtenida (Dev) y el número de instancias en las cuales el método alcanza la mejor solución (#mejor).

	Prev_GRASP	GRASP
Valor	194109,8	233841,7
#mejor	6	9
Dev	10,99 %	2,84 %

Cuadro 1. Comparación de los métodos GRASP

Finalmente, hemos realizado un último test preliminar al método GRASP en el cual comparamos los diferentes resultados obtenidos si utilizamos en la lista de candidatos sólo nodos, sólo aristas o si incluimos ambos. El valor medio de la solución en los tres casos es similar, sin embargo, la mejor solución siempre se encuentra en el caso en el que la lista de candidatos está formada por los nodos, en el caso que la lista de candidatos esté formada por sólo aristas, la desviación media con respecto a la mejor solución es de un 5,41 % y finalmente la que está compuesta tanto por aristas como por nodos tiene una desviación del 8,73 %.

Nuestro experimento final, será comparar todos los métodos en el conjunto de las 50 instancias. En la tabla 2 observamos el porcentaje de desviación media a la mejor solución obtenida en estos experimentos. En ella, se observa que el método AdTS_SO es el que mejores resultados obtiene, encontramos la mejor solución utilizando este método con las 40 instancias RanReal en los dos tamaños, mientras que en las instancias de DB se obtiene la mejor solución en 8 de los 10 ejemplos, el método GRASP obtiene la mejor solución en 2 ocasiones, y exhibe una desviación de 0,08 % en contraste del 0,33 % que se observa en el método Prev_GRASP. Tal y como observamos en los experimentos previos, Prev_GRASP mejora sobre GRASP en los casos RanReal tamaño medio ($n = 240$), obteniendo mejor desviación media de un 6,82 % respecto al 12,09 % del GRASP, sin embargo, mientras que el método GRASP se mantiene en una desviación media de alrededor del 12 % al 14 %, Prev_GRASP aumenta su desviación hasta un 36,99 % en los casos RanReal de tamaño $n = 480$.

	Prev_GRASP	AdTS_SO	GRASP
DB	0,33 %	0,03 %	0,08 %
RR240	6,82 %	0,00 %	12,09 %
RR480	36,99 %	0,00 %	13,29 %

Cuadro 2. Resultado de las desviaciones de cada uno de los métodos a la mejor solución encontrada

6. Conclusiones

La adaptación AdTS_SO ha resultado ser la mejor. Nuestro GRASP simplificado ha conseguido superar a Prev_GRASP en los propios ejemplos propuestos por ellos y también en las instancias más grandes, por último, decir que Prev_GRASP solamente ha conseguido superar a nuestro GRASP en RR240 por lo que no se justifica la mixtura en nodos−aristas en la CL.

7. Agradecimientos

Este trabajo ha sido financiado parcialmente por el Ministerio de Educación, Cultura y Deporte (PRX12/00016), por el Ministerio de Economía y Competitividad (TIN2012-35632) y por la Generalitat Valenciana, Proyecto Prometeo (2013/049).

Referencias

1. Deng, Y. and J.F. Bard A reactive GRASP with path relinking for the capacitated clustering *Journal of Heuristics*, 17:119–152, 2011.
2. Feo, T., Goldschmidt, O. and Khellaf, M., One-half approximation algorithms for the k-partition problem *Operation Research*. 40:170–173, 1992.
3. Festa, P., Resende, M.G.C., An annotated bibliography of GRASP, Part I: Algorithms *International Transactions in Operational Research*. 16:1–24, 2009.
4. Gallego, M., Laguna, M., Martí, R. and Duarte, A. Tabu search with strategic oscillation for the maximally diverse grouping problem *Journal of the Operational Research Society*. 64(5): 724–734, (2013)

Predicción de tráfico mediante co-evolución de Redes Neuronales de Funciones de Base Radial y selección de variables de entrada

V. M. Rivas[1], Elisabeth Parras-Gutierrez[1], M. G. Arenas[2], P. A. Castillo[2], P. García Sánchez[2], J.J. Merelo[2], P. García-Fernández[3]

[1] Dept. de Informática, Universidad de Jaén, Jaén, España
[2] Dept. Arquitectura y Tecnología de los Computadores, Universidad de Granada, España
[3] Dept. Electrónica y Tecnología de los Computadores, Universidad de Granada, España
vrivas@ujaen.es

Resumen Este trabajo presenta un novedoso sistema de detección de flujo de tráfico, y muestra los resultados del algoritmo L-Co-R en la predicción de la serie temporal que el citado flujo de tráfico representa. La predicción de la serie temporal se ha realizado comparando con algoritmos existentes y muestra la adecuación del algoritmo, L-Co-R, ante el problema de la predicción de la serie temporal captada.

1. Introducción

El desarrollo de nuevos sistemas que informen de las condificiones del tráfico y del flujo de vehículos en las carreteras, es una apuesta de la Dirección General de Tráfico (DGT) dentro del territorio nacional y de las administracciones homólogas fuera del territorio nacional. Ejemplos de esta apuesta son los proyectos desarrollados para informar lo más exactamente posible acerca del estado del tráfico, como el que lleva a cabo Google Maps que proporciona desde hace no mucho tiempo (2011) información del estado del tráfico en las ciudades más importantes de algunos paises, aunque el proyecto inicial empezó sólo en algunas ciudades de EEUU (2009). Otros ejemplos nacionales son la apuesta de la DGT por la investigación en tráfico financiando proyectos y promocionando encuentros sobre movilidad y seguridad vial.

Paralelamente a esta circunstancia, la población actual está evolucionando y se está convirtiendo en una población permanentemente informada, conectada y ubícua, puesto que en España existen actualmente más de 52 millones de clientes de telefonía móvil (`http://goo.gl/WTPN3`). Esta característica permite que casi cualquier conductor utilice a diario esos dispositivos de comunicación y por lo tanto lo lleven siempre con ellos mismos para comunicarse tanto de forma hablada como escrita, informarse, consultando cualquier dato que les pueda interesar (gracias a las tarifas de datos), o utilizando cualquier otro servicio accesible desde el dispositivo.

En España existían en 2012 165.568 Km de vías (último dato oficial publicado) [1], lo que da una idea muy clara de la complejidad de una red, puesto que, aunque sea sólo por su extensión, necesita muchísimo trabajo para su coordinación, su mantenimiento y su gestión en general. Por lo que al organismo que gestiona la red le es de mucha utilidad tener información lo más precisa posible, por básica que sea, sobre los usuarios que la utilizan, para poder preveer situaciones de saturación, o de evacuación en caso de alguna emergencia. Datos tan básicos como el número de vehículos que pasan por cada vía o a qué hora hay más afluencia para intentar regular la señalización de cada vía podrían facilitar el trabajo de los operarios.

En este trabajo se presenta un sistema de recogida de información de estado del tráfico y se analizan los datos recogidos incluyendo una predicción inicial de las series temporales de datos recogidas. Una serie temporal es un conjunto de datos ordenados de forma cronológica que sirve de base para realizar una predicción de valores futuros dentro de esa misma serie temporal. La predicción de series temporales es un campo de investigación activo dentro de areas como la estadística o la investigación operativa [5], [4], y son datos presentes tanto en el campo de la ingeniería como en la biología, economía o las ciencias sociales entre otras.

La captación de datos se basa en la detección de dispositivos *bluetooth* (BT), en diferentes vías, lo que nos proporciona una descripción de las condiciones de tráfico en tiempo real y un conjunto de datos válidos para utilizarlos con técnicas de predicción de series temporales. La captación de datos incluye tanto dispositivos BT empotrados en algún tipo de vehículo como dispositivos de usuarios particulares que lleven el bluetooth activado. El principal dato recolectado de cada dispositivo es la dirección MAC que se encripta y se almacena, lo que permite identificar de forma unívoca el dispositivo, pero no el usuario, por lo que la privacidad de la persona usuaria del dispositivo está totalmente reservada. Esta identificación única puede permitir la identificación del dispositivo en varios puntos de la ciudad, lo que puede dar una idea del tipo de desplazamientos que ese usuario ha realizado.

Concretamente el dispositivo utilizado para la captación de datos es Intelify (`http://www.intelify.net/`) que ofrece una solución compacta para captar dispositivos BT y transmitir la información a un servidor remoto manteniendo un índice de detección adecuado (incluye un 8.5 % de error en la captación). Intelify es un dispositivo autónomo que puede ser instalado en el área donde se desean captar datos. Dispone de varios sensores y capta la información para enviarla a través de una conexion 3G. Los datos recopilados que se incluyen en este trabajo han sido captados por 5 puntos diferentes (nodos 1-5) durante un total de 60 días.

La información que se capta se organiza en una entidad que podemos denominar *paso*. Un *paso* será un dispositivo detectado en un determinado nodo de captación a una determinada hora de inicio y que se terminó de captar a una determinada hora de fin. Es decir, con cuatro datos, nodo, dispositivo, hora de inicio y hora de fin se puede categorizar un paso. La mac del dispositivo incluye

la identificación de la marca, modelo y algunas características más del dispositivo que ayuda a su categorización dentro del abanico de dispositivos captados por Intelify.

En este primer trabajo se han aplicado varios algoritmos de predicción de series temporales ya validados en la bibliografía y se presentan los resultados iniciales para predicción a corto plazo (1 día) sin modificar, tratar o adecuar la serie de datos inicialmente recopilada por el dispositivo de captación.

El objetivo final de este proceso de investigación inicial es predecir el flujo de tráfico que tendrán las vías monitorizadas con este tipo de dispositivo o cualquier otro que proporcione una información simple y fiable a cerca del paso de vehículos por un determinado punto. Esta predicción permitirá, por ejemplo, poder tomar decisiones acerca del mejor itinerario a elegir para un determinado desplazamiento, tanto para los usuarios de las vías como para las administraciones a la hora de planificar señalización, e intervención en las vías monitorizadas.

El artículo a partir de este punto se organiza en secciones. La sección 2 repasa el estado del arte tanto de los algoritmos de de predicción de series temporales como de los dispositivos de captación de datos existentes actualmente. Por su parte, la sección 3 presenta los resultados obtenidos en la predicción a corto plazo del flujo del tráfico. Por último en la última sección se presentan unas conclusiones iniciales y los pasos que se pondrán en práctica como líneas futuras de trabajo.

2. Estado del arte

En esta sección se repasarán de forma resumida los dispositivos de captación de tráfico existentes y posteriormente introduciremos un pequeño estado del arte sobre los algoritmos de predicción de series temporales que están vigentes.

Las tecnologías que permiten detectar el flujo de tráfico se clasifican en dos grupos: intrusivas y no intrusivas. Las intrusivas implican una modificación física de la vía que monitoriza y esta característica se convierte en su principal inconveniente, puesto que la vía debe ser cerrada para su instalación además de un posterior manteninimiento. En cualquier caso estas técnicas sólo cuentan vehículos que atraviesan el sistema de monitorización, sin identificarlos de ninguna forma. En este grupo se incluyen por ejemplo los sensores piezoelectricos o los tubos neumáticos. Las tecnologías no intrusivas no producen interrupción del tráfico, puesto que su colocación no está físicamente en el pavimento, sino en zonas adyacentes de la vía como paneles informativos, carteles de señalización, puestos de emergencias, etc. En este último grupo se incluyen los radares, la señalización lumínica o las cámaras de monitorización.

El inconveniente principal de este sistema de monitorización es que no permiten identificar los vehículos de forma unívoca para poder construir patrones de desplazamiento o matrices de frecuencia de destinos para un conjunto de orígenes dados.

Respecto al estado del arte de los algoritmos de predicción, existen muchos trabajos relacionados con el tema y que han sido ampliamente utilizados con con-

juntos de datos públicos con los que han sido testeados por sus desarrolladores. Todos los algoritmos de predicción intentan modelar un sistema numérico que represente el comportamiento de la serie y con el pueda predecirla. Estos modelos pueden ser lineales o no lineales. Dentro del grupo de los lineales hay varios, pero el más conocido es el llamado método ARIMA [3]. ARIMA es un modelo simple y cuyo funcionamiento está ampliamente demostrado, pero no funciona muy bien con muchas aplicaciones a series temporales reales, puesto que su ajuste no es fácil y lo debe de realizar operarios especializados. Esta es la principal razón de que surgieran los métodos no lineales.

Los modelos no lineales se ajustan mejor a aplicaciones reales aunque la facilidad de uso sigue siendo escasa [6], de ahí que existan propuestas donde se recomienda que se le debe dar prioridad a la facilidad de uso frente a otros aspectos más técnicos [4].

La predicción de series temporales también se ha abordado desde otros campos, como técnicas de Soft-Computing, [7], [8]; Técnicas Fuzzy [9], [10]; redes neuronales, [11]; Regresión [12]; y sistemas expertos [13]. En nuestro caso nos hemos centrado en métodos no lineales basados en la utilización de Redes Neuronales, puesto que tal y como afirma Tang [14] son métodos que ofrecen una mayor flexibilidad y efectividad a la hora de predecir una serie temporal real.

En concreto el algoritmo desarrollado, L-Co-R [2] utiliza Redes Neuronales de Base Radial (RBFNs). L-Co-R [2] presenta varias ventajas con respecto a otros métodos[4]. La princial es que coevoluciona los datos de entrada de la serie (lag) junto con la red neuronal que se va a utilizar para predecir. Este enfoque coevolutivo no ha sido desarrollado en la literatura, puesto que otros autores coevolucinan las diferentes partes de la red o simplemente no utilizan el enfoque coevolutivo a la predicción sino a otro tipo de problemas como la clasificación. Aún así, si que existen autores que utilizan el mismo tipo de redes de forma no coevolutiva como [16,17,18] que aplican la predicción al precio del aceite de oliva.

Para realizar la predicción se utilizan periodos de tiempo de la serie de tamaño finito llamados "lags". Los lags deben representar perfectamente el comportamiento de la serie y su relación y según el teorema de Takens [19] estableciendo correctamente estos lags, el comportamiento del espacio multidimensional construido a partir de dicho lag, puede representar de forma correcta el comportamiento de la serie temporal real.

Existen métodos automáticos que intentan determinar el lag más adecuado como pueden ser el algoritmo QIEHI [20], [21]. Otros autores han ido más allá, buscando no sólo el lag, sino las características externas que afectan a la serie y que con un preprocesamiento previo de la serie temporal, puede desembocar en la elección de un lag más adecuado para la predicción, [22]. También se ha realizado este estudio después de la predicción, [23], incluyendo un post procesamiento de los datos que modifica el lag de salida del método.

[4] Estas ventajas y el algoritmo al completo están detallados en la referencia anterior y no se detallan en esta publicación por falta de espacio

En la propuesta que se presenta en este artículo, se utiliza para la predicción un método no lineal que utiliza redes de base radial (RBFs) para determinar de forma cooperativa el lag y la predicción de la serie temporal a la que se aplique el algoritmo, si necesidad de ningún pre o pro procesamiento.

3. Experimentos y resultados

En esta sección se analizan los datos recolectados y se obtienen los resultados estadísticos iniciales a cerca de la predicción realizada.

Debido al gran número de métodos existentes para realizar predicción, hemos seleccionado sólo 6 de ellos para poder realizar comparaciones entre los diferentes resultados. En esta selección se han incluido métodos lineales y no lineales. En concreto la selección es ARIMA, Croston, Theta, Spline y L-Co-R y como método de control hemos seleccionado la media. Todos los métodos excepto L-Co-R están incluidos dentro del paquete estadístico R y son ampliamente conocidos y esta es la razón de por qué han sido seleccionados, puesto que así cualquier otro investigador puede comprobar sus resultados si dispone de R. L-Co-R [2] es un algoritmo coevolutivo desarrollado que ha generado resultados muy buenos dentro de la predicción a corto plazo. Se trata de un método no lineal basado en evolucionar RBFs y Lags simultáneamente para adecuar la red al lag elegido y el lag elegido a la red que el método construye.

Los experimentos se han realizado utilizando las cinco series de datos descritas en la sección anterior. Los primeros 53 días de la serie se han utilizado para entrenar los diferentes métodos y poder construir el modelo predictivo, mientras que los 7 días restantes se han utilizado para testear la predicción generada. La predicción se genera con un horizonte de 1, es decir conocidos los datos de la serie con n elementos, el algoritmo predice el elemento $n + 1$.

Para cada método y cada serie, se han calculado una serie de medidas de error para poder comparar la predicción de los diferentes métodos:

- *Mean Absolute Error (MAE):*

$$MAE = mean(\mid e_t \mid) \tag{1}$$

- *Mean Absolute Percentage Error (MAPE):*

$$MAPE = mean(\mid p_t \mid) \tag{2}$$

- *Median Absolute Percentage Error (MdAPE):*

$$MdAPE = median(\mid p_t \mid) \tag{3}$$

- *Mean Absolute Scaled Error (MASE):*

$$MASE = mean(\mid q_t \mid) \tag{4}$$

- *Mean Squared Error (MSE):*

$$MSE = \frac{1}{n}\sum_{i=1}^{n} {e_t}^2 \tag{5}$$

donde Y_t *es el dato observado en tiempo* $t = 1, ..., n$*;* F_t *es la predicción de* Y_t*;* e_t *es el error de predicción (i.e.* $e_t = Y_t - F_t$*);* $p_t = 100e_t/Y_t$ *es el porcentaje de error y* $q_t = \dfrac{e_t}{\dfrac{1}{n-1}\sum_{i=2}^{n} \mid Y_i - Y_{i-1} \mid}$

Los resultados mostrados del algoritmo L-Co-R son medias de error de 30 ejecuciones con cada una de las series temporales estudiadas

La tabla 1 muestra los diferentes valores de error para cada uno de los métodos y para cada serie. Destacados en el texto aparecen el mejor (menor) valor obtenido para cada nodo y cada medida de error. Tal y como se puede apreciar en la tabla, la tendencia es siempre que el mejor algoritmo es justo el algoritmo L-Co-R puesto que obtiene los menores errores para las medidas MAE, MAPE, MdAPE y MASE (excepto para el nodo 2), aunque para otras medidas como MSE no hay un claro algoritmo ganador.

Los resultados de cada método se han comparado utilizando dos medidas no paramétricas: El test de Friedman y el test de Iman-Davenport mediante las cuales podemos determinar si existen diferencias significativas entre los métodos o no, estableciendo así un ranking de métodos de mejor a peor en el proceso de predicción. Los resultados revelan que existen diferencias significativas cuando el valor $P - value$ es mejor que 0.05.

El ranking formado por los valores resultados de los test no-paramétricos, determinan cual es el mejor método. Una vez que este método se ha encontrado, se puede comparar de nuevo con el método de Holm [25]. En la tabla 3 se resumen los resultados así como la existencia de diferencias significativas entre los métodos. En este caso, los resultados muestran que el algoritmo L-Co-R es el mejor para cuatro de las medidas de error utilizadas (MAE, MAPE, MASE, y MdAPE), aunque no se puede realizar esta afirmación si lo comparamos con el método ARIMA, donde no se encuentran para estas series, diferencias significativas en los resultados de predicción.

4. Conclusiones y trabajo futuro

En este trabajo se presenta la utilización de un dispositivo de captación de emisiones BT para recopilar series temporales relacionadas con el flujo del tráfico en cualquier tipo de vía. Concretamente se han monitorizado 6 vías obteniendo las correspondientes series para cada uno de los dispositivos (Nodo 1-5).

Utilizando estas series temporales reales que incluyen el número de dispositivos captados por día, se ha realizado un estudio de las series para predecir a corto plazo (1 día) cual será el flujo de tráfico en cada uno de los nodos. Se han

Nodo 1	*MAE*	*MAPE*	*MASE*	*MdAPE*	*MSE*
ARIMA	514,83	40,42	1,14	30,76	424860,5
CROSTON	509,91	39,28	1,13	33,8	387403,9
THETA	528,59	41,16	1,17	32,78	443898,5
SPLINE	550,02	43,87	1,21	30,1	488504,9
MEAN	456,95	38,57	1,01	26,11	318623,5
L-Co-R	**372,77**	**23,66**	**0,81**	**18,46**	**231731,8**
Nodo 2	*MAE*	*MAPE*	*MASE*	*MdAPE*	*MSE*
ARIMA	433,52	26,64	**1,03**	22,38	**310105,3**
CROSTON	532,87	27,98	1,26	27,89	398979,9
THETA	435,58	26,77	1,03	22,55	312496,1
SPLINE	535,86	34,53	1,27	24,7	447557,4
MEAN	489,57	24,44	1,16	28,64	361070,5
L-Co-R	**429,78**	**11,41**	1,04	**11,98**	311561,7
Nodo 3	*MAE*	*MAPE*	*MASE*	*MdAPE*	*MSE*
ARIMA	373,17	35,11	1,32	27,54	**150293,6**
CROSTON	377,42	37,03	1,34	27,89	160970,9
THETA	394,95	38,11	1,4	29,41	175710,7
SPLINE	384,68	38,86	1,36	29,36	184807,2
MEAN	373,18	35,11	1,32	27,53	150307,8
L-Co-R	**363,3**	**10,56**	**0,84**	**17,01**	182469,9
Nodo 4	*MAE*	*MAPE*	*MASE*	*MdAPE*	*MSE*
ARIMA	3168,36	40,62	1,04	18,99	21792093
CROSTON	3789,4	56,83	1,25	54,33	25037291
THETA	3085,3	38,14	1,01	25,42	21510757
SPLINE	3758,06	39,45	1,24	43,53	28745309
MEAN	4545,17	92,73	1,5	40,07	26612429
L-Co-R	**540,59**	**6,26**	**0,07**	**4,14**	**1120317,2**
Nodo 5	*MAE*	*MAPE*	*MASE*	*MdAPE*	*MSE*
ARIMA	5012,95	43,8	1,13	20,19	**29974781**
CROSTON	5583,14	46,32	1,26	26,12	35082013
THETA	5779,37	45,78	1,3	28,25	35859271
SPLINE	5878,35	46,26	1,32	28,69	37288518
MEAN	5013,48	43,81	1,13	20,19	29980578
L-Co-R	**4978,08**	**24,59**	**0,87**	**9,37**	35875329

Cuadro 1: Valores obtenidos para las cinco medidas de error (MAE, MAPE, MASE, MdAPE y MSE) para los métodos de predicción seleccionados (ARIMA, Croston, Theta, Spline, Mean y L-Co-R) para todos los nodos. En negrita se resalta el mejor resultado para cada caso.

seleccionado 5 métodos de predicción con los que se han calculado 5 medidas de error.

Los resultados obtenidos demuestran que la predicción con el algoritmo L-Co-R a corto plazo es significativamente mejor que las predicciones obtenidas

Error	*Friedman P-value*	*Iman-Davenport P-value*
MAE	0.0052	0.0003
MAPE	0.0080	0.0008
MASE	0.0141	0.0030
MdAPE	0.0063	0.0005
MSE	0.0213	0.0066

Cuadro 2: Resultado del valor *P-value* para los test de Friedman y Iman-Davenport para cada medida de error. Un valor de *P-value* menor que 0.05 (todos en este caso) indica que existen diferencias significativas entre los errores calculados para cada método.

con el resto de los métodos para todas las medidas de error seleccionadas menos una.

Este trabajo abre un abanico de posibilidades de análisis de la gran cantidad de datos captados y la posibilidad de desarrollar métodos de predicción no sólo a corto plazo, sino a medio y largo plazo, lo que facilitará la tarea de previsión del flujo de tráfico en las vías monitorizadas y por consiguiente una mejor gestión del tráfico en dichas vías por parte de las autoridades competentes.

Este trabajo se está desarrollando gracias a la financiación de los proyectos TIC-3903, P08-TIC-3928, CEI2013-P-14, TIN2011-28627-C04-02, y del proyecto FEDER de la Unión Europea con título "Sistema de Información y Predicción de bajo coste y autónomo para conocer el Estado de las Carreteras en tiempo real mediante dispositivos distribuidos" (SIPEsCa) del Programa Operativo FEDER de Andalucía 2007-2013. Asimismo, queremos mostrar nuestro agradecimiento al personal e investigadores de la Agencia de Obra Pública de la Junta de Andalucía, Consejería de Fomento y Vivienda, por su dedicación y profesionalidad.

Referencias

1. Dirección General de Tráfico. Evolución 1970-2012. Ministerio de Fomento, Mayo 2013 http://www.fomento.gob.es/MFOM/LANG_CASTELLANO/DIRECCIONES_GENERALES/CARRETERAS/CATYEVO_RED_CARRETERAS/
2. Elisabet Parras-Gutierrez, M. G. Arenas, V. M. Rivas, M. J. del Jesus, Coevolution of lags and RBFNs for time series forecasting: L-Co-R algorithm, Soft Computing 16(6), (2012) 919-942.

Error	*Método de Control*	*Ranking*	*Método*	*p*	*Hipótesis nula*
MAE	L-Co-R	1	ARIMA	0.176	Aceptada
		2	MEAN	0.028	**Rechazada**
		3	THETA	0.007	**Rechazada**
		4	CROSTON	0.007	**Rechazada**
		5	SPLINE	2.003E-4	**Rechazada**
MAPE	L-Co-R	1	ARIMA	0.090	Aceptada
		2	MEAN	0.0.063	Aceptado
		3	THETA	0.011	**Rechazada**
		4	CROSTON	0.002	**Rechazada**
		5	SPLINE	3.857E-4	**Rechazada**
MASE	L-Co-R	1	ARIMA	0.398	Aceptada
		2	MEAN	0.063	Aceptado
		3	THETA	0.028	**Rechazada**
		4	CROSTON	0.018	**Rechazada**
		5	SPLINE	7.232E-4	**Rechazada**
MdAPE	L-Co-R	1	ARIMA	0.176	Aceptada
		2	MEAN	0.042	**Rechazada**
		3	THETA	0.004	**Rechazada**
		4	SPLINE	0.002	**Rechazada**
		5	CROSTON	7.232E-4	**Rechazada**
MSE	ARIMA	1	L-Co-R	0.500	Aceptada
		2	MEAN	0.398	Aceptado
		3	CROSTON	0.176	Aceptado
		4	THETA	0.176	Aceptado
		5	SPLINE	7.232E-4	**Rechazada**
SMAPE	ARIMA	1	L-Co-R	0.866	Aceptada
		2	MEAN	0.310	Aceptado
		3	CROSTON	0.091	Aceptado
		4	THETA	0.043	**Rechazada**
		5	SPLINE	0.006	**Rechazada**

Cuadro 3: Resultados del test de Holm para cada medida de error. Valores de p menores o iguales a 0.05 indican que los resultados de error calculados por el algoritmo de control y los demás métodos, se pueden considerar significativamente diferentes.

3. G. Box, G. Jenkins, Time series analysis: forecasting and control, San Francisco: Holden Day, 1976.
4. J. De Gooijer, R. Hyndman, 25 years of time series forecasting, International Journal of Forecasting 22 (3) (2006) 443–473.
5. R. Fildes, K. Nikolopoulos, S. Crone, A. Syntetos, Forecasting and operational research: a review, Journal of the Operational Research Society 59 (2008) 1150–1172.
6. M. Clements, P. Franses, N. Swanson, Forecasting economic and financial time-series with non-linear models, International Journal of Forecasting 20 (2) (2004) 169–183.
7. B. Samanta, Prediction of chaotic time series using computational intelligence, Expert Systems with Applications 38 (9) (2011) 11406–11411.

8. S. Zhu, J. Wang, W. Zhao, J. Wang, A seasonal hybrid procedure for electricity demand forecasting in china, Applied Energy 88 (11) (2011) 3807–3815.
9. W. Qiu, X. Liu, H. Li, A generalized method for forecasting based on fuzzy time series, Expert Systems with Applications 38 (8) (2011) 10446–10453.
10. C. Wang, A comparison study between fuzzy time series model and arima model for forecasting taiwan export, Expert Systems with Applications 38 (8) (2011) 9296–9304.
11. T. Yu, K. Huarng, A neural network-based fuzzy time series model to improve forecasting, Expert Systems with Applications 37 (4) (2010) 3366–3372.
12. K. Kavaklioglu, Modeling and prediction of turkey's electricity consumption using support vector regression, Applied Energy 88 (1) (2011) 368–375.
13. P. K. Dash, A. C. Liew, S. Rahman, G. Ramakrishna, Building a fuzzy expert system for electric load forecasting using a hybrid neural network, Expert Systems with Applications 9 (3) (1995) 407–421.
14. Z. Tang, C. de Almeida, P. Fishwick, Time series forecasting using neural networks vs. box-jenkins methodology, Simulation 57 (5) (1991) 303–310.
15. V. Rivas, J. Merelo, P. Castillo, M. Arenas, J. Castellano, Evolving rbf neural networks for time-series forecasting with evrbf, Information Sciences 165 (3-4) (2004) 207 – 220.
16. H. Hippert, J. Taylor, An evaluation of bayesian techniques for controlling model complexity and selecting inputs in a neural network for short-term load forecasting, Neural Networks 23 (3) (2010) 386–395.
17. C. Lee, C. Ko, Time series prediction using rbf neural networks with a nonlinear time-varying evolution pso algorithm, Neurocomputing 73 (1-3) (2009) 449–460.
18. M. Perez-Godoy, P. Pérez-Recuerda, M. Frías, A. Rivera, C. Carmona, M. Parras, Co^2rbfn for short and medium term forecasting of the extra-virgin olive oil price, in: J. González, D. Pelta, C. Cruz, G. Terrazas, N. Krasnogor (Eds.), Proceedings of Nature Inspired Cooperative Strategies for Optimization, Vol. 284, Springer Berlin/Heidelberg, 2010, pp. 113–125.
19. F. Takens, Dynamical Systems and Turbulence, Lecture Notes In Mathematics, Vol. 898, Springer, New York, NY, 1980, Ch. Detecting strange attractor in turbulence, pp. 366–381.
20. R. Araújo, A quantum-inspired evolutionary hybrid intelligent apporach fo stock market prediction, International Jorunal of Intelligent Computing and Cybernetics 3 (10) (2010) 24–54.
21. R. Araújo, Hybrid intelligent methodology to design translation invariant morphological operators for brazilian stock market prediction, Neural Networks 23 (10) (2010) 1238–1251.
22. R. García-Pajares, J. Benitez, G. Sainz Palmero, Feature selection form time series forecasting: a case study, in: Proceedings of 8th International Conference on Hybrid Intelligent Systems, 2008, pp. 555–560.
23. A. Maus, J. C. Sprott, Neural network method for determining embedding dimension of a time series, Communications in Nonlinear Science and Numerical Simulation 16 (8) (2011) 3294–3302.
24. R. Fildes, The evaluation of extrapolative forecasting methods, International Journal of Forecasting 8 (1) (1992) 81–98.
25. S. Holm, A simple sequentially rejective multiple test procedure, Scandinavian Journal of Statistics 6 (2) (1979) 65–70.

Sistemas de trading en la recesión económica: gramáticas evolutivas, relación fitness-objetivos, y sectores empresariales

Iván Contreras[1], J. Ignacio Hidalgo[1], and Laura Núñez-Letamendia[2] F.Javier Diego[3]

[1] Facultad de Informática, Universidad Complutense de Madrid, ivancontreras@pas.ucm.es,hidalgo@dacya.ucm.es
[2] IE Business School, Madrid Laura.Nunez@ie.edu
[3] Universidad Politécnica de Madrid javier.diego@upm.com

Resumen Los Sistemas de Trading son una herramienta fundamental en los mercados de inversión actuales. Estos sistemas se han afianzado por dos razones principales: Los mercados son influidos por una gran cantidad de variables (desastres naturales, políticas de gobierno, etc.) y los inversores pueden ser afectados por una gran cantidad de factores emocionales (psicología de mercado). Los Sistemas de Trading permiten el manejo de grandes volúmenes de datos y capaces de abstraerse de la psicología físico-emocional que afecta a los inversionistas. En este trabajo implementamos un Sistema de Trading bassado en gramáticas evolutivas que es validado en la actual recesión económica. Utilizaremos tres diferentes funciones de fitness que nos proporcionaran cinco versionnes del Sistemas de Trading diferentes, tres de ellos mono-objetivo y 2 multi-objetivo. Obtendremos reglas que nos proporcionan series de señales de inversión que validaremos en el 2012 donde analizaremos los diferentes rendimientos obtenidos. Además se realiza un análisis sectorial de las compañias y se correlacciona a las funciones objetivo con los diferentes sectores de actividad económica.

1. Gramáticas evolutivas y Sistemas de Trading

Los mercados de inversión son sistemas dependientes de infinidad de variables, en los cuales una pequeña variación puede cambiar drasticamente los movimientos de una activo bursátil. A raiz de esta complejidad y la intención de evitar la psicología de mercado sugen los Sistemas de Trading (STs). Un ST es un conjunto de reglas específicas, o parámetros, que determinan la entrada y salida de un activo en bolsa. Los STs se diseñan como sistemas de optimización que inspeccionan un complejo espacio de búsqueda en busca de las series optimas de toma de decisiones de inversión. Este entorno adverso sugiere a las metaheurísticas como una de las mejores soluciones para encontrar buenas soluciones en un corto periodo de tiempo. La aplicación de metaheurísticas en ha tenido un rápido desarrollo tanto en el mundo científico como en el profesional. Como consecuencia de este desarrollo, muchos sistemas de inversión se han

visto beneficiados por el apoyo de complejos sistemas de computación capaces de analizar grandes cantidades de datos e información. La literatura en torno a las metaheurísticas es amplia, nosotros nos centraremos en una de las ramas más conocidas y con más éxito entre este tipo de métodos: las metaheurísticas evolutivas.

El actual estado del arte contiene multitud de trabajos que implementan Algoritmos Evolutivos (AEs) como motor de los STs, que por falta de espacio es imposible citar. Debido a la antiguedad de la mayoría de las bases de datos y la actual recesión económica, muchos de estos trabajos no son extensibles a la actual situación socio-económica. Como alternativa a los métodos basados en AGs surgen las Gramáticas Evolutivas (GEs) [13], que se presentan como una limpia y eficiente solución que encaja adecuadamente en el contexto de la complejidad de los mercados financieros. Las GEs son una metodología evolutiva relativamente nueva dentro de las metaheurísticas, y que tiene la suficiente identidad con respecto a las propuestas anteriores [13] para ser una seria competencia. Resumidamente se podría definir como un tipo de algoritmo evolutivo diseñado para evolucionar programas informáticos definidos por una gramática, por lo general en notación Backus–Naur Form (BNF). La metodología más similar es la Programación Genética (PG), que también es capaz de evolucionar programas de computadoras. La disimilitud principal que hacen de las GEs una solución atractiva y elegante es la diferencia entre el fenotipo y el genotipo. Así, en lugar de representar los programas en estructura arbórea, las GEs presentan un cromosoma compuesto por tokens que conectan directamente con una norma especificada de la gramática. El cromosoma en sí se considera el genotipo y el código real derivado de los tokens se llama fenotipo. El enfoque que ofrecen las GEs se convierte en un método atractivo gracias a su flexibilidad. Esta característica es la principal ventaja de esta metodología y está estrechamente correlacionada con la gran modularidad que una gramática bien estructurado proporciona.

A pesar de ser una metodología relativamente nueva, existe una amplia cantidad de trabajos de investigación. La universidad de Limerick a través del grupo de BDS ha realizado un excelente trabajo en torno a las GEs. Por ejemplo, en [8] los autores proponen el uso de una gramática adaptativa a las series de datos analizadas, de manera que las reglas de la gramática están en constante evolución durante la ejecución del ST y la llegada de nuevos datos. Otros trabajos de este grupo se centran en evaluar los beneficios posibles en mercados de cambio de divisa o en diferentes indices bursátiles como en [3] [9] entre otros. Otros autores también han propuesto STs basados en GEs, como en [1], donde el autor implementa un sistema que usa la co-evolución de reglas de entrada, de salida y de stoploss para movimientos de venta y compra. En [2] se actualiza este trabajo usando una función de fitness mas compleja propuesta por P.Saks en [15]. Todos estos trabajos han inspirado una nueva implementación para nuestro ST que presentaremos más adelante.

La aplicación de las GEs no se limita al ambito de los mercados bursátiles, estas han sido igualmente aplicadas fructuosamente en otros campos. Por ejemplo, en

[11] se utilizan para la generación de modelos óptimos de redes bioquímicas , en [14] son aplicadas a la resolución de identidades trigonométricas, e incluso para la composición automática de obras musicales [6].
El resumen anterior, ofrece algunas de las propuestas que podemos encontrar en la literatura de GEs y STs que se han utilizado hasta la fecha actual. El contexto de este trabajo no recae en una metodología innovadora correlacionada con las GEs, el principal objetivo ha sido analizar los resultados que ofrece un ST en un período de recesión de la económica y proponer un método que funcione correctamente en un ambiente hostil como en el que nos encontramos actualmente.
En [5] se analizo una pequeña muestra de empresas españolas con un ST basado en GE y guiado por una función de fitness (rendimiento acumulado). En este trabajo analizaremos los resultados que nos proporcionara un ST guiado por diferentes funciones de fitness. Por una parte utilizaremos las funciones de fitness independientemente unas de otras, analizando así varias versiones del ST. Más tarde combinaremos las mismas funciones de fitness para analizar el ST como un sistema multiobjetivo. Además en [5] el análisis de las empresas relacionadas con la construcción provocaba un empeoramiento generalizado de los rendimientos, por lo que en este trabajo analizaremos nuestro conjunto de empresas en función del area en el que la empresa desarrolla sus actividades. Este trabajo establecerá uno de los puntos de partida para un ST capaz de realizar un análisis exhaustivo de grandes regiones en busca de las compañias con más rendimiento.
Las aportaciones de este artículo son dos:

1. Expandir los resultados del trabajo presentado anteriormente [5]. Ampliando numero de empresas analizadas y funciones de fitness, incluyendo el análisis de STs multiobjetivo.
2. Las empresas que cotizan en la bolsa son diferentes, cada una tienen una rentabilidad y riesgo. Proponemos y analizamos la evaluacion de nuestro conjunto de empresas por los sectores en las que estas trabajan.

La organización de las siguientes secciones del artículo se presenta a continuación: La Sección 2 muestra la implementación de los principales componenetes de nuestro AE. En la Sección 2 se presenta la metodología del ST para la obtención de las reglas de inversión. La Sección 3 presenta un análisis de los resultados experimentales de los diferentes STs. Finalmente, concluimos el artículo con las conclusiones y algunas lineas de trabajo futuro en la Sección 4.

2. Implementación y metodología del ST basado en GE

Durante los últimos años las GEs se están asentando como una poderosa herramienta dentro de los AEs. Esta técnica de computación evolutiva fue promovida por C.Ryan, JJ. Collins and M. O'Neill en 1998 [13]. Como comentamos en la introducción, las GEs se diseñaron para conseguir una buena solución mediante la evolución (basada en grámaticas) de código de programación. En nuestro ST, la solución consiste en encontrar una serie de movimientos de inversión que nos proporcione una cantidad de beneficio favorable. La gramática esta compuesta

por una serie de reglas basadas en ITBs y como todo AE, estará guiada por una función de fitness que evaluará la bondad de las soluciones generadas.

2.1. Funciones de fitness

En este trabajo utilizamos varias funciones de Fitness que evaluaremos, individualmente y combinadamente, para poder comparar los resultados de rendimiento entre ellas. Las funciones de fitness utilizadas son el Rendimiento Acumulado (RA), el Indice de Sharpe (IS) y el Coeficiente de Correlación entre la curva de inversión Perfecta y la Real (CCPR).
Cada una de nuestras inversiones estará representada por una señal que nuestra función de fitness evaluará (comprar, vender o nula). La primera función que guiará nuestro sistema de trading es el rendimiento acumulado de las inversiones:

$$RA = \prod_{i=1}^{n}(1 + R_i) \tag{1}$$

La segunda funcción de fitness evalúa ganancias y riesgos en un solo valor, bien conocido como el Indice de Sharpe:

$$IS = \frac{RA - R_f}{\sigma^{(R_i - R_f)/|(R_i - R_f)|}} \tag{2}$$

Por ultimo utilizaremos el indice de correlación entre la inversión perfecta y la evaluada:

$$CCPR_i = \sum_{i=1}^{n}(\frac{(X_i - \overline{X})(Y_i - \overline{Y})}{((n-1)(\sigma_X)(\sigma_Y))} \tag{3}$$

Donde R_i denota el rendimiento del dia i como sigue:

$$R_i = \begin{cases} \frac{P_i - P_{i-1}}{P_{i-1}} & \text{Si el ST da una señal de compra} \\ \frac{-(P_i - P_{i-1})}{P_{i-1}} & \text{Si el ST da una señal de venta} \\ \emptyset & \text{Si la señal del ST es neutral} \end{cases} \tag{4}$$

Y donde:
- R_f es el rendimiento libre de riesgo.
- σ es la desviación estándar
- X es la curva de ganancias perfecta.
- Y es la curva de ganancias real
- P_i es el precio de cierre en el día i.

2.2. Metodología del ST

El ST basado en GE esta dirigido principalmente por seis ITBs y seis pesos correlacionados con cada indicador (W_1, W_2, W_3, W_4, W_5 and W_6). Para este trabajo hemos seleccionado los siguientes ITBs: El Cruce de Medias Moviles (MAC), el Volumen-Precio (VP), la Media Movil de Convergencia-Divergencia

(MACD), los Soportes y Resistencias (SR) y el Indice de Fuerza Relativa (RSI). Cabe denotar que el RSI se utiliza como 2 indicadores separados, por un lado para identificar las divergencias que señalan los cambios de tendencias(RSId) y por otro lado para identificar los niveles de sobre-compra y sobre-venta (RSIo). Cada uno de estos indicadores proporcionan una señal de compra, venta o neutral. Mediante los pesos correlacionados a cada indicador se atribuye un peso especifico a cada ITB, de manera que la importancia de las señales obtenidas por cada indicador se vera modificada por estos pesos que ponderaran la señal. Los indicadores han sido seleccionados por su utilidad en el mundo profesional y académico de las finanzas. Si el lector esta interesado en obtener descripciones completas de los ITBs citados, pueden acudir a [4] donde se podrán encontrar amplias explicaciones de estos. La combinación de las señales de los indicadores y reglas que se construyan como solución a partir de la gramática proporcionarán un valor para cada día de inversión. Este valor se comparará con unos umbrales que se han debido establecer previamente en el ST, los umbrales de compra y venta. Estos umbrales pueden variar entre 0 y 6/-6, esta representación proporciona un sistema agresivo en los valores extremos y uno conservador en los valores medios. De esta forma, si este valor excede al umbral de compra, preestablecido por el usuario, el sistema dará una señal de compra. En caso contrario, si este valor es menor que el umbral de venta el sistema dará una señal de venta. Si el valor se encuentra entre los dos valores umbrales, el sistema proporcionará una señal neutral. El sistema proporcionará así, señales de inversión a corto o a largo.

El proceso de trabajo que establece nuestro ST se puede resumir de la siguiente forma:

1. El inversor selecciona el conjunto de ITBs
2. Se establecen los umbrales de compra y venta (bajos= agresivos, altos=conservadores)
3. Para cada compañia i
   a) Aplicar el ST basado en GE a los datos *insample* para obtener una solución A
   b) Aplicar la solución A, es decir el conjunto de reglas, a los datos *outsample*:
      1) Si $A_j > Umbral_{compra}$ Compra.
      2) elseif $A_j < Umbral_{venta}$ Vende.
      3) else Neutral, no emite señal.
   c) Calcular el rendimiento dado por los movimientos de compra/venta.

2.3. Implementación de la GE

Usamos una codificación entera para el genotipo con una implementación clásica de los operadores del algoritmo evolutivo. Generamos la descendencia mediante el operador de cruce (probabilidad de 0.8), que como demostró O'Neill en [12] es crucial para el correcto desarrollo para el algoritmo. La implementación de este operador es la cruza por un punto, que ya ha demostrado ser un operador exitoso y capaz de conseguir buenos resultados en el proceso de intercambiar

bloques de información genética. La mutación se implementa mediante la conocida *Integer Flip Mutation* con una probabilidad de 0.02. La implementación incluye el operador de *wrapping*. Este operador permite reutilizar el cromosoma para obtener estructuras de reglas mas amplias influyendo asi en la calidad y diversidad de los individuos generados como muestra Huggoson en los resultados experimentales de [10]. Como se ha explicado anteriormente, las GEs usan gramáticas para construir un conjunto de reglas que guiaran el ST en la busqueda de series optimas de inversión. En este trabajo se han utilizado varias gramáticas y por simplicidad exponemos como ejemplo la versión reducida de la gramática testeada:

```
<codigo>      ::=  <codigo><indicador> | <indicador>
<indicator> ::=  <MAC> | <MACD> | <RSId> | <RSIo>| <SR> | <VP>

<MAC>   ::=   "MAC("<Param1>","<Param2>","<Peso>");"
<MACD> ::=   "MACD("<Param3>","<Param4>","<Param5>","<Peso>");"
<VP>  ::=  "VP("<Param6>","<Peso>");"
<SR>   ::=   "SR("<Param7>","<Param8>","<Param9>");"
<RSIo> ::=   "RSI(false,"<Param10>","<Param11>","<Param12>","<Peso>");"
<RSId> ::=   "RSI(true," <Param13> ","<Param14>","<Peso> ");"
```

* Parámetros (ParamX) y pesos (Peso) son números aleatorios en cierto rango.

3. Resultados Experimentales

3.1. Validación y experimentación

La funcionalidad de un ST es generar series de toma de decisiones de inversión a través del análisis de datos históricos. La selección de estos datos es una importante decisión en el proceso de desarrollo del sistema. La calidad de los resultados estará estrechamente correlacionada con la estabilidad y rentabilidad de las distintas tendencias del mercado elegido. La importancia de los datos históricos en estos sistemas provoca que obtener ganancias en un periodo de crecimiento generalizado o estabilidad económica, por regla general, es más sencillo que hacerlo en un mercado en declive o inestable. Por lo tanto, el ya excitante desafío de desarrollar un TS capaz de predecir el intrincado comportamiento de la bolsa se convierte en un desafío aún más complicado e interesante en un entorno económico hostil. Los datos utilizados para el analisis (in-sample) se comprenden entre 2001 y 2011, y los resultados de prueba para validar el funcionamiento del ST (out-sample) comprenden el año 2012.

De esta forma, nosotros validaremos el ST con datos actuales en uno de los crack económicos más importantes, la actual recesión económica. Los datos de precios diarios de cierre y volumenes son pertenecientes a cuatro importantes países de la UE: España, Francia, Alemania y Reino Unido. El número de compañias usadas en los experimentos es treinta y seis, y son seleccionadas aleatoriamente de la base de datos por pais. Las compañias que forman nuestra base

de datos han sido listadas mediante Orbis y escogidas mediante los siguientes criterios: Compañias activas con al menos 5 años de datos históricos en la bolsa (Excluyendo financieras); Y su actual capitalización de mercado por encima de 25.000.000 euros.

3.2. Resultados experimentales

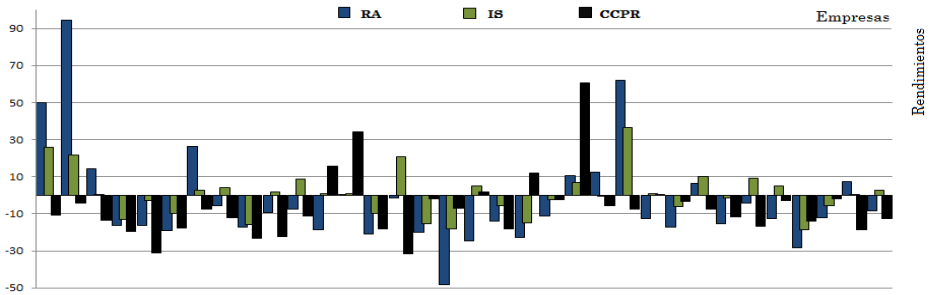


Figura 1: Rendimientos proporcionados por los 3 STs, uno por cada función de fitness, de las 36 empresas .

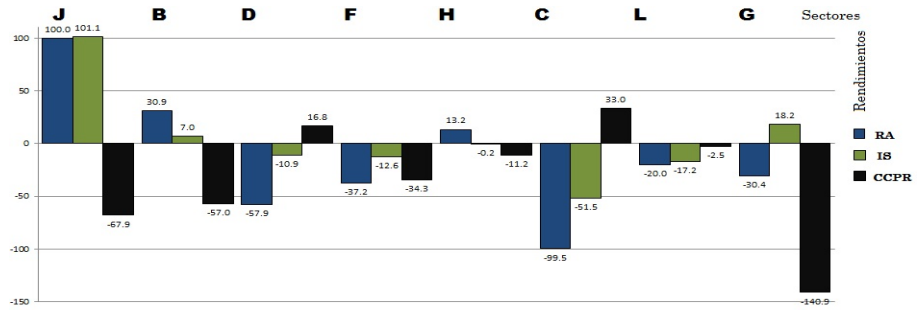


Figura 2: Rendimientos proporcionados por los 3 STs, uno por cada función de fitness, de los 8 sectores.

La figura 1 muestra los resultados de las inversiones en las 36 empresas listadas en los países de Alemania, Reino Unido, España y Francia en el año 2012. El eje Y representa el rendimiento y el X las diferentes empresas. Cada empresa consta de tres resultados, uno por cda función de fitness: el RA, el IS y el CCPR. La gráfica muestra perdidas y ganancias en diferentes empresas, si se decidiese invertir en una cartera con todas estas empresas, los rendimientos acumulados para cada versión del ST serían: un -100 % para el RA, un 25 % para el IS y un

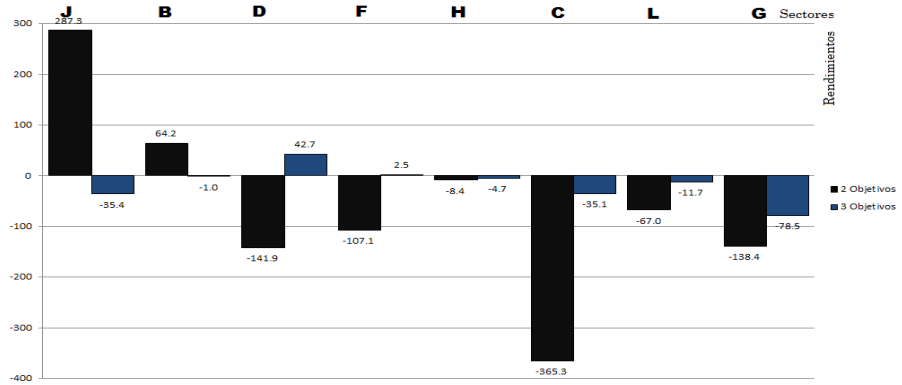


Figura 3: Rendimientos proporcionados por los dos STs multi-objetivo de los 8 sectores.

-229 % para el CCPR. La más simple conclusión de estos resultados es no utilizar el RA ni el CCPR, y utilizar siempre el IS como función de fitness de nuestro AE. Sin embargo, un estudio detallado de la gráfica hace entrever un desarrollo más complejo de esta idea. En terminos generales el IS es él que más se eleva en rendimientos, pero para algunas empresas se observan mejores resultados con los STs basados en las funciones de RA ó CCPR, ya sea por que genera más ganancias o por el contrario por que genera menos perdidas. Esta observación nos hace cuestionarnos el comportamiento selectivo que el ST muestra con diferentes empresas dependiendo de la función de fitness utilizada en el AE.

La bolsa esta formada por multitud de compañias ocupadas en las actividades más dispares. Los acontecimientos como la actual recesión, el "boom"del ladrillo, el encarecimiento del petroleo, las sequías, etc afectan de una forma diferente a cada una de ellas. Sin embargo, existen empresas que trabajan más correlacionadas unas con otras, es decir, las tendencias y comportamientos de BMW y Volkswagen son más dependientes unos de otros, que los que podría tener la misma BMW con Telefónica. Por este motivo, en este trabajo vamos a realizar el mismo experimento pero dividiendo las empresas en diferentes sectores según la clasificación NACE Rev.2. La Nomenclatura estadística de actividades económicas de la Comunidad Europea (NACE) es el sistema de clasificación de las actividades económicas usado en la Unión Europea. Esta clasificación se basa en unidades estadísticas que corresponden a una actividad económica específica (o a un grupo de actividades similares), que conforman un grupo económico, es decir una industria o sector económico. NACE Rev. 2 está estructurada en 21 secciones. Hemos dividido las empresas que conforman nuestra base de datos en los sectores correspondientes a esta clasificación y hemos realizado el mismo experimento solo que con las empresas agrupadas por sectores. Los sectores que han dado lugar de la selección de nuestras empresas son los siguientes: J - Información y comunicación; B - Petroleo y otras industrias extractivas; D -

Electricas; F - Construcción; H - Transporte y almacenamiento; C - Fabricación; L - Actividades inmobiliarias; G - Comercio

En la figura 2 se pueden observar los rendimientos (eje X) de las 8 sectores correspondientes a las empresas (eje Y). Los resultados que se muestran son motivadores, si cada uno de los STs hubiera invertido en empresas de sectores determinados, los rendimientos serían muy diferentes. Utilizando RA para los sectores J y B el ST obtendría unos beneficios de un 130 %. Usando IS para los sectores J y G los beneficios obtenidos ascenderían a 119 %. Y por último usando el ST con el fitness de CCPR para los sectores D y C los rendimientos quedarían en un 50 %. Estos resultados nos inducen a pensar que un análisis previo de los sectores o combinado en la gramática junto a las funciones de fitness proporcionará soluciones que encajen mejor y arrojen mejores rendimientos. Por otro lado, nos anima a realizar otra serie de experimentos combinando las funciones de fitness en un solo ST, es decir, transformando nuestro ST en un ST multiobjetivo.

La Figura 3 muestra los resultados de las series de inversiones en los 8 sectores analizados. Como en las figuras anteriores el eje X representa los rendimientos y el eje Y los diferentes sectores. Cada sector contara con dos resultados, el primero sera la combinación de dos objetivos, el IS y el AR. El segundo resultado será la combinación de las tres funciones de fitness que se han utilizado en este trabajo (IS, AR y CCPR). El algoritmo de selección ha pasado a implementarse mediante un NGSA-II [7] utilizando los mismos operadores y probabilidades del anterior AE. Los resultados muestran un incremento de las ganancias en aquellos sectores donde ambos objetivos conseguían buenos resultados, por ejemplo en el sector J ,donde el RA y el IS conseguían aproximadamente un 100 % de rentabilidad, gracias a la combinación de los dos objetivos las ganancias aumentan hasta un 287 %. Sin embargo, aquellos sectores donde no coinciden en términos de ganancias los resultados empeoran. Véase el sector G, donde el AR obtenía un 18 % de beneficios y el IS un 33 % de perdidas, y que acaba presentando unas demoledoras perdidas de -138 %. Estos resultados de nuevo apoyan la idea de construir un ST capaz de elegir donde utilizar unas funciones de fitness especificas a un sector o grupos de compañias específicas.

4. Conclusiones y trabajo futuro

En este trabajo hemos testeado un ST capaz de obtener beneficios en un ambiente hostil como la presente recesión económica. Además hemos comparado los rendimientos del ST con diferentes funciones de fitness para el mismo conjunto de datos. Hemos analizado el conjunto de compañias por sectores categorizados en diferentes actividades empresariales. Estos resultados experimentales nos proporcionan información de como las funciones de fitness pueden ser más o menos rentables dependiendo del sector evaluado. Además, hemos implementado una solución multiobjetivo que es capaz de aumentar los beneficios o perdidas dependiendo de la utilidad de las funciones por separado en los diferentes sectores empresariales. Este trabajo nos anima a ampliar este análisis y establecer

unas lineas de futuro en pos de una implementación capaz de decidir en que compañias o sectores invertir y que función de fitness utilizar.

Referencias

1. K. Adamu and S. Phelps. Modelling financial time series using grammatical evolution. In *Proceedings of the Workshop on Advances in Machine Learning for Computational Finance*, London, UK, 2009.
2. Kamal Adamu and Steve Phelps. Coevolution of technical trading rules for high frequency trading. In *Proceedings of the World Congress on Engineering*, pages 96–101, 2010.
3. Anthony Brabazon and Michael O'Neill. Evolving technical trading rules for spot foreign-exchange markets using grammatical evolution. *Computational Management Science*, 1(3):311–327, October 2004.
4. Robert W. Colby and Thomas A. Meyers. *The encyclopedia of technical market indicators*. Irwin, New York [u.a.], 1988.
5. Iván Contreras, José Ignacio Hidalgo, and Laura Núñez-Letamendia. Combining technical analysis and grammatical evolution in a trading system. In *EvoApplications*, 2013.
6. Alfonso Ortega de la Puente, Rafael Sánchez Alfonso, and Manuel Alfonseca Moreno. Automatic composition of music by means of grammatical evolution. *SIGAPL APL Quote Quad*, 32(4):148–155, June 2002.
7. Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evolutionary Computation*, 6(2):182–197, 2002.
8. Ian Dempsey, Michael O'Neill, and Anthony Brabazon. Live trading with grammatical evolution. In *GECCO 2004 Workshop Proceedings*, Seattle, Washington, USA, 26-30 June 2004.
9. Ian Dempsey, Michael O'Neill, and Anthony Brabazon. Adaptive trading with grammatical evolution. In *Proceedings of the 2006 IEEE Congress on Evolutionary Computation*, pages 9137–9142, Vancouver, 6-21 July 2006. IEEE Press.
10. Jonatan Hugosson, Erik Hemberg, Anthony Brabazon, and Michael O'Neill. Genotype representations in grammatical evolution. *Appl. Soft Comput.*, 10(1):36–43, January 2010.
11. Jason H. Moore and Lance W. Hahn. Petri net modeling of high-order genetic systems using grammatical evolution. *Biosystems*, 72(12):177 – 186, 2003. Computational Intelligence in Bioinformatics.
12. M. O'Neill and C. Ryan. *Grammatical Evolution: Evolutionary Automatic Programming in an Arbitrary Language*. Kluwer Academic Publishers, 2003.
13. C. Ryan, J. J. Collins, and M. O'Neill. Grammatical evolution: Evolving programs for an arbitrary language. In *Lecture Notes in Computer Science 1391. First European Workshop on Genetic Programming 1998.*, 1998.
14. Conor Ryan, Michael O'Neill, and Jj Collins. Grammatical evolution: Solving trigonometric identities. In *In Proceedings of Mendel '98: 4th International Conference on Genetic Algorithms, Optimization Problems, Fuzzy Logic, Neural Networks and Rough Sets*, pages 111–119, 1998.
15. Philip Saks and Dietmar G. Maringer. Evolutionary money management. volume 5484 of *Lecture Notes in Computer Science*, pages 162–171. Springer, 2009.

Aprendizaje de reglas difusas TSK de orden 1 mediante búsqueda local

Javier Cózar, Luis delaOssa y José A. Gámez

Departamento de Sistemas Informáticos - I^3A
Universidad de Castilla-La Mancha - Albacete (02071) - España
{javier.cozar,luis.delaossa,jose.gamez}@uclm.es

Resumen La metodología COR (*Cooperative Rules*) utiliza un algoritmo de optimización para buscar el conjunto de reglas difusas que minimiza el error de predicción para un conjunto de datos. Debido a que las reglas difusas solamente interactúan con reglas adyacentes, los algoritmos de búsqueda local son adecuados para llevar a cabo este proceso.
En este trabajo, se adaptarán los algoritmos de búsqueda local existentes, que aprenden reglas lingüísticas de tipo Mamdani o TSK-0, para el aprendizaje de reglas de tipo TSK-1, en las que el consecuente es un polinomio de grado 1.
Los resultados muestran que el algoritmo propuesto permite obtener sistemas de tamaño reducido, en los que la predicción mejora a la obtenida con el algoritmo clásico de aprendizaje mediante mínimos cuadrados.

Keywords: Reglas difusas TSK, aprendizaje automático, modelado difuso

1. Introducción

Existen diversos algoritmos que derivan las reglas de un Sistema Basado en Reglas Difusas (SBRDs) a partir de un conjunto de datos. Algunos de ellos utilizan métricas que evaluan las reglas de forma individual [11,4] y determinan el conjunto final de reglas en base a estas medidas. La calidad individual de las reglas es un factor relevante. Sin embargo, la salida producida por un SBRD ante una entrada se obtiene mediante la combinación de las salidas generadas por cada una de las reglas disparadas. A este hecho se le denomina *cooperación*, y es la clave de la potencia y expresividad de los SBRDs.

Existen diferentes propuestas, como la metodología COR [2], que aprenden la base de reglas mediante la evaluación y búsqueda de conjuntos de reglas, teniendo en cuenta la cooperación. El principal problema de estas técnicas es que el espacio de entrada crece exponencialmente con el tamaño del problema y del número de etiquetas usadas para modelar las variables difusas. Además, la evaluación de cada conjunto de reglas es costosa, ya que supone construir la base de reglas y procesar todo el conjunto de datos sobre la misma.

Los métodos descritos en [6,5] proponen el uso de algoritmos de búsqueda local para encontrar la base de reglas en SBRDs lingüísticas de tipo Mamdani [6], y sistemas de tipo Takagi-Sugeno-Kang de orden 0 [5]. Además de la calidad de las soluciones en términos de error, estos algoritmos llevan a cabo la búsqueda de modo más eficiente que los basados en poblaciones. Principalmente, esto se debe al hecho de que un cambio local en un sistema de reglas (una sola regla) tiene una repercusión acotada, y puede evaluarse con un menor coste.

En el caso de los sistemas TSK de orden 1 (TSK-1), en los que el consecuente es un polinomio de grado 1, el consecuente de cada regla se aprende de manera independiente mediante mínimos cuadrados [7]. En este trabajo, proponemos la adaptación de los algoritmos citados anteriormente [6,5] para el aprendizaje de reglas TSK-1 con cooperación. Para ello, es necesario modificar el modo en que se generan los consecuentes de las reglas.

Este artículo se divide en 5 secciones además de esta introducción. En la sección 2 se describen los SBRDs tipo TSK-1. Después, las secciones 3 y 4 explican la metodología COR y la adaptación de los algoritmos de búsqueda local para el aprendizaje de este tipo de reglas. Posteriormente, la sección 5 presenta el estudio experimental del método propuesto. Finalmente, la sección 6 resume las principales conclusiones obtenidas, y describe líneas de trabajo futuras.

2. Sistemas basados en reglas difusas tipo TSK-1

En los SBRDs, las reglas están formadas por predicados del tipo X es F, donde X es una variable del problema, y F es un conjunto difuso definido sobre el dominio de dicha variable. En el caso de los Sistemas Basados en Reglas Difusas Lingüísticas (SBRDLs) [12,9], las reglas están formadas por predicados del tipo X es A, donde A es una etiqueta lingüística, es decir, un término asociado con un conjunto difuso determinado, y previamente definido en el dominio de X, que junto con la definición de la partición difusa y las etiquetas, se denomina variable lingüística [12].

El uso de variables lingüísticas hace que los modelos de SBRDLs sean más fáciles de interpretar por expertos humanos. Sin embargo, la limitación en el número de conjuntos difusos que pueden ser utilizados para construir las reglas -solo aquellos que forman parte de la definición de las variables lingüísticas- limitan su rendimiento en cuanto a capacidad de predicción.

Existen diferentes variaciones sobre los SBRDLs que ayudan a mejorar su precisión con una pequeña pérdida de interpretabilidad. Una de las más importantes consiste en el uso de reglas TSK [10]. En estos sistemas, el consecuente de cada regla R_s es una función polinómica de las variables de entrada, $P_s(X_1, \ldots, X_n)$. El orden de un SBRDL TSK se refiere al grado del polinomio P_s. Así, en sistemas TSK-1 el consecuente es un polinomio de primer grado, es decir, una combinación lineal de los valores de entrada.

Por lo tanto, una regla de tipo TSK-1 se expresa como:

$$R_s : \text{Si } X_1 \text{ es } A_1^s \& \ldots \& X_n \text{ es } A_n^s \text{ entonces } Y = a_{s_1} X_1 + \cdots + a_{s_n} X_n + b_s$$

Dada una instancia $e_l = (x_1^l, \ldots, x_n^l, y^l)$, la salida en un sistema TSK formado por el conjunto de reglas $\mathcal{RB}$ se obtiene como la media ponderada de las salidas individuales producidas por cada regla $R_s \in \mathcal{RB}$:

$$\hat{y}^l = \frac{\sum_{R_s \in \mathcal{RB}} h_s^l P_s(x_1^l, \ldots, x_n^l)}{\sum_{R_s \in \mathcal{RB}} h_s^l} \tag{1}$$

donde $h_s^l(e_l) = T(A_1^s(x_1^l), \ldots, A_n^s(x_n^l))$ es el grado de emparejamiento de la instancia e_l con la regla R_s, T es una T-norma [1], y $P_s(x_1^l, \ldots, x_n^l)$ es el consecuente (polinomio) de la regla R_s.

3. La metodología de Reglas Cooperativas para reglas TSK-1

Los algoritmos de aprendizaje de SBRDL que se limitan a la derivación de las reglas toman como entrada dos elementos:

- Un *Conjunto de datos* $\mathcal{E} = \{e_1, \ldots, e_l, \ldots, e_N\}$, donde $e_l = (x_1^l, \ldots, x_n^l, y^l)$.
- La *definición de las variables lingüísticas*: sus dominios, las particiones difusas y los términos difusos $\mathcal{A}_i$ para cada variable de entrada X_i. En el caso de reglas tipo Mamdani, la definición incluye los términos difusos $\mathcal{B}$ para la variable de salida Y [2]. Sin embargo, esto no es necesario para las reglas TSK.

A partir de dichas entradas, estos métodos generan un conjunto de reglas candidatas - sus antecedentes -, de modo que cada instancia $e_l \in \mathcal{E}$ dispare al menos una regla. Después, generalmente de forma simultánea, se eligen algunas de estas reglas y se fijan sus consecuentes.

Formalmente, se define un subespacio difuso del dominio de entrada, $S_s = (A_1^s, \ldots, A_i^s, \ldots, A_n^s)$, con $A_i^s \in \mathcal{A}_i$, como aquel que contiene cualquier instancia e_l tal que $\mu_{A_1^s}(x_1^l) \cdot \ldots \cdot \mu_{A_n^s}(x_n^l) > 0$. Por otra parte, se denomina $\mathcal{E}^s \subseteq \mathcal{E}$ al conjunto de instancias del conjunto de entrenamiento que pertenecen a un subespacio de entrada.

El número máximo de subespacios de entrada (y de posibles reglas del sistema) es $N_s = \prod_{i=1}^{n} |\mathcal{A}_i|$. Sin embargo, los métodos basados en cobertura de los datos generan reglas que solo cubren un conjunto de subespacios S^+.

Generalmente, S^+ se obtiene a partir de una *Rejilla Difusa*. En este caso un subespacio es incluído en S^+ si contiene al menos un ejemplo, es decir:

$$S_s \in S^+ \text{ si } \exists e_l \in \mathcal{E} | \mu_{A_1^s}(x_1^l) \cdot \ldots \cdot \mu_{A_n^s}(x_n^l) > 0.$$

[1] En este trabajo consideraremos la T-norma del mínimo

[2] En este estudio solo consideraremos una variable de salida, aunque este número puede ser mayor que 1.

Los algoritmos de aprendizaje generan reglas candidatas de modo que para cada $S_s \in S^+$, se genera una regla

$$R_s : \text{Si } X_1 \text{ es } A_1^s \& \ldots \& X_n \text{ es } A_n^s \text{ entonces } Y = \ldots$$

Una vez determinadas las reglas candidatas, se eligen los consecuentes. Algoritmos como Wang y Mendel [11] o ALM [4] para reglas tipo Mamdani; Inductive Method [8] para el caso de reglas TSK-0; o Mínimos cuadrados [7] para el caso de reglas TSK-1, fijan el consecuente de manera independiente para cada regla. Sin embargo, al evaluar de forma individual la calidad de cada regla no se tiene en cuenta la forma en la que los SBRDs procesan las entradas. Una instancia $e_l = (x_1^l \cdot \ldots \cdot x_n^l)$ dispara toda regla R_s tal que $\mu_{A_1^s}(x_1^l) \cdot \ldots \cdot \mu_{A_n^s}(x_n^l) > 0$. Si $\mathcal{R}^l$ es el conjunto de reglas disparadas por la entrada e_l. La salida $\hat{y}^l$ es obtenida como una combinación de las salidas individuales producidas por cada regla en $\mathcal{R}^l$ (expresión 1). Por lo tanto, la calidad de una regla R_s debería ser evaluada en el contexto de $\mathcal{R}^l$. Esto implica que el rendimiento de R_s depende de aquellas reglas R_t tal que $\mathcal{E}^s \cap \mathcal{E}^t \neq \emptyset$, es decir, de aquellas reglas que cubran alguno de los ejemplos cubiertos por R_s.

Hay algunos enfoques que consideran la cooperación entre reglas en el aprendizaje. En la metodología de Reglas Cooperativas (COR) [2] se trata de encontrar la combinación óptima de consecuentes mediante un algoritmo de búsqueda. En el caso de las reglas tipo Mamdani, la búsqueda puede ser llevada a cabo mediante cualquier algoritmo de optimización combinatoria, como Algoritmos Genéticos [3] u Optimización Basada en Colonia de Hormigas [1]; mientras que para reglas de tipo TSK-0 se han utilizado búsqueda local [5], aunque también se puede utilizar cualquier algoritmo de optimización numérica.

Al margen del algoritmo utilizado, debe evaluarse cada configuración o base de reglas candidata. Para ello, puede utilizarse cualquier medida de error. En este estudio hemos considerado el Error Cuadrático Medio *MSE*. Sea $\hat{y}_o^l$ la salida producida por el SBRDL que contiene el conjunto de reglas $\mathcal{RB}_o$ al procesar la instancia e_l. El error producido por el sistema con el conjunto de reglas $\mathcal{RB}_o$ al procesar el conjunto de datos $\mathcal{E}$, $MSE_o(\mathcal{E})$, se obtiene como:

$$MSE_o(\mathcal{E}) = \sum_{l=1}^{|\mathcal{E}|} \frac{(\hat{y}_o^l - y^l)^2}{N} \tag{2}$$

4. Algoritmo de búsqueda local

El algoritmo descrito en [6] usa algoritmos de búsqueda local para encontrar un conjunto final de reglas tipo Mamdani. Esta propuesta se basa en que, dada una base de reglas $\mathcal{RB}_o$, un cambio en el consecuente de una regla R_s^o puede repercutir en las reglas R_t^o tales que $\mathcal{E}^s \cap \mathcal{E}^t \neq \emptyset$. Aunque teóricamene este cambio podría propagarse y alterar el valor óptimo del consecuente para todas las reglas en $\mathcal{RB}_o$, la repercusión suele limitarse a algunas reglas adyacentes.

El algoritmo 1 muestra el esquema de funcionamiento de la búsqueda local. La vecindad de una base de reglas $\mathcal{RB}_o$, $\mathcal{N}(\mathcal{RB}_o)$, se define como todos aquellos conjuntos de reglas que se obtienen cambiando el consecuente de una sola regla de $\mathcal{RB}_o$. Una vez que la solución inicial es generada, aleatoriamente o mediante cualquier otro método como [11,4], el algoritmo lleva a cabo un *HillClimbing*, aplicando en cada paso el movimiento en la vecindad que produce el mayor decremento en el error (MSE). El algoritmo se detiene cuando ningún cambio en el conjunto de reglas conlleva una disminución del error.

Algoritmo 1 Búsqueda local con COR

```
procedimiento BúsquedaLocalCOR(E,DBF)
    S+ ← generarSubespaciosEntrada(E,DBF)
    RB_o ← generarSoluciónInicial(S+,E,DBF)
    nuevoError ← MSE_o(E)
    error ← ∞
    mientras nuevoError < error hacer
        error ← nuevoError
        N(RB_o) ← generaVecinos(RB_o)
        para RB_p ∈ N(RB_o) hacer
            si MSE_p(E) < nuevoError entonces
                nuevoError ← MSE_p(E)
                RB_o ← RB_p
            fin si
        fin para
    fin mientras
    devuelve RB_o
fin procedimiento
```

Además de obtener mejores resultados, en términos de error, que un algoritmo genético, el algoritmo de búsqueda local es considerablemente más eficiente [6]. La evaluación de una solución consiste en construir la base de reglas $\mathcal{RB}_o$ y, para cada instancia $e_l \in \mathcal{E}$, procesar todas las reglas $R_s^o \in \mathcal{RB}_o$, produciendo el conjunto de salidas difusas $\mathcal{B}^l$, que deben ser agregadas y defuzificadas. En el caso de la búsqueda local, un vecino $\mathcal{RB}_p \in \mathcal{N}(\mathcal{RB}_o)$ se obtiene mediante el cambio de una sola regla de $\mathcal{RB}_o$, por lo que solo es necesario recalcular las salidas producidas para las instancias cubiertas por esa regla para obtener $MSE_p(\mathcal{E})$ a partir de $MSE_o(\mathcal{E})$.

Por otra parte, se puede evitar una gran cantidad de cálculos. Si $\mathcal{N}_s(\mathcal{RB}_o)$ es el subconjunto de vecinos de $\mathcal{RB}_o$ obtenido mediante el reemplazo de la regla R_s^o, y $\mathcal{RB}_p \in \mathcal{N}_s(\mathcal{RB}_o)$, no es necesario evaluar todos los elementos en $\mathcal{N}(\mathcal{RB}_p)$. De hecho, si $\mathcal{E}^s \cap \mathcal{E}^t = \emptyset$, no hay necesidad de calcular $\mathcal{N}_t(\mathcal{RB}_p)$. Si $\mathcal{RB}_{o'} \in \mathcal{N}_t(\mathcal{RB}_o)$, $\mathcal{RB}_{p'} \in \mathcal{N}_t(\mathcal{RB}_p)$, y $\mathcal{E}^s \cap \mathcal{E}^t = \emptyset$, entonces:

$$MSE_{o'}(\mathcal{E}) - MSE_o(\mathcal{E}) = MSE_{p'}(\mathcal{E}) - MSE_p(\mathcal{E}).$$

Como se muestra en [6], esto permite evitar el cómputo de un gran número de evaluaciones en el proceso de búsqueda.

4.1. Adaptación del vecindario para el aprendizaje de reglas TSK-1

La adaptación de la metodología COR al caso de los sistemas TSK-1 es inmediata, y se basa en el algoritmo para sistemas TSK-0 propuesto en [5]. En lugar de buscar un consecuente de entre un conjunto de etiquetas discretas (como sucede con las reglas tipo Mamdani [6]), en el caso de reglas TSK-1 se han de encontrar polinomios de orden 1:

$$P_s^o(X_1, \ldots, X_n) = a_{s_1} X_1 + \cdots + a_{s_n} X_n + b_s$$

Cada solución $\mathcal{RB}_o$ puede ser representada por una matriz de números reales, en la que cada fila $s = 1 \ldots |S^+|$ se corresponde con la regla R_s, y contiene los coeficientes $(\{b_s^o, a_{s_1}^o, \ldots, a_{s_n}^o\})$ de un polinomio $P_s^o(X_1, \ldots, X_n)$. De modo que cualquier método de búsqueda que trabaje sobre el dominio de los números reales puede ser utilizado par establecer los consecuentes.

En el caso de sistemas TSK-1 no es posible llevar a cabo una búsqueda local directamente, ya que cada consecuente es un polinomio cuyos coeficientes son números reales. Por tanto, $|\mathcal{N}(RB_o)| = \infty$. Sin embargo, dada una base de reglas $\mathcal{RB}_o$, es posible obtener el consecuente óptimo -que de lugar a un menor error- para una regla determinada $R_s^o \in \mathcal{RB}_o$.

Sea $\mathcal{E}^s$ el conjunto de ejemplos cubiertos por R_s^o; $e_l \in \mathcal{E}^s$; y $\mathcal{R}^l \subset \mathcal{RB}_o$ el conjunto de reglas disparadas por e_l. La salida producida por el sistema $\mathcal{RB}_o$ al procesar e_l se obtiene como:

$$\hat{y}_o^l = \frac{\sum_{R_s^o \in \mathcal{R}^l} h_s^l v_s^l}{\sum_{R_s^o \in \mathcal{R}^l} h_s^l}$$

donde $v_s^l = P_s(e_l) = a_{s_1} x_1^l + \cdots + a_{s_n} x_n^l + b_s$ es la salida producida por la regla R_s cuando es disparada por la instancia e_l.

Si se considera que $HV_{o-s}^l = \sum_{R_t^o \in \mathcal{R}^l | R_t^o \neq R_s^o} h_t^l v_t^l$, y que $H_o^l = \sum_{R_t^o \in \mathcal{R}^l} h_t^l$, la ecuación anterior puede expresarse como:

$$\hat{y}_o^l = \frac{\sum_{R_t^o \in \mathcal{R}^l | R_t^o \neq R_s^o} h_t^l v_t^l + h_s^l v_s^l}{\sum_{R_t^o \in \mathcal{R}^l} h_t^l} = \frac{HV_{o-s}^l + h_s^l v_s^l}{H_o^l} = \frac{HV_{o-s}^l}{H_o^l} + \frac{h_s^l}{H_o^l} v_s^l \qquad (3)$$

Teniendo en cuenta todo el conjunto de casos cubiertos por la regla, el objetivo es encontrar el polinomio que minimice $y^l - \hat{y}_o^l \ \forall e_l \in \mathcal{E}^s$. Dicho polinomio puede obtenerse directamente mediante mínimos cuadrados.

Reescribiendo la ecuación 3 en forma vectorial, con el objetivo de aplicar mínimos cuadrados, tenemos que:

$$\hat{y}_o^l - \frac{HV_{o-s}^l}{H_o^l} = \frac{h_s^l}{H_o^l} v_s^l = \left| \frac{h_s^l}{H_o^l} \ \frac{h_s^l}{H_o^l} x_1^l \ \cdots \ \frac{h_s^l}{H_o^l} x_n^l \right| \cdot \left| b_s \ a_{s_1} \ \cdots \ a_{s_n} \right|^T$$

Por lo tanto, para todo el conjunto de instancias $\mathcal{E}^s$:

$$\begin{vmatrix} \hat{y}_o^1 - \frac{HV_{o-s}^1}{H_o^1} \\ \vdots \\ \hat{y}_o^l - \frac{HV_{o-s}^l}{H_o^l} \\ \vdots \\ \hat{y}_o^{|\mathcal{E}^s|} - \frac{HV_{o-s}^{|\mathcal{E}^s|}}{H_o^{|\mathcal{E}^s|}} \end{vmatrix} = \begin{vmatrix} \frac{h_s^1}{H_o^1} & \frac{h_s^1}{H_o^1}x_1^1 & \cdots & \frac{h_s^1}{H_o^1}x_n^1 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{h_s^l}{H_o^l} & \frac{h_s^l}{H_o^l}x_1^l & \cdots & \frac{h_s^l}{H_o^l}x_n^l \\ \vdots & \vdots & \vdots & \vdots \\ \frac{h_s^{|\mathcal{E}^s|}}{H_o^{|\mathcal{E}^s|}} & \frac{h_s^{|\mathcal{E}^s|}}{H_o^{|\mathcal{E}^s|}}x_1^{|\mathcal{E}^s|} & \cdots & \frac{h_s^{|\mathcal{E}^s|}}{H_o^{|\mathcal{E}^s|}}x_n^{|\mathcal{E}^s|} \end{vmatrix} \cdot \begin{vmatrix} b_s \\ a_{s_1} \\ \vdots \\ a_{s_n} \end{vmatrix}$$

Es importante remarcar que tanto h_s^l como H_o^l solo necesitan ser calculados una vez, pues no cambian durante todo el proceso de búsqueda.

Aunque es posible determinar el consecuente óptimo para una regla, es muy importante considerar la no inclusión de la misma en aras de mejorar la interpretabilidad del sistema. Esto se consigue considerando un consecuente adicional representado como $\emptyset$. De cara al algoritmo de búsqueda local, esto implica que $|\mathcal{N}_s(\mathcal{RB}_o)| = 2$, ya que los vecinos asociados a cada regla se obtienen mediante la sustitución del consecuente por el óptimo, o mediante la eliminación de la regla (consecuente $\emptyset$). Esto implica que $|\mathcal{N}(\mathcal{RB}_o)| = 2(|\mathcal{RB}_o| - 1)$, por lo que el tamaño de vecindad es pequeño.

Sin embargo, el algoritmo puede llevar a cabo un número de iteraciones excesivo, aunque hay un punto tras el cual las mejoras producidas son prácticamente insignificantes. Hay dos estrategias que pueden solucionar este problema. La primera consiste en parar la búsqueda cuando la mejora producida no alcanza un umbral determinado. La segunda, utilizada en [5] y en este trabajo, evita evaluar vecinos cuando la diferencia entre el mejor consecuente para una determinada regla y el consecuente usado en el sistema actual no alcanza un umbral. Puesto que los consecuentes son polinomios, la diferencia entre dos polinomios se define como la suma de la diferencia en valor absoluto entre cada par de coeficientes.

5. Experimentación

Para probar los algoritmos propuestos, se han tomado cuatro conjuntos de datos para modelado del repositorio KEEL[3]. Uno de ellos, *autoMPG*6, tiene 5 variables de entrada; otros dos, *machineCPU* y *dee*, tienen 6; por último, *autoMPG*8 tiene 7.

El algoritmo tomado como base para la comparativa utiliza *Mínimos cuadrados* directamente para obtener el polinomio óptimo para cada regla. Este algoritmo, usado en [7], no considera cooperación. En cuanto al algoritmo de búsqueda local para aprendizaje de reglas TSK-1 con cooperación (*COR-Búsqueda Local*) ha sido implementado siguiendo el esquema propuesto en la sección anterior, e

[3] http://sci2s.ugr.es/keel/

incluye todas las optimizaciones comentadas para evitar evaluaciones innecesarias. Como punto de partida de la búsqueda, se han considerado dos opciones: la base de reglas obtenida por medio de *Mínimos cuadrados* de manera independiente para cada regla (*COR-Búsqueda Local*); y la base de reglas vacía, en la que cada consecuente es fijado inicialmente a $\emptyset$ (*COR-Búsqueda Local-Vacío*). Para todos los algoritmos, las variables lingüísticas se han modelado mediante particiones simétricas triangulares con 3 y 5 etiquetas.

Los resultados han sido obtenidos ejecutando 30 ejecuciones independientes sobre cada problema. En cada una de ellas, el conjunto de datos de entrenamiento se ha obtenido seleccionando el 80 % de las instancias de manera aleatoria, utilizando como semilla el número de ejecución, y dejando el 20 % restante para el test. La tabla 1 muestra los resultados obtenidos.

A simple vista, puede apreciarse que el sistema de partida utilizado en el algoritmo de búsqueda local tiene una incidencia determinante en los resultados obtenidos. Es por esto que ambas propuestas se analizarán por separado.

COR-Búsqueda Local.

Como puede observarse en los resultados de error en el entrenamiento para los dos primeros algoritmos, aplicar la búsqueda local con cooperación sobre el modelo obtenido con mínimos cuadrados permite incrementar sustancialmente el ajuste del modelo. Esta mejora es muy importante en casi todos los casos, y no depende del número de etiquetas. Sin embargo, los resultados de error para el test apuntan a cierto sobreajuste en algunos casos, aunque en otros el algoritmo mejora también. En relación al número de reglas, la búsqueda local produce cierto descenso, aunque éste no es excesivamente imporante.

COR-Búsqueda Local-Vacío.

El aumento en la capacidad de ajuste del modelo permite simplificar éste de manera muy importante. Así, al iniciar la búsqueda local desde el conjunto de reglas vacío, puede observarse que la mejora en el ajuste a los datos de entrenamiento se traslada ahora también a los de test, ya que la disminución del error con respecto a los dos primeros algoritmos es muy significativa especialmente para los problemas *machineCPU* y *dee*. Además, dicha mejora se logra a la vez que se reduce sustalcialmente el número de reglas. Reducción que en algunos casos implica el descarte de más del 90 % de éstas.

Cabe apuntar también que, para el caso de las reglas TSK-1, parece beneficioso utilizar una granularidad menor en las particiones. Esta tendencia se mantiene al utilizar los algoritmos de búsqueda local que, en el caso de utilizar 3 etiquetas por variable, aprenden modelos muy reducidos.

machineCPU

3 etiquetas	Error Entr.	Error Test	#Reglas	#Sistemas eval.	#Instancias Proc.
Mínimos cuadrados	169,63	133,00	251,07	1,00	167,00
COR-Búsqueda Local	3,38	138,48	249,77	12408,13	901761,60
COR-Búsqueda Local-Vacío	2,49	23,45	37,63	3799,63	249106,87
5 etiquetas					
Mínimos cuadrados	311,21	311,03	590,60	1,00	167,00
COR-Búsqueda Local	62,14	292,02	585,70	22279,43	930558,93
COR-Búsqueda Local-Vacío	0,88	29,90	82,77	10891,63	406747,80

autoMPG6

3 etiquetas	Error Entr.	Error Test	#Reglas	#Sistemas eval.	#Instancias Proc.
Mínimos cuadrados	3,58	4,06	196,33	1,00	313,00
COR-Búsqueda Local	0,77	8,08	171,13	7458801,33	519593419,07
COR-Búsqueda Local-Vacío	1,56	4,97	89,77	743175,23	48276154,13
5 etiquetas					
Mínimos cuadrados	5,30	7,36	794,97	1,00	313,00
COR-Búsqueda Local	0,11	9,39	764,57	1218775,57	29661239,30
COR-Búsqueda Local-Vacío	0,27	7,72	271,63	848192,93	20241095,30

dee

3 etiquetas	Error Entr.	Error Test	#Reglas	#Sistemas eval.	#Instancias Proc.
Mínimos cuadrados	2,35	2,36	516,27	1,00	292,00
COR-Búsqueda Local	1,80	2,04	510,50	6341,13	308590,50
COR-Búsqueda Local-Vacío	0,14	0,88	174,27	1175879,67	70063701,53
5 etiquetas					
Mínimos cuadrados	2,37	2,37	3057,37	1,00	292,00
COR-Búsqueda Local	1,57	2,26	3041,30	27608,10	239439,93
COR-Búsqueda Local-Vacío	0,00	1,14	215,27	70752,83	791715,00

autoMPG8

3 etiquetas	Error Entr.	Error Test	#Reglas	#Sistemas eval.	#Instancias Proc.
Mínimos cuadrados	4,37	6,16	819,00	1,00	313,00
COR-Búsqueda Local	0,33	7,56	669,83	3440450,80	96438238,80
COR-Búsqueda Local-Vacío	0,70	6,73	143,20	646604,50	16688036,07
5 etiquetas					
Mínimos cuadrados	6,19	9,70	2708,70	1,00	313,00
COR-Búsqueda Local	0,39	11,31	2678,93	140476,13	1621330,67
COR-Búsqueda Local-Vacío	0,10	7,82	206,70	132184,57	1336617,40

Tabla 1. Resultados (media de 30 ejecuciones) para los cuatro problemas: Error de entrenamiento, error de test, número de reglas del sistema, número de sistemas evaluados y número de instancias procesadas.

6. Conclusiones finales y trabajo futuro

En este trabajo se ha presentado un algoritmo basado en búsqueda local eficiente para aprender reglas tipo TSK-1 considerando la cooperación entre reglas. Los resultados obtenidos muestran que el algoritmo propuesto mejora de manera sustancial la capacidad de ajuste de los sistemas obtenidos. A pesar de que esta mejora puede acarrear un problema de sobreajuste para algunos problemas, posibilita reducir la complejidad del modelo. De hecho, al utilizar el conjunto vacío como punto de comienzo de la búsqueda local, se consigue una reducción importante en el número de reglas, disminuyendo además, sustancialmente en algunos casos, el error de test.

En futuros trabajos, pretendemos incluir regularización al proceso de mínimos cuadrados con el fin de obtener funciones polinómicas más reducidas en los consecuentes, mejorando así la interpretabilidad de los sistemas obtenidos. Por otra parte, se extenderá el trabajo a problemas con un mayor número de variables, en los que la generación de reglas candidatas se ha de abordar de modo distinto.

Agradecimientos. Este estudio ha sido parcialmente financiado por el Gobierno de España (MINECO) y fondos FEDER bajo el proyecto TIN2010-20900-C04-03. Javier Cózar ha sido también financiado por el MICINN a través de la beca FPU12/05102.

Referencias

1. J. Casillas, O. Cordón, I. Fernández de Viana, and F. Herrera. Learning cooperative linguistic fuzzy rules using the best-worst ant system algorithm. *International Journal of Intelligent Systems*, 20:433–452, 2005.
2. J. Casillas, O. Cordón, and F. Herrera. Cor: A methodology to improve ad hoc data-driven linguistic rule learning methods by inducing cooperation among rules. *IEEE Transactions on Systems, Man and Cybernetics. Part B: Cybernetics.*, 32(4):526–537, 2002.
3. J. Casillas, O. Cordón, and F. Herrera. *Technologies for constructing intelligent systems: Tasks*, chapter Different approaches to induce cooperation in fuzzy linguistic models under the COR methodology, pages 321–334. Physica-Verlag GmbH, Heidelberg, Germany, Germany, 2002.
4. O. Cordón and F. Herrera. A proposal for improving the accuracy of linguistic modeling. *IEEE Transactions on Fuzzy Systems*, 8(3):335–344, 2000.
5. J. Cózar, L. de la Ossa, and J.M. Puerta. Learning cooperative tsk-0 fuzzy rules using fast local search algorithms. In *CAEPIA*, pages 353–362, 2011.
6. L. delaOssa, J.A. Gámez, and J.M. Puerta. Learning cooperative fuzzy rules using fast local search algorithms. In *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2010)*, pages 2134–2141, 2006.
7. F. Herrera and P. Villar. Un algoritmo genético para aprendizaje de un sistema basado en reglas difusas tipo takagi-sugeno. In *V Congreso Espa nol sobre Metaheurísticas and Algoritmos Evolutivos y Bioinspirados(MAEB07)*, pages 543–548, 2007.
8. H. Ishibuchi, K. Nozaki, H. Tanaka, Y. Hosaka, and M. Matsudai. Empirical study on learning in fuzzy systems by rice taste analysis. *Fuzzy Sets Syst.*, 64(2):129–144, June 1994.
9. E.H. Mamdani and S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, 7:1–13, 1975.
10. T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications for modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, 15(1):116–132, 1985.
11. L.X. Wang and J.M. Mendel. Generating fuzzy rules by learning from examples. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(6):1414–1427, 1992.
12. L.A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning. *Information Science*, 8:199–249, 1975.

Validación Inteligente para la Sincronización de Semáforos Basada en Feature Models

Javier Ferrer, José García-Nieto, Enrique Alba, and Francisco Chicano

Dept. de Lenguajes y Ciencias de la Computación,
Universidad de Málaga, E.T.S.I. Informática,
{ferrer,jnieto,eat,chicano}@lcc.uma.es

Resumen El concepto de *Smart City* o Ciudad Inteligente engloba el conjunto de acciones y servicios, basados en las tecnologías de información y comunicación, que se ofrecen en un núcleo urbano. En este sentido, el uso de técnicas bioinspiradas para la gestión del flujo del tráfico mediante la sincronización de semáforos podría constituir uno de los aspectos más innovadores en los entornos urbanos en el futuro. No obstante, la programación automática de semáforos requiere además de un proceso de validación de las soluciones generadas, dado que afectan a la seguridad de miles de usuarios. En este trabajo se propone una estrategia de validación basada en los Modelos de Características (*Feature Models*) para generar automáticamente diversos escenarios para la comprobación de la robustez de los programas de semáforos. Como caso de estudio, se realiza la validación de programas de semáforos en el área urbana de la ciudad de Málaga, generados mediante cuatro algoritmos de optimización (PSO, DE, Random Search y SCPG). El resultado es información validada sobre el programa de semáforos que mejor actúa para un mayor porcentaje de situaciones de tráfico diferentes.

Keywords: Modelos de Características, Metaheurísticas, *Pairwise Testing*, Priorización, Semáforos

1. Introduction

Hoy en día, todas las iniciativas para el desarrollo de la Ciudad Inteligente o *Smart City* [1] reciben un gran interés institucional y mediático por el impacto que generan sobre la sociedad en general. El concepto de *Smart City* engloba una serie de acciones y servicios coordinados, cuyo objetivo es conseguir la mejora de eficiencia a nivel energético, el consumo y la utilización sostenible de recursos, así como la aparición de mejores procesos operativos. En este sentido, la gestión del flujo del tráfico mediante la sincronización de semáforos constituye uno de los aspectos más sensibles en los entornos urbanos. En la actualidad existen pocas propuestas basadas en técnicas inteligentes para la sincronización de semáforos que obtienen soluciones eficientes [2,3,4]. No obstante, dichas soluciones están determinadas por instancias o escenarios de prueba con condiciones muy específicas y de variabilidad limitada. Por tanto, disponer de una estrategia de validación para diversos escenarios es crucial para este tipo de procesos, ya que determinarían la robustez de las soluciones aportadas.

Con esta motivación, proponemos en este trabajo una estrategia de validación basada en Modelos de Características (*Feature Models* - FM) [5], que sirven para representar adecuadamente sistemas de alta variabilidad. En este trabajo hemos desarrollado un modelo de gestión de tráfico, el cual recoge toda la variabilidad de características (posibles situaciones de tráfico) y la plasma en diferentes escenarios de movilidad urbana.

Entre las principales características que afectan al tráfico podemos enumerar: diferentes condiciones meteorológicas, tipología de vehículos, grado de pericia de los conductores, tiempo de simulación, etc. No obstante, el análisis de todos los posibles escenarios de tráfico es inviable debido a la gran cantidad de combinaciones de características a tener en cuenta. Por tanto, vamos a utilizar *Prioritized Genetic Solver* (PGS) [6], un algoritmo evolutivo de optimización diseñado para generar el conjunto mínimo de configuraciones o escenarios que cumplen el criterio de cobertura *pairwise*. Además, no todas las características tienen la misma probabilidad de ocurrencia o importancia, por lo que nuestro modelo de características ha sido ampliado con asignación de prioridades a características.

Como caso de estudio, vamos a centrarnos en un área urbana de Málaga. Para esto, utilizaremos el simulador de tráfico SUMO (Simulator of Urban Mobility) [7]. SUMO ofrece una fuente de información constante sobre el flujo de vehículos con: velocidades, consumos, emisiones, tiempos de viaje, etc.; y nos permite configurar escenarios realistas (como en el caso de Málaga) siguiendo direcciones y patrones reales de movilidad. En concreto, vamos a comprobar la robustez de programas de semáforos, previamente generados mediante cuatro algoritmos de optimización: *Particle Swarm Optimization* (PSO) [8], *Differential Evolution* (DE) [9], *Random Search* (RS) y *SUMO Cycle Programs Generator* (SCPG) [2]. Nuestro modelo de validación nos ayudará a seleccionar cual (o cuales) de estas soluciones es la más adecuada para un mayor número de escenarios diferentes de tráfico.

Las principales contribuciones de este artículo son las siguientes:

- Desarrollo de un Modelo de Características priorizado para la representación de un sistema de gestión (semafórica) de tráfico con características ponderadas acorde a su importancia o probabilidad de ocurrencia.
- Generación automática de escenarios ponderados para la validación de programas de semáforos por medio del algoritmo PGS.
- Análisis de los programas de semáforos para el caso de estudio de la ciudad de Málaga previamente generados por los algoritmos de optimización (PSO, DE, Random Search, SCPG) en dichos escenarios.

Este artículo se organiza de la siguiente manera. La Sección 2 presenta los conceptos previos sobre representación de variabilidad en Modelos de Características. La sección 3 describe el algoritmo PGS para la generación de escenarios de validación a partir del FM. La Sección 4 detalla el procedimiento experimental llevado a cabo y los análisis de resultados. Por último, en la Sección 5 se enumeran las conclusiones y el trabajo futuro.

2. Representación de Sistemas con Alta Variabilidad

Los modelos de características son un estándar *de facto* para modelar las características comunes y variables de un sistema (representadas por cajas etiquetadas) y sus relaciones (representadas con líneas) formando una estructura de tipo árbol, especificando así un conjunto de combinaciones de características que dan lugar a múltiples configuraciones diferentes [5]. Cada característica, aparte de la raíz, tiene una sola característica padre y puede tener un conjunto de características hijas. Nótese que una característica hija sólo puede ser incluida en una configuración si y sólo si, su padre es incluido también. Para ilustrar estos conceptos vamos a usar un FM (Figura 1) que hemos generado para representar la gestión de tráfico. Entre las principales características contamos con diferentes condiciones meteorológicas, tipología de vehículos, grado de pericia y reacción de los conductores, tiempo y cantidad de vehículos en la simulación. Por todo esto, el modelo representa 960 escenarios válidos diferentes, y éste nos servirá para generar un subconjunto de escenarios adecuado para testar los programas de semáforos.

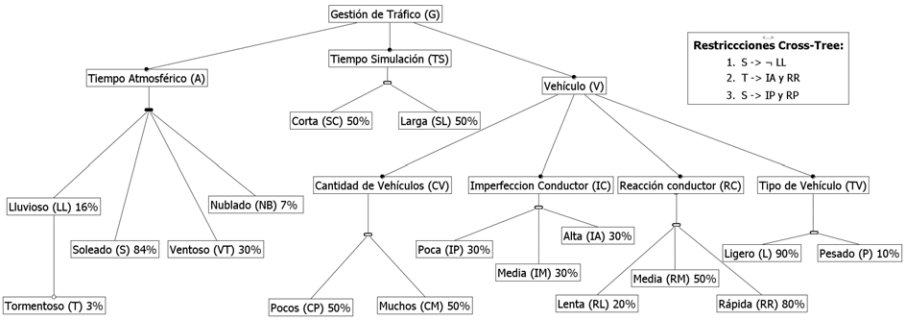


Figura 1: Modelo de gestión de tráfico con prioridades

Hay 4 tipos de relaciones jerárquicas entre características:

- *Características opcionales:* son representadas con un círculo vacío e indica que esta característica puede o no ser seleccionada si su padre es seleccionado. En nuestra instancia sería la característica "tormentoso" (T).
- *Características obligatorias:* son representadas con un circulo relleno, y deben ser seleccionadas si su padre es seleccionado. En nuestra instancia serían obligatorias: G, A, TS, V, CV, IC, RC y TV.
- *Relaciones Or-inclusivas:* son representadas como óvalos independientes rellenos que abarcan un conjunto de líneas que conectan la característica padre con las características hijas. Indican que al menos una característica debe ser seleccionada si el padre es seleccionado. En nuestra instancia, la relación de A con sus hijos es de este tipo.
- *Relaciones Or-exclusivas:* son representadas como óvalos independientes vacíos que abarcan un conjunto de líneas que conectan la característica padre con

las características hijas. Indican que exactamente una característica debe ser seleccionada si el padre es seleccionado. En nuestra instancia la relaciones de TS, CV, IC, RC y TV con sus respectivos hijos son Or-exclusivas.

Además de las relaciones padre-hijo, las características se pueden relacionar también con otras ramas del modelo de características, son las restricciones conocidas como *Cross-Tree Constraints (CTC)* [10]. Estas restricciones, así como las impuestas por las relaciones jerárquicas entre características, son expresadas y comprobadas usando lógica proposicional. Para el FM analizado en este trabajo hemos indicado las tres restricciones que se consideran en el área superior derecha de la Figura 1.

Estos modelos de características pueden ser ampliados para tener en cuenta una ponderación asociada a cada característica. En nuestra instancia, indicamos el peso de cada característica opcional en porcentaje, porque las características obligatorias siempre deben estar presentes (no necesitan ser ponderadas). A continuación resumimos la terminología básica que usamos durante este trabajo.

Una configuración o escenario es un par $[sel, \overline{sel}]$ donde sel y $\overline{sel}$ son respectivamente los conjuntos de características seleccionadas y no seleccionadas. Con respecto a las prioridades, un par priorizado *pc* está compuesto por dos características f_1 y f_2, tales que el peso del par es calculado como $pc.w = f_1.w * f_2.w$. Consecuentemente, la prioridad de una configuración es calculada como la suma de los pares priorizados que se generan de ella. Cabe recordar que una configuración es válida para un FM, *sii* dicha configuración no contradice ninguna restricción del FM.

Las pruebas de interacción combinatorias son un enfoque que construye *test suites* que nos llevan a probar de forma sistemática todas las configuraciones de un sistema [11]. En este trabajo vamos a utilizar el enfoque *pairwise*, que es el más utilizado en testing combinatorio y esta basado en la asunción de que la mayoría de errores originados en un parámetro es causado por la interacción de dos valores [12]. Así que este criterio es satisfecho si todos los pares de características $([(f_1, f_2), (f_1, \overline{f_2}), (\overline{f_1}, f_2) y (\overline{f_1}, \overline{f_2})])$ están presentes en al menos un escenario. Cuando se aplica esta técnica a FMs, la idea es seleccionar un conjunto de escenarios válidos, donde los errores posibles se manifiestan con alta probabilidad, sin tener que probar de forma exhaustiva todas las posibles configuraciones (960 en este caso de estudio). Además, gracias a la priorización, vamos a testar en primer lugar los escenarios con aquellas características que son más frecuentes o importantes en nuestros sistemas.

3. Generador de Escenarios: Prioritized Genetic Solver

El algoritmo *Prioritized Genetic Solver (PGS)* es un algoritmo evolutivo que construye una *test suite* (conjunto de escenarios) teniendo en cuenta prioridades y restricciones durante la generación. Este trabajo adapta y extiende el algoritmo *Genetic Solver* propuesto por Ferrer et al. [6] para trabajar con FMs y restricciones. PGS es un algoritmo genético constructivo que añade una nueva

configuración a la solución parcial en cada iteración hasta que todas las parejas de características hayan sido cubiertas. PGS se ha implementado empleando un framework de desarrollo llamado jMetal [13] y una herramienta para el manejo de FMs llamada FAMA [14]. Revisar la referencia del algoritmo [6] para más detalles.

El FM que vamos a explorar con este algoritmo es el presentado anteriormente en la Figura 1. Para fijar una prioridad a cada una de las características, se han seguido los siguientes criterios: con respecto al tiempo atmosférico se han utilizado datos del Instituto Nacional de Meteorología, concretamente de la estación meteorológica del aeropuerto de Málaga. Sobre la cantidad de vehículos ligeros y pesados, hemos recurrido a datos públicos sobre tráfico de la Delegación de Movilidad del Ayuntamiento de Málaga (http://movilidad.malaga.eu/). Los pesos del resto de características se han establecido atendiendo a la importancia de las mismas. Recordamos al lector que las características obligatorias tendrían un peso del 100 %, puesto que son obligatorias en toda configuración válida.

Tras la ejecución del algoritmo, PGS ha generado las configuraciones o escenarios que podemos ver en la Tabla 1, cumpliendo con el criterio de cobertura *pairwise*. En dicha tabla podemos ver que se han generado 10 escenarios diferentes para testar. Se han marcado las características que están presentes en cada escenario. En la última columna tenemos el peso total que aporta cada una de las configuraciones a la test suite. Por ejemplo, vemos como el escenario 1 (E1) aporta un 63,59 % de peso, mientras que los 4 últimos escenarios (E7-E10) suponen tan sólo alrededor de un 1 %. Esto quiere decir que hay que tener en cuenta escenarios diversos, pero corregir las deficiencias en algunos escenarios será mas importante que en otros.

E	G	A	LL	T	S	VT	NB	TS	SC	SL	V	CV	CP	CM	IC	IP	IM	IA	RC	RL	RM	RR	TV	L	P	P %
E1	✓	✓			✓			✓		✓	✓	✓		✓	✓		✓		✓			✓	✓	✓		63,59
E2	✓	✓			✓	✓		✓	✓		✓	✓	✓		✓	✓			✓		✓		✓	✓		22,37
E3	✓	✓	✓				✓	✓	✓		✓	✓	✓		✓			✓	✓			✓	✓		✓	7,09
E4	✓	✓	✓	✓		✓		✓		✓	✓	✓		✓	✓			✓	✓	✓			✓	✓		3,32
E5	✓	✓	✓	✓			✓	✓	✓		✓	✓		✓	✓		✓		✓		✓		✓		✓	1,41
E6	✓	✓			✓		✓	✓		✓	✓	✓	✓		✓	✓			✓	✓			✓		✓	1,21
E7	✓	✓				✓		✓		✓	✓	✓		✓	✓			✓	✓		✓		✓		✓	0,43
E8	✓	✓	✓			✓	✓	✓		✓	✓	✓		✓	✓	✓			✓			✓	✓	✓		0,34
E9	✓	✓	✓	✓		✓		✓	✓		✓	✓	✓		✓		✓		✓	✓			✓		✓	0,23
E10	✓	✓	✓	✓		✓	✓	✓		✓	✓	✓		✓	✓			✓	✓			✓	✓		✓	0,01

Tabla 1: Escenarios computados por el PGS con criterio de cobertura *pairwise*

4. Validación Experimental

En esta sección vamos a validar de forma experimental los programas de ciclos de semáforos generados por una serie de algoritmos (PSO, DE, Random Search y SCPG) [2] para la generación de programas de ciclos en semáforos.

Figura 2: Instancia de Málaga exportada al formato de SUMO

4.1. Caso de Estudio: Málaga

La instancia a partir de la cual se generaron todos los escenarios, así como los programas de ciclos optimizados para los semáforos, corresponde a un área urbana de aproximadamente 0,75 km^2 físicamente localizada en la ciudad de Málaga. Esta instancia consiste en una serie de ficheros con información proveniente de mapas digitales (OpenStreetMap) como: reglas de tráfico regulado, situación de las señales y otros elementos, posición de semáforos, direcciones, calles, intersecciones, glorietas, etc. Además, se ha determinado el número de vehículos circulando, así como sus velocidades a partir de las especificaciones de la Delegación de Movilidad del Ayuntamiento de Málaga. Esta información se obtuvo a partir de puntos sensorizados en ciertas calles de la ciudad calculando así una medida de la densidad de tráfico en varios intervalos temporales.

La ilustración de la Figura 2 muestra el área seleccionada para las ciudad tratada con sus correspondientes capturas de OpenStreetMap y SUMO, describiendo el proceso de generación de la instancia. Situada en la zona entre el centro de la ciudad y el puerto, está compuesta por calles con diferentes anchuras y longitudes, además de varias glorietas. Contiene intersecciones con un número de semáforos entre 4 y 16. El número de intersecciones estudiadas es de 70, con un total de 312 semáforos y entre 250 y 500 vehículos circulando. En nuestro estudio, cada uno de los vehículos recorre su propia ruta desde el origen hasta su destino circulando con una velocidad máxima de 50 km/h (velocidad típica en la áreas urbanas). Las rutas fueron generadas a priori de manera uniforme. El tiempo de simulación es de 500 o 1000 segundos según el escenario.

En cuanto al conductor de un vehículo, hay dos aspectos principales que caracterizan su conducción; el grado de imperfección al volante y el tiempo de reacción ante los estímulos. En SUMO estas características están representadas por 2 parámetros. El parámetro σ es un número real en el rango $[0 \dots 1]$ que pondera la destreza del conductor. Un valor de $\sigma = 1$ representa un conductor con muchas imperfecciones y un valor de $\sigma = 0$ representa el comportamiento de un conductor perfecto. Por otro lado tenemos el parámetro τ, que mide el tiempo de reacción en segundos del conductor ante cualquier estímulo. Además, tenemos otros parámetros como la duración de la simulación, la cantidad de vehículos y el porcentaje de vehículos ligeros frente a pesados. Un vehículo pesado supone una aceleración, desacelaración y velocidad máxima menor (35km/h) que un vehículo ligero. Las características opcionales seleccionadas a partir de nuestro modelo de gestión de tráfico van a generar escenarios de tráfico con distintas

características. En la Tabla 2 mostramos un resumen de lo que supone activar cada característica.

Características	LL	T	S	VT	NB	SC	SL	CP	CM
Parámetros	$\sigma+ = 0{,}1$ $\tau+ = 1$	$\sigma+ = 0{,}1$ $\tau+ = 1$	$\sigma- = 0{,}1$ $\tau- = 1$	$\sigma+ = 0{,}1$ $\tau+ = 1$	$\sigma+ = 0{,}1$ $\tau+ = 1$	500s	1000s	250 u.	500 u.
Características	IP	IM	IA	RP	RM	RA	L	P	-
Parámetros	$\sigma- = 0{,}1$	–	$\sigma+ = 0{,}1$	$\tau+ = 1$	–	$\tau- = 1$	95 %vl	85 %vl	-

Tabla 2: Activación de características y peso correspondiente

SUMO genera información acerca de la simulación de una instancia, a partir de la cual calculamos una serie de métricas. En este trabajo usamos las siguientes: vehículos que llegan a su destino (VLL), vehículos que no llegan a su destino (VNLL), CO2 emitido a la atmósfera (CO2), óxidos de Nitrógeno emitidos a la atmósfera (NOx), combustible consumido en el trayecto (FUEL) y tiempo global de viaje(GJT).

4.2. Configuración de los experimentos

Como comentamos anteriormente, hemos seleccionado las soluciones aportadas por 4 algoritmos [2] para someterlos a estudio: PSO en su versión *Standard 2011* [15], DE con mutación *DE/rand/1*, Random Search y SCPG [7]. Los tres primeros algoritmos son no-deterministas, por lo que hemos seleccionado 30 programas de semáforos (soluciones) para cada uno, generadas a priori para el caso de estudio. El cuarto algoritmo es el de la solución propuesta por SUMO (determinista), donde realizamos una única simulación por escenario propuesto (10 simulaciones de E1 a E10). A continuación, en el caso de los algoritmos no-deterministas, para cada escenario propuesto vamos a realizar además 30 ejecuciones independientes más por cada uno de los 30 programas de semáforos generados por los algoritmos. Todo esto suma 27.010 simulaciones independientes (3 algoritmos $\times$ 10 escenarios $\times$ 30 configuraciones de semáforos $\times$ 30 ejecuciones independientes + 10 ejecuciones de SCPG).

4.3. Análisis de los resultados

En la Figura 3 podemos observar el rendimiento de los algoritmos para las métricas VLL, CO2, NOx, Fuel y GJT por vehículo y todos los escenarios. Cada una de las métricas está normalizada para su correcta visualización. En esta figura podemos apreciar que PSO obtiene el máximo número de vehículos que llegan a destino con menor consumo de combustible (Fuel), y arrojando a la atmósfera menor cantidad de CO2 y NOx. A pesar de que el tiempo global de viaje por cada vehículo es máximo para PSO, esto se debe a que también los vehículos con destinos lejanos son capaces llegar a él. En esta figura también podemos apreciar que SCPG es el peor algoritmo porque tiene consumos altos de Fuel, CO2 y NOx, a pesar de que la media de vehículos que llegan a su destino es mínima.

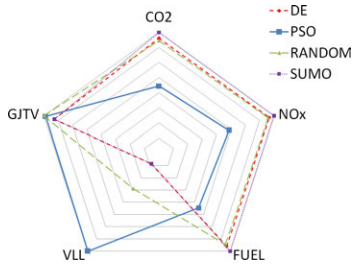


Figura 3: Diagrama normalizado de estrella

	PSO	DE	RANDOM	SCPG
Escenario 1	**455.00** (13)	269.64 (4)	333.16 (1)	248.64
Escenario 2	**197.51** (3)	125.29 (26)	142.19 (4)	113.39
Escenario 3	**172.45** (3)	111.00 (26)	125.00 (17)	98.67
Escenario 4	**260.32** (20)	172.03 (26)	187.84 (24)	145.61
Escenario 5	**133.39** (14)	96.80 (26)	100.00 (20)	78.90
Escenario 6	**241.90** (2)	186.25 (22)	214.58 (0)	170.19
Escenario 7	**368.90** (20)	205.90 (4)	246.39 (24)	194.13
Escenario 8	**341.41** (2)	200.93 (4)	235.32 (24)	184.19
Escenario 9	**147.54** (3)	96 (11)	107.87 (17)	91.26
Escenario 10	**272.32** (20)	172.87 (26)	191.35 (20)	146.41

Tabla 3: Mejor media de VLL y configuración que genera este resultado.

En la Tabla 3 podemos observar la media de vehículos que llegan a su destino para cada escenario y algoritmo. Destacamos entre paréntesis el programa de semáforos de cada algoritmo que obtiene el mejor resultado promediando 30 ejecuciones independientes. Podemos observar que los programas generados por PSO siempre obtienen la mejor media de vehículos que llegan a su destino. Destacamos además que el programa generado que obtiene el mejor resultado no es siempre el misma. Por ejemplo, para PSO, los programas 3 y 20 consiguen el mejor resultado en 3 de 10 escenarios. Esto quiere decir que no existe un programa de semáforos que sea el mejor para todos los posibles escenarios de tráfico. Incluso el programa 13, que consigue el mejor resultado para el escenario más probable (Escenario 1 con 63,59 %), no vuelve a ser el mejor para ningún otro escenario. Esta misma interpretación se da para los demás algoritmos. Por ello, pensamos que es necesario el entrenamiento de estos algoritmos para considerar los aspectos variables del tráfico que hemos plasmado en 10 escenarios diferentes.

En la Figura 4 observamos los diagramas de caja para 4 métricas representativas: C02 por vehículo, GJT por vehículo, VLL y VNLL. En estos diagramas no hemos tenido en cuenta los algoritmos, sino que hemos analizado por escenario para poner de manifiesto la variabilidad que podemos encontrar según las circunstancias del tráfico. Por un lado, analizamos el escenario más favorable atendiendo a los vehículos que llegan a destino, que es el Escenario 1. En este escenario los programas de semáforos analizados hacen que la mayoría de los vehículos lleguen al destino, arrojando una mínima cantidad de CO2 y con GJT por vehículo cercano a 500sg. Este escenario es tan favorable porque asume una las condiciones meteorológicas es soleada (S), con tiempo de simulación largo (SL), y con mayor cantidad de vehículos ligeros (VL). Por otro lado, el escenario más desfavorable es el Escenario 5, porque llegan muy pocos vehículos en comparación con los que no llegan a su destino, por lo que se arroja en proporción mucho CO2 a la atmósfera. Este escenario es tan desfavorable porque las condiciones meteorológicas son de lluvia (LL), tormenta (T) y niebla (NB), con simulación corta (SC), con 500 vehículos (CM), y un mayor porcentaje de vehículos pesados (P).

Este análisis nos da una clara idea de la necesidad de aplicación de las técnicas de priorización para tener en cuenta todos los posibles escenarios de tráfico.

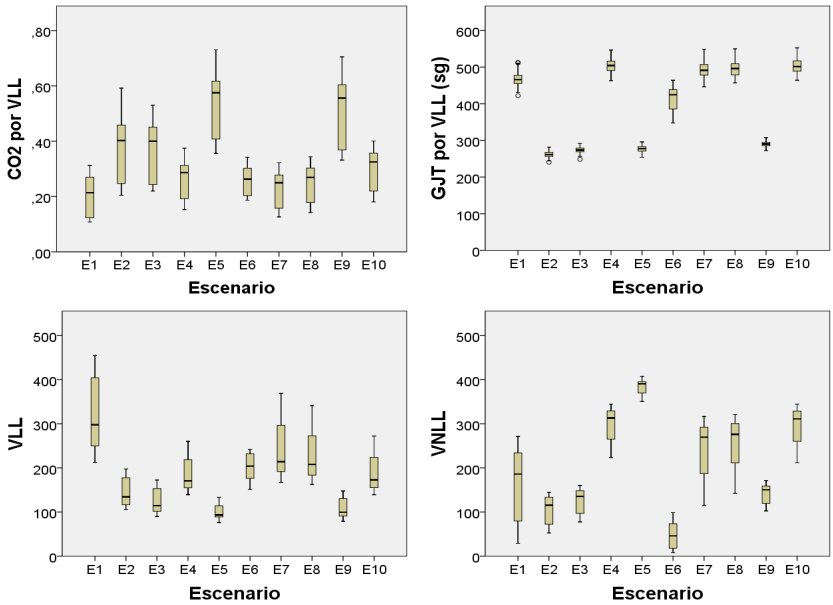


Figura 4: Diagramas de caja de CO2, GJT, VLL, y VNLL por escenario.

Anteriormente sólo se han considerado unas únicas situaciones de tráfico y eso provoca la generación de programas de semáforos limitados a condiciones ideales de tráfico. A partir de ahora podemos proponer algoritmos para generar programas de semáforos optimizados que se adapten a más situaciones diferentes de tráfico, ponderando aquellas que sean más probables en el día a día, pero sin olvidar las situaciones excepcionales.

5. Conclusiones

En este trabajo hemos propuesto una estrategia para la validación de programas de ciclo de semáforos. Hemos partido de un Modelo de Características que recoge la variabilidad que existe en un sistema complejo como es la gestión del tráfico, para, a continuación, mediante un algoritmo de optimización (PGS), generar un conjunto mínimo de escenarios que cumplen que todos los posibles pares de características se han validado. Además, hemos comprobado cómo se adaptan los programas de semáforos generados por los cuatro algoritmos propuestos PSO, DE, Random Search y SCPG. Este sistema de validación nos permite elegir un programa específico para situaciones de tráfico diferentes.

Analizando los resultados podemos concluir que PSO ha sido el mejor algoritmo generando programas de semáforos para todos los escenarios. No obstante, no existe un programa concreto, de los 30 generados por PSO, que haya sido el mejor para todos los escenarios. De hecho, para un programa generado por PSO, como máximo es el mejor en sólo 3 de 10 escenarios. Este hecho nos indica que la ponderación de cada escenario debe ser tenida en cuenta para la elección del mejor programa de semáforos generado por los algoritmos.

Una línea de trabajo futuro sería considerar varios escenarios diferentes en la evaluación del *fitness* de los algoritmos, para hacer que se generen programas de semáforos más robustos y adaptativos a las condiciones cambiantes de tráfico. Los escenarios no son igualmente probables, así que proponemos también considerar estos pesos de los escenarios para convertir el *fitness* en una función agregativa, ponderada según el resultado que obtengamos en la generación de los escenarios, a partir del Modelo de Características.

Acknowledgments Este trabajo está parcialmente financiado por el Ministerio de Economía y Competitividad (MINECO) del Gobierno de España y fondos FEDER con el proyecto roadME con TIN2011-28194 (`http://roadme.lcc.uma.es`). J. Ferrer y J. García-Nieto disfrutan de sendas becas con códigos BES-2012-055967 y BES-2009-018767 del MINECO.

Referencias

1. Caragliu, A., Del Bo, C., Nijkamp, P.: Smart cities in europe. Journal of Urban Technology **18**(2) (2011) 65–82
2. García-Nieto, J., Olivera, A., Alba, E.: Optimal cycle program of traffic lights with particle swarm optimization. IEEE Trans. on Evolutionary Computation (2013)
3. Medina, J.S., Moreno, M.G., Royo, E.R.: Applying a Traffic Lights Evolutionary Optimization Technique to a Real Case: Las Ramblas Area in Santa Cruz de Tenerife. IEEE Trans. on Evo. Comp. **12**(1) (2008) 25 –40
4. Nagatani, T.: Effect of speed fluctuations on a green-light path in a 2D traffic network controlled by signals. Physica A: Statistical Mechanics and its Applications **389**(19) (2010) 4105 – 4115
5. Kang, K., Cohen, S., Hess, J., Novak, W., Peterson, A.: Feature-Oriented Domain Analysis (FODA) Feasibility Study. Technical Report CMU/SEI-90-TR-21, Software Engineering Institute, Carnegie Mellon University (1990)
6. Ferrer, J., Kruse, P.M., Chicano, J.F., Alba, E.: Evolutionary algorithm for prioritized pairwise test data generation. In: GECCO, ACM (2012) 1213–1220
7. Krajzewicz, D., Bonert, M., Wagner, P.: The open source traffic simulation package SUMO. RoboCup 2006 Infrastructure Simulation Competition (2006)
8. Kennedy, J., Eberhart, R.C.: Swarm Intelligence. Morgan Kaufmann Publishers, San Francisco, California (2001)
9. Price, K., Storn, R., Lampinen, J.: Differential Evolution: A Practical Approach to Global Optimization. Natural Computing. Springer (2005)
10. Benavides, D., Segura, S., Cortés, A.R.: Automated analysis of feature models 20 years later: A literature review. Inf. Syst. **35**(6) (2010) 615–636
11. Cohen, M.B., Dwyer, M.B., Shi, J.: Constructing interaction test suites for highly-configurable systems in the presence of constraints: A greedy approach. IEEE Trans. Software Eng. **34**(5) (2008) 633–650
12. Stevens, B., Mendelsohn, E.: Efficient software testing protocols. In: Proceedings of the 1998 conference of the Centre for Advanced Studies on Collaborative research. CASCON '98, IBM Press (1998) 22–
13. Durillo, J.J., Nebro, A.J.: jMetal: A java framework for multi-objective optimization. Advances in Engineering Software **42**(10) (2011) 760 – 771
14. Trinidad, P., Benavides, D., Ruiz-Cortes, A., Segura, S., Jimenez, A.: Fama framework. In: SPLC '08. 359–359
15. Clerc, M.: Standard PSO 2011. Technical Report [online] http://www.particleswarm.info/, Particle Swarm Central (January 2011)

Generador de Instancias de Problemas de Optimización Combinatoria Basados en Permutaciones

Leticia Hernando, Alexander Mendiburu, and Jose A. Lozano

Intelligent Systems Group,
Departmento de Ciencias de la Computación e Inteligencia Artificial,
Universidad del País Vasco UPV/EHU, San Sebastián
{leticia.hernando,alexander.mendiburu,ja.lozano}@ehu.es

Resumen El diseño de instancias de problemas de optimización combinatoria es un aspecto muy importante en el área de la optimización. En la evaluación o comparación entre algoritmos, uno puede encontrarse con que su comportamiento es diferente dependiendo de las propiedades de las instancias con las que se trabaja. Por ejemplo, existen métodos para estimar el número de óptimos locales de una instancia que dependen directamente del tamaño de las bases de atracción de los óptimos locales. Por lo tanto, es interesante poder disponer de un conjunto grande de instancias de diferentes características. Para ello, se han propuesto métodos que diseñan instancias de problemas de optimización combinatoria para espacios continuos o incluso binarios. Además, nosotros propusimos un generador de instancias cuando se trata del espacio de permutaciones. Este generador utiliza el modelo probabilístico conocido como modelo de Mallows y proporciona instancias con características específicas que afectan al número de óptimos locales o a los tamaños de sus bases de atracción. En este artículo comprobamos que nuestro generador es suficientemente flexible para generar instancias de niveles de complejidad muy diferentes. Estudiamos los parámetros que intervienen en él, observando la influencia que tienen en las instancias que se generan.

1. Introducción

La generación de instancias de Problemas de Optimización Combinatoria (POCs) es un factor esencial en la comparación y análisis de diferentes algoritmos metaheurísticos, así como para evaluar algoritmos de estimación del número de óptimos locales de una instancia. El diseño de un generador de instancias moldeable es de gran relevancia puesto que modificando el valor de sus parámetros se permite el control de las propiedades de las instancias.

Dada la relevancia de este tema, se pueden encontrar en la literatura varias propuestas. En [1] se propone un generador para espacios binarios, y en un trabajo más reciente [2] se muestra un modelo que genera funciones multimodales para optimización en espacios continuos. Particularmente, el estudio desarrollado en [3] tiene una gran relevancia para nuestro generador. Los autores proponen

un generador de instancias para un espacio de búsqueda continuo basado en una combinación de Gaussianas, que puede ser modelado por unos parámetros. Basándonos en ese trabajo, propusimos un generador de instancias de POCs cuyo espacio de búsqueda es el espacio de las permutaciones [4]. Este generador utiliza una combinación de un modelo probabilístico con diferentes parámetros para permutaciones llamado el modelo de Mallows.

En este trabajo, estudiamos la flexibilidad del generador para crear instancias de diferentes niveles de complejidad. Para medir la complejidad de las instancias, las resolveremos con un algoritmo de búsqueda local, considerando que son más complejas cuantos más óptimos locales tienen. Además, creemos que esta medida de dificultad puede ser también adecuada para evaluar algoritmos poblaciones. Estudiamos la importancia de cada uno de los factores y su influencia directa con características de la instancia como el tamaño de las bases de atracción de los óptimos locales. En la parte de la experimentación observamos cómo varían estas propiedades según modificamos los parámetros.

El resto del artículo está organizado del siguiente modo. El modelo de Mallows se explica en la Sección 2. En la Sección 3, presentamos el generador de instancias y damos tres ejemplos para obtener instancias de diferentes niveles de complejidad. La experimentación se presenta en la Sección 4 donde se observa la evolución de las propiedades de la instancia al variar los valores de los parámetros. Finalmente, las conclusiones y líneas futuras se exponen en la Sección 5.

2. Modelo de Mallows

El modelo de Mallows [5] es un modelo probabilístico exponencial para permutaciones y está basado en una distancia. Esta distribución viene definida por dos parámetros: la permutación central σ_0, y el parámetro de dispersión θ. Sea Ω el conjunto de todas las permutaciones de tamaño n, la distribución de Mallows se define para cada $\sigma \in \Omega$ como: $p(\sigma) = e^{-\theta d(\sigma_0,\sigma)}/Z(\theta)$ donde $Z(\theta) = \sum_{\sigma\in\Omega} e^{-\theta d(\sigma_0,\sigma)}$ es un término de normalización y $d(\sigma_0,\sigma)$ es la distancia entre la permutación central σ_0 y σ. Una de las distancias utilizadas normalmente es la Kendall tau. Dadas dos permutaciones σ_1 y σ_2, esta métrica cuenta el mínimo número de swaps adyacentes necesarios para convertir σ_1 en σ_2. Bajo esta distancia, el término de normalización $Z(\theta)$ tiene forma cerrada y no depende de σ_0: $Z(\theta) = \prod_{j=1}^{n-1}(1 - e^{-(n-j+1)\theta})/(1 - e^{-\theta})$.

Nótese que si $\theta > 0$, entonces σ_0 es la permutación con la probabilidad más alta. Las demás permutaciones $\sigma \in \Omega - \{\sigma_0\}$ tienen probabilidad inversa-exponencialmente proporcionales a θ y a su distancia a σ_0. Por lo tanto, la distribución de Mallows puede ser considerada análoga a la distribución Gaussiana pero en el espacio de permutaciones.

3. Generador de Instancias

En [4] se presentó un generador de instancias para POCs basados en permutaciones. Este generador está constituido por diferentes parámetros que, según

su valor, proporcionan diferentes características en las instancias. En esta sección se explica teóricamente en qué consiste este generador basado en una combinación de modelos de Mallows y que consta de la resolución de un problema de programación lineal. La función a optimizar para este problema será elegida de manera adecuada para poder obtener la instancia con las características deseadas. También mostramos tres ejemplos de cómo modificar los parámetros para obtener dichas propiedades.

El generador de instancias presentado utiliza $3m$ parámetros que son: m permutaciones centrales $\{\sigma_1, ..., \sigma_m\}$, m parámetros de dispersión $\{\theta_1, ..., \theta_m\}$ y m pesos $\{w_1, ..., w_m\}$. Se genera un modelo de Mallows $p_i(\sigma|\sigma_i, \theta_i), \forall i \in \{1, ..., m\}$ para cada una de las m permutaciones centrales σ_i y para cada parámetro de dispersión θ_i. A cada permutación del espacio de búsqueda $\sigma \in \Omega$ se le debe asignar un valor de función objetivo. Este valor se define de la siguiente manera: $f(\sigma) = \max\limits_{1 \leq i \leq m} \{w_i p_i(\sigma|\sigma_i, \theta_i)\}$. Variando estos parámetros de manera adecuada, obtendremos instancias con diferentes propiedades, y por lo tanto, muy distintos niveles de complejidad. En esta sección se analizan estas propiedades y la manera de obtener instancias que las satisfagan.

El primer punto a tener en cuenta, y que tiene mucha relevancia a la hora de crear instancias de diferentes niveles de complejidad, es el número de óptimos locales. Todo óptimo local será una permutación central, puesto que son a las que mayor valor de función objetivo se les asigna por la propia definición de función de Mallows. Un factor importante a observar, es si se cumplirá que todas las permutaciones centrales σ_i's serán óptimos locales. Es decir, que si generamos una instancia con m funciones de Mallows, ésta tenga exactamente m óptimos locales. Para que $\{\sigma_1, ..., \sigma_m\}$ sean óptimos locales, tienen que cumplir que $d(\sigma_i, \sigma_j) \geq 2$, $\forall i \neq j$, puesto que dos permutaciones vecinas no pueden ser óptimos locales al mismo tiempo. Además, un problema con el que podemos encontrarnos es que no todas las permutaciones centrales sean óptimos locales, puesto que puede haber alguna que quede escondida bajo una función de Mallows centrada en otra permutación distinta. Esto ocurre cuando el máximo valor encontrado para una permutación central σ_i viene dado por la función de Mallows $p_j(\sigma_i|\sigma_j, \theta_j)$ que está centrada en otra permutación central σ_j. En este caso, la función de Mallows $p_i(\sigma|\sigma_i, \theta_i)$ centrada en σ_i desaparece y σ_i deja de ser óptimo local. Luego una segunda restricción que se debe cumplir para que toda permutación central sea óptimo local es que el valor de función objetivo de σ_i tiene que ser alcanzado por el i-ésimo modelo de Mallows. Es decir:

$$f(\sigma_i) = \max_{1 \leq k \leq m} \{w_k p_k(\sigma_i|\sigma_k, \theta_k)\} = w_i p_i(\sigma_i|\sigma_i, \theta_i) = w_i \frac{e^{-\theta_i d(\sigma_i, \sigma_i)}}{Z(\theta_i)} = \frac{w_i}{Z(\theta_i)} \quad (1)$$

Además, por la propia definición de óptimo local se tiene que cumplir la siguiente restricción:

$$f(\sigma_i) > f(\sigma), \quad \forall \sigma \quad \text{tal que} \quad d(\sigma_i, \sigma) = 1. \quad (2)$$

Para que (2) se satisfaga, y teniendo en cuenta (1), necesitamos que se cumpla lo siguiente:

Algorithm 1 Algoritmo para generar instancias

1. Se elige m
2. Se eligen las permutaciones centrales: $\sigma_1, \sigma_2, ..., \sigma_m$
3. Se eligen los parámetros de dispersión: $\theta_1, \theta_2, ..., \theta_m$
4. Se resuelve el problema de programación lineal en los pesos w_i, $\forall i \in \{1, ..., m\}$:

$$min/max\{G(w_1, ..., w_m)\}$$

$$\frac{w_i}{Z(\theta_i)} > w_j \frac{e^{-\theta_j (d(\sigma_i, \sigma_j) - 1)}}{Z(\theta_j)} \quad (\forall i, j \in \{1, ..., m\})$$

5. Se asigna a cada $\sigma \in \Omega$ el valor de función objetivo: $f(\sigma) = \max\limits_{1 \leq i \leq m} \{w_i \frac{e^{-\theta_i d(\sigma_i, \sigma)}}{Z(\theta_i)}\}$

$\frac{w_i}{Z(\theta_i)} > w_j p_j(\sigma),\ \forall \sigma$ tal que $d(\sigma_i, \sigma) = 1$, para j=1,...,m.

Sin embargo, sabiendo que si $\sigma \in \Omega$ es tal que $d(\sigma_i, \sigma) = 1$, se cumple que $d(\sigma_j, \sigma) = d(\sigma_j, \sigma_i) - 1$, o bien, $d(\sigma_j, \sigma) = d(\sigma_j, \sigma_i) + 1$, la ecuación (2) se puede escribir de la siguiente manera:

$$\frac{w_i}{Z_i(\theta_i)} > \frac{w_j}{Z_j(\theta_j)} e^{-\theta_j (d(\sigma_i, \sigma_j) - 1)}, \quad j \in \{1, 2, ..., m\}, i \neq j. \tag{3}$$

La elección de los parámetros θ_i's es de gran importancia puesto que están relacionados con las características que tendrá la instancia obtenida. Nótese que una vez fijados los parámetros θ_i's, las desigualdades anteriores son lineales en los w_i's. Por tanto, los valores de los w_i's se pueden obtener como solución de un problema de programación lineal. No obstante, no tenemos definida ninguna función objetivo a maximizar/minimizar para nuestro problema de programación lineal. Esta función la escogeremos siguiendo ciertos criterios que dependen de las características que queremos que tenga la instancia que estamos creando. A continuación se explica con detalle tres ejemplos de posibles funciones objetivo del problema de programación lineal, así como se dan nociones de cómo deben ser elegidos los θ_i's para crear instancias de tres niveles de complejidad diferentes al resolverse éstas con algoritmos de búsqueda local: instancias fáciles donde el óptimo global tiene una base de atracción mucho mayor que los demás óptimos locales, instancias difíciles donde el óptimo global tiene una base de atracción mucho menor que estos, e instancias donde las bases de atracción de todos los óptimos locales son similares. Véase Algoritmo 1 para observar cómo actúa el generador de manera general.

3.1. Óptimo Global con Base de Atracción Grande

En este apartado estudiamos la adecuada elección de los parámetros para obtener una instancia que resulte sencilla al resolverse con un algoritmo basado en una búsqueda local. Es decir, en la que la base de atracción del óptimo global contenga una gran proporción de soluciones del espacio de búsqueda. Nos basamos en la elección de tres factores: distribución de las permutaciones centrales,

elección de la función objetivo a optimizar para el problema de programación lineal y la elección de los valores de los parámetros de dispersión θ_i.

Suponemos que $\sigma_1, \sigma_2, ...\sigma_m$ son los óptimos locales donde, sin pérdida de generalidad, asumimos que σ_1 es el óptimo global. Escogiendo σ_1 lo más alejado posible del resto de permutaciones centrales favorecemos que éste tenga una base de atracción grande, pues podrá contener en su base de atracción a todas las permutaciones σ tales que $d(\sigma_1, \sigma) < d(\sigma_1, \sigma_j)$ siendo σ_j la permutación central más cercana a σ_1. En esta situación, la diferencia entre el valor de función objetivo de σ_1 y el del resto de permutaciones centrales podría considerarse como la función objetivo a maximizar para el problema de programación lineal: $G_{\mathrm{M}axGO} = max\left\{\frac{w_1}{Z(\theta_1)} - \sum_{i=2}^{m}\frac{w_i}{Z(\theta_i)}\right\}$. De esta manera ayudamos a que las permutaciones σ tales que $d(\sigma_1, \sigma) < d(\sigma_1, \sigma_j)$ terminen en la base de atracción de σ_1 y no de σ_j. Por último, tenemos en cuenta que cuanto mayor es θ más rápido decrece el valor asignado por la función de Mallows, y menor el valor de los vecinos de la permutación central. Luego en este caso, consideraríamos un valor pequeño para θ_1 mientras que lo elegimos grande para θ_i, $i \in \{2, ..., m\}$.

3.2. Óptimo Global con Base de Atracción Pequeña

Estudiamos ahora la elección de los parámetros cuando se quiere obtener una instancia donde el óptimo global tiene una base de atracción pequeña. Como en el caso anterior debemos elegir una correcta distribución de las permutaciones centrales, una función objetivo a optimizar adecuada para el problema de programación lineal, y los valores de los parámetros de dispersión θ_i que conviene.

Asumimos que σ_1 es el óptimo global, escogiéndolo esta vez lo más cercano posible del resto de permutaciones centrales. El razonamiento es análogo al caso anterior. Esto es, cuanto más cerca esté σ_1 de σ_j, $j = 2, ..., m$, menor es la cantidad de permutaciones que hay entre ellas que son las que mayor probabilidad podrían tener de estar en la base de atracción de σ_1. En este caso, se propone minimizar la diferencia entre el valor de función objetivo del óptimo global σ_1 y el de las permutaciones que están a distancia 1 de él: $G_{\mathrm{M}inGO} = min\left\{\frac{w_1}{Z(\theta_1)} - \frac{1}{m-1}\sum_{i=2}^{m} w_i \cdot \frac{1}{Z(\theta_i)} e^{-\theta_i(d(\sigma_1,\sigma_i)-1)}\right\}$.De este modo, al haber elegido el óptimo global cercano a los demás óptimos locales, y al estar minimizando la diferencia de valor de función objetivo con sus permutaciones vecinas, se obliga a que las demás permutaciones centrales tengan valores altos de función objetivo. Por lo tanto, las permutaciones σ tales que $1 < d(\sigma_1, \sigma) < d(\sigma_1, \sigma_j)$, siendo σ_j otra permutación central distinta a σ_1, tienen altas probabilidades de estar en la base de atracción de σ_j. Por lo explicado en el caso anterior, elegimos un valor de θ_1 mucho mayor que el de θ_i, $\forall i \in \{2, ..., m\}$.

3.3. Óptimos Locales con Tamaños Similares de Base de Atracción

Por último, estamos interesados en crear instancias en las que todos los óptimos locales tienen bases de atracción de un tamaño similar. Este es un caso favorable para muchos algoritmos de estimación del número de óptimos locales.

Considerando que lo que se quiere obtener son óptimos locales con tamaños similares de bases de atracción, éstos deberán estar repartidos de una manera lo más uniformemente posible por el espacio de búsqueda. De esta forma, se evita que un óptimo local demasiado alejado del resto pueda atraer en su base de atracción un gran número de soluciones. Sin pérdida de generalidad, consideramos que σ_1 es el óptimo global, y que σ_m es el óptimo local con menor valor de función objetivo. La función a minimizar en este caso es la diferencia entre el valor de función objetivo de σ_1 y el de σ_m (e implícitamente minimizar la diferencia de los valores de función objetivo de todos los óptimos locales): $G_{SimAB} = \text{mín}\left\{\frac{w_1}{Z(\theta_1)} - \frac{w_m}{Z(\theta_m)}\right\}$. De esta manera se fuerza a que todos los óptimos locales tengan un valor de función objetivo muy similar, y no haya grandes diferencias y por tanto, no tienda ninguno a atraer demasiadas soluciones a su base de atracción. Por este mismo motivo, en este caso consideramos que es adecuado tomar todos los θ_i $(i = 1, ..., m)$ iguales.

4. Experimentación

El objetivo de la experimentación es estudiar la importancia de la elección de los parámetros del modelo presentado y de la función a optimizar que interviene en el generador. Si controlamos estos parámetros, seremos capaces de controlar la dificultad de las instancias al resolverlas con algoritmos de búsqueda local, y que intuimos también con algoritmos poblacionales. En esta sección mostramos experimentos que corroboran lo explicado teóricamente en la Sección 3.

4.1. Diseño de los Experimentos

Trabajamos en el espacio de permutaciones de tamaño n=6 y generamos instancias para 10, 20 y 30 modelos de Mallows basados en la distancia Kendall tau, considerando las cinco siguientes opciones para elegir los valores de θ_i:

1. Elegimos m valores de θ_i según una distribución uniforme $U[1.5, 4.5]$.
2. Elegimos m valores de θ_i según una distribución normal $N(2, 1)$.
3. Elegimos $\theta_i = 3$, $\forall 1 \leq i \leq m$.
4. Elegimos $\theta_1 = 1.5$ y $\theta_i = 3.5$, $\forall 2 \leq i \leq m$, suponiendo σ_1 el óptimo global.
5. Elegimos $\theta_1 = 3.5$ y $\theta_i = 1.5$, $\forall 2 \leq i \leq m$, suponiendo σ_1 el óptimo global.

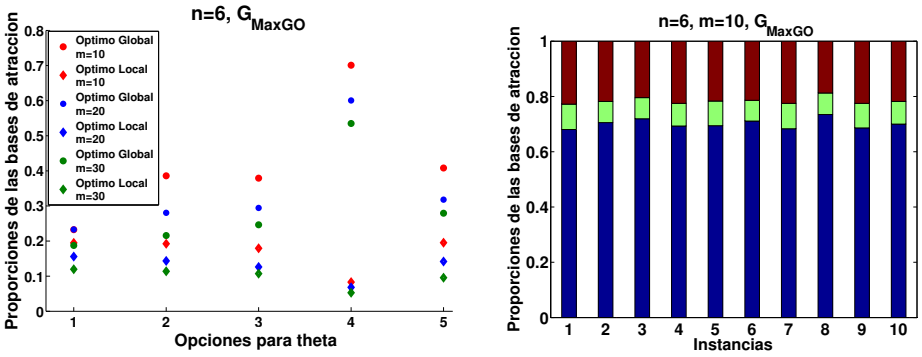
Consideramos las funciones objetivo a optimizar en el problema de programación lineal: G_{MaxGO}, G_{MinGO} y G_{SimAB} (secciones 4.2, 4.3 y 4.4, respectivamente).

4.2. Óptimo Global con Base de Atracción Grande

Generamos 10 instancias para el tamaño de permutación $n = 6$ para cada una de las opciones indicadas en el apartado 4.1, para dar valores a los diferentes θ_i. Calculamos los valores de los pesos w_i resolviendo el problema de programación lineal, con las restricciones (3) y considerando la función a optimizar: G_{MaxGO}.

Las permutaciones centrales son elegidas de la siguiente manera: se parte de una permutación inicial σ_1 y se obtienen todas las permutaciones σ_i tales que $d(\sigma_1, \sigma_i) = 2$. A continuación, se cogen las permutaciones σ_j tales que $d(\sigma_1, \sigma_j) = 4$. Y así sucesivamente, seleccionando siempre las permutaciones a distancia par para evitar que dos de ellas estén a distancia 1 y, por lo tanto, no puedan ser óptimos locales al mismo tiempo. Cuando $m - 1$ permutaciones han sido añadidas, el proceso se detiene. Finalmente, la última permutación central es la permutación que está a la máxima distancia posible de σ_1 $(n(n-1)/2)$.

Con nuestro generador obtenemos el valor de función objetivo de cada solución del espacio de búsqueda. Podemos conocer qué soluciones son óptimos locales, así como las que están contenidas en su base de atracción. Teniendo en cuenta las proporciones de los tamaños de estas bases de atracción, en la Figura 1(a) mostramos, para cada opción 1-5 de la elección de los valores de θ, la media de las proporciones de los tamaños de las bases de atracción del óptimo global (punto), así como del óptimo local con mayor base de atracción (rombo), de las 10 instancias. Las instancias creadas con 10, 20 y 30 óptimos locales se representan en rojo, en azul y en verde respectivamente. Observamos que los mejores resultados se obtienen para la opción número 4, que efectivamente es la que proponíamos en la teoría. Evidentemente, a medida que aumenta el número de óptimos locales, la proporción disminuye. Para visualizar mejor estos resultados, en la Figura 1(b) hemos representado para la opción número 4 de la elección de los valores de θ_i, las proporciones obtenidas en las 10 instancias al elegir 10 óptimos locales. La proporción del tamaño de la base de atracción del óptimo global está indicado en azul, mientras que la del óptimo local con mayor base de atracción está en verde para cada una de las instancias.



(a) (b)

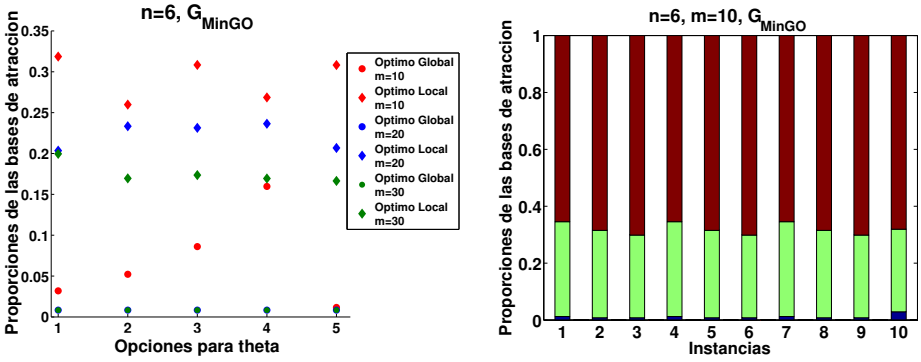
Figura 1. Resultados para 10 instancias generadas para tamaño de permutación n=6 y m=10, 20 y 30 funciones de Mallows, optimizando G_{MaxGO}. En 1(a) se representa la media de 10 proporciones obtenidas para el optimo global y el óptimo local con mayor tamaño de base de atracción para las 5 opciones de θ. En 1(b) se representan las proporciones del tamaño de las bases de atracción de estos óptimos para las 10 instancias cuando se elige la opción 4 para los θ_i

4.3. Óptimo Global con Base de Atracción Pequeña

Generamos 10 instancias para permutaciones de tamaño $n = 6$ para las 5 opciones diferentes mencionadas para dar valores a θ_i $(1 \leq i \leq m)$. Utilizamos G_{MinGO} en el problema de programación lineal para obtener los pesos w_i.

Para elegir las permutaciones centrales comenzamos con una inicial σ_1 que será considerada el óptimo global, y se obtienen todas las permutaciones σ_i tales que $d(\sigma_1, \sigma_i) = 2$. A continuación, se cogen las permutaciones σ_j tales que $d(\sigma_1, \sigma_j) = 4$. Y así sucesivamente, siempre cogiendo las permutaciones a distancia par. El proceso se detiene en el momento en el que se tiene m permutaciones.

En la Figura 2(a) mostramos la media de las proporciones de los tamaños de las bases de atracción del óptimo global (punto) y del óptimo local con mayor base de atracción (rombo) de las 10 instancias para cada opción 1-5. Se representan los resultados para las instancias creadas con 10, 20 y 30 óptimos locales, en rojo, azul y verde respectivamente. Cuando el numero de óptimos locales es menor ($n = 10$), la opción para la que se obtiene menor tamaño de base de atracción del óptimo global, es la opción 5 que es la que se mencionaba en la Sección 3.2. Cuando $m = 20$ y 30, el tamaño de la base de atracción del óptimo global es el mínimo posible en todos los casos. Esto es, que la base de atracción solamente contiene a él mismo y a las permutaciones que están a distancia 1. Elegimos la opción 5 en la Figura 2(b) para representar las proporciones obtenidas en las 10 instancias al elegir 10 óptimos locales. La proporción del tamaño de la base de atracción del óptimo global está indicado en azul, mientras que la del óptimo local con mayor base de atracción en verde (en rojo, la proporción restante).



(a) (b)

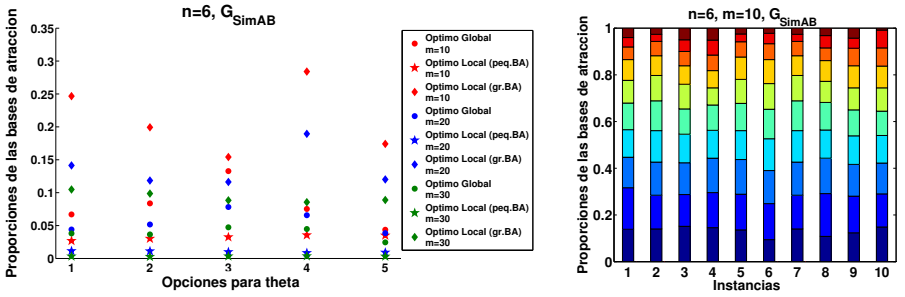
Figura 2. Resultados para 10 instancias generadas para tamaño de permutación n=6 y m=10, 20 y 30 funciones de Mallows, optimizando G_{MinGO}. En 2(a) se representa la media de 10 proporciones obtenidas para el optimo global y el óptimo local con mayor tamaño de base de atracción para las 5 opciones de θ. En 2(b) se representan las proporciones del tamaño de las bases de atracción de estos óptimos para las 10 instancias cuando se elige la opción 5 para los θ_i

4.4. Óptimos Locales con Tamaños Similares de Base de Atracción

Por último hemos generado también 10 instancias para el tamaño de permutación $n = 6$ considerando las opciones 1-5 para dar valores a los diferentes θ_i. Para calcular los valores de los pesos w_i hemos considerado optimizar G_{SimAB}.

Para obtener las permutaciones centrales, comenzamos con una permutación inicial σ_1 que será el óptimo global, después se elige una permutación aleatoria σ_2 tal que $d(\sigma_1, \sigma_2) = 2$. A continuación, se coge una permutación σ_3 tal que $d(\sigma_1, \sigma_3) = 4$. Y así sucesivamente, se van cogiendo permutaciones σ_i tales que $d(\sigma_1, \sigma_i) = 2 \cdot (i - 1)$. Cuando se llega a elegir una permutación que está a la distancia máxima $(n(n-1)/2$, o bien, $n(n-1)/2 - 1)$, se vuelve a empezar tomando una permutación σ_j tal que $d(\sigma_1, \sigma_j) = 2$, $d(\sigma_1, \sigma_{j+1}) = 4$, etc. El proceso finalizará cuando tengamos m permutaciones.

En la Figura 3(a) se muestran, para cada opción 1-5 de la elección de los valores de θ, la media de las proporciones de los tamaños de las bases de atracción del óptimo global (punto), así como del óptimo local con mayor base de atracción (rombo), y del óptimo local con menor base de atracción (estrella) de las 10 instancias. Se representan los resultados para las instancias creadas con 10, 20 y 30 óptimos locales, en rojo, azul y verde respectivamente. Si se toma la diferencia entre las proporciones de los dos óptimos locales con mayor y menor tamaño de base de atracción, resulta menor para la opción 3, tal y como se preveía en la Sección 3.3. Se observa en la Figura 3(b) al detalle las proporciones de las diferentes bases de atracción de todos los óptimos locales para la opción 3 de los valores de θ_i, para 10 óptimos locales. El óptimo global está representado en azul más oscuro, y siguiendo de abajo a arriba se pueden distinguir las proporciones de los tamaños de las bases de atracción de los diferentes óptimos locales, ordenadas éstas de mayor a menor.



(a) (b)

Figura 3. Resultados para 10 instancias generadas para tamaño de permutación n=6 y m=10, 20 y 30 funciones de Mallows, optimizando G_{SimAB}. En 3(a) se representa la media de 10 proporciones obtenidas para el optimo global y los óptimos locales con mayor y menor tamaño de base de atracción para las 5 opciones de θ. En 3(b) se representan las proporciones del tamaño de las bases de atracción de lo óptimos locales para las 10 instancias cuando se elige la opción 3 para los θ_i

5. Conclusiones y trabajo futuro

En este trabajo, mostramos la importancia de cada uno de los parámetros del generador de instancias de problemas de optimización combinatoria basados en permutaciones propuesto en [4]. Damos tres ejemplos para elegir estos factores de manera adecuada para obtener la instancia con el nivel de complejidad deseado. Básicamente, fijado el número m de modelos de Mallows, estudiamos los tamaños de las bases de atracción de los óptimos locales al modificar la distribución de los óptimos locales en el espacio de búsqueda, la función objetivo a optimizar para el problema de programación lineal y la elección de los parámetros θ_i $(i = 1, ..., m)$.

Conseguimos crear instancias de tres niveles de complejidad muy difierentes: instancias donde la base de atracción del óptimo global es grande, pequeña, y donde los tamaños de las bases de atracción de los óptimos locales son similares. Se observa que la distancia a la que está el óptimo global respecto del resto de óptimos locales, está directamente relacionada con el tamaño de su base de atracción. Además las propiedades de la instancia varían según cambiamos las funciones objetivo a optimizar, así como según variamos los valores de θ_i. Concluimos, que si θ es grande para cierto óptimo local, esto provoca que el tamaño de su base de atracción sea menor que si se le asigna un θ pequeño.

Basándonos en estos resultados, consideramos que la herramienta es lo suficientemente flexible como para generar instancias de dificultad similar a las encontradas en problemas tales como Traveling Salesman Problem, Flowshop Scheduling Problem, etc. Más aún, consideramos que nuestro modelo es suficientemente flexible como para representar la complejidad de problemas reales.

6. Agradecimientos

Este trabajo es parcialmente financiado por Saiotek y programas de Grupos de Investigación 2013-18 (IT-609-13) (Gobierno Vasco), TIN2010-14931 (Ministerio de Ciencia y Tecnología), red en biomedicina computacional COMBIOMED (Instituto de Salud Carlos III) y por el Proyecto NICaiA PIRSES-GA-2009-247619 (Comisión Europea). Leticia Hernando tiene beca del Gobierno Vasco.

Referencias

1. De Jong, K.A., Potter, M.A., Spears, W.M.: Using problem generators to explore the effects of epistasis. In: Seventh International Conference on Genetic Algorithms, Morgan Kaufmann (1997) 338–345
2. Rönkkönen, J., Li, X., Kyrki, V., Lampinen, J.: A framework for generating tunable test functions for multimodal optimization. Soft Computing (2010) 1–18
3. Gallagher, M., Yuan, B.: A general-purpose tunable landscape generator. IEEE Transactions on Evolutionary Computation **10**(5) (October 2006) 590–603
4. Hernando, L., Mendiburu, A., Lozano, J.A.: Generating Customized Landscapes in Permutation-based Combinatorial Optimization Problems. In: Learning and Intelligent OptimizatioN Conference (LION 7), Catania, Italy (2013)
5. Mallows, C.L.: Non-null ranking models. Biometrika **44**(1-2) (1957) 114–130

Estudiando la Creatividad mediante Algoritmos Evolutivos Desconectados

Lilian Navarro[1], Francisco Fernández de Vega[2], Cayetano Cruz[2], and Francisco Chávez[2]

[1] Escuela de Arte y Superior de Diseño, C/. Paseo de arte y oficios, S/N, 06800, Mérida, España
leanan79@hotmail.com
[2] Universidad de Extremadura, C/. Santa Teresa de Jornet, 38, 06800, Mérida, España
fcofdez@unex,es.es, ccruz@unex,es.es, fchavez@unex,es.es

Resumen Este artículo presenta una aplicación no convencional de un Algoritmo Evolutivo con el objetivo de estudiar los procesos que influyen en la creación artística humana. Hemos extraído los principales componentes de un algoritmo evolutivo y los hemos utilizado como un conjunto de reglas que deben ser seguidas por los artistas cuando crean arte. El artículo muestra cómo este tipo de AE desconectado podría convertirse en una herramienta muy útil para la comprensión de procesos creativos, permitiendo en el futuro añadir nuevos conocimientos humanos en AEs convencionales dedicados a la creación del arte y el diseño.

Keywords: Creatividad, Algoritmos Evolutivos Desconectados

1. Introducción

Los procesos naturales han servido frecuentemente para inspirar a artistas y arquitectos [2], pero también a científicos e ingenieros que buscan en la naturaleza soluciones creativas a sus problemas [13], [4].

Entre los métodos disponibles, los Algoritmos Evolutivos (AEs), y específicamente los Algoritmos Evolutivos Interactivos (AEIs) [16], ya han sido aplicados en el proceso de creación artística [15] aunque como reconoce McCormack todavía hay problemas pendientes relacionados con la creatividad [9] en este contexto.

La intención de éste artículo es profundizar en el conocimiento de la creación artística mediante la configuración de una metodología destinada a la creación artística humana basada en Algoritmos Evolutivos. Mediante la ejecución de un algoritmo evolutivo desconectado, un grupo de artistas, evolucionan una obra de arte colectiva, y se estudia el tipo de pensamientos, técnicas y soluciones generadas por los agentes humanos de una manera evolutiva. La idea es incorporarlo en el futuro en herramientas software.

El artículo se organiza de la siguiente manera: la sección 2 describe la conexión entre los AEs y el proceso de creación artística. La sección 3 describe

la metodología, mientras que en la sección 4 se presentan los resultados obtenidos y las lecciones aprendidas en la experiencia. La sección 5 incluye nuestras conclusiones y el futuro trabajo.

2. Creación

Aunque inicialmente concebidos como métodos de optimización y búsqueda de soluciones, los AEs y sus versiones interactivas también se han entendido como una fuente de creatividad, experimentando en la última década una explosión en aplicaciones en el mundo del arte, la música y el diseño [1]. Hoy día los AEIs se consideran útiles en procesos creativos [9], y posibles modelos de creación artística [3].

Pero ciertamente, la creatividad exhibida por los artistas con frecuencia cuenta con la capacidad de hacer uso de la singularidad como un concepto útil, de los conocimientos necesarios para entender su valor, y su aplicación para generar soluciones creativas [5]. Sin embargo hasta ahora, los AEIs solamente han tomado prestado de los artistas su capacidad para realizar evaluaciones estéticas, de modo que toman el papel de función de evaluación en el algoritmo, y como reconoce McCormack [10], existen una serie de problemas aún no superados en las aplicaciones en el mundo del arte, siendo los fundamentales la evaluación estética automática, y el problema de crear arte reconocido como tal por los seres humanos. Esta última cuestión se relaciona con el debate general sobre el test de Turing para la inteligencia artificial. Podemos emplear una prueba similar al Turing Test para Arte Artificial: Arte generado por ordenadores que no se distingue del generado por humanos. Una discusión similar se puede encontrar en [11].

Este trabajo pretende una mejor comprensión del tipo de operaciones mentales y aplicaciones prácticas desarrolladas por los artistas para que el conocimiento que podamos obtener sea útil en el futuro para mejorar los AE cuando son aplicados en el mundo del arte. Hemos tenido en cuenta trabajos previos y herramientas desarrolladas en esta línea, tal como Picbreeder [15]. Sin embargo, nosotros presentamos una nueva metodología consistente en tres etapas: (i) desconectar el AE para que sea ejecutado completamente por los artistas, (ii) indagar en el tipo de operaciones genéticas que los artistas aplican en su proceso creativo, y (iii) tratar de considerar posibles implementaciones futuras de algunos de estos operadores en AEs estándar.

Dado que empleamos agentes humanos en la metodología descrita a continuación, y que se requieren varios meses para un único experimento, se decidió analizar los resultados obtenidos por una sola ejecución del algoritmo, que permitirá tanto extraer conclusiones preliminares, a la vez que diseñar una buena planificación para una serie de futuras ejecuciones que permita la confirmación de las conclusiones.

Para llevar a cabo el experimento, se invitó a un grupo de cinco artistas, todos ellos Licenciados en Bellas Artes, y algunos de ellos Doctores. Al seguir un conjunto específico de normas se generó una obra colectiva que mostramos

a continuación, a la vez que analizamos desde el punto de vista de los AEs los resultados obtenidos.

En resumen, este artículo presenta una nueva metodología para mejorar los Algoritmos Evolutivos estándar, además de una aplicación específica al proceso de creación de obras de arte. Se incluye información sobre el experimento realizado y se lleva a cabo un análisis de los resultados obtenidos, con conclusiones preliminares para una futura mejora de AE dedicados a la creación artística.

3. Metodología

Antes de iniciar el experimento, reunimos un grupo de cinco artistas a los que se mostraron los fundamentos de los AEs. Pedimos primero a cada artista elegir una pintura de la historia del arte, para formar una población inicial de cinco imágenes (ver Figura 1). Después, y durante 10 semanas, los artistas trabajaron de forma independiente, y otra persona actuó como coordinador: cada artista generó una nueva obra a partir de dos imágenes elegidas libremente de la generación anterior, utilizando las operaciones de cruce y mutación que el artista considerara útiles. No se especificó qué debía ser evaluado en el proceso de selección, ni la forma en que normalmente la mutación y el cruce se codifican en los AE. También solicitamos a los artistas que completaran una ficha con las ideas tomadas en consideración y aplicadas al crear cada nueva imagen. El coordinador recibió semanalmente las cinco nuevas obras, construyó con ellas la nueva generación, y envió de nuevo a los artistas para que pudieran trabajar en una nueva obra.

Figura 1. Población inicial.

Un resumen de los principales objetivos y el proceso de realización es el siguiente:

- Crear una serie de obras de arte evolutivas generadas por cinco agentes humanos (artistas) empleando una versión adaptada del Algoritmo Evolutivo. El objetivo del proceso es el estudio de la dinámica del proceso creativo desarrollado por los seres humanos desde el punto de vista de los Algoritmos Evolutivos.
- Cada artista aporta una imagen inicial con comentarios del trabajo personal del artista, que forma parte de la población inicial.

- Cada artista producirá en cada generación, cada semana, una obra de arte sin firmar, en el formato único de 40x40cm, en papel de alto gramaje empleando cualquier técnica manual: dibujo, pintura, collage, etc... (fotografías directas o esculturas no están permitidas).
- Cada obra de arte es el resultado de la aplicación de cualquier tipo de mutación y de cruce de dos imágenes seleccionadas de la generación anterior.
- Los artistas enviarán por correo electrónico al coordinador, con una versión digital de la obra en un archivo de 2MB, fotografiada o escaneada y una ficha con la lectura objetiva y subjetiva de las imágenes seleccionadas como padres del nuevo trabajo, la evaluación numérica de los diferentes parámetros, y una descripción general del tipo de operaciones de mutación y de cruce aplicadas.
- El coordinador, una vez recibida la nueva generación completa, la hará pública en una carpeta compartida de Dropbox, por lo que cada artista podrá analizarla y empezar la siguiente generación.
- Al finalizar el experimento, junto con el análisis de los resultados, también se incluirá la obra de arte completa, única e indisoluble para ser expuesta, o vendida con el acuerdo de todos los participantes.

Desde el punto de vista de la persona que actuó como coordinador, y una vez que el experimento se ha ejecutado, no se encuentra ninguna diferencia cuando se compara con un experimento de AE estándar: Cada generación surgía un nuevo conjunto de individuos que heredaban características de la generación previa. La expectativa era aún mayor, dado el tipo de individuos generados: obras de arte cuyas imágenes digitales llegaban por correo electrónico.

El proyecto se inició el martes 23 de Octubre de 2012, con el envío de las imágenes iniciales, y terminó el Martes, 1 de Enero de 2013.

La forma empleada para recopilar la información incluyó los campos expuestos en la figura 3, que será analizada en la próxima sección.

4. Experimentos y resultados

Los resultados que se incluyen a continuación corresponden al único experimento realizado siguiendo la metodología explicada anteriormente. En las cifras presentadas, codificamos cada una de las operaciones genéticas de los artistas utilizando los siguientes colores (artista-color): #1 violeta, #2 verde, #3 azul claro, #4 gris, #5 azul claro.

Llevamos a cabo un doble análisis de los resultados: se analiza primero el experimento como el transcurso de un Algoritmo Evolutivo y después, nos fijamos en el conjunto de individuos generados como obra de arte colectiva. Se estudia el resultado mediante el análisis de la información, tanto objetiva como subjetiva, proporcionada por los artistas cuando reflexionan sobre su forma de trabajar.

4.1. A. La obra de arte desde una perspectiva evolutiva

Consideramos aquí el experimento como la evolución de una serie de imágenes. La población inicial para el experimento se compuso con obras de arte conocidas, propuestas por cada uno de los artistas. La Figura 1 muestra las imágenes

seleccionadas, que incluye obras de Boticelli, Leonardo da Vinci, Schiele y Millais. Sorprendentemente, dos artistas seleccionaron obras del mismo autor, Schiele. Es también significativa la inexistencia de obras abstractas que podrían ser consideradas más cercanas a la búsqueda aleatoria de las poblaciones iniciales de un AE, incluyendo aquí los AEI y otras herramientas dedicadas al arte y el diseño, como Picbreeder, que comienza generando una serie de diseños abstractos.

Los artistas evaluaron cada semana las imágenes de la generación anterior, sin información sobre el autor, seleccionado dos de ellas como fuente de inspiración, y aplicando operaciones de cruce y mutación para producir un nuevo individuo para la próxima generación. Se repitió el procedimiento durante 10 generaciones, hasta cumplir los criterios de finalización: 10 semanas de trabajo completo. La figura 2 muestra los resultados obtenidos, con conexiones entre generaciones: cada una de las líneas muestra cómo una imagen de la generación i ha sido seleccionada como fuente de inspiración para otra nueva imagen en la generación $i + 1$, lo que indica cuál actúa como padre y cuál como hijo.

Una de las obras de arte procedentes de la generación 2 no está conectada con la generación anterior. Esto puede ser considerado como un tipo de fallo de funcionamiento en el algoritmo. Se corrigió el error -notificándolo al artista- y ya no volvió a repetirse. La genealogía para cada obra de arte es muy útil cuando analizamos sus comportamientos y operaciones. Vamos a considerar información cuantitativa y cualitativa. La Figura 3 muestra el número de veces que cada artista ha seleccionado el trabajo de los demás como fuente de inspiración, seleccionando obras de la generación anterior. De un total de 10 generaciones, 2 selecciones por artista, 5 artistas, 100 selecciones realizadas en total, vemos que dos artistas han obtenido la mayor influencia en la obra de arte conjunta: violeta y gris, conteniendo casi el 50 % de las operaciones de selección. El artista azul claro apenas ha influido en sus compañeros, recibiendo menos de 10 selecciones. Aunque este tipo de información normalmente no se procesa en los AEs estándar, la consideramos de interés aquí para ver la influencia de cada uno de los artistas en el proceso de evolución. La importancia de esta información ya ha sido reconocida en algunos AEIs cuando el trabajo de colaboración se lleva a cabo por diferentes usuarios [15], [17].

Analizando la información, vemos que los artistas pocas veces eligen su propio trabajo como inspiración. Sólo en la primera generación, todos los artistas escogieron su propia obra como uno de los padres para su siguiente trabajo. Nos encontramos con un comportamiento general: los artistas generalmente se basan en las obras de sus compañeros. Los artistas #4 y #5 prefieren elegir obras ajenas: El artista #5 nunca eligió a una de sus obras como inspiración, lo cual parece que fue útil para mejorar la diversidad en su propio trabajo, mientras que el Artista #4 es el que selecciona con más frecuencia su propio trabajo como inspiración, incluso cuando no selecciona de forma explícita su propia obra, aplica la misma técnica en cada generación, manteniendo implícita su propia influencia, y reduciendo claramente la diversidad de sus individuos generados a lo largo del proceso. Sólo tres imágenes han sido descartadas. Esto es completamente extraño para un AE estándar, donde generalmente sólo unos pocos individuos

Figura 2. Genealogía del trabajo (izquierda), imágenes principales (derecha)

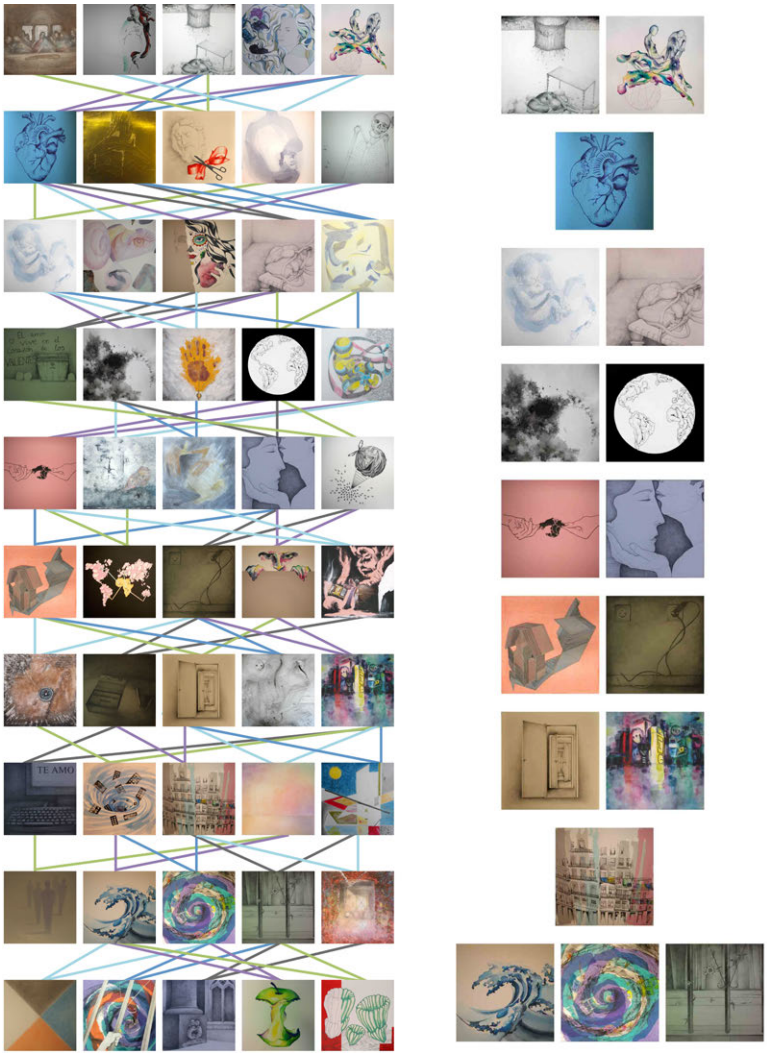
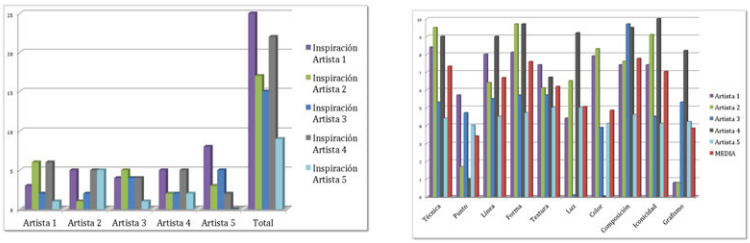


Figura 3. ¿Quién selecciona a quien? (izquierda) Objetivos (derecha)



son seleccionados y utilizados para la creación de las nuevas generaciones. Los individuos generalmente permanecen en la población y tienen oportunidades de ser seleccionados más adelante, pero en nuestro caso, los individuos de una generación anterior se descartan una vez creada la nueva generación. La sorpresa en nuestro experimento es que casi todos los individuos han contribuido de manera uniforme en el proceso creativo, aunque sólo puedan permanecer durante una generación. La conclusión principal de este tipo de resultado es que la diversidad obtenida por los artistas aplicando un AE es enorme. También se observa a lo largo del experimento una preferencia por obra figurativa sobre la abstracta, ya que se han generado pocas obras de arte abstracto. Curiosamente, esto es opuesto a los resultados producidos por los AEI estándar, donde las operaciones matemáticas se codifican y se aplican, generando una especie de obra abstracta, que por medio de filtración y selección humana hace tender hacia elementos figurativos.

A partir de la tercera generación, podemos observar un resultado más unificado en cuanto al color y técnica aplicada, en comparación con la anterior generación, lo que podría interpretarse como un fenómeno de convergencia. Hemos encontrado que la influencia entre los artistas es cada vez mayor y esto se refleja en las imágenes. Por ejemplo, el dibujo en grafito en claroscuro, que fue la única técnica utilizada a lo largo del experimento por el Artista # 4, influye en la pérdida de color en las generaciones. En general, los artistas se centraron en las formas y los objetos suprimiendo el fondo.

Una pregunta difícil es si las operaciones pueden ser conscientemente implementadas por agentes artificiales, sin embargo, no es difícil aplicar medidas específicas considerando que el color o la forma producen cambios cuando se codifican como operadores de mutación.

En las últimas generaciones el color y la textura son valores mucho más apreciados. Además, los objetos tienen un fuerte carácter geométrico, una clara conexión con el tipo de cosas que los AEIs codifican: la descripción matemática de las formas, junto con los componentes del color. Se observan trazos más sueltos en los dibujos, posiblemente debido a la práctica contínua (Ver Figura 2).

4.2. B. Génesis de la obra desde una perspectiva artística

La idea ahora es tratar de entender el proceso creativo desde el punto de vista de los artistas. El análisis de las fichas enviadas de cada obra por los artistas lo hacemos teniendo primero en cuenta una lectura objetiva de la sintaxis de las imágenes, y luego realizamos la lectura subjetiva, tratando de extraer conclusiones de interés.

Lectura Objetiva La Figura 3 nos muestra la relevancia de cada uno de los elementos incluidos en la ficha al crear la nueva imagen. Los artistas coinciden en que la técnica empleada es un elemento fundamental en la etapa de selección. En cuanto a los elementos gráficos que se utilizan, el punto no es muy valorado; la línea, por el contrario, ha sido muy apreciada debido a su capacidad para definir contornos, marcar volúmenes, o resaltar el fondo. Las formas del objeto se definen con las líneas utilizadas.

Las texturas utilizadas son tanto visuales como táctiles, con técnicas como el collage. Al analizar el tratamiento de la luz encontramos diversidad, ya que si se aplica claroscuro la luz se convierte en un elemento clave, pero cuando se dibujan sólo las formas, la luz se suprime completamente.

Hay preferencia por la utilización del claroscuro, por la influencia de uno de los artistas. Aunque el color no es una de las características más valoradas en la ficha (véase la figura 3), se aplican comportamientos específicos en relación con el color, como su tendencia a aparecer en las últimas generaciones. La composición es la característica más apreciada por los artistas, ya que es esencial cuando se trata de distribuir el espacio disponible para realizar la obra.

La iconicidad es otra característica importante a considerar en el proceso de creación, ya que los artistas seleccionan trabajos previos como fuente de inspiración, tratando de conectar con las experiencias personales, y por lo tanto sugieren ideas que se desarrollarán en su obra. Los artistas tienen así en cuenta información semántica obtenida a partir de las imágenes seleccionadas. La iconicidad no sólo cumple con una función estética, sino también con la de transmitir sentimientos o conceptos mediante el uso constante de metáforas extraídas de las imágenes.

La Figura 4 es útil para analizar cómo se inspiran los diferentes artistas. Podemos ver imágenes diferentes con la misma inspiración, y el resultado obtenido es completamente diferente.

Lectura Subjetiva Los artistas transmiten a través de las imágenes temas universales como el amor, la angustia, el recorrido de la vida, la espiritualidad, la injusticia social, el miedo, etc, utilizando sus experiencias personales para incluir elementos con los que representar los conceptos. Al principio del experimento, los artistas eran más moderados al representar sentimientos o ideas sugeridas en la obra anterior, se basaban más en la forma. Pero, como el trabajo maduró durante varias semanas, se desarrolló una especie de entrenamiento tanto físico como mental, donde la inspiración estaba más vinculada al mundo de las ideas.

Figura 4. Diferentes resultados con la misma imagen de inspiración

En base al doble análisis anterior, podemos resumir los principales resultados de este experimento a la vez que sugerimos algunas propuestas específicas desde el punto de vista de los Algoritmos Evolutivos:

- Inicialización: el trabajo figurativo debe ser un componente principal para la población inicial y una fuerza que lidera todo el proceso (en oposición a poblaciones abstractas).
- Proceso de selección: (i) Alentar a los agentes para seleccionar resultados generados por otros agentes, y favoreciendo la participación de todos los individuos.
- Cruce y mutación: La composición y otros conceptos artísticos deben ser incluidos, así como la semántica e información simbólica que permiten nuevas operaciones genéticas.
- Co-evolución: permite a diferentes especies utilizar diferentes técnicas y operaciones.
- Propiedades de Auto-organización: Es positiva la existencia de conciencia que permite la auto-reflexión sobre los resultados obtenidos y la elección adecuada del camino a seguir.
- Salida: Permitir las diferentes técnicas artísticas y soportes materiales (no sólo creaciones digitales).

Esperamos poder confirmar estos hallazgos en nuevos experimentos, e iniciar el diseño y la implementación de nuevos AE dedicados al arte en un futuro trabajo.

5. Conclusiones

Este trabajo ha mostrado un nuevo enfoque para el estudio humano de la creatividad mediante el uso de Algoritmos Evolutivos Desconectados. Un conjunto de cinco artistas han desarrollado a lo largo de diez semanas una obra de arte colectivo siguiendo las reglas de un AE. El experimento nos ofrece varias conclusiones preliminares: (i) la diversidad observada en el resultado es bastante diferente al que se observa en los AE interactivos: diferentes materiales, técnicas, temas, ... han sido empleados y generados. (ii) Hemos aprendido que las

operaciones de mutación y cruce empleados por los artistas son muy diferentes de las que normalmente se ejecutan en AEI destinados a generar arte y diseño, además de un enfoque figurativo que se ha seguido desde la inicialización de la población. (iii) El conocimiento, las experiencias vitales, los sentimientos y la auto-reflexión, son algunos de los componentes principales que aplican artistas a lo largo del proceso, lo que nos lleva a considerar que se requiere un gran paso en los AE para emular el proceso creativo desarrollado por los seres humanos.

6. Agradecimientos

Este trabajo ha sido financiado por el Ministerio de Ciencia e Innovación, bajo el proyecto ANYSELF (TIN2011-28627-C04), la Universidad de Extremadura y el Gobierno de Extremadura Consejería de Economía Comercio e Innovación y fondos FEDER, proyecto GRU10029.

Referencias

1. P. Bentley: Evolutionary Design by Computers. Morgan Kaufmann: Springer-Verlag (1999).
2. G. M. Codignola: Generative recognizable codes from gaudi (2000)
3. P. Dahlstedt: Turn-based evolution as a proposed implementation of artistic creative process. IEEE Congress on Evolutionary Computation, 1–7 (2012)
4. G. W. Flake: The Computational Beuty of Nature. Springer-Verlag (2000)
5. J. Guilford: Creativity. American Psycologist, 444–454 (1950)
6. Iamus and London-Symphony-Orchestra: Iamus [audio cd] (2012)
7. A. Kosorukoff: Human based genetic algorithm (2001)
8. B. A. Limb CJ: Neural substrates of spontaneous musical performance: An fmri study of jazz improvisation. PLos ONE (2008)
9. J. MacCormack and M. d'Inverno (eds): Computers and Creativity. Springer-Verlag (2012)
10. J. McCormack: Open problems in evolutionary music and art. EvoWorkshops, 428–436 (2005)
11. K. McGuire: A turing test for generative art. Proceedings XIV Generative Art Conference (2011)
12. S. Y. Y. L. Nickerson, Jeffrey: Structures for creativity: The crowdsorcing of design. Proceedings CHI 2011 Workshop on Crowdsourcing and Human Computation (2011)
13. P. Prusinkiewicz, A. Lindenmayer, J. S. Hanan *et al.*: The algorithmic beauty of plants. Springer-Verlag (1990)
14. J. Romero and P. Machado, Eds.: The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music. Natural Computing Series (2007)
15. J. Secretan, N. Beato, D. B. D'Ambrosio, A. Rodriguez, A. Campbell, J. T. Folsom-Kovarik, and K. O. Stanley: Picbreeder: A case study in collaborative evolutionary exploration of design space. Evol. Comput. 19(3), 373–403 (2011)
16. H. Takagi: Interactive evolutionary computation. Proccedings of the 5th Int'l Conf. on Soft Computing, 41–50 (1998)
17. M. G. Valdez, L. Trujillo, F. F. de Vega, J. J. Merelo, and G. Olague: Evospace-interactive: A framework to develop distributed collaborative-interactive evolutionary algorithms for artistic design. Proceedings EvoMusart (2013)

Scatter Search aplicado aplicada al problema de la minimización del *profile* de matrices y grafos

J. Sánchez-Oro[1], A. Duarte[1], M. Laguna[2], y R. Martí[3]

[1] Dept. Ciencias de la Computación, Universidad Rey Juan Carlos (Spain)
[2] Leeds School of Business, University of Colorado Boulder (USA)
[3] Dept. de Estadística e Investigación Operativa, Universidad de Valencia (Spain)
jesus.sanchezoro@urjc.es, abraham.duarte@urjc.es, laguna@colorado.edu, rafael.marti@uv.es

Resumen En este trabajo se propone un algoritmo basado en *Scatter Search* para resolver el problema de la minimización del *Profile*. Se utiliza la estrategia *Path Relinking* como método base para la generación de nuevas soluciones mediante combinación. El problema del Profile (PMP) es NP-duro y tiene aplicaciones relevantes en técnicas de análisis numérico basadas en la manipulación de matrices de gran tamaño. Fue propuesto a comienzos de los 70 pero en el estado del arte no se encuentra ningún método eficiente para la resolución del problema. La amplia experimentación llevada a cabo en este trabajo demuestra que se ha cumplido el objetivo de establecer un nuevo estándar en la solución del PMP.

Keywords: profile minimization problem, metaheurísticas, scatter search, path relinking

1. Introducción

Se define $G = (V, E)$ como un grafo conexo no dirigido, siendo V el conjunto de vértices ($|V| = n$) y E el conjunto de aristas ($|E| = m$) de G. La asignación de una etiqueta única ($\varphi(v) = \{1, 2, ..., n\}$) a cada uno de los vértices de G conforma una ordenación lineal φ de los vértices de G. Una solución al PMP de un grafo G puede representarse mediante una ordenación lineal de los vértices de G. Siendo $Adj(v)$ el conjunto de vértices adyacentes a v, se define $f(v, \varphi, G)$ como el mínimo entre la posición del vértice adyacente a v con la menor etiqueta en φ y la propia etiqueta de v. En términos matemáticos,

$$f(v, \varphi, G) = \min \left(\varphi(v), \underset{u \in Adj(v)}{\arg\min} \varphi(u) \right).$$

El $VertexCut$ de un vértice v del grafo G en una ordenación φ queda definido como $VC(v, \varphi, G) = \varphi(v) - f(v, \varphi, G)$. El *profile* de una ordenación lineal φ de un grafo G ($P(\varphi, G)$) se define como la suma de los $VertexCut$ de cada vértice

en la ordenación. Es decir,

$$P(\varphi, G) = \sum_{v \in V} VC(v, \varphi, G)$$

El objetivo del PMP es encontrar una ordenación lineal $\varphi^\star$ tal que $P(\varphi^\star, G) = \min_{\varphi \in \phi} P(\varphi, G)$. La Figura 1 muestra un ejemplo de un grafo conexo no dirigido (izquierda) así como una posible ordenación lineal del mismo, $\varphi = \{C, E, B, A, D\}$ (derecha). Además, se muestra la evaluación de los $VertexCut$ de cada vértice. El *profile* de la ordenación mostrada se calcula como la suma de todos los $VertexCut$, es decir: $P(\varphi) = 0 + 1 + 0 + 3 + 3 = 7$.

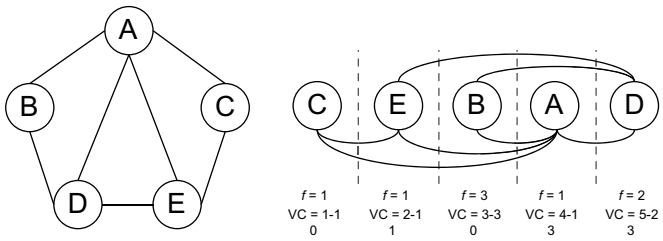


Figura 1. Ejemplo de un grafo (izquierda) y una posible ordenación lineal del mismo, con el cálculo de los $VertexCut$ de cada vértice

El PMP fue propuesto originalmente como una aproximación para reducir el espacio necesario para almacenar matrices dispersas [26], donde se ha probado que es equivalente al problema del *SumCut* [1]. El PMP también tiene aplicaciones en la mejora del rendimiento de operaciones sobre sistemas de ecuaciones no lineales [25], el Proyecto Genoma Humano [14], arqueología [15], recuperación de la información [2] y reconocimiento de patrones [14].

El PMP es un problema equivalente al *Interval Graph Completion Problem* [19], el cual es NP-completo, como se demostró en [10]. Sin embargo, la solución óptima de algunos tipos de grafos se puede obtener en tiempo polinómico. Por ejemplo, en [19] se proponen varios algoritmos que encuentran la solución del PMP en tiempo polinómico para grafos de tipo camino, rueda, bipartido completo y árboles D4 (árboles con diámetro 4). De la misma manera, en [12] se presenta un algoritmo que encuentra la solución óptima del PMP para grafos triangulares.

El primer heurístico encontrado en el estado del arte referente al PMP consiste en un procedimiento constructivo [3] conocido como *Reverse Cuthill-McKee Algorithm* (RCM), basado en construir una estructura de nivel de los vértices. En [22] se mejora el algoritmo del RCM cambiando la selección del nodo raíz dentro de una estructura de nivel más general. Este método se conoce como GPS. En [11] se presenta el algoritmo GK que utiliza un pseudo-diámetro para producir una nueva estructura de nivel, mejorando los resultados de GPS en calidad,

pero necesitando un mayor tiempo de cómputo. Finalmente, en [17] se introduce una nueva implementación que mejora la calidad tanto de GK como de GPS. El mejor método heurístico para el PMP encontrado en la literatura aparece en [18] y utiliza la metodología del Recocido Simulado (*Simulated Annealing - SA*) combinada con los métodos constructivos descritos anteriormente.

2. Scatter Search

Scatter Search (SS) [16] es una metaheurística que incluye cinco métodos para construir, mantener y transformar una población de soluciones. Tres de estos métodos (construcción diversa, mejora y combinación) dependen directamente del problema, por lo que sus estrategias se basan en la información presente en el mismo. Por otra parte, la actualización del conjunto de referencia y la generación de subconjuntos tienen implementaciones estándar que son independientes del problema. El estado del arte de SS se mantiene en constante crecimiento con un gran número de nuevos trabajos (ver [8], [20], [6] y [7]). El Algoritmo 1 muestra el marco general de la metodología *Scatter Search*.

Algoritmo 1 Metodología Scatter Search

```
1: Construcción diversa
2: Mejora
3: Actualización del RefSet
4: while criterio de finalización no alcanzado do
5:    Generación de los subconjuntos
6:    Combinación
7:    Mejora
8:    Actualización del RefSet
9: end while
```

El algoritmo comienza con la aplicación del método de construcción diversa (paso 1) y el método de mejora (paso 2), generando una población de P soluciones que se utilizan para construir el conjunto de referencia inicial ($RefSet$) de tamaño b. El $RefSet$ debe contener tanto soluciones de alta calidad como soluciones diversas, por lo que en su actualización (paso 3) se seleccionan las $b/2$ mejores soluciones de P y, tras ello, las $b/2$ soluciones más diversas con respecto al $RefSet$ del conjunto restante ($P \setminus RefSet$). Es importante tener en cuenta que la selección de las soluciones más diversas de un conjunto es un problema NP-duro (ver [5]) y, por lo tanto, este paso se lleva a cabo de manera heurística. En concreto, una vez que se han seleccionado las $b/2$ mejores soluciones al $RefSet$, las siguientes soluciones se añaden una a una. La primera solución seleccionada es la solución más diversa respecto a las soluciones actuales del $RefSet$. La medida de la diversidad se representa mediante una función que maximiza la mínima distancia entre la solución que se está evaluando y el conjunto de soluciones donde va a ser agregada (en este caso, el $RefSet$). La distancia entre

una solución φ y el $RefSet$ se define como la suma mínima de las diferencias absolutas de las etiquetas asignadas a cada vértice en la solución φ y cada una de las soluciones φ' del $RefSet$. En términos matemáticos,

$$d(\varphi, RefSet) = \min_{\varphi' \in RefSet} \left(\sum_{v \in V} |\varphi(v) - \varphi'(v)| \right).$$

Como las etiquetas de una solución representan posiciones, el cálculo de la distancia es equivalente a la distancia posicional [4]. Una vez que se ha seleccionado una solución φ del conjunto $P \setminus RefSet$, ésta se añade al $RefSet$ y el proceso se repite hasta completar el $RefSet$ con b soluciones, eligiendo en cada paso la solución $\varphi^\star$ con la máxima distancia a las soluciones actuales del $RefSet$. Es decir,

$$\varphi^\star = \arg\max_{\varphi \in P \setminus RefSet} d(\varphi, RefSet)$$

En este trabajo, durante la etapa de generación de subconjuntos (paso 5) se generan todos los pares de soluciones que no han sido combinados anteriormente. La actualización del $RefSet$ del paso 8 es diferente a la del paso 3. Para mantener la diversidad entre las soluciones, una solución φ se admite si mejora la mejor solución del $RefSet$, o si mejora la peor solución y además la distancia de φ con la solución más cercana es mayor que un umbral predefinido $dthresh$. Si finalmente la solución φ puede entrar al $RefSet$, lo hace sustituyendo la solución más cercana con un valor de función objetivo menor o igual que φ. El proceso finaliza cuando no hay nuevas soluciones que puedan formar parte del $RefSet$. Es decir, si durante una iteración determinada del método, el $RefSet$ no cambia tras la actualización del paso 8, el algoritmo termina.

3. Construcción de soluciones diversas

En este trabajo se presentan cuatro métodos de construcción de soluciones diversas para construir una población P de soluciones, basados en la metodología GRASP [9,23,24]. Se define U como el conjunto de vértices no etiquetados y $L = V \setminus U$ como el conjunto de vértices ya etiquetados. Inicialmente, todos los vértices están en U. El primer método, C1, comienza seleccionando el vértice $v \in U$ de menor grado, seleccionando al azar en caso de empate. El vértice elegido v recibe la primera etiqueta, por lo que $\varphi(v) = 1$, y se actualizan los conjuntos U y L ($U = U \setminus \{v\}$ y $L = L \bigcup \{v\}$). Tras ello, se construye una lista de candidatos CL formada por los vértices adyacentes a v que no han sido etiquetados. En términos matemáticos, $CL = \{u : (v, u) \in E | v \in U\}$.

Para cada vértice v se evalúa la siguiente función voraz: $g_1(v) = |N_L(v)| - |N_U(v)|$, donde $N_L(v) = \{u \in L : (v, u) \in E\}$ y $N_U(v) = \{u \in U : (v, u) \in E\}$. La función voraz $g_1(v)$ mide la necesidad de dar al vértice v la siguiente etiqueta. La función considera que es más urgente etiquetar un vértice que tiene a la mayoría de sus adyacentes ya etiquetado. En la metodología GRASP las

construcciones son semi-voraces, y la implementación contiene una lista de candidatos restringida (RCL). La RCL incluye a los mejores candidatos de la CL que van a tener la misma probabilidad de ser elegidos:

$$RCL = \left\{v \in CL : g_1(v) > g_1^{min} + \alpha_1 \left(g_1^{max} - g_1^{min}\right)\right\}$$

donde g_1^{min} y g_1^{max} son el mínimo y el máximo valor de $g_1(v) \forall v \in CL$, respectivamente. Se elige un vértice v de la RCL de forma aleatoria y, tras asignarle la siguiente etiqueta disponible, los conjuntos U y L se actualizan. CL también se actualiza añadiendo los vértices sin etiquetar adyacentes a v. La construcción termina cuando todos los vértices han sido etiquetados.

El segundo método constructivo (C2) es semejante a C1, pero modificando la implementación de la selección semi-voraz. En este caso, la RCL se construye mediante una muestra aleatoria de vértices de la CL, y, tras ello, se selecciona el vértice v con mayor valor de $g_1(v)$. El tamaño de la RCL se establece con un parámetro α_2, que representa un porcentaje del tamaño de la lista de candidatos. Es decir, la RCL tendrá un tamaño de $\alpha_2 \cdot |CL|$ vértices. Como en C1, una vez que un vértice ha sido elegido, se le asigna la siguiente etiqueta y se actualizan los conjuntos U y L.

La función voraz $g_1(v)$ no tiene en cuenta los valores de las etiquetas asignadas a los vértices adyacentes a v. Por ejemplo, si i es la siguiente etiqueta disponible, $g_1(v)$ no utiliza información acerca de las etiquetas que los elementos de $N_L(v)$ han recibido (entre 1 y $i-1$). Claramente, es más importante etiquetar el vértice v si sus adyacentes han recibido etiquetas que son mucho menores que i. La nueva función voraz $g_2(v)$ sí que aprovecha esta información:

$$g_2(v) = (|N_L(v)| - |N_U(v)|) \sum_{u \in N_L(v)} |\varphi(v) - \varphi(u)|$$

Esta nueva función voraz se utiliza para crear dos nuevos métodos constructivos: C3 y C4. C3 es similar a C1, y C4 es similar a C2, pero utilizando g_2 como función voraz.

4. Búsqueda local

En este artículo se proponen dos métodos de búsqueda en una vecindad: uno definido por intercambios y otro por inserciones. El $intercambio(i, j)$ de dos nodos v, u tal que $\varphi(v) = i$ y $\varphi(u) = j$ se define como la modificación de las etiquetas de dichos nodos, de manera que el vértice v recibe la etiqueta de u y viceversa. Por otra parte, una $insercion(i, j)$ es el movimiento del vértice v situado en la posición i ($\varphi(v) = i$) a la posición j. Como ambos movimientos producen una vecindad de tamaño $O(n^2)$, es necesario evaluar los movimientos de manera eficiente para identificar el mejor de ellos.

En la evaluación del movimiento $insercion(i, j)$ para $j > i$, es sencillo ver que solo hay que reevaluar el $VertexCut$ de los vértices v tal que $i \leq \varphi(v) \leq n$. Por tanto, si se almacenan los $VertexCut$ de todas las posiciones, el cálculo de

un movimiento se basa en actualizar aquellas posiciones relevantes y el valor de la suma de los $VertexCut$. Además, la actualización de los $VertexCut$ es acumulativa. Es decir, si primero evaluamos el movimiento $insercion(i, j)$, debemos reevaluar el $VertexCut$ de aquellos vértices situados entre las etiquetas i y n. Si tras ese movimiento evaluamos $insercion(i, j + 1)$ sin tener en cuenta el anterior movimiento, necesitaríamos evaluar otra vez los $VertexCut$ de los mismos vértices. Sin embargo, observando los cambios en los $VertexCut$ en estos movimientos, se puede determinar que los únicos cambios tras el segundo movimiento se producen en los vértices v con etiquetas $\varphi(v) \geq j + 1$.

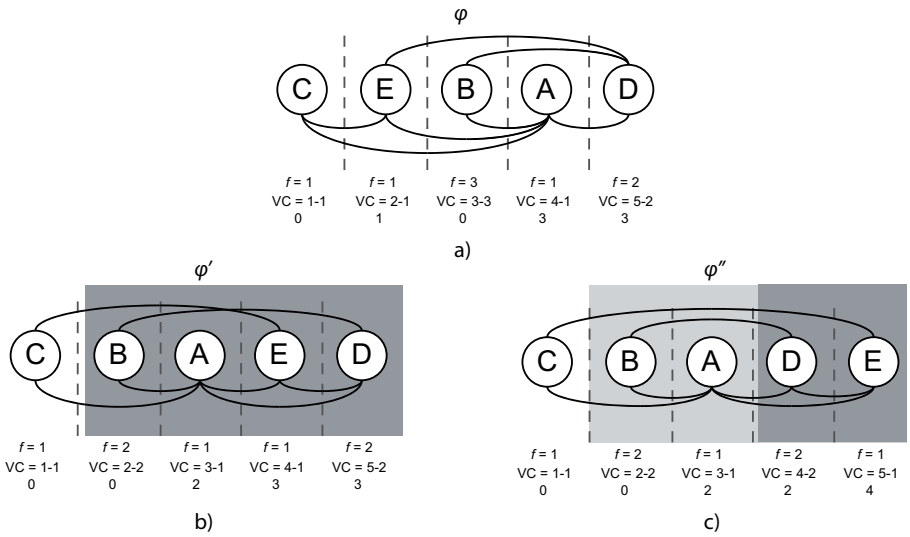


Figura 2. Ejecución de dos inserciones consecutivas: $insercion(2, 4)$ (b) y $insercion(2, 5)$ (c) a partir de una ordenación lineal (a). Se resalta con un sombreado oscuro las posiciones de las que hay que recalcular el $VertexCut$ y con sombreado claro las posiciones que no es necesario calcular gracias a la implementación propuesta.

La Figura 4 muestra un ejemplo de inserción de un vértice en dos posiciones consecutivas. En concreto, la Figura 4.b muestra la inserción del vértice E en la posición 4, partiendo de la ordenación $\varphi = \{C, E, B, A, D\}$ de la Figura 4.a. Por otro lado, la Figura 4.c muestra la inserción del vértice B en la posición 5, partiendo de la misma ordenación (Figura 4.a). En ambas inserciones, es necesario volver a calcular el $VertexCut$ de las posiciones $p \geq \varphi(E)$. Sin embargo, la implementación eficiente propuesta en este trabajo permite aprovechar los cálculos realizados para la evaluación de los $VertexCut$ en la Figura 4.b, de forma que en la segunda inserción (Figura 4.c) tan solo es necesario evaluar las posiciones $p \geq \varphi'(E)$. Es decir, se evalúan dos posiciones en lugar de cuatro. Por lo tanto, para aprovechar esta evaluación eficiente de la inserción, el movimiento

$insercion(i, j)$ se implementa como una secuencia de intercambios de vértices en posiciones consecutivas.

5. Métodos de combinación

Se propone un método de combinación (CM1) y una variante (CM2). El método propuesto sigue la estrategia de *Path Relinking* (PR) [13]. PR trata de construir caminos entre dos soluciones donde la función objetivo es solo uno de los elementos utilizados para determinar la dirección. Aprovecha el principio de que la vecindad de una solución buena posiblemente contenga otras soluciones de alta calidad si se explora de una manera diferente.

Se define φ_s como la solución inicial y φ_t como la solución guía o destino. CM1 comienza identificando el conjunto D de vértices que tienen etiquetas diferentes en φ_s y φ_t, es decir, $D = \{v : \varphi_s(v) \neq \varphi_t(v)\}$. Se genera un conjunto de soluciones intercambiando las etiquetas del vértice v y el vértice que tiene la etiqueta $\varphi_t(v)$ en la solución inicial, para todos los vértices $v \in D$. Este movimiento hace que φ_s se acerque a φ_t. Este paso genera $|D|$ soluciones de las cuáles seleccionamos la mejor teniendo en cuenta el valor de la función objetivo. La solución elegida se convierte en la nueva solución inicial y continúa el proceso hasta que $D = \emptyset$, es decir, hasta que $\varphi_s = \varphi_t$. Este procedimiento, llamado *Greedy Path Relinking* en [24] devuelve la mejor solución encontrada en el camino.

CM2 es una variante de CM1, donde se modifica el criterio de selección de soluciones en el camino generado. En este caso, en lugar de elegir la mejor solución del conjunto de soluciones generado en un paso del camino, se elige una solución aleatoria para favorecer la diversificación. Al igual que CM1, CM2 devuelve la mejor solución encontrada en el camino.

6. Resultados experimentales

Esta sección describe los experimentos computacionales que se han llevado a cabo para probar los algoritmos propuestos. Todos los algoritmos se han desarrollado en Java SE 6 y ejecutados en un Intel Core i7 2600 a 3.4 GHz con 2 GB de memoria RAM. Los resultados de los algoritmos RCM [3] y SA [18] del estado del arte se han obtenido ejecutando el código C original proporcionado por los autores. El conjunto de instancias utilizado consiste en 262 instancias divididas en tres subconjuntos, y están disponibles en `www.optsicom.es/pmp`. Los tres subconjuntos de instancias se describen a continuación:

- **HB:** Se compone de 73 instancias de la colección *Harwell-Boeing Sparse Matrix Collection* [21]. Contiene instancias obtenidas de problemas de sistemas lineales, mínimos cuadrados y cálculo de autovalores en una amplia variedad de disciplinas científicas. El número de vértices de las instancias varía entre 24 y 960 y el número de aristas entre 34 y 3721.
- **K-Graphs:** Este conjunto contiene 98 grafos bipartidos con un número de vértices comprendido entre 4 y 142 y un número de aristas entre 3 y 5016.

El conjunto de vértices V de un grafo bipartido se puede dividir en dos subconjuntos V_1 y V_2 de forma que no existen aristas entre dos nodos del mismo subconjunto de vértices. El valor óptimo de estos grafos se conoce por construcción y viene determinado por $n_1 n_2 + \frac{1}{2} n_1(n_1 - 1)$ para $n_1 \leq n_2$, donde $n_1 = |V_1|$ y $n_2 = |V_2|$

- **D4-Trees:** Consta de 91 instancias basadas en árboles con diámetro 4, un rango de vértices entre 10 y 100 nodos y un rango de aristas comprendido entre 9 y 99. Las soluciones óptimas se conocen por construcción, y su valor es $|E| + \sum_{i=3}^{k} (\gamma(v_i) - 1)$.

Los experimentos llevados a cabo se pueden dividir en dos partes. En la primera parte se lleva a cabo un análisis factorial de los algoritmos propuestos sobre un subconjunto de las instancias utilizadas para seleccionar la mejor combinación de constructivo, búsqueda local y método de combinación. En concreto, el subconjunto de instancias de la experimentación preliminar se compone de las 30 instancias con menos de 250 vértices obtenidas del conjunto HB.

La mejor combinación resultante del experimento factorial es la del constructivo C1 junto con la búsqueda basada en inserciones y el método de combinación CM1, obteniendo una desviación del 0.80 %, 20 de las mejores soluciones encontradas y un tiempo de ejecución de 6.59 segundos, seguido de C3+inserciones+CM1, con un 0.75 % de desviación, 17 mejores soluciones encontradas y 6.78 segundos de tiempo de ejecución. El algoritmo utilizado para el experimento final es por tanto C1+inserciones+CM1.

En el experimento final se comparan los mejores algoritmos encontrados en el estado del arte con el algoritmo propuesto en este trabajo (SS) utilizando el conjunto completo de instancias. En concreto, se utilizan los algoritmos Reverse Cuthill-McKee (RCM) y Simulated Annealing (SA). El experimento reporta información acerca de el valor promedio de la función objetivo (F.O.), la desviación estándar respecto a la mejor solución conocida (Dev), el número de mejores soluciones encontradas (#Mejores) y el tiempo de ejecución en segundos. La Tabla 1 muestra los resultados obtenidos en el experimento.

Algoritmo	**F.O.**	**Dev(%)**	**#Mejores**	**Tiempo**
RCM	2886.31	69.00	107	69.78
SA	2837.06	18.62	135	111.27
SS	2346.52	0.10	234	60.22

Tabla 1. Comparativa del algoritmo propuesto con el estado del arte

Como se puede observar, el algoritmo propuesto obtiene mejores resultados en todas las estadísticas. Además, la baja desviación del método propuesto indica que, en el reducido número de instancias donde nuestro algoritmo no encuentra la mejor solución conocida, proporciona soluciones de alta calidad. Analizando los conjuntos de instancias por separado (D4-Trees, K-Graphs y HB) se observa que el conjunto de los K-Graphs no parece ser un reto para ninguno de los algoritmos,

ya que en todos los casos obtienen el valor óptimo (salvo el algoritmo SA en una instancia) en un intervalo de tiempo reducido (menos de un segundo en el caso de SS y RCM, y 6 segundos en el caso de SA). Sin embargo, en el conjunto D4-Tree el algoritmo SS ya muestra su potencial, obteniendo 89 de los 91 valores óptimos, seguido del SA (31), y RCM (2). En el conjunto más difícil de los tres, HB, donde no existe óptimo conocido, el algoritmo SS sigue dominando al resto de algoritmos, obteniendo 67 de las mejores soluciones y una desviación del 0.38 %, seguido del SA (7 mejores soluciones, 28.93 %) y el RCM (7 mejores soluciones, 31.31 %). El tiempo de ejecución requerido por SS y RCM es similar, siendo SS ligeramente inferior, mientras SA necesita aproximadamente el doble de tiempo.

Finalmente, para validar los resultados obtenidos, se aplican test no paramétricos a los algoritmos utilizados para comprobar si existen diferencias significativas entre ellos. En primer lugar se aplica el test de Friedman, obteniendo un p-valor de 0.000, lo que claramente indica que existen diferencias significativas entre los métodos propuestos. Los rankings producidos por el test son 1.16 (SS), 2.37 (SA) y 2.81(RCM). A continuación se aplica el test de Wilcoxon y de signos para comparar los mejores algoritmos del experimento, SS y SA. En ambos casos se obtiene un p-valor de 0.000, lo que demuestra que sí existen diferencias significativas entre ambos algoritmos.

7. Conclusiones

En este trabajo se presentan diferentes algoritmos basados en la metodología *Scatter Search* y *Path Relinking* para resolver el problema del *Profile*. En concreto, se proponen cuatro métodos constructivos, dos búsquedas locales y dos métodos de combinación de soluciones. Se presenta una experimentación factorial para elegir la mejor combinación de métodos para el algoritmo (constructivo, búsqueda local y método de combinación). Los resultados obtenidos muestran que el objetivo del trabajo se ha cumplido, mejorando los resultados encontrados en el estado del arte. Los resultados están respaldados por diferentes test estadísticos no paramétricos que confirman la calidad del algoritmo propuesto.

Agradecimientos

Este trabajo ha sido parcialmente financiado por la Comunidad de Madrid a través del proyecto con referencia S2009/TIC-1542 y por el Ministerio de Educación y Ciencia a través del proyecto con referencia (TIN2009-07516). Nos gustaría agradecer al Prof. Robert R. Lewis su colaboración compartiendo el código fuente de los métodos RCM y SA.

Referencias

1. Agrawal A, Klein P, Ravi R. Ordering problems approximated: single-processor scheduling and interval graph completion. Automata, Languages and Programming, 510:751–762 (1991).

2. Botafogo RA. Cluster analysis for hypertext systems. Proceedings of the 16th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, 116–125 (1993).
3. Cuthill E, McKee J. Reducing the bandwitdh of sparse symmetric matrices. ACM'69 Proceedings of the 1969 24th national conference, 157–172 (1969).
4. Das A, Roberts M. Metric distances of permutations. `www.cra.org/Activities/craw\_archive/dmp/awards/2004/Das/paper.ps`, (2004).
5. Duarte A, Martí R. Tabu Search and GRASP for the Maximum Diversity Problem. European Journal of Operational Research, 178:71–84 (2007).
6. Duarte A, Martí R, Pardo EG, Pantrigo JJ. Scatter Search for the cut width minimization problem. Annals of Operations Research, 199(1):285–304 (2012).
7. Duarte A, Glover F, Martí R, Gortázar F. Hybrid scatter stabu search for unconstrained global optimization. Annals of Operations Research, 183:95–123 (2011).
8. Gallego M, Duarte A, Laguna M, Martí R. Hybrid heuritics for the maximum diversity problem. Computational Optimization and Applications, 44(3):411-426 (2009).
9. Feo TA, Resende MGC. A probabilistic heuristic for a computationally difficult set covering problem. Operations Research Letters, 8(2):67–71 (1989).
10. Garey MR, Johnson DS. Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H.Freeman and Co., (1979).
11. Gibbs NE. A hybrid profile reduction algorithm. ACM Trans. Math. Softw., 2(4):378–387 (1976).
12. Guan G, Williams KL. Profile minimization of triangulated triangles. Discrete Mathematics, 260:69-76 (2003).
13. Glover F, Laguna M. Tabu Search. Kluwer Academic Publishers: Boston, (1997).
14. Karp R. Mapping the genome: some combinatorial problems arising in molecular biology. STOC'93 Proceedings of the twenty-fifth annual ACM symposium on Theory of computing, 278–285 (1993).
15. Kendall R. Incidence matrices, interval graphs and seriation in archaeology. Pacific Journal of Mathematics, 28:565–570 (1992).
16. Laguna M, Martí R. Scatter Search: methodology and implementations in C. Clubber Academic Publisher (2003).
17. Lewis J. The Gibbs-Poole-Stockmeyer and Gibbs-King algorithms for reordering sparse matrices. ACM Transactions on Mathematical Software (TOMS), 8:190–194 (1982).
18. Lewis R. Simulated annealing for profile and fill reduction. International Journal for Numerical Methods in Engineering, 37(6):905-925 (1994).
19. Lin Y, Yuan J. Profile minimization problem for matrices and graphs. Acta Math. Appl. Sinica, English Series, Yingyong Shuxue-Xuebas, 10(1):107–112 (1994).
20. Martí R, Duarte A, Laguna M. Advanced scatter search for the max-cut problem. INFORMS Journal on Computing, 21(1):26-38 (2009).
21. `http://math.nist.gov/MatrixMarket/`
22. Poole W, Stockmeyer P, Gibbs N. An algorithm for reducing the bandwitdh and profile of a sparse matrix. SIAM Journal on Numerical Analysis, 13:236–250 (1976).
23. Resende MGC, Smith SH, Feo TA. A greedy randomized adaptive search procedure for maximum independent set. Operations Research, 42:860–878 (1994).
24. Resende MGC, Martí R, Gallego M, Duarte A. GRASP and path relinking for the max-min diversity problem. Computers and Operations Research, 37:498-508 (2010).
25. Saad Y. Iterative methods for sparse linear systems. SIAM, ISBN-13:978-0-898715-34-7 (2003).
26. Tewarson, R. Sparse matrices. Academic Press: New York (1973).

Optimización del Problema de Planificación de Áreas de Localización con el Algoritmo NSGAII

Víctor Berrocal-Plaza*, Miguel A. Vega-Rodríguez, and Juan M. Sánchez-Pérez

Dept. de Tecnología de los Computadores y de las Comunicaciones,
Universidad de Extremadura, Escuela Politécnica, Campus Universitario S/N, 10003,
Cáceres, Spain
{vicberpla, mavega, sanperez}@unex.es

Resumen En este artículo se presenta nuestra adaptación del algoritmo evolutivo multiobjetivo Non-dominated Sorting Genetic Algorithm II (NSGAII) para optimizar una de las tareas de gestión más importantes en cualquier red de comunicaciones móviles, la gestión de la localización de los abonados. De entre las diferentes estrategias de gestión estudiamos aquella basada en Áreas de Localización por ser una de las más utilizadas. Esta estrategia define un problema de optimización multiobjetivo con dos funciones de coste. En trabajos anteriores, otros autores optaron por combinar linealmente estas dos funciones con el fin de reducir la complejidad del problema. Sin embargo, esta técnica tiene asociados varios inconvenientes. Es por eso que proponemos la adaptación de uno de los algoritmos de optimización multiobjetivo con más presencia en la literatura. Además, con una aproximación multiobjetivo se obtiene un amplio abanico de soluciones entre las que el operador de red puede seleccionar aquella que mejor se adapte al estado real de la red. A través de un estudio experimental demostramos que nuestra propuesta mejora a los algoritmos desarrollados por otros autores al mismo tiempo que se obtiene una amplia variedad de soluciones.

Keywords: Gestión de Localización Móvil, Non-dominated Sorting Genetic Algorithm II, Optimización Multiobjetivo, Problema de Planificación de Áreas de Localización.

1. Introducción

Las Redes Públicas Móviles Terrestres (PLMNs) son redes de telefonía móvil capaces de proporcionar servicios a una gran cantidad de abonados con escasos recursos radioelélectricos. Para ello, la zona de cobertura se divide en varias zonas de pequeño tamaño (conocidas como celdas), entre las que los recursos

* Este trabajo ha sido parcialmente subvencionado por el Ministerio de Economía y Competitividad y por el Fondo Europeo de Desarrollo Regional (FEDER) bajo el contrato TIN2012-30685 (proyecto BIO). Víctor Berrocal-Plaza disfruta del apoyo de la beca de investigación FPU-AP2010-5841, proporcionada por Ministerio de Educación.

disponibles son reutilizados de forma sistemática [1]. Además, estas redes deben ser capaces de prestar servicios en todo momento a usuarios estáticos o en movimiento, independiente de la posición en la que se encuentren. Esto hace que la gestión de la localización del abonado sea una de las tareas más importantes en cualquier red de telefonía móvil. Toda estrategia de gestión de localización consta de dos procedimientos principales: gestión de la actualización de la localización del abonado y gestión de la paginación o radiobúsqueda [2]. La actualización de la localización del abonado se utiliza para rastrear automáticamente el movimiento de los abonados por la red móvil. Existen varios métodos para determinar cuándo un usuario debe actualizar su localización (o celda en la que se encuentra): no actualizar nunca, cada vez que se detecta cambio de celda (actualizar siempre), cada cierto tiempo (actualización periódica), cada cierto número de celdas visitadas (actualización basada en distancia), cada vez que se entra en una celda determinada (actualización basada en celdas de notificación), cada cierto número de cambios de celda (actualización basada en movimiento), o cada vez que se cambia de área (actualización basada en Áreas de Localización). Por otra parte, la red móvil utiliza el procedimiento de paginación o radiobúsqueda para determinar la celda exacta en la que se encuentran los abonados que tienen una llamada entrante. Este procedimiento consiste en el envío de mensajes de búsqueda (ya sea secuencial o simultáneamente) entorno a la última localización conocida.

De entre todas las estrategias de gestión de localización, una de las más utilizadas es aquella basada en Áreas de Localización con un procedimiento de paginación simultáneo. Esta estrategia define un problema de optimización multiobjetivo (conocido como Problema de Planificación de Áreas de Localización, LAPP) perteneciente al grupo de problemas NP-Completos debido al gran tamaño del espacio objetivo [3]. Teniendo en cuenta esto último y considerando que el crecimiento anual de la demanda de servicios móviles sigue una tendencia exponencial, hace que sea realmente interesante el uso de metaheurísticas que nos permitan reducir el tráfico de señalización asociado a la gestión de la localización.

En trabajos anteriores, otros autores recurrieron a la combinación lineal de las funciones de coste para poder simplicar el problema de optimización y, de esta forma, poder aplicar metaheursticas mono-objetivo (SOAs: Single-objective Optimization Algorithms). Sin embargo, la combinación lineal tiene una serie de inconvenientes, tal y como se explica en la Sección 4. Es por eso que en este trabajo se propone la adaptación de un algoritmo de optimización multiobjetivo (the Non-dominated Sorting Genetic Algorithm II, NSGA-II) para evitar estos inconvenientes y, al mismo tiempo, obtener un conjunto de soluciones entre las que el operador de red puede seleccionar aquella que mejor se adapte al estado real de la red de señalización.

El resto del documento se organiza como se describe a continuación. La Sección 2 muestra los trabajos relacionados con el trabajo presentado en este artículo. La Sección 3 define y describe el Problema de Planificación de Áreas de Localización. Nuestra adaptación del algoritmo NSGA-II se presenta en la Sección

4. La Sección 5 muestra los resultados experimentales y las comparativas con los trabajos realizados por otros autores. Finalmente, nuestras conclusiones y trabajos futuros se discuten en la Sección 6.

2. Trabajos Relacionados

En la literatura podemos encontrar varios autores que tratan el Problema de Planificación de Áreas de Localización (LAPP), sin embargo en ninguno de ellos se utiliza una aproximación multiobjetivo. P. Gondim en [3] definió un Algoritmo Genético (GA) para encontrar configuraciones de Áreas de Localización (LAs) quasi-óptimas. P. Demestichas et al. [4] desarrollaron tres metaheurísticas mono-objetivo (Tabu Search (TS), Simulated Annealing (SA) y GA) para estudiar el comportamiento de las LAs en diferentes entornos. Lamentablemente, no se puede realizar una comparativa con estos dos trabajos previos debido a que sus redes de test no se encuentran disponibles. Posteriormente, J. Taheri y A. Y. Zomaya crearon y estudiaron con diferentes metaheurísticas (red neural de Hopfield (HNN) [5], SA [6], GA [7] y combinaciones de GA con HNN (GA-HNN) [8]) varias redes de test en las que los patrones de movilidad y llamada de usuario son similares a los que se pueden encontrar en redes reales. Estas redes también se estudiaron en [9] con el algoritmo Differential Evolution (DE), y serán las que estudiaremos en este trabajo.

3. Gestión de Áreas de Localización

La estrategia de gestión de localización basada en Áreas de Localización es una de las más utilizadas para el rastreo automático de los abonados por la red móvil. Por definición, un Área de Localización (LA) es un grupo continuo y no solapado de celdas de red. El objetivo principal de esta estrategia es acotar los procedimientos de actualización de localización y radiobúsqueda a nivel de LA. Es decir, un abonado sólo actualizará su localización (o celda en la que se encuentra) cuando se mueve a una nueva LA, y la paginación sólo se realiza en la última LA actualizada. De esta forma, el principal reto de esta estrategia es encontrar las configuraciones de LAs que minimicen simultáneamente el número de actualizaciones de localización (o coste de actualización, LU_{cost}) y el número de mensajes de paginación (o coste de paginación, PA_{cost}). Debe tenerse en cuenta que ambos costes (LU_{cost} y PA_{cost}) son conflictivos. Para reducir el LU_{cost} al mínimo, todas las celdas de la red deben pertenecer al mismo LA (i.e. no hay actualización de localización), lo que da lugar a un PA_{cost} máximo pues el procedimiento de paginación debe realizarse en toda la red cada vez que un abonado recibe una llamada. Y viceversa, el PA_{cost} es mínimo cuando la red conoce la celda exacta en la que se encuentra cada abonado en todo momento (i.e. la paginación sólo se realiza en una única celda), pero para ello es necesario que los usuarios actualicen su posición cada vez que cambian de celda (i.e. cada celda de red pertenece a un LA diferente) lo que provoca un LU_{cost} máximo.

Por tanto, el Problema de Planificación de Áreas de Localización define un problema de optimización multiobjetivo con dos funciones de coste (véase ecuaciones 1 y 2). La ecuación 1 representa la primera función de coste, en esta ecuación $\rho_{i,j}$ es una variable binaria que es igual a 1 sólo cuando la celda i y la celda j pertenecen a LAs diferentes, $N_{i,j}$ es el número de abonados que se mueven desde la celda i a la celda j, y $NCell$ es el número total de celdas de red. Y la ecuación 2 caracteriza a la segunda función de coste, en esta última ecuación NIC_i es el número de llamadas entrantes de la celda i, NA_k es el número de celdas pertenecientes al LA_k, CA_k es un vector que almacena las celdas pertenecientes al LA k, y $NArea$ es el número total de LAs.

$$f_1 = min\left\{LU_{cost} = \sum_{i=1}^{NCell}\sum_{j=1}^{NCell}\rho_{i,j} \times N_{i,j}\right\}, \tag{1}$$

$$f_2 = min\left\{PA_{cost} = \sum_{k=1}^{NArea}\sum_{i\in CA_k} NIC_i \times NA_k\right\}. \tag{2}$$

En trabajos anteriores [5,6,7,8,9], estas dos funciones de coste se combinaron en una única función con el objetivo de reducir la complejidad del problema de optimización (véase la ecuación 3, donde β es un coeficiente de ponderación para asignar más prioridad a uno de los dos objetivos). Sin embargo, la combinación lineal de las funciones de coste tiene una serie de inconvenientes (véase la Sección 4). Es por eso que en este trabajo se propone el uso de un algoritmo de optimización multiobjetivo.

$$f^{SOA}(\beta) = \beta f_1 + f_2. \tag{3}$$

4. Optimización Multiobjetivo

Todo problema de optimización multiobjetivo es un problema de optimización en el que se deben optimizar de forma simultánea dos o mas funciones objetivo conflictivas o contradictorias, como por ejemplo el Problema de Planificación de Áreas de Localización (LAPP). En este tipo de problemas, el reto es encontrar un amplio abanico de soluciones, cada una de ellas relacionada con un compromiso específico entre los objetivos. Sin embargo, algunos autores optan por reducir la complejidad del problema de optimización utilizando para ello la combinación lineal de las funciones de coste involucradas (véase la ecuación 3). Esta técnica permite simplicar la labor del optimizador pero tiene varios inconvenientes asociados. En primer lugar, se requiere de un conocimiento muy preciso del problema para configurar correctamente el/los coeficiente(s) de ponderación ($\beta \in \Re$). En segundo lugar y centrándonos en el problema estudiado (LAPP), la combinación lineal no tiene en cuenta que el valor adecuado de β puede ser diferente para diferentes estados de la red de señalización (la carga de la red es variable en el tiempo). Y en tercer y último lugar, el resultado de

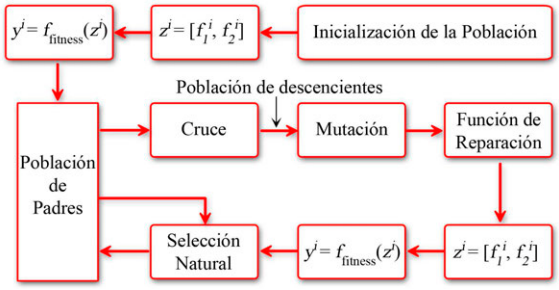


Figura 1. Descomposición de tareas de NSGAII

optimizar la ecuación 3 es una solución específica para el valor de β utilizado, requiriéndose de una ejecución independiente del optimizador para cada valor de este coeficiente.

Con el objetivo de evitar todos estos inconvenientes, en este trabajo se propone una adaptación del algoritmo Non-dominated Sorting Genetic Algorithm II (NSGAII [10], uno de los más utilizados en el campo de la optimización multiobjetivo) para optimizar el problema de gestión de Áreas de Localización. Este algoritmo es un algoritmo genético elitista con una función de fitness que nos permite ordenar las soluciones obtenidas en función de su calidad en el contexto multiobjetivo. La función de fitness del algoritmo NSGAII consta de dos terminos principales: la *ordenación no dominada* y la *distancia de crowding*. Mediante la *ordenación no dominada* las distintas soluciones obtenidas se ordenan en *Frentes* de tal forma que en un mismo *Frente* no se puede establecer el concepto de *dominancia*. Por definición, una solución z^i se dice que *domina* a otra solución z^j ($z^i \prec z^j$) cuando z^i mejora a z^j en al menos un objetivo y z^i iguala a z^j en el resto de objetivos. Una vez finalizada la ordenación por *Frentes*, se utiliza la *distancia de crowding* para determinar cuáles son las mejores soluciones dentro de un mismo *Frente*. Las mejores soluciones serán aquellas con el mayor y/o menor valor de las funciones de coste y aquellas que estén más separadas con respecto a sus soluciones inmediatamente más próximas en el espacio objetivo. La finalidad de esta última clasificación es favorecer la dispersión de los *Frentes*.

La Figura 1 presenta la descomposición de tareas del algoritmo NSGAII. Como todo algoritmo genético, NSGAII es una metaheurística poblacional que utiliza los operadores evolutivos de los sistemas biológicos para mejorar de forma iterativa un conjunto de soluciones. Por definición, una metaheurística poblacional es toda aquella metaheurística que gestiona una o más poblaciones de individuos, donde cada individuo codifica una solución factible del problema. Como se observa en la Figura 1, el primer paso que se debe realizar es la inicialización y la evaluación de la primera población de padres, el procedimiento de inicialización se describe en la Sección 4.1. Y a continuación, se ejecuta un método iterativo hasta que se alcanza la condición de parada definida, que en este trabajo es el *Número Máximo de Generaciones*. La primera tarea dentro de este método es la operación de cruce, mediante la cual los individuos de la población padre se

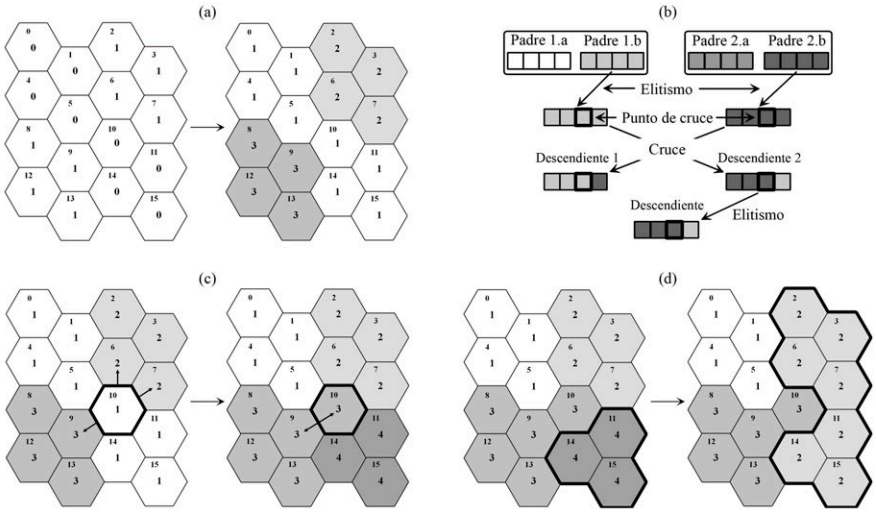


Figura 2. Operadores evolutivos: (a) inicialización del individuo, (b) operación de cruce, (c) mutación a nivel de gen, (d) mutación a nivel de Áreas de Localización

combinan para formar la población de descendientes, véase la Sección 4.2. Posteriormente, se aplica la operación de mutación sobre la población de descendientes obtenida en el paso anterior, esta operación está definida en la Sección 4.3. Tras estas dos operaciones (cruce y mutación), se utiliza una función de reparación cromosómica para transformar soluciones inviables en soluciones factibles. Y por último, se seleccionan los mejores individuos obtenidos hasta el momento como la población de padres de la siguiente generación, este procedimiento se conoce como selección natural.

4.1. Representación e Inicialización del Individuo

En este trabajo, cada individuo de la población es un vector de enteros que almacena el Área de Localización asociada con cada celda de red. En la inicialización de la población, cada uno de estos vectores se llena con un patrón aleatorio de 0s y 1s, el cual se utilizará para conocer la configuración de LAs. Para ello, y teniendo en cuenta la definición de Área de Localización (grupo continuo y no solapado de celdas de red), cada Área de Localización estará constituída por grupos continuos de celdas con el mismo valor de vector. Un ejemplo de este procedimiento se puede observar en la Figura 2.a, donde el número situado en la parte superior izquierda es el identificador de celda y el número central representa el Área de Localización asignada a esa celda.

4.2. Operación de Cruce

La operación de cruce se realiza con probabilidad P_C para generar una población de descendientes del mismo tamaño que la población de padres. En este

trabajo se ha implementado un cruce elitista en el que el número de puntos de cruce es seleccionado al azar en el intervalo [1, 4]. La Figura 2.b muestra la operación de cruce utilizada en este trabajo. Primero se seleccionan cuatro individuos agrupados en parejas y se elige como padre el mejor individuo de cada grupo. A continuación, cada padre se divide en fragmentos y se combina con los fragmentos del otro padre con el fin de obtener dos descendientes. Finalmente, se almacena en la población de descendientes el mejor de estos dos nuevos individuos.

4.3. Operación de Mutación

La operación de mutación se realiza con probabilidad P_M sobre la población de descendientes. Esta operación modifica levemente la información genética de la descendencia con el fin de explorar zonas del espacio objetivo desconocidas. En este trabajo hemos desarrollado dos operaciones de mutación denominadas *mutación a nivel de gen* y *mutación a nivel de LA*. En la *mutación a nivel de gen* se fusiona una celda frontera (celda que es frontera entre dos o más LAs) con su LA vecina de menor tamaño. Un ejemplo de esta mutación se muestra en la Figura 2.c. Y la *mutación a nivel de LA* consiste en fusionar el LA más pequeña con su LA vecina de menor tamaño. La Figura 2.d muestra un ejemplo de este tipo de mutación.

5. Resultados Experimentales

En esta sección se muestran nuestros resultados experimentales y una comparativa con los trabajos desarrollados por otros autores. Se han estudiado cuatro redes de test de diferente complejidad: LA25 (red de 5×5 celdas), LA35 (red de 5×7 celdas), LA49 (red de 7×7 celdas) y LA63 (red de 7×9 celdas). Todas estas redes han sido desarrolladas a partir de modelos de movilidad de usuario y patrones de llamada realistas, lo que las hace muy interesantes [5,6,7,8]. Además, estas mismas redes han sido estudiadas con diferentes metaheurísticas, todas ellas del campo de la optimización mono-objetivo: Red Neural de Hopfield (HNN) [5], Simulated Annealing (SA) [6], Algoritmo Genético (GA) [7], combinaciones de GA con HNN (GA-HNN) [8] y Evolución Diferencial (DE) [9]. Sin embargo, con el propósito de evitar los inconvenientes de la combinación lineal de las funciones objetivo, en este trabajo se propone la adaptación de un algoritmo de optimización multiobjetivo (NSGAII) para encontrar configuraciones quasi-óptimas de Áreas de Localización. Por lo que para poder comparar nuestra propuesta con los trabajos realizados por otros autores debemos buscar en nuestro *Frente de Pareto* la solución que mejor optimiza la función objetivo utilizada por estos algoritmos mono-objetivo, que es la ecuación 3 con β igual a 10 ($f^{SOA}(\beta = 10)$). Y será esta solución la que se comparará con la mejor solución encontrada en [5,6,7,8,9].

Para realizar una comparativa justa con los trabajos [5,6,7,8,9], utilizamos la misma condición de parada (*Máximo Número de Generaciones* = 5000 generaciones) y el mismo tamaño de población ($N_{pop} = 250$ individuos). El resto de

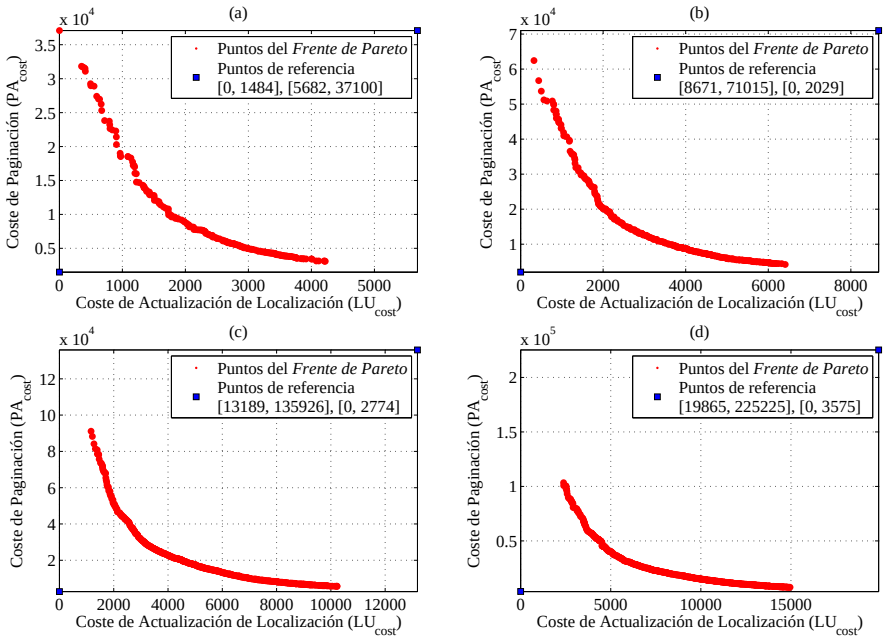


Figura 3. *Frentes de Pareto*: (a) LA25, (b) LA35, (c) LA49, (d) LA63

parámetros de NSGAII se han configurado por medio de un estudio paramétrico de 30 ejecuciones independientes por experimento. La configuración de parámetros que proporciona el mayor *Hipervolumen* es: $P_C = 0{,}8$ y $P_M = 0{,}4$. El *Hipervolumen* (I_H) es un indicador de calidad multiobjetivo que, en el caso de dos objetivos, mide el área del espacio objetivo dominada por el *Frente de Pareto* y acotada por los *puntos de referencia* (puntos calculados por medio de los valores máximo y mínimo de cada función de coste, ecuaciones 1 y 2) [11]. En el Problema de Planificación de Áreas de Localización, estos puntos se obtienen por medio de las configuraciones de LAs extremas: cuando todas las celdas de red pertenecen al mismo LA, y cuando cada celda de red pertenece a un LA diferente.

El Cuadro 1 muestra los resultados del estudio estadístico realizado con la configuración anterior. En este estudio se han calculado la media y la desviación estándar del *Hipervolumen* obtenido en 30 ejecuciones independientes de nuestro algoritmo. Una representación gráfica de los *Frentes de Pareto* cuyo I_H es el más próximo al I_H medio se muestra en la Figura 3. En esta figura se puede observar que el PA_{cost} crece más rápido que el LU_{cost} y que la densidad de puntos del *Frente de Pareto* disminuye a medida que aumenta el PA_{cost}. Esto último puede ser debido a que el número de soluciones factibles disminuye a medida que aumenta el tamaño de las Áreas de Localización. Una vez obtenido los *Frentes de Pareto*, el siguiente paso es buscar en cada *Frente de Pareto* la solución que mejor optimiza la función de coste utilizada en los trabajos

Cuadro 1. Datos estadísticos del *Hipervolumen* (I_H)

Red de test	$media\,\{I_H(\%)\}$	$\sigma\,\{I_H(\%)\}$
LA25	72,07	0,009
LA35	77,31	0,946
LA49	79,16	1,299
LA63	79,14	1,863

Cuadro 2. Comparación con otros trabajos. Optimización de $f^{SOA}(\beta = 10)$

Red de test	$NSGAII$	$GA^{[7]}$	$HNN^{[5]}$	$DE^{[9]}$	$GA-HNN_1^{[8]}$	$GA-HNN_2^{[8]}$	$GA-HNN_3^{[8]}$	$SA^{[6]}$
LA25	26990	28299	27249	26990	26990	26990	26990	26990
LA35	39832	40085	39832	39859	40117	39832	39832	42750
LA49	60849	61938	63516	61037	62916	62253	60696	60694
LA63	89085	90318	92493	89973	92659	91916	91819	90506

anteriores [5,6,7,8,9] (que es la ecuación 3 con β igual a 10, $f^{SOA}(\beta = 10)$), para finalmente comparar esta solución con la mejor solución encontrada en estos trabajos. El Cuadro 2 muestra la comparativa con los mejores resultados obtenidos por otros autores. En este cuadro se observa que nuestra adaptación de NSGAII mejora a los algoritmos mono-objetivo implementados en [5,6,7,8,9] en las dos redes de test más complejas (a excepción de los algoritmos SA [6] y GA-HNN_3 [8] en la red LA49). Tarea que queda lejos de ser trivial, ya que nuestra adaptación de NSGAII (especializada en encontrar un amplio abanico de soluciones) debe competir contra algoritmos especializados en encontrar una única solución.

6. Conclusiones y Trabajo Futuro

Este artículo presenta nuestra adaptación de un algoritmo evolutivo multiobjetivo (el algoritmo Non-dominated Sorting Genetic Algorithm II, NSGAII [10]) para optimizar una de las tareas más importantes en cualquier red de comunicaciones móviles, la gestión de la localización de los abonados. De entre las distintas estrategias de gestión se ha optado por estudiar aquella basada en Áreas de Localización, puesto que es una de las más utilizadas en las redes móviles actuales. Esta estrategia define un problema de optimización multiobjetivo conocido como Problema de Planificación de Áreas de Localización (LAPP). El principal aporte de este trabajo, y que lo distingue de los estudios realizados por otros autores, es que se propone una técnica de optimización multiobjetivo para evitar los diversos inconvenientes derivados de la combinación lineal de las funciones de coste. Además, con una aproximación multiobjetivo se obtiene una amplia gama de soluciones (*Frente de Pareto*) entre las cuales el operador de red puede seleccionar aquella que mejor se adapte al estado real de la red

de señalización en cada momento. Con el fin de conocer la calidad de nuestra propuesta, se han estudiado cuatro redes de prueba de diferente complejidad y se han comparado nuestros resultados con los mejores resultados obtenidos en [5,6,7,8,9]. En este estudio experimental se demuestra que nuestra adaptación de NSGAII, al mismo tiempo que obtiene un amplio abanico de soluciones, supera a los algoritmos mono-objetivo desarrollados por otros autores en las dos redes de test más complejas. Tarea nada trivial, ya que se está comparando con algoritmos especializados en encontrar una única solución.

Como trabajo futuro se propone mejorar los operadores evolutivos presentados en este trabajo para explorar zonas del espacio objetivo con valores extremos de ambas funciones de coste. Además, podría ser interesante adaptar otros algoritmos de optimización multiobjetivo y estudiar otras redes de test.

Referencias

1. Agrawal, D., Zeng, Q.: Introduction to Wireless and Mobile Systems. Cengage Learning (2010)
2. Kyamakya, K., Jobmann, K.: Location management in cellular networks: classification of the most important paradigms, realistic simulation framework, and relative performance analysis. IEEE Transactions on Vehicular Technology **54**(2) (2005) 687–708
3. Gondim, P.: Genetic algorithms and the location area partitioning problem in cellular networks. In: Procedings of the IEEE 46th Vehicular Technology Conference on Mobile Technology for the Human Race. Volume 3. (1996) 1835–1838
4. Demestichas, P., Georgantas, N., Tzifa, E., Demesticha, V., Striki, M., Kilanioti, M., Theologou, M.E.: Computationally efficient algorithms for location area planning in future cellular systems. Computer Communications **23**(13) (2000) 1263–1280
5. Taheri, J., Zomaya, A.Y.: The use of a hopfield neural network in solving the mobility management problem. In: Proceedings of The IEEE/ACS International Conference on Pervasive Services. (2004) 141–150
6. Taheri, J., Zomaya, A.Y.: A simulated annealing approach for mobile location management. In: Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium. (2005) 194–194
7. Taheri, J., Zomaya, A.Y.: A genetic algorithm for finding optimal location area configurations for mobility management. In: Proceedings of the The IEEE Conference on Local Computer Networks 30th Anniversary. LCN '05, Washington, DC, USA, IEEE Computer Society (2005) 568–577
8. Taheri, J., Zomaya, A.Y.: A combined genetic-neural algorithm for mobility management. J. Math. Model. Algorithms (2007) 481–507
9. Almeida-Luz, S.M., Vega-Rodríguez, M.A., Gómez-Púlido, J.A., Sánchez-Pérez, J.M.: Differential evolution for solving the mobile location management. Appl. Soft Comput. **11**(1) (January 2011) 410–427
10. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. Trans. Evol. Comp **6**(2) (April 2002) 182–197
11. Coello, C.A.C., Lamont, G.B., Veldhuizen, D.A.V.: Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation). Springer-Verlag New York, Inc., Secaucus, NJ, USA (2006)

Metaheurísticas para Smart Mobility: Reducción de Emisiones y Consumo de Carburantes en el Tráfico Urbano

José García-Nieto and Enrique Alba

Dept. de Lenguajes y Ciencias de la Computación,
Universidad de Málaga, E.T.S.I. Informática,
B. Louis Pasteur 35, 29071, España
{jnieto,eat}@lcc.uma.es

Resumen Hoy en día, la mejora del tráfico vehicular supone una labor ineludible en nuestras ciudades a la hora de mitigar problemas como la excesiva emisión de gases contaminantes y el consumo no sostenible de carburantes. El concepto de Smart Mobility enmarca el conjunto de acciones, técnicas e infraestructuras encaminadas a la mejora de la movilidad y organización del tráfico en las ciudades. En este contexto, proponemos en este trabajo una estrategia de optimización basada en técnicas metaheurísticas de Inteligencia Colectiva para la programación de ciclos en los semáforos desplegados en grandes zonas metropolitanas. De esta forma, pretendemos mejorar el flujo del tráfico de los vehículos con el objetivo global de reducir las emisiones de gases contaminantes (CO_2 y NO_x) y el consumo de combustible en las áreas observadas. En este estudio seguimos el modelo de referencia en la Unión Europea para las emisiones de tráfico HBEFA y nos centramos en dos áreas urbanas localizadas en las ciudades de Málaga y Sevilla. Tras varias comparativas entre diferentes técnicas de optimización y con soluciones dadas por expertos en el dominio, comprobamos cómo nuestra propuesta es capaz de obtener reducciones significativas en términos de emisiones contaminantes y de consumo de carburante.

Keywords: Programación de Semáforos, Particle Swarm Optimization, Simulador de Tráfico SUMO, Modelo de Emisiones de Tráfico HBEFA

1. Introducción

En las últimas décadas, el elevado nivel de contaminación del aire y el consumo de hidrocarburos derivado del tráfico urbano están desembocando en serios problemas que directamente afectan a la salud de los ciudadanos, a la economía global y al mantenimiento de nuestras ciudades. Por tanto, la mejora de la movilidad supone una tarea fundamental en las grandes áreas metropolitanas, lo cual repercute en la gestión eficiente del tráfico vehicular. Debido a que la modernización de las infraestructuras físicas es muy costosa y no es siempre abordable, los esfuerzos en este sentido deben dirigirse en gran medida hacia una planificación

eficiente de los semáforos. De esta forma se puede agilizar, con un bajo coste, la circulación de vehículos en las ciudades [7,8,9]. No obstante, el número de semáforos operativos está creciendo y su planificación conjunta es cada vez más compleja debido a la gran cantidad de combinaciones de periodos de colores que se deben manejar. Por tanto, la utilización de sistemas automáticos inteligentes para la programación óptima de ciclos de semáforos es una tarea indispensable en la gestión del tráfico vehicular.

En este sentido, el empleo de técnicas inteligentes de optimización, como los algoritmos metaheurísticos [1], ha ido ganando importancia con el tiempo ya que se ha demostrado su adecuación en la planificación de semáforos [3,9,10]. Sin embargo, en ningún caso (según nuestro conocimiento) se ha tratado la optimización de los programas de ciclos desde la perspectiva medioambiental, para la reducción de emisiones contaminantes y de consumo de fuel. Además, en la mayoría de los casos, el uso de dichos sistemas inteligentes se ha limitado por lo general a la optimización de instancias académicas pequeñas, con uno o dos cruces y con 8 semáforos como mucho, o que están diseñados en exclusiva para zonas muy específicas, impidiendo estudios de ciudades completas.

Por tanto, la motivación principal de este trabajo es la de proponer una estrategia de optimización, basada en un algoritmo de Cúmulo de Partículas (o PSO, Particle Swarm Optimization) [5], para la obtención de programas de ciclos de semáforos eficientes desde el punto de vista de varios factores medioambientales: emisiones de CO_2 y NO_x, además de la cantidad total de combustible consumido por los vehículos. Lo haremos además sobre grandes áreas urbanas y para un elevado número de semáforos y vehículos, demostrando que las metaheurísticas son una tecnología clave para el escalado futuro en este dominio de aplicación.

PSO es un algoritmo de fácil configuración que por lo general desarrolla una rápida convergencia a soluciones de calidad [5]. Esto es una característica deseable ya que nos permite obtener soluciones satisfactorias en fases tempranas de optimización. Para la evaluación de los programas de ciclos generados (codificados como vectores solución), hemos utilizado el simulador de tráfico microscópico SUMO (Simulator of Urban Mobility) [6], mediante el que se obtiene una fuente de información constante sobre el flujo de vehículos. Esto es fundamental para poner usar un algoritmo avanzado y hacer planificación automática. Además, SUMO nos permite trabajar con el modelo estándar de emisiones HBEFA (HandBook Emission FActors) [4], a través del cual obtenemos la información sobre las emisiones de polución y el consumo de combustible, que será utilizada por el algoritmo de optimización para la evaluación de soluciones.

Para este estudio hemos utilizado dos instancias localizadas en los centros urbanos de Málaga y Sevilla, con cientos de semáforos y diferentes densidades de tráfico (entre 250 y 500 vehículos circulando en tiempo de simulación). Los resultados y comparaciones con otras técnicas: Evolución Diferencial (DE) [11], Random Search (RANDOM) y Sumo Cycle Program Generator (SCPG) [6], ponen de relieve las mejoras significativas obtenidas por nuestra propuesta en términos de emisiones de CO_2 y NO_x, así como de consumo de combustible.

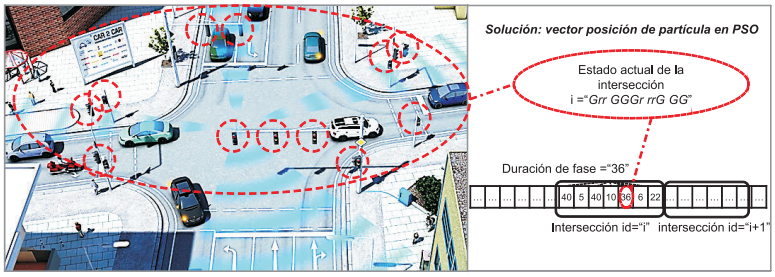


Figura 1. Programa de ciclos en una intersección

Este trabajo se organiza de la siguiente manera. En la Sección 2 se presenta el problema de la programación óptima de ciclos en semáforos y se introduce el modelo de emisiones HBEFA. La Sección 3 describe nuestra estrategia de optimización. La Sección 4 está dedicada a los experimentos y análisis de soluciones. Las conclusiones globales y el trabajo futuro se resumen en la Sección 5.

2. Programas de Ciclos y Modelo de Emisión HBEFA

En la red vial de tráfico, los semáforos se sitúan en las intersecciones y controlan el flujo del tráfico mediante sus programas de estados de colores y periodos de duración. Dentro de una misma intersección, todos los semáforos se gobiernan por un programa común, ya que deben estar necesariamente sincronizados por motivos de seguridad. Además, para todos los semáforos en una misma intersección, la combinación de colores durante un periodo de ciclo debe ser siempre válida, siguiendo las reglas de tráfico específicas con el fin de evitar colisiones y accidentes en general. En nuestro modelo trabajamos exclusivamente con combinaciones válidas de estados de colores para cada intersección, las cuales permanecen constantes durante el proceso de optimización. Así se evita la aparición de combinaciones de estado no válidas y se restringe la optimización únicamente a la búsqueda de estados factibles.

En este contexto, nuestro principal objetivo consiste en encontrar **programas de ciclos optimizados** para todos los semáforos situados en una determinada área urbana. Definimos un ciclo como el periodo de tiempo en el que un conjunto de semáforos (en un cruce) permanecen con sus estados de luces/colores fijos. Al mismo tiempo, estos programas deben también coordinarse con semáforos en intersecciones adyacentes, mejorando así el flujo de vehículos que circulan conforme a la regulación vial establecida.

- *Codificación.* Siguiendo la especificación de SUMO para la programación de ciclos, hemos codificado las posibles soluciones como vectores de naturales. De esta forma, cada elemento del vector (variable) representa una duración de fase de los semáforos implicados en una determinada intersección. La Figura 1 muestra un ejemplo de codificación representando la intersección "*i*", la cual consta de 12 semáforos con estado actual *"Grr GGGr rrG GG"*, es decir 7 semáforos

en verde (G) y los otros 5 en rojo (r), durante 36 segundos. La siguiente fase cambiará el estado de los semáforos a otra combinación válida *"yGG rrry GGr yy"* (siendo 'y' ámbar), durante 6 segundos. Tras finalizar la última fase en una intersección (el número de fases en cada intersección puede ser diferente), vuelve a tomar la primera actuando de manera cíclica. Esta representación nos permite además tener en cuenta la interdependencia entre variables, no sólo para duraciones de fase en una misma intersección, sino que también para semáforos en intersecciones adyacentes.

Por otra parte, SUMO nos permite simular factores medioambientales basados en el modelo HBEFA (HandBook Emission Factors for Road Transport) [4]. A través de este modelo podemos simular diferentes condiciones de vehículos con información sobre aceleraciones, deceleraciones, frenado, velocidades máximas, así como su repercusión en las emisiones basadas en HBEFA: hidrocarburos, partículas emitidas, niveles de CO, CO_2, NO_x y otros agentes de polución. En este estudio nos hemos centrado en los agentes contaminantes CO_2 y NO_x, además del consumo de combustible ($Fuel = F$).

- *Función de Fitness.* Para evaluar cada programa de ciclos se ha formulado la siguiente función de fitness $f(s)$, mediante la cual se considera la información del tráfico obtenida por simulación con datos reales de la ciudad:

$$f(s) = \frac{\| \sum_{V_l} CO_2(s) \|_l + \| \sum_{V_l} NO_x(s) \|_l + \| \sum_{V_l} F(s) \|_l}{\#edges} + \rho \cdot \overline{V}(s) \qquad (1)$$

El objetivo principal consiste en minimizar los niveles de emisiones y consumo general en los escenarios estudiados. Como puede observarse en la Ecuación 1, en primer término se consideran los cómputos de cada indicador de emisión por vehículo (V), normalizados por la longitud de la vía (l) en la que se obtienen dichos niveles de emisión, CO_2 y NO_x en mg/s, además del consumo F en l/s. Esta suma de tres términos se divide además entre el número de vías/calles ($\#edges$) transitadas por los vehículos. Finalmente, se considera el producto de los vehículos que no llegan al destino en tiempo de observación ($\overline{V}$), por una constante de gasto medioambiental ρ. Debido a que no es posible computar los niveles medioambientales de los vehículos que quedan atascados (no llegan en tiempo de simulación), se ha calculado esta constante $\rho = 0{,}75$ como el gasto aproximado medio por vehículo en recorrer su ruta en diferentes condiciones de densidad de tráfico.

3. Estrategia de Optimización

Nuestra estrategia de optimización se compone por dos partes principales: la optimización con Particle Swarm Optimization (PSO) [5] y la obtención de datos desde SUMO. Para el desarrollo de nuestro PSO hemos seguido las especificaciones del último estándar 2011 [2]. En este algoritmo, cada posición de partícula x^i se actualiza cada iteración g mediante la Ecuación 2.

$$x^i_{g+1} = x^i_g + v^i_{g+1} \qquad (2)$$

donde el término v_{g+1}^i se calcula por la actualización de la velocidad:

$$v_{g+1}^i = w \cdot v_g^i + Gr_g^i - x_g^i + HS(Gr, \parallel Gr - \boldsymbol{x_g} \parallel) \tag{3}$$

$$con \;\; Gr_g^i = \frac{x_g^i + p_g'^i + l_g'^i}{3}, \;\; p_g'^i = x_g^i + c \cdot (p_g^i - x_g^i), \;\; l_g'^i = x_g^i + c \cdot (l_g^i - x_g^i) \tag{4}$$

En esta fórmula, p_g^i es la mejor solución que la partícula i ha obtenido en toda su historia, l_g^i es la mejor partícula dentro de un vecindario de dimensión $k = 3$ seleccionada aleatoriamente (uniforme) del cúmulo y w es el factor de inercia (decrece linealmente con el número de iteraciones de 0,5 a 0,1). El coeficiente de aceleración $c > 1$ es un valor aleatorio normal con $\mu = 1/2$ y $\rho = 1/12$. Este coeficiente se actualiza de nuevo para cada componente en el vector de velocidad. Por último, HS es un generador de números aleatorios en una hiper-esfera (Hyper-Sphere), con Gr como centro de gravedad. Es decir, Gr se calcula como el punto equidistante entre p'_g, l'_g y x_g. Con este mecanismo se consigue la independencia sobre el sistema de coordenadas, mejorando este aspecto sobre las anteriores versiones de PSO [2].

Debido a que la optimización de programas de ciclos se codifica mediante vectores de números naturales (representando ciclos de duración de fase), hemos utilizado el método de cuantificación (*quantisation*) provisto en la especificación del estándar PSO 2011 [2]. El método de cuantificación se aplica a cada nueva partícula (generada con la Ecuación 2) y transforma las variables continuas a discretas. Básicamente consiste en un cuantificador uniforme de Mid-Thread como se especifica en la Ecuación 5, con paso de cuantificación $\Delta = 0{,}5$.

$$Q(x) = \Delta \cdot \lfloor x/\Delta + 0{,}5 \rfloor \tag{5}$$

La simulación se utiliza para asignar un valor de calidad *fitness* a las soluciones codificadas en las partículas, para un determinado escenario urbano. Esta simulación se lleva a cabo instanciando SUMO con datos de la ciudad. Así, cuando el algoritmo PSO genera una nueva solución con un nuevo programa de ciclos, se inicia SUMO para procesar distintos escenarios de tráfico sobre la ciudad con: calles, direcciones, semáforos, vehículos, rutas, etc., respecto al nuevo programa de ciclos. Tras el procesado de diferentes escenarios, SUMO devuelve los valores necesarios para computar la función de *fitness* (Ecuación 1). Cada nuevo programa de ciclos se carga en cada proceso de simulación. En este sentido, lo demandado actualmente en ingeniería civil son programas de ciclos constantes para áreas y para franjas horarias preestablecidas (hora punta, horario nocturno, festivos, etc.), lo cual nos llevó a abordar el problema con este enfoque.

4. Experimentos

En este trabajo hemos generado dos escenarios de tráfico con información real a partir de mapas digitalizados (ilustrados en la Figura 2). Estos escenarios están físicamente localizados en las ciudades de Málaga y Sevilla cubriendo áreas similares de aproximadamente 0,75 km^2.

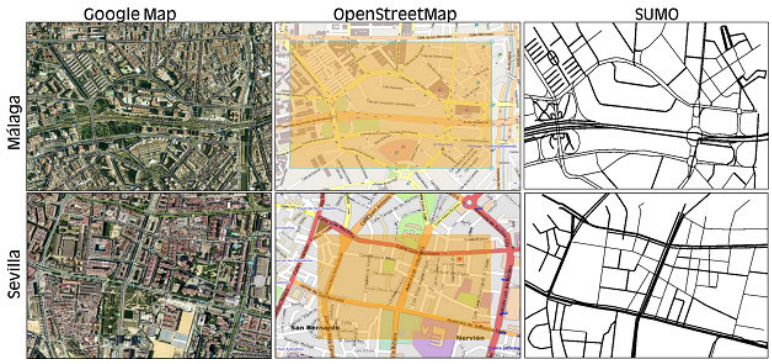


Figura 2. Instancias de Málaga y Sevilla exportadas al formato de SUMO

La información utilizada para ello consiste en: reglas de tráfico regulado, situación de las señales, direcciones, calles, intersecciones, etc. Además, se ha determinado el número de vehículos circulando, así como sus velocidades a partir de las especificaciones de las Delegaciones de Movilidad de estas ciudades [1]. Dichas áreas contienen las siguientes características:

1. *Málaga*. Localizado entre las avenidas Andalucía y Aurora, este escenario (Figura 2, arriba) se compone de avenidas y calles con diferente longitud y número de vias, además de varias rotondas. Contiene intersecciones incluyendo desde 4 a 16 semáforos cada una.
2. *Sevilla*. Cubre el céntrico distrito de Nervión (Figura 2, abajo). Se compone de intersecciones con entre 4 y 17 semáforos cada una.

El número de intersecciones estudiadas es de 70 en ambos escenarios aunque, cabe clarificar, que el número de semáforos no es el mismo ya que dichas intersecciones son muy diferentes entre sí, sumando 304 semáforos en Málaga y 323 en el caso de Sevilla. Además, para cada escenario hemos generado dos instancias con diferentes densidades de tráfico con 250 y 500 vehículos circulando en un tiempo de estudio de 1.200 segundos (iteraciones de microsimulación). Este tiempo de duración fue determinado como el tiempo máximo que tarda un coche en completar su ruta, incluso si éste debiera parar en todos los semáforos que encontrara y con la densidad de tráfico inicial. Cuando un vehículo abandona el recorrido del escenario, ya no volverá a aparecer de nuevo durante el tiempo de observación. Cada uno de los vehículos recorre su propia ruta con una velocidad máxima de 50 km/h (velocidad típica en la áreas urbanas).

Para la ejecución del algoritmo hemos utilizado la implementación en C++ de PSO (Standard 2011) que provee la biblioteca MALLBA [1]. La fase de simulación se realiza mediante el simulador SUMO en su versión 0.12.0 para Linux.

[1] Málaga(http://movilidad.malaga.eu/), Sevilla (http://www.trajano.com/). Esta información se obtuvo a partir de puntos sensorizados en ciertas calles de la ciudad, calculando así una medida de la densidad de tráfico en varios intervalos temporales.

Los experimentos se realizaron en clusters de máquinas, en su mayoría equipadas con procesadores Quad Core (4 núcleos), 2GB de RAM y S.O. Linux Debian.

Para cada escenario se realizaron 30 ejecuciones independientes de nuestro PSO. El tamaño del cúmulo es de 30 partículas, realizando 300 iteraciones de optimización (resultando de esta manera 9.000 evaluaciones de SUMO) por ejecución e instancia. Como ya comentamos en la sección anterior, el tamaño de la partícula depende directamente del número de fases en cada programa de ciclos de los semáforos, siendo 312 y 368 para las instancias de Málaga y Sevilla, respectivamente.

De manera adicional, hemos implementado para esta experimentación un algoritmo de Evolución Diferencial (DE) [11] en su versión Canónica $DE\backslash rand\backslash 1$ y otro de Random Search, los dos en el ámbito de la biblioteca MALLBA. De esta forma, sometiendo estos algoritmos al mismo procedimiento experimental que a PSO esperamos obtener algunos indicios sobre la potencia de nuestra propuesta, respecto a otra metaheurística (en el caso de DE) con modelo de aprendizaje cercano al de PSO, así como respecto al un algoritmo puramente estocástico, como en el caso de Random Search (RAND). El número máximo de evaluaciones a realizar por estos dos algoritmos fue también de 9.000.

Por último, añadimos un cuarto algoritmo a la comparación que consiste en el generador de programas de ciclos de SUMO (SCPG). Este último algoritmo consiste básicamente en asignar a las duraciones de fase de los semáforos nuevos valores en el rango $(6, 31)$, de acuerdo a tres factores diferentes: la proporción de estados en verde de las fases, el número de vías de entrada en las intersecciones y el tiempo de frenado de los vehículos cuando se acercan a los semáforos[2]. De esta forma, podemos contrastar también el rendimiento de nuestra propuesta frente al modelo humano que SCPG representa.

4.1. Resultados y Comparaciones

En primer lugar, analizamos en esta sección el comportamiento de nuestra propuesta desde el punto de vista de su rendimiento computacional. En la Tabla 1 podemos ver los resultados, en cuanto a valor de calidad (*fitness*), obtenidos por nuestro PSO. Estos resultados consisten en el máximo (Max.), la mediana (Med.), el mínimo (Min.), la media (Mea.) y la desviación Típica (Std.) de las distribuciones obtenidas para los escenarios de Málaga y Sevilla, con densidades de tráfico de 250 y 500 vehículos. Junto a estos resultados se disponen además los obtenidos utilizando DE, RAND y SCPG, siguiendo el mismo procedimiento de optimización y sobre las diferentes instancias. Además, en esta misma tabla, las últimas filas para cada instancia contienen el resultado de realizar tests estadísticos (no-paramétricos) [12] de Friedman y Holm para determinar qué algoritmo muestra mejor *ranking* y si existen diferencias significativas con respecto a éste y los demás. En concreto, estos resultados consisten en los *p-valores ajustados* por el test de Holm (HA_p) para un valor de confianza estadística $\alpha = 0{,}05$.

[2] La información completa sobre el funcionamiento del algoritmo SCPG en [6].

Tabla 1. Valores de calidad obtenidos para Málaga y Sevilla con 250 y 500 vehículos

Ciudad	Valor	Densidad de tráfico (Número de vehículos)							
		250				500			
		PSO	DE	RAND	SCPG	PSO	DE	RAND	SCPG
Málaga	Max,	**1,04E+02**	1,08E+02	1,11E+02	1,05E+02	**2,95E+02**	3,06E+02	3,42E+02	5,59E+02
	Med,	**9,76E+01**	1,02E+02	1,08E+02	1,05E+02	**2,41E+02**	2,62E+02	3,07E+02	5,59E+02
	Min,	**9,12E+01**	9,83E+01	1,04E+02	1,05E+02	**1,92E+02**	2,25E+02	2,66E+02	5,59E+02
	Mea,	**9,72E+01**	1,02E+02	1,08E+02	1,05E+02	**2,41E+02**	2,64E+02	3,04E+02	5,59E+02
	Std,	**3,00E+00**	1,93E+00	1,50E+00	0,00E+00	**2,50E+01**	1,94E+01	1,95E+01	0,00E+00
	HA_p	-	*1,42E-02*	*1,78E-08*	*1,33E-16*	-	*1,09E-01*	*5,20E-06*	*5,49E-16*
Sevilla	Max,	**2,04E+02**	2,19E+02	2,16E+02	3,96E+02	**5,89E+02**	5,91E+02	5,58E+02	7,27E+02
	Med,	**1,73E+02**	1,89E+02	2,10E+02	3,96E+02	**4,74E+02**	5,00E+02	5,34E+02	7,27E+02
	Min,	**1,54E+02**	1,79E+02	1,97E+02	3,96E+02	**4,21E+02**	4,79E+02	5,09E+02	7,27E+02
	Mea,	**1,78E+02**	1,92E+02	2,08E+02	3,96E+02	**4,75E+02**	5,10E+02	5,36E+02	7,27E+02
	Std,	**1,49E+01**	9,08E+00	5,79E+00	0,00E+00	**3,17E+01**	2,96E+01	9,55E+00	0,00E+00
	HA_p	-	*7,18E-02*	*1,52E-07*	*6,82E-17*	-	*9,32E-03*	*9,58E-07*	*1,89E-17*

Como primera observación de la Tabla 1, podemos destacar que nuestra estrategia de optimización con PSO obtiene el mejor fitness (minimizando) para todos lo escenarios de tráfico, siendo las diferencias entre las distribuciones de resultados de los algoritmos estadísticamente significativas. Existe una excepción en el escenario de Sevilla con 250 vehículos para el cual, los resultados de DE son estadísticamente similares (*p-valor = 7,18E-02*) a los de PSO, aunque con peor *ranking* para DE. Precisamente, el algoritmo DE obtiene el segundo mejor *ranking*, seguido por RAND y SCPG.

En este sentido, una segunda interesante observación consiste en que todos los algoritmos de búsqueda con componente estocástica obtienen, por lo general, mejor resultado que el propio generador de SUMO (SCPG), de operación determinista. Podemos ver que incluso el algoritmo RAND genera, tras cierto numero de simulaciones, mejores configuraciones de semáforos que SCPG. Estos resultados nos dan cierta confianza respecto a la eficacia de nuestra estrategia de optimización, ya que en la gran mayoría de los experimentos, PSO nos proporciona beneficios cuantitativos respecto al método utilizado actualmente.

4.2. Análisis de los Factores Medioambiantales

En esta sección pasamos a analizar el impacto directo sobre los factores medioambientales estudiados. Para ello utilizaremos los programas de ciclos resultantes tras la experimentación.

Tal y como se contempla en el modelo HBEFA [4], las emisiones de CO_2 y NO_x son los compuestos químicos con más concentración detectados en los tests de tubo de escape en vehículos de gasolina y diesel, respectivamente[3]. Por tanto, para este estudio hemos contabilizado las trazas de estos dos compuestos generados por todos los vehículos circulando en nuestros escenarios. La Figura 3 (derecha) contiene las gráficas de barras de concentraciones generales (g/s) de los mejores programas de ciclos generados por cada algoritmo, para los escenarios de Málaga y Sevilla. Además, se disponen también los valores de consumo de carburantes ($Fuel$ en l/s).

[3] Las emisiones de CO son proporcionales a las de CO_2 en HBEFA.

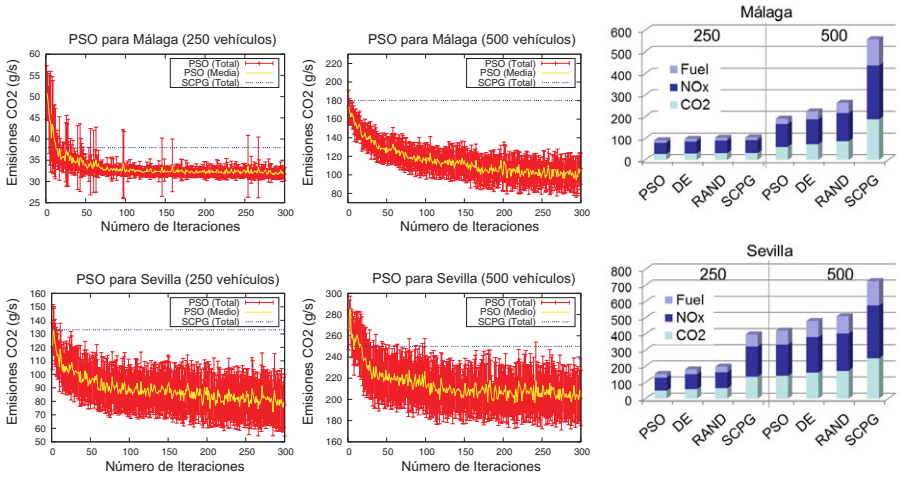


Figura 3. Trazas de emisión de CO_2 (izq.) y coste medioambiental general (der.)

De manera general, se puede observar cómo el gasto medioambiental de los programas de ciclos generados por PSO es inferior al de las demás estrategias. Si bien, es SCPG el que provoca un empeoramiento cualitativo mayor respecto a todos los factores de polución y consumo. Estas diferencias se incrementan para los casos con más densidad de vehículos (500), lo cual refleja que el margen de mejora de PSO podría ser aún mayor en escenarios de gran escala, con miles de vehículos y grandes áreas metropolitanas. En concreto, en el caso de Sevilla con 500 vehículos, la mejora obtenida por PSO para el CO_2 con respecto a SCPG es del 43,2 %; la mejora respecto al NO_x es del 40,9 %; y la mejora respecto al consumo de carburante es del 43,4 %.

Como caso particular, en la Figura 3 (izquierda-centro) se disponen las gráficas de evolución de CO_2 para cada instancia producidas por los programas de ciclos generados mediante PSO. En el caso de Sevilla, se puede observar una mayor variabilidad en los programas de ciclos obtenidos por PSO que para el escenario de Málaga. Esto nos da idea de la diferencia en la complejidad entre los dos escenarios. No obstante, para ambos casos, ya desde las primeras etapas de optimización, nuestra propuesta consigue mejorar los niveles de polución con respecto a la solución de SUMO (representada mediante líneas de puntos).

5. Conclusiones

En este trabajo, proponemos una estrategia de optimización basada en PSO y simulación con SUMO para la generación de programas de ciclos de semáforos. El objetivo es reducir las emisiones de CO_2 y NO_x, así como el consumo de carburantes mediante la mejora del tráfico rodado. Como casos de estudio hemos seguido modelos de tráfico realistas localizados en dos áreas de $0{,}75km^2$ en las ciudades de Málaga y Sevilla, con densidades de tráfico de 250 y 500 vehículos.

Como conclusión global podemos destacar que nuestra propuesta de optimización consigue mejoras significativas en términos de emisiones de polución y de consumo de combustible, respecto a los programas de los expertos en el área (utilizadas en SUMO). Dependiendo de la instancia, la mejora para los tres factores (CO_2, NO_x y $Fuel$) está en el rango entre el 12 % y 43 %. Si extrapolamos estas cantidades al caso real de la ciudad de Málaga, con aproximadamente 600.000 habitantes y 2,15 toneladas CO_2 habitante/año de tráfico rodado[4], el ahorro aproximado rondaría las 510.000 toneladas (40 % menos) de CO_2 que se dejarían de emitir por año en esta ciudad.

Como trabajo futuro, pretendemos extender el estudio a instancias de aún mayor escala (cubriendo ciudades enteras) y utilizando algoritmos con operadores de diseño específico al problema.

Agradecimientos Este trabajo está parcialmente financiado por el Ministerio de Economía y Competitividad (MINECO) del Gobierno de España y fondos FEDER con el proyecto roadME con TIN2011-28194 (`http://roadme.lcc.uma.es`). José García-Nieto disfruta de una beca con código BES-2009-018767 del MINECO.

Referencias

1. E. Alba, G. Luque, J. García-Nieto, G. Ordoñez, and G. Leguizamón. Mallba: a software library to design efficient optimisation algorithms. *International Journal of Innovative Computing and Applications*, 1(1):74–85, 2007.
2. M. Clerc and J. Kennedy. Standard PSO 2011. Particle Swarm Central Site [online] http://www.particleswarm.info/, January 2011.
3. J. García-Nieto, A. Olivera, and E. Alba. Optimal cycle program of traffic lights with particle swarm optimization. *IEEE Trans. on Evo. Comp.*, In Press, 2013.
4. M. Keller. Handbook of Emission Factors for Road Transport (HBEFA) 3.1. quick reference. Technical report, INFRAS, 2010.
5. J. Kennedy and R. C. Eberhart. *Swarm Intelligence*. Morgan Kaufmann Publishers, San Francisco, California, 2001.
6. D. Krajzewicz, M. Bonert, and P. Wagner. The open source traffic simulation package SUMO. *RoboCup 2006 Infrastructure Simulation Competition*, 2006.
7. X. Li, G. Li, S. Pang, X. Yang, and J. Tian. Signal timing of intersections using integrated optimization of traffic quality, emissions and fuel consumption: a note. *Transportation Research Part D: Transport and Environment*, 9(5):401 – 407, 2004.
8. J. McCrea and S. Moutari. A hybrid macroscopic-based model for traffic flow in road networks. *European Journal of Operational Research*, 207(2):676–684, 2010.
9. J. Sánchez Medina, M. Galán Moreno, and E. Rubio Royo. Applying a Traffic Lights Evolutionary Optimization Technique to a Real Case: Las Ramblas Area in Santa Cruz de Tenerife. *IEEE Trans. on Evo. Comp.*, 12(1):25 –40, 2008.
10. L. Peng, M. Wang, J. Du, and G. Luo. Isolation niches particle swarm optimization applied to traffic lights controlling. In *48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, pages 3318 –3322, dec. 2009.
11. K. Price, R.M. Storn, and J.A. Lampinen. *Differential Evolution: A Practical Approach to Global Optimization*. Natural Computing. Springer, 2005.
12. D. J. Sheskin. *Handbook of Parametric and Nonparametric Statistical Procedures*. Chapman & Hall/CRC, 2007.

[4] Fuentes Dictamen para el año 2008, Área de Medio Ambiente Diputación de Málaga

Estudio comparativo de distintos métodos de umbral en regresión ordinal

Pedro Antonio Gutiérrez, María Pérez-Ortiz, Javier Sánchez-Monedero y César Hervás-Martínez*

Universidad de Córdoba, Dept. de Informática y Análisis Numérico
Campus de Rabanales, Edificio Albert Einstein, 14071 - Córdoba, España
Grupo de investigación AYRNA: http://www.uco.es/ayrna
{pagutierrez,i82perom,jsanchezm,chervas}@uco.es

Resumen En este trabajo se presenta un estudio comparativo de una familia de métodos de clasificación en el contexto de regresión ordinal, los conocidos como métodos de umbral. El término regresión ordinal hace referencia a aquellos problemas de clasificación en los que las categorías de la variable discreta a predecir guardan una relación natural de orden. Al contrario que los métodos tradicionales de clasificación nominal que ignoran el orden de las etiquetas, los métodos de umbral están diseñados para aprovecharlo. En concreto, se basan en proyectar los datos a una línea recta mediante un modelo lineal o no lineal y aprender un conjunto de umbrales que dividen dicha línea en las distintas categorías, incluyendo así la información de orden de forma directa. Aunque la versión lineal de los métodos de umbral es bien conocida y utilizada en áreas como ciencias sociales, su uso no es tan popular en el área de aprendizaje automático. Sin embargo, existen algunos clasificadores no lineales basados en estas ideas. Este artículo introduce las características de los métodos existentes y ofrece un estudio comparativo experimental de los mismos.

Keywords: regresión ordinal, métodos de umbral, máquinas de vectores soporte, análisis discriminante

1. Introducción

La tarea de clasificación o de predicción de valores numéricos a partir de patrones previamente etiquetados es uno de los temas centrales de investigación en aprendizaje automático [1]. Sin embargo, existe un área que ha recibido relativamente poca atención, la denominada regresión ordinal o clasificación ordinal, que aborda la tarea de predecir variables discretas formadas por categorías en las que existe un orden natural. Por ejemplo, las encuestas de satisfacción de estudiantes normalmente incluyen la evaluación de los docentes en una escala

* Este trabajo ha sido subvencionado en parte por el proyecto TIN2011-22794 del Ministerio Español, Comisión Ciencia y Tecnología (MICYT), fondos FEDER y el proyecto P2011-TIC-7508 de la "Junta de Andalucía" (España).

del tipo $\{malo, medio, bueno, muy\ bueno, excelente\}$. De esta forma, a estas categorías se les confiere una información de orden, por ejemplo, un profesor de la clase *medio* está mejor valorado (o es mejor) que un profesor de la clase *malo*, pero la clase *bueno* es mejor que ambas. Cuando tratamos este tipo de problemas dos hechos son decisivos: los costes de los errores de clasificación no son los mismos (es claro que confundir un profesor *excelente* con un profesor *malo* debería penalizarse mucho más que confundirlo con uno *muy bueno*) y la información de orden debería (y, como veremos en este artículo, puede) utilizarse para obtener modelos más precisos.

Los problemas de regresión ordinal son muy frecuentes en muchas áreas de investigación y han sido muchas veces tratados como problemas de clasificación nominal lo que puede llevar a soluciones no óptimas. Por citar algunas de estas áreas, medicina [2,3,4], interfaz cerebro-computador [5], valoración de crédito [6,7] o reconocimiento facial [8], son ejemplos típicos donde la variable objetivo puede ser de naturaleza ordinal. Todos los trabajos citados incluyen metodologías que tienen en cuenta el orden de las categorías, demostrando una mejora en el rendimiento respecto a clasificadores nominales pese a que el enfoque tradicional en dichas áreas es ignorar este orden.

El problema de regresión ordinal consiste en predecir la etiqueta y de un vector de entrada $\mathbf{x}$, donde $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^K$ e $y \in \mathcal{Y} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_Q\}$, es decir, $\mathbf{x}$ está en un espacio K-dimensional e y en un espacio de Q etiquetas diferentes que se corresponden con las categorías. El objetivo es encontrar una regla de clasificación o función $f : \mathcal{X} \to \mathcal{Y}$ para predecir las etiquetas de nuevos patrones, dado un conjunto de entrenamiento de N puntos, $D = \{(\mathbf{x}_i, y_i), 1 \leq i \leq N\}$. Hasta el momento, todo es similar a la clasificación nominal, pero un orden natural en las etiquetas aparece en la regresión ordinal, $\mathcal{C}_1 \prec \mathcal{C}_2 \prec \ldots \prec \mathcal{C}_Q$, donde $\prec$ es la relación de orden impuesta por el problema. Además, muchos algoritmos de regresión ordinal hacen uso del rango de la etiqueta ordinal, es decir, la posición de la misma en la escala ordinal, que puede expresarse con la función $\mathcal{O}(\cdot)$, de forma que $\mathcal{O}(\mathcal{C}_q) = q, 1 \leq q \leq Q$.

En Estadística, desde hace varias décadas, las bases de datos de regresión ordinal han sido tradicionalmente tratadas utilizando una función de enlace sobre un modelo lineal, de manera que dicha función estima las probabilidades de pertenencia a cada clase respetando las restricciones de orden [9]. Este tipo de modelos pertenecen al grupo conocido como *métodos de umbral*, en el sentido que aprenden una proyección lineal o no lineal que sitúa todos los patrones en una línea recta (lo que puede verse como una aproximación de la variable latente subyacente en la variable categórica tratada) y un vector de umbrales que estiman el rango de variable latente que corresponde a cada categoría. En la última década, han empezado a surgir métodos de aprendizaje automático, que, con una filosofía similar, adaptan mecanismos de clasificación nominal bien conocidos para abordar bases de datos de regresión ordinal, como por ejemplo, las Máquinas de Vectores Soporte (SVM) [10,11], los Procesos Gaussianos [12] o el Análisis Discriminante [13], demostrando, en general, un mejor rendimiento que los métodos nominales tradicionales.

El objetivo del presente trabajo es ofrecer un resumen de los principales *métodos de umbral* existentes en la literatura y realizar una comparativa experimental sobre un conjunto de bases de datos de prueba. El resto del trabajo se organiza como sigue: en la sección 2 se presenta un resumen de los métodos que serán comparados en la sección 3. Las conclusiones se incluyen en la sección 4.

2. Métodos de umbral

A menudo, en el contexto de regresión ordinal, se asume que existe una variable latente que origina las etiquetas con las que nos encontramos (por ejemplo, si las etiquetas son *bebé, niño, adulto, anciano*, la variable latente que generó esas etiquetas es la edad de la persona). Los métodos que aprovechan dicha hipótesis son denominados métodos de umbral y se pueden considerar los más populares entre los existentes [14]. Estos métodos estiman dos cosas:

- Una función $f(\mathbf{x})$ que intenta predecir la naturaleza de estos valores reales subyacentes.
- Un conjunto de umbrales $\mathbf{b} = (b_1, b_2, \ldots, b_{Q-1}) \in \mathbb{R}^{Q-1}$ que representan intervalos en el rango de $f(\mathbf{x})$ y que deben satisfacer la restricción $b_1 \leq b_2 \leq \ldots \leq b_{Q-1}$. Estos umbrales delimitan cada una de las clases.

Se pueden considerar dos familias de métodos de umbral, a) los modelos discriminativos que estiman directamente la probabilidad a posteriori $P(y|\mathbf{x})$ o aprenden una función que transforme $\mathbf{x}$ a etiquetas de clase; y b) los modelos generativos que aprenden un modelo de la probabilidad conjunta $P(\mathbf{x}, y)$ de los patrones con características de entrada $\mathbf{x}$ y la etiqueta y, para luego utilizar un procedimiento Bayesiano que realice la predicción en base la probabilidad estimada $P(y|\mathbf{x})$.

2.1. Métodos discriminativos

En este contexto se pueden distinguir diferentes propuestas:

- Modelos de enlace acumulado (*Cumulative link models*, CLMs): estos modelos provienen de un contexto puramente estadístico, siendo el más conocido el modelo de posibilidades proporcionales (*Proportional Odds Model*, POM) propuesto por McCullagh en 1980 [15]. Los CLMs [16] son una familia de modelos en la que se sitúa el POM y que extienden la regresión logística binaria para predecir categorías en una escala ordinal. De esta forma, se basan en estimar probabilidades acumuladas de pertenencia a varias categorías $P(y \preceq \mathcal{C}_j|\mathbf{x})$ que se pueden relacionar de forma directa con las probabilidades de pertenencia a una sola categoría:

$$P(y \preceq \mathcal{C}_q|\mathbf{x}) = P(y = \mathcal{C}_1|\mathbf{x}) + \ldots + P(y = \mathcal{C}_q|\mathbf{x}),$$
$$P(y = \mathcal{C}_q|\mathbf{x}) = P(y \preceq \mathcal{C}_q|\mathbf{x}) - P(y \preceq \mathcal{C}_{q-1}|\mathbf{x}),$$

  con $1 \leq q \leq Q$ y considerando por definición que $P(y \preceq \mathcal{C}_Q|\mathbf{x}) = 1$. Estos modelos tienen la siguiente formulación general:

$$g^{-1}\left(P(y \preceq \mathcal{C}_q|\mathbf{x})\right) = b_q - \mathbf{w}^T\mathbf{x}, 1 \leq q \leq Q,$$

donde $g^{-1} : [0, 1] \to (-\infty, +\infty)$ es una función monótona a menudo conocida como la función de enlace inversa y b_q es el umbral para la clase $\mathcal{C}_q$. Este método de umbral considera por tanto un modelo lineal para la función $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$. De esta forma, se predice la etiqueta $\mathcal{C}_q$ si y solo si $f(\mathbf{x}) \in [b_{q-1}, b_q]$, donde la función f y $\mathbf{b} = (b_0, b_1, ..., b_{Q-1}, b_Q)$ deben determinarse a partir de los datos de entrenamiento, asumiendo que $b_0 = -\infty$ y $b_Q = +\infty$. La función inversa de enlace más comúnmente escogida es la función `logit` (inversa de la función logística) aunque existen otras posibilidades como la `probit`, `log-log` complementaria, `log-log` negativa o la `cauchit` [16]. El principal problema con este tipo de modelos (que quedará reflejado en la sección experimental) es que, al ser lineales, son muy poco flexibles y pueden producir resultados peores en tareas de clasificación.

– Máquinas de Vectores Soporte (*Support Vector Machines*, SVMs): por sus buenos resultados en generalización, los modelos SVM son bastante populares en el contexto de regresión ordinal. Es bien conocido que las SVM son formuladas de forma natural para clasificación binaria y que su extensión a problemas multiclase nominales se basa en descomposiciones de tipo *OneVsOne* o *OneVsAll*, cuya validación a nivel teórico puede ser problemática. Sin embargo, la extensión de las SVM a problemas de regresión ordinal es directa y natural, de manera que el problema de optimización de las SVM puede reformularse como la búsqueda de $Q - 1$ hiperplanos paralelos, todos ellos representados por un vector de proyección común $\mathbf{w}$ y los umbrales escalares $b_1 \leq \ldots \leq b_{Q-1}$, de esta forma separando los datos en clases ordenadas. En este sentido, Chu y Keerthi [17,11] extendieron un trabajo anterior para proponer una formulación que permitiese optimizar $\mathbf{w}$ y todos los umbrales, asegurando la restricción $b_1 \leq \ldots \leq b_{Q-1}$. Lo hicieron con dos enfoques distintos, introducir de forma explícita dicha restricción en la formulación (*Support Vector Ordinal Regression with Explicit Constraints*, SVOREX) y considerar que todas las clases contribuían al error de cada punto, lo que demostraron resultaba en un satisfacción automática de las restricciones para la solución óptima (*SVOR with Implicit Constraints*, SVORIM).
– Análisis discriminante: esta metodología sobradamente conocida en el entorno del aprendizaje automático también ha sido reformulada para regresión ordinal [13]. Para ello, se introduce una restricción que impone una separación mínima en las proyecciones más cercanas de dos puntos pertenecientes a clases adyacentes en la escala ordinal. Esta restricción preserva la información de orden en las categorías. El método se conoce como *Kernel Discriminant Learning for Ordinal Regression*, (KDLOR) [13,4].

2.2. Métodos generativos

El único método que hemos considerado en este grupo es el de Procesos Gaussianos para Regresión Ordinal (*Gaussian Processes for Ordinal Regression*, GPOR) [12] que modela la variable latente $f(\mathbf{x})$ utilizando Procesos Gaussianos y luego estima todos los parámetros mediante un procedimiento Bayesiano. Los valores la función latente $\{f(\mathbf{x}_i)\}$ se asume que son dados por variables aleatorias

indexadas por el patrón de entrada en un Proceso Gaussiano de media cero. Se utilizan funciones *kernel* para aproximar la covarianza entre las funciones que se corresponden a dos puntos.

3. Experimentación

El estudio experimental comparativo pretende ofrecer una idea del rendimiento de los distintos métodos sobre un conjunto de bases de datos de prueba.

3.1. Bases de datos utilizadas

En regresión ordinal, el repositorio de bases de datos más utilizado es el proporcionado por Chu et al. [12] que incluye distintas bases de datos de regresión. Estas bases de datos no son realmente bases de datos de regresión ordinal, sino bases de datos de regresión donde la variable objetivo se ha discretizado en un conjunto de categorías. Dado que estos conjuntos de datos omiten parte de los problemas a los que nos enfrentamos cuando las bases de datos son realmente bases de datos de clasificación ordinal (desbalanceo de clases, ausencia de una variable latente explícita...), hemos utilizado en su lugar una serie de bases de datos de los repositorios UCI [18] y `mldata.org` [19]. En total disponemos de 17 bases de datos para nuestro estudio, cuyas características podemos ver resumidas en el Cuadro 1 (en esta tabla, se pueden reconocer muchas bases de datos que otros trabajos han considerado sistemáticamente como nominales, ignorando así la información de orden).

3.2. Algoritmos seleccionados

Hemos considerado los algoritmos que se han analizado en la sección anterior:

- El modelo lineal POM [20], utilizando la función de enlace `logit`.
- Las formulaciones de SVM para regresión ordinal [11], tanto la de restricciones explícitas (SVOREX) como la de restricciones implícitas (SVORIM).
- El algoritmo KDLOR propuesto en [13].
- El método GPOR [12] incluyendo un mecanismo de determinación automática de la relevancia de cada variable, tal y como proponen los autores.

Además, hemos considerado el algoritmo de clasificación nominal SVM (*C-Support Vector Classifier*, *C*-SVC), con la descomposición *OneVsOne*.

3.3. Medidas de evaluación consideradas y optimización de los parámetros

Las medidas más comunes en regresión ordinal son el error absoluto medio (*Mean Absolute Error*, MAE) y el error cero-uno medio (MZE, *Mean Zero-one Error*) [12,11,13]. El MZE es el ratio de error del clasificador, sin tener en

Cuadro 1. Características de los conjuntos de datos utilizados

BDD.	nº Pat.	nº Attr.	nº Classes	Distribución
contact-lenses (CL)	24	6	3	(15, 5, 4)
pasture (PA)	36	25	3	(12, 12, 12)
squash-stored (SS)	52	51	3	(23, 21, 8)
squash-unstored (SU)	52	52	3	(24, 24, 4)
tae (TA)	151	54	3	(49, 50, 52)
newthyroid (NT)	215	5	3	(30, 150, 35)
balance-scale (BS)	625	4	3	(288, 49, 288)
SWD (SW)	1000	10	4	(32, 352, 399, 217)
car (CA)	1728	21	4	(1210, 384, 69, 65)
bondrate (BO)	57	37	5	(6, 33, 12, 5, 1)
toy (TO)	300	2	5	(35, 87, 79, 68, 31)
eucalyptus (EU)	736	91	5	(180, 107, 130, 214, 105)
LEV (LE)	1000	4	5	(93, 280, 403, 197, 27)
automobile (AU)	205	71	6	(3, 22, 67, 54, 32, 27)
winequality-red (WR)	1599	11	6	(10, 53, 681, 638, 199, 18)
ESL (ES)	488	4	9	(2, 12, 38, 100, 116, 135, 62, 19, 4)
ERA (ER)	1000	4	9	(92, 142, 181, 172, 158, 118, 88, 31, 18)

cuenta la ordinalidad, es decir:

$$MZE = \frac{1}{N}\sum_{i=1}^{N}[\![y_i^* \neq y_i]\!] = 1 - Acc,$$

donde y_i es la etiqueta real, y_i^* es la etiqueta predicha y Acc es la precisión del clasificador. Esta medida toma valores en el intervalo $[0, 1]$. En contraposición, el MAE si que tiene en cuenta la ordinalidad de la variable objetivo:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\mathcal{O}(y_i) - \mathcal{O}(y_i^*)|,$$

donde $|\mathcal{O}(y_i) - \mathcal{O}(y_i^*)|$ es la distancia absoluta entre las etiquetas reales y predichas. El MAE varia desde 0 hasta $Q - 1$ (que sería la máxima desviación en número de categorías).

Para evaluar los distintos modelos, se consideraron 30 particiones aleatorias de las bases de datos, con un 75 % de los patrones en el conjunto de entrenamiento y un 25 % en el conjunto de generalización (tal y como se sugiere en [21]). Las particiones fueron las mismas para todos los algoritmos y se hicieron de forma estratificada, respetando las proporciones de cada clase en el conjunto original para cada uno de los subconjuntos.

Es muy importante realizar una correcta optimización de todos los parámetros que intervienen en los distintos algoritmos. Cuando el número de parámetros

no es alto y los algoritmos no son costosos, es aconsejable realizar una selección basada en una validación interna sobre el conjunto de entrenamiento. Esto es, los datos de entrenamiento son utilizados para realizar un proceso de validación tipo *k-fold*, obteniendo una estimación del error para cada combinación de parámetros. Posteriormente, se seleccionan los valores de los parámetros que den como resultado el menor error de validación en este proceso *k-fold* y se reentrena el modelo utilizando el conjunto de entrenamiento completo y la combinación seleccionada. Es necesario decidir, aún así, un conjunto de valores posibles para cada parámetro, dando lugar a un *grid* de combinación (de tantas dimensiones, como número de parámetros). En nuestro caso, los valores seleccionados son:

- El *kernel* Gaussiano se utilizó para los métodos no lineales SVC1V1, SVOREX, SVORIM y KDLOR. Los valores considerados para el ancho del *kernel* son $\sigma \in \{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$.
- El parámetro de coste C tomó sus valores en $C \in \{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$ y $C \in \{10^{-1}, 10^{0}, 10^{1}\}$, para los métodos SVM y KDLOR, respectivamente.
- El algoritmo KDLOR requiere optimizar un parámetro adicional u que se suma a la diagonal principal de las matrices de varianzas-covarianzas con el objetivo de evitar la singularidad de las mismas. Los valores considerados fueron $u \in \{10^{-6}, 10^{-3}, \ldots, 10^{-2}\}$.

En cuanto al algoritmo GPOR, su formulación permite que los valores de sus parámetros se optimicen automáticamente durante el proceso de aprendizaje.

3.4. Resultados

Los resultados obtenidos en los conjuntos de test, se muestran en el Cuadro 2 y en el Cuadro 3, mediante los estadísticos media y desviación típica. Además, el Cuadro 4 incluye los tiempos totales para realizar en proceso de aprendizaje completo (selección de los parámetros mediante validación, entrenamiento y generalización). Estas tablas incluyen el promedio del rendimiento y la desviación típica para todas las bases de datos, como resumen de su comportamiento. Del análisis de estos resultados se pueden extraer las siguientes conclusiones:

- El rendimiento del algoritmo SVC1v1 en cuanto a la medida MZE es muy bueno. Sin embargo, la descomposición realizada por SVC1v1 no tiene en cuenta el orden de las clases y eso hace que su rendimiento respecto a la medida MAE sea peor en comparación a SVOREX o SVORIM. Esta medida debería ser la más importante cuando consideramos problemas de regresión ordinal, ya que tiene en cuenta la magnitud de los errores cometidos.
- El rendimiento del algoritmo GPOR es bastante bajo tanto en MZE como en MAE. Los algoritmos de tipo SVM son los que obtienen un mejor balance entre ambas medidas, y en especial, el algoritmo SVOREX. Como se puede observar, los métodos discriminativos (SVOREX y SVORIM) mejoran a los generativos (GPOR). El menor coste computacional de los métodos de umbral se asocia al método POM, seguido por el SVOREX, pero, sin embargo, obtiene rendimiento bajo debido a su carácter lineal.

Cuadro 2. Resultados de MZE en el conjunto de generalización.

BDD.	SVC1V1	POM	SVOREX	SVORIM	KDLOR	GPOR
CL	,**3222**$_{,\mathbf{1233}}$	,3833$_{,1704}$	,*3500*$_{,1103}$	,3611$_{,0987}$	,3833$_{,1759}$	,3944$_{,0927}$
PA	,**2963**$_{,\mathbf{1439}}$	,5037$_{,1537}$	,3333$_{,1167}$	,*3148*$_{,1170}$	,3370$_{,1254}$	,4778$_{,1778}$
SS	,3538$_{,1205}$	,6179$_{,1518}$	,3564$_{,1421}$	,**3487**$_{,\mathbf{1396}}$	,*3513*$_{,1190}$	,5487$_{,1005}$
SU	,**2077**$_{,\mathbf{1248}}$	,6513$_{,1425}$	,2615$_{,1188}$	,2641$_{,1207}$	,*2487*$_{,1273}$	,3564$_{,1622}$
TA	,4307$_{,0624}$	,4956$_{,0773}$	,**4211**$_{,\mathbf{0582}}$	,*4237*$_{,0571}$	,4237$_{,0576}$	,6719$_{,0407}$
NT	,0352$_{,0254}$	,*0278*$_{,0222}$	,0315$_{,0239}$	,0290$_{,0221}$	,**0247**$_{,\mathbf{0202}}$	,0340$_{,0244}$
BS	,0295$_{,0145}$	,0945$_{,0186}$	,*0021*$_{,0056}$	,**0013**$_{,\mathbf{0042}}$	,1631$_{,0240}$	,0342$_{,0118}$
SW	,**4169**$_{,\mathbf{0315}}$	,4321$_{,0296}$	,4291$_{,0314}$	,*4291*$_{,0301}$	,5092$_{,0293}$	,4223$_{,0305}$
CA	,**0057**$_{,\mathbf{0048}}$	,8425$_{,3063}$	,*0126*$_{,0053}$	,0127$_{,0048}$	,0465$_{,0083}$	,0371$_{,0095}$
BO	,*4467*$_{,0530}$	,6556$_{,1605}$	,4822$_{,0904}$	,4644$_{,0773}$	,4689$_{,0830}$	,**4222**$_{,\mathbf{0320}}$
TO	,0516$_{,0247}$	,7107$_{,0255}$	,**0173**$_{,\mathbf{0127}}$	,*0227*$_{,0141}$	,1142$_{,0305}$	,0462$_{,0223}$
EU	,*3585*$_{,0269}$	,8507$_{,0157}$	,3620$_{,0281}$	,3600$_{,0303}$	,3667$_{,0287}$	,**3145**$_{,\mathbf{0341}}$
LE	,**3675**$_{,\mathbf{0243}}$	,3767$_{,0280}$	,3745$_{,0197}$	,*3741*$_{,0210}$	,4507$_{,0346}$	,3877$_{,0301}$
AU	,**2468**$_{,\mathbf{0520}}$	,5333$_{,1942}$	,3410$_{,0751}$	,3282$_{,0714}$	,*3083*$_{,0622}$	,3891$_{,0726}$
WR	,*3683*$_{,0223}$	,4028$_{,0154}$	,3779$_{,0201}$	,3817$_{,0199}$	,**3496**$_{,\mathbf{0200}}$	,3942$_{,0148}$
ES	,3036$_{,0320}$	,2945$_{,0336}$	,2877$_{,0336}$	,**2836**$_{,\mathbf{0339}}$	,3462$_{,0368}$	,*2869*$_{,0307}$
ER	,7344$_{,0231}$	,7439$_{,0211}$	,*7140*$_{,0262}$	,7592$_{,0211}$	,7949$_{,0346}$	,**7121**$_{,\mathbf{0270}}$
Media	,**2927**$_{,\mathbf{1869}}$	,5069$_{,2343}$	,*3032*$_{,1912}$	,3034$_{,1963}$	,3345$_{,1856}$	,3488$_{,2096}$

El mejor resultado está en negrita y el segundo mejor en cursiva

4. Conclusiones

Este trabajo ha realizado un análisis de los principales métodos de umbral que podemos encontrar en la literatura de regresión ordinal. Se ha incluido una comparativa experimental de gran parte de dichos métodos utilizando repositorios habituales de bases de datos. El mejor balance entre buena clasificación y baja desviación respecto a la etiqueta real es obtenido por SVOREX. El modelo lineal POM muestra un rendimiento bajo, pero su coste computacional es reducido. Así, este método puede considerarse para bases de datos grandes y cuando se requiera interpretabilidad para los modelos obtenidos.

Referencias

1. Cherkassky, V., Mulier, F.M.: Learning from Data: Concepts, Theory, and Methods. Wiley-Interscience (2007)
2. Bender, R., Grouven, U.: Using binary logistic regression models for ordinal data with non-proportional odds. Journal of clinical epidemiology **51**(10) (1998) 809–816
3. Fan, Y.: Ordinal ranking for detecting mild cognitive impairment and alzheimer's disease based on multimodal neuroimages and CSF biomarkers. In Liu, T., Shen, D., Ibanez, L., Tao, X., eds.: Proceedings of the First International Workshop on Multimodal Brain Image Analysis (MBIA2011). Volume 7012 of Lecture Notes in Computer Science., Springer Berlin Heidelberg (2011) 44–51

Cuadro 3. Resultados de MAE en el conjunto de generalización.

BDD.	SVC1V1	POM	SVOREX	SVORIM	KDLOR	GPOR
CL	$0{,}522_{0{,}218}$	$0{,}533_{0{,}253}$	$\mathit{0{,}483}_{\mathit{0{,}134}}$	$\mathbf{0{,}467_{0{,}148}}$	$0{,}517_{0{,}225}$	$0{,}511_{0{,}175}$
PA	$\mathbf{0{,}296_{0{,}144}}$	$0{,}585_{0{,}204}$	$0{,}333_{0{,}117}$	$\mathit{0{,}315}_{\mathit{0{,}117}}$	$0{,}341_{0{,}130}$	$0{,}489_{0{,}190}$
SS	$0{,}377_{0{,}138}$	$0{,}813_{0{,}248}$	$\mathit{0{,}367}_{\mathit{0{,}147}}$	$\mathbf{0{,}354_{0{,}142}}$	$0{,}374_{0{,}145}$	$0{,}626_{0{,}148}$
SU	$\mathbf{0{,}215_{0{,}135}}$	$0{,}826_{0{,}230}$	$0{,}262_{0{,}119}$	$0{,}264_{0{,}121}$	$\mathit{0{,}251}_{\mathit{0{,}132}}$	$0{,}356_{0{,}162}$
TA	$0{,}543_{0{,}104}$	$0{,}628_{0{,}116}$	$0{,}466_{0{,}061}$	$\mathit{0{,}461}_{\mathit{0{,}060}}$	$\mathbf{0{,}461_{0{,}065}}$	$0{,}861_{0{,}155}$
NT	$0{,}035_{0{,}025}$	$\mathit{0{,}028}_{\mathit{0{,}022}}$	$0{,}031_{0{,}024}$	$0{,}029_{0{,}022}$	$\mathbf{0{,}025_{0{,}020}}$	$0{,}034_{0{,}024}$
BS	$0{,}030_{0{,}015}$	$0{,}107_{0{,}021}$	$\mathit{0{,}002}_{\mathit{0{,}006}}$	$\mathbf{0{,}001_{0{,}004}}$	$0{,}164_{0{,}025}$	$0{,}034_{0{,}012}$
SW	$\mathbf{0{,}440_{0{,}036}}$	$0{,}450_{0{,}030}$	$0{,}446_{0{,}031}$	$0{,}445_{0{,}032}$	$0{,}579_{0{,}035}$	$\mathit{0{,}440}_{\mathit{0{,}032}}$
CA	$\mathbf{0{,}007}_{\mathit{0{,}006}}$	$1{,}451_{0{,}548}$	$\mathit{0{,}013}_{\mathit{0{,}005}}$	$\mathit{0{,}013}_{\mathit{0{,}005}}$	$0{,}046_{0{,}008}$	$0{,}038_{0{,}010}$
BO	$0{,}640_{0{,}097}$	$0{,}947_{0{,}321}$	$\mathit{0{,}622}_{\mathit{0{,}092}}$	$\mathbf{0{,}613_{0{,}081}}$	$0{,}629_{0{,}082}$	$0{,}624_{0{,}062}$
TO	$0{,}052_{0{,}025}$	$0{,}981_{0{,}039}$	$\mathbf{0{,}017_{0{,}013}}$	$\mathit{0{,}023}_{\mathit{0{,}014}}$	$0{,}114_{0{,}030}$	$0{,}046_{0{,}022}$
EU	$0{,}410_{0{,}036}$	$1{,}939_{0{,}254}$	$0{,}403_{0{,}036}$	$\mathit{0{,}394}_{\mathit{0{,}036}}$	$0{,}401_{0{,}032}$	$\mathbf{0{,}331_{0{,}038}}$
LE	$\mathbf{0{,}402_{0{,}028}}$	$\mathit{0{,}409}_{\mathit{0{,}030}}$	$0{,}411_{0{,}021}$	$0{,}411_{0{,}024}$	$0{,}512_{0{,}039}$	$0{,}422_{0{,}031}$
AU	$\mathbf{0{,}362_{0{,}098}}$	$0{,}953_{0{,}687}$	$0{,}416_{0{,}091}$	$0{,}400_{0{,}093}$	$\mathit{0{,}388}_{\mathit{0{,}082}}$	$0{,}594_{0{,}131}$
WR	$\mathit{0{,}411}_{\mathit{0{,}023}}$	$0{,}435_{0{,}017}$	$0{,}417_{0{,}019}$	$0{,}419_{0{,}020}$	$\mathbf{0{,}393_{0{,}021}}$	$0{,}425_{0{,}017}$
ES	$0{,}321_{0{,}038}$	$0{,}310_{0{,}038}$	$0{,}302_{0{,}035}$	$\mathbf{0{,}300_{0{,}040}}$	$0{,}365_{0{,}043}$	$\mathit{0{,}301}_{\mathit{0{,}035}}$
ER	$1{,}286_{0{,}069}$	$\mathit{1{,}218}_{\mathit{0{,}050}}$	$\mathbf{1{,}209_{0{,}057}}$	$1{,}222_{0{,}044}$	$1{,}783_{0{,}099}$	$1{,}241_{0{,}051}$
Media	$0{,}373_{0{,}303}$	$0{,}742_{0{,}487}$	$\mathit{0{,}365}_{\mathit{0{,}287}}$	$\mathbf{0{,}361_{0{,}289}}$	$0{,}432_{0{,}391}$	$0{,}434_{0{,}317}$

El mejor resultado está en negrita y el segundo mejor en cursiva

4. Pérez-Ortiz, M., Gutiérrez, P.A., García-Alonso, C., Salvador-Carulla, L., Salinas-Pérez, J.A., Hervás-Martínez, C.: Ordinal classification of depression spatial hotspots of prevalence. In: Proceedings of the 11th International Conference on Intelligent Systems Design and Applications (ISDA). (nov. 2011) 1170 –1175
5. Yoon, J.W., Roberts, S.J., Dyson, M., Gan, J.Q.: Bayesian inference for an adaptive ordered probit model: An application to brain computer interfacing. Neural Networks **24**(7) (2011) 726–734
6. Kim, K.j., Ahn, H.: A corporate credit rating model using multi-class support vector machines with an ordinal pairwise partitioning approach. Computers and Operations Research **39**(8) (2012) 1800–1811
7. Dikkers, H., Rothkrantz, L.: Support vector machines in ordinal classification: An application to corporate credit scoring. Neural Network World **15**(6) (2005) 491–507
8. Rudovic, O., Pavlovic, V., Pantic, M.: Multi-output laplacian dynamic ordinal regression for facial expression recognition and intensity estimation. In: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2012) 2634–2641
9. Anderson, J.A.: Regression and ordered categorical variables. Journal of the Royal Statistical Society. Series B (Methodological) **46**(1) (1984) 1–30
10. Herbrich, R., Graepel, T., Obermayer, K.: Large margin rank boundaries for ordinal regression. In Smola, A., Bartlett, P., Schölkopf, B., Schuurmans, D., eds.: Advances in Large Margin Classifiers, Cambridge, MA, MIT Press (2000) 115–132
11. Chu, W., Keerthi, S.S.: Support Vector Ordinal Regression. Neural Computation **19**(3) (2007) 792–815

Cuadro 4. Tiempo total en segundos necesario para el proceso completo: validación de los parámetros, entrenamiento y generalización.

BDD.	SVC1V1	POM	SVOREX	SVORIM	KDLOR	GPOR
CL	$11{,}8_{0{,}7}$	$\mathit{0{,}3}_{0{,}0}$	$0{,}5_{0{,}0}$	$10{,}5_{0{,}4}$	$9{,}3_{1{,}6}$	$\mathbf{0{,}3_{0{,}4}}$
PA	$15{,}3_{2{,}4}$	$\mathbf{0{,}4_{0{,}0}}$	$\mathit{0{,}5}_{0{,}0}$	$10{,}4_{0{,}4}$	$9{,}3_{4{,}2}$	$5{,}9_{3{,}6}$
SS	$14{,}0_{1{,}8}$	$\mathit{0{,}6}_{0{,}0}$	$\mathbf{0{,}6_{0{,}0}}$	$10{,}5_{0{,}4}$	$5{,}7_{0{,}4}$	$16{,}0_{15{,}5}$
SU	$13{,}1_{1{,}2}$	$\mathbf{0{,}6_{0{,}0}}$	$\mathbf{0{,}6_{0{,}0}}$	$10{,}4_{0{,}3}$	$\mathit{6{,}1}_{3{,}0}$	$17{,}8_{14{,}6}$
TA	$13{,}8_{0{,}7}$	$\mathbf{1{,}3_{0{,}1}}$	$\mathit{3{,}8}_{0{,}2}$	$25{,}4_{8{,}7}$	$70{,}4_{2{,}8}$	$153{,}2_{117{,}8}$
NT	$16{,}8_{2{,}4}$	$\mathbf{0{,}4_{0{,}1}}$	$\mathit{1{,}4}_{0{,}0}$	$11{,}3_{0{,}3}$	$73{,}6_{9{,}3}$	$152{,}2_{68{,}2}$
BS	$\mathit{19{,}2}_{0{,}3}$	$\mathbf{0{,}5_{0{,}2}}$	$39{,}3_{0{,}5}$	$76{,}4_{11{,}0}$	$383{,}1_{21{,}0}$	$1983{,}6_{1265{,}9}$
SW	$\mathit{35{,}0}_{2{,}3}$	$\mathbf{0{,}5_{0{,}0}}$	$778{,}3_{40{,}5}$	$245{,}5_{45{,}5}$	$726{,}3_{85{,}9}$	$11586{,}5_{3691{,}7}$
CA	$\mathit{101{,}8}_{1{,}0}$	$\mathbf{3{,}0_{2{,}2}}$	$1732{,}2_{69{,}3}$	$158{,}1_{3{,}4}$	$2436{,}8_{56{,}9}$	$332220{,}5_{763433{,}0}$
BO	$12{,}5_{0{,}4}$	$\mathbf{0{,}5_{0{,}1}}$	$\mathit{0{,}7}_{0{,}0}$	$10{,}6_{0{,}4}$	$9{,}2_{3{,}2}$	$16{,}6_{13{,}6}$
TO	$14{,}9_{1{,}9}$	$\mathbf{0{,}3_{0{,}0}}$	$\mathit{8{,}0}_{0{,}2}$	$23{,}3_{1{,}0}$	$109{,}5_{42{,}5}$	$151{,}8_{113{,}6}$
EU	$\mathit{59{,}0}_{0{,}4}$	$\mathbf{9{,}0_{9{,}7}}$	$123{,}5_{2{,}7}$	$139{,}8_{5{,}2}$	$912{,}5_{104{,}5}$	$56038{,}6_{45377{,}7}$
LE	$\mathit{32{,}3}_{0{,}9}$	$\mathbf{0{,}8_{0{,}2}}$	$795{,}3_{38{,}1}$	$586{,}2_{53{,}6}$	$762{,}3_{90{,}6}$	$4965{,}5_{1772{,}8}$
AU	$16{,}1_{0{,}3}$	$\mathit{7{,}6}_{3{,}8}$	$\mathbf{6{,}8_{0{,}3}}$	$31{,}6_{5{,}7}$	$70{,}8_{21{,}7}$	$2127{,}6_{1717{,}2}$
WR	$\mathit{101{,}9}_{2{,}9}$	$\mathbf{0{,}8_{0{,}0}}$	$1889{,}0_{78{,}9}$	$1062{,}2_{69{,}7}$	$3072{,}3_{130{,}0}$	$48272{,}9_{14360{,}1}$
ES	$\mathit{17{,}4}_{0{,}4}$	$\mathbf{0{,}5_{0{,}1}}$	$115{,}0_{14{,}8}$	$125{,}2_{49{,}7}$	$241{,}0_{15{,}6}$	$586{,}2_{318{,}6}$
ER	$\mathit{37{,}7}_{0{,}6}$	$\mathbf{0{,}5_{0{,}0}}$	$1477{,}0_{142{,}9}$	$3102{,}3_{140{,}6}$	$811{,}2_{41{,}5}$	$6217{,}1_{1077{,}6}$
Media	$\mathit{31{,}3}_{29{,}3}$	$\mathbf{1{,}6_{2{,}6}}$	$410{,}2_{668{,}9}$	$331{,}7_{764{,}6}$	$571{,}1_{889{,}7}$	$27324{,}2_{80372{,}2}$

El mejor resultado está en negrita y el segundo mejor en cursiva

12. Chu, W., Ghahramani, Z.: Gaussian processes for ordinal regression. J. of Machine Learning Research **6** (2005) 1019–1041
13. Sun, B.Y., Li, J., Wu, D.D., Zhang, X.M., Li, W.B.: Kernel discriminant learning for ordinal regression. **22**(6) (2010) 906–910
14. Verwaeren, J., Waegeman, W., De Baets, B.: Learning partial ordinal class memberships with kernel-based proportional odds models. Computational Statistics & Data Analysis **56**(4) (2012) 928–942
15. McCullagh, P.: Regression models for ordinal data. Journal of the Royal Statistical Society. Series B (Methodological) **42**(2) (1980) 109–142
16. Agresti, A.: Categorical Data Analysis. 2 edn. John Wiley and Sons (2002)
17. Chu, W., Keerthi, S.S.: New approaches to support vector ordinal regression. In: In ICML'05: Proceedings of the 22nd international conference on Machine Learning. (2005) 145–152
18. Asuncion, A., Newman, D.: UCI machine learning repository (2007)
19. PASCAL: Pascal (Pattern Analysis, Statistical Modelling and Computational Learning) machine learning benchmarks repository (2011)
20. McCullagh, P., Nelder, J.A.: Generalized Linear Models. 2nd edn. Monographs on Statistics and Applied Probability. Chapman & Hall/CRC (1989)
21. Prechelt, L.: PROBEN1: A set of neural network benchmark problems and benchmarking rules. Technical Report 21/94, Fakultät für Informatik (Universität Karlsruhe) (1994)

MOPNAR: Algoritmo Evolutivo Multi-Objetivo para Extraer Reglas de Asociación Cuantitativas Positivas y Negativas

D. Martín[1] *, A. Rosete[1], J. Alcalá-Fdez[2], and F. Herrera[2]

[1] Dpto. de Inteligencia Artificial e Infraestructura de Sistemas,
Instituto Superior Politécnico J.A Echeverría, Cujae, 19390 La Habana, Cuba
{dmartin,rosete}@ceis.cujae.edu.cu
[2] Dpto. de Ciencias de la Computación e Inteligencia Artificial,
CITIC-UGR, Universidad de Granada, 18071 Granada, España
{jalcala,herrera}@decsai.ugr.es

Resumen Los métodos de extracción de reglas de asociación cuantitativas consideran normalmente un único objetivo, sin embargo, recientemente algunos autores han abordado este proceso como un problema multi-objetivo para optimizar varias medidas a la vez. Además, muchos de estos métodos se centran solo en extraer reglas positivas sin poner atención a reglas negativas que pueden ser interesantes. En este trabajo nosotros proponemos MOPNAR, un nuevo algoritmo evolutivo multi-objetivo para extraer reglas de asociación cuantitativas positivas y negativas. Esta propuesta extiende el modelo MOEA/D-DE para realizar un aprendizaje evolutivo de los intervalos de los atributos y una selección de condiciones para cada regla, maximizando tres objetivos: rendimiento, interés y comprensibilidad. Además, MOPNAR introduce un proceso de reinicialización y una población externa para almacenar todas las reglas no dominadas, provocar diversidad y mejorar el cubrimiento de la base de datos. Los resultados obtenidos sobre varias bases de datos reales muestran su efectividad.

Keywords: Minería de Datos, Reglas de Asociación Cuantitativas, Reglas de Asociación Negativas, Algoritmo Evolutivo Multi-Objetivo

1. Introducción

Las reglas de asociación han sido una de las técnicas de minería de datos (MD) [11] más utilizada para extraer conocimiento interesante a partir de bases de datos (BD). Estas reglas representan e identifican dependencias entre los atributos de una BD. Son expresiones del tipo $X \rightarrow Y$, donde X y Y son conjuntos de ítems (parejas atributo-valor) y cumplen que $X \cap Y = \oslash$. Esto significa que

* Este trabajo está soportado en parte por el proyecto nacional TIN2011-28488, por el programa CEI del MICINN bajo el proyecto PYR-2012-13 CEI BioTIC GENIL (CEB09-0010), y por el proyecto regional P10-TIC-6858.

si todos los ítems de X están en un registro de la BD, entonces todos los ítems de Y están también en la registro con una alta probabilidad, y X y Y no tienen ningún ítem en común [1].

La mayoría de los trabajos previos han sido enfocados sobre BDs con valores binarios, sin embargo las BDs en las aplicaciones del mundo real están compuestos normalmente por valores cuantitativos, por lo cual varios estudios han sido presentados para extraer reglas de asociación cuantitativas (RACs) [17]. Muchos de estos métodos solo consideran ítems positivos en la extracción de reglas de asociación sin tener en cuenta los negativos que también son interesantes al expresar, por ejemplo, la ausencia de Y ante la presencia de X ($X \rightarrow \neg Y$). Las reglas de asociación negativas pueden incluir ítems negativos en el antecedente, en el consecuente o en ambos. Algunos investigadores han propuesto métodos para extraer RACs positivas y negativas (RACPNs) [3].

Muchos algoritmos evolutivos han sido propuestos para extraer RACs [6,13] porque son considerados una de las técnicas de búsqueda más exitosas para problemas complejos y han demostrado ser muy buenos en el aprendizaje y la extracción de conocimiento. Estos métodos suelen considerar un único objetivo para medir la calidad de las reglas. Sin embargo, recientemente se ha planteado la extracción de reglas de asociación como un problema multi-objetivo teniendo en cuenta varios objetivos en el proceso de extracción de dichas reglas [4]. Los algoritmos evolutivos multi-objetivos (AEMOs) son un mecanismo interesante para tratar problemas de naturaleza multi-objetivo por lo que algunos AEMO han sido aplicados para extraer RACs, donde cada solución del frente de Pareto representa una RAC [4].

En este trabajo proponemos un nuevo AEMO, llamado MOPNAR, para extraer un conjunto reducido de RACPNs con un buen equilibrio entre el número de reglas, el soporte y el cubrimiento de la BD. Nuestra propuesta extiende el AEMO basado en descomposición MOEA/D-DE [12] para realizar un aprendizaje evolutivo de los intervalos de los atributos y una selección de las condiciones para cada regla, maximizando tres objetivos: rendimiento, interés y comprensibilidad. Además, MOPNAR introduce un proceso de reinicialización y una población externa (PE) al modelo evolutivo, con el fin de promover diversidad en la población, almacenar todas las reglas no dominadas encontradas y mejorar el cubrimiento de la BD.

La organización del trabajo es como sigue. En la siguiente sección se introducen definiciones básicas de las RACPNs y de algunas de sus medidas de calidad. En la Sección 3 se detalla el algoritmo evolutivo propuesto para obtener RACPNs. En la Sección 4 se muestran los resultados obtenidos sobre cuatro BDs reales. En la Sección 5 se presentan las conclusiones.

2. Reglas de Asociación Positivas y Negativas

Las reglas de asociación obtenidas en BDs con valores numéricos son denominadas RAC [17], donde cada ítem es un par *atributo-intervalo*. Por ejemplo, una RAC positiva es $Salario \in [3000, 3500] \rightarrow NumCoches \in [3, 4]$. Muchos de

los estudios clásicos presentan dificultades para descubrir RACs debido a que los atributos cuantitativos contienen muchos valores distintos. Para evitar este problema un método normalmente utilizado es dividir el dominio de los atributos numéricos en intervalos, lo cual es un problema crítico en la extracción de RACs porque la información no está clasificada. Además, dichos intervalos pueden tener una gran influencia sobre las reglas obtenidas. Por esta razón, varios investigadores han propuesto métodos para aprender los intervalos de las RACs [3,4,6,13].

La mayoría de estos algoritmos se han centrado en las reglas de asociación positivas sin ponerle atención a las negativas. Sin embargo, estas últimas también pueden ser interesantes puesto que ofrecen un nuevo conocimiento para apoyar la toma de decisiones. Asimismo, pueden incluir ítems negativos dentro del antecedente($\neg X \rightarrow Y$), del consecuente ($X \rightarrow \neg Y$), o de ambos ($\neg X \rightarrow \neg Y$). Destacar que, las reglas de asociación negativas deben incluir al menos un ítem negativo en la regla. Por ejemplo, una RAC negativa puede ser $Altura \in [110, 150] \rightarrow Edad \in \neg[0, 5]$.

Las medidas el soporte y la confianza son las más utilizadas para evaluar las RACs. El soporte y la confianza de una regla $X \rightarrow Y$ se define como:

$$Soporte(X \rightarrow Y) = SOP(XY) \tag{1}$$

$$Confianza(X \rightarrow Y) = SOP(XY)/SOP(X) \tag{2}$$

donde $SOP(I)$ es la probabilidad de que un conjunto de ítems I aparezca en la BD. Las técnicas clásicas para extraer reglas de asociación intentan obtener reglas con valores de soporte y confianza mayores que un mínimo de soporte (minSop) y una mínima confianza (minConf). Sin embargo, varios autores han señalado algunos inconvenientes de este marco de trabajo que conduce a encontrar reglas de baja calidad [8]. La confianza no detecta independencia estadística o dependencia negativa entre los ítems debido a que no tiene en cuenta el soporte del consecuente. Asimismo, los ítems con soporte muy alto pueden generar reglas de baja calidad porque cualquier conjunto de ítems parece ser un buen predictor de ellos. Por esta razón, en la literatura se han propuesto otras medidas de calidad para la selección y ranking de ejemplos de acuerdo con su interés potencial para el usuario [10]. A continuación describimos brevemente las medidas que han sido utilizadas en este trabajo.

La medida Lift [15] representa el ratio entre la confianza de la regla y la confianza esperada de la regla. Esta medida toma valores en el intervalo [0,∞) donde valores menores de 1 representan dependencia negativa, 1 representa independencia y mayores que 1 representan dependencia positiva. Lift de una regla $X \rightarrow Y$ se define como:

$$Lift(X \rightarrow Y) = SOP(XY)/(SOP(X)SOP(Y)) \tag{3}$$

El Factor de Certeza (FC) [16] mide la variación de la probabilidad de que Y esté en una registro cuando se consideran solo los registros donde está X. FC toma valores en el intervalo [-1,1] donde valores positivos y negativos representan

dependencia positiva y negativa respectivamente y 0 representa independencia. Esta medida para una regla $X \rightarrow Y$ se define de tres maneras dependiendo de si la confianza es menor, mayor o igual que $SOP(Y)$:

Si $Confianza(X \rightarrow Y) > SOP(Y)$

$$(Confianza(X \rightarrow Y) - SOP(Y))/(1 - SOP(Y)) \tag{4}$$

Si $Confianza(X \rightarrow Y) < SOP(Y)$

$$(Confianza(X \rightarrow Y) - SOP(Y))/SOP(Y) \tag{5}$$

Sino es 0

Netconf [2] evalúa una regla basándose en el soporte de la regla, del antecedente y del consecuente. Netconf obtiene valores en el intervalo [-1,1] donde valores positivos y negativos representan dependencia positiva y negativa respectivamente y 0 representa independencia. Netconf de una regla $X \rightarrow Y$ se define como:

$$Netconf(X \rightarrow Y) = (SOP(XY) - SOP(X)SOP(Y))/(SOP(X)(1 - SOP(X))) \tag{6}$$

3. MOPNAR: AEMO para extraer RACPNs

En esta sección se describe nuestra propuesta, MOPNAR, para obtener un conjunto de RACPNs con un buen equilibrio entre el número de reglas, el soporte y el cubrimiento de la BD.

3.1. Modelo Evolutivo Multi-Objetivo MOEA/D-DE

El método propuesto extiende el modelo evolutivo de MOEA/D-DE [12] para extraer RACPNs, e introduce una PE y un proceso de reinicialización para almacenar todas las reglas no dominadas encontradas, provocar diversidad en la población y mejorar el cubrimiento de las BDs. Así, este modelo primero genera un vector de pesos para cada subproblema, que son utilizados para calcular el valor del enfoque de descomposición de cada subproblema. Luego se selecciona un conjunto de vecinos para cada vector de pesos, conteniendo T vectores de pesos más cercanos. Después el algoritmo genera una población inicial, inicializa el punto de referencia para cada objetivo, con los mejores valores encontrados hasta el momento e inicializa la PE con las reglas no dominadas de la población inicial. A continuación se generan dos hijos aplicando los operadores de cruce, mutación y reparación. Estos se aplican sobre una solución de la población y sobre otra solución seleccionada aleatoriamente con una probabilidad δ entre su vecindad y la población (δ es definida por el usuario). Estos hijos se utilizan para actualizar los puntos de referencias y sustituir algunas soluciones de la población actual que tengan los peores valores del enfoque de descomposición. Téngase en cuenta que el máximo número de soluciones que se pueden sustituir por una solución hija es limitado y debe ser mucho menor que T. Estos pasos se repiten para cada solución de la población y se actualiza la PE en la cual se mantienen

todas las reglas no dominadas encontradas hasta el momento y se eliminan las reglas redundantes para evitar solapamientos. Finalmente, se aplica el proceso de reinicialización al final de una generación cuando el número de nuevas soluciones en la población es menor que $\alpha\,\%$ ($\alpha\,\%$ definido por el usuario, normalmente al $5\,\%$). En ese momento se marcan los registros cubiertos por las reglas que están en la PE y a partir de los no marcados se aplica de nuevo el proceso de inicialización (ver Subsección 3.3). Este proceso nos permite realizar una buena exploración del espacio de búsqueda y mejorar el cubrimiento de la BD. Todo este proceso se repite hasta que se cumpla la condición de parada (ver [12] para más información). Al culminar el proceso se devuelven todas las reglas de la PE que viene dado por la totalidad del frente.

En [12] los autores presentaron tres enfoques de descomposición donde recomendaron el propuesto por Tchebycheff [14], el cual es el utilizado en este trabajo.

3.2. Objetivos

Nuestra propuesta maximiza tres objetivos: rendimiento, interés y comprensibilidad. Rendimiento es el producto entre el soporte y el FC (ver Sección 2), lo cual nos permite obtener un conjunto de reglas con un buen equilibrio entre reglas locales y generales. Destacar que solamente nos interesan las reglas con dependencia fuerte [8] entre los ítems porque representan dependencias positivas entre ellos y evitan el problema del soporte (ver Sección 2). Además, teniendo en cuenta que las reglas de asociación negativas nos permiten representar dependencias negativas, solo obtendremos aquellas reglas con FC > 0.

El Interés intenta medir como de interesante es una regla, permitiendonos extraer solo aquellas reglas que sean interesante para el usuario. En este trabajo hemos utilizado la medida de interés lift (ver Sección 2), la cual nos permite detectar la dependencia negativa, positiva o independencia entre los ítems y su rango de valores no está limitado, permitiendo representar mejor las diferencias entre las reglas y reducir el número de empates.

La comprensibilidad pretende medir lo fácil de comprender que puede ser una regla [9]. Las reglas cuando involucran muchos atributos pueden resultar difíciles de comprender. En este trabajo hemos usado la medida de comprensibilidad de una regla $X \rightarrow Y$ según el número de atributos que contiene. Esta se define como sigue, donde $Atrib_{X \rightarrow Y}$ es el número de atributos involucrados en la regla.

$$Comprensibilidad(X \rightarrow Y) = 1/Atrib_{X \rightarrow Y} \tag{7}$$

3.3. Esquema de Codificación y Población Inicial

Un cromosoma es un vector de n genes que representa los atributos e intervalos de una regla, donde n es el número de atributos de la BD. Nuestra propuesta utiliza una codificación posicional en la que el i-ésimo gen codifica el i-ésimo atributo. Cada gen consta de cuatro partes: ac indica si un gen es considerado en la regla; pn indica si el intervalo es positivo o negativo; li y ls representan

el límite inferior y superior del intervalo del atributo respectivamente. Hay que tener en cuenta que si el atributo es nominal, li y ls serán iguales, representando solo un valor del atributo nominal. Por tanto un cromosoma C_T se codifica de la siguiente manera:

$$C_T = Gen_1 Gen_2 \dots Gen_n, \quad Gen_i = (ac_i, pn_i, li_i, ls_i), \;\; i = 1, \dots, n$$

Para evitar que el aumento de los intervalos cubra la totalidad del dominio hemos definido *amplitud*, el cual representa el tamaño máximo que el intervalo de un atributo puede alcanzar. La *amplitud* de un atributo i se define como:

$$Amplitud_i = (Max_i - Min_i)/\gamma \tag{8}$$

donde γ es un valor dado por el experto que determina el equilibrio entre la generalización y la especificidad de las reglas, y Min_i y Max_i son los valores de mínimo y máximo del dominio del atributo i, respectivamente. Para los intervalos negativos la amplitud representa el tamaño mínimo que el intervalo de un atributo puede alcanzar.

La población inicial estará compuesta por un conjunto de reglas que contienen un solo atributo en el consecuente y presentan un buen cubrimiento de la BD. Para crear la población inicial, primero se selecciona aleatoriamente los atributos que formarán parte del antecedente y del consecuente de la regla. Después se selecciona si el intervalo será positivo o negativo. Luego se generan los intervalos que tendrán un tamaño igual al 50 % de la amplitud de cada atributo y centrados en un registro seleccionado. Finalmente se marcan los registros que cubre la regla en la BD. Este proceso se repite para todos los registros que no han sido marcados hasta que se complete la población inicial. Si todos los registros se han marcado y la población inicial no se ha completado, se vuelven a desmarcar todos los registros y el proceso se repite hasta que la población inicial sea completada.

3.4. Operadores Genéticos

Esta propuesta utiliza los operadores genéticos de cruce, mutación y reparación. Primero, el operador de cruce genera dos hijos intercambiando aleatoriamente los genes de los padres (exploración). Después, el operador de mutación selecciona aleatoriamente un gen del cromosoma. De este gen, selecciona al azar uno de los límites del intervalo y aumenta o disminuye su valor de manera aleatoria, y modifica los valores de ac y pn aleatoriamente. Por último, el operador de reparación corrige las reglas que tengan más de un atributo en el consecuente o no tengan antecedente o consecuente. Si el consecuente contiene más de un atributo, uno de ellos se selecciona aleatoriamente para ser el consecuente y el resto pasan al antecedente. Si no hay ningún atributo en el antecedente y/o consecuente, se seleccionan al azar entre los atributos que no han sido considerados en la regla. Además para obtener reglas más simples este operador decrementa el tamaño de los intervalos mientras el número de registros cubiertos sea igual que el cubierto por los intervalos originales. Destacar que, si el intervalo es negado, este se incrementa reduciendo el dominio que cubre.

Tabla 1: BDs consideradas en el estudio experimental

Nombres	*#Atrib(R/E/N)*	*#Reg*
House_16H [3]	17 (10/7/0)	22784
Movement Libras	91 (90/0/1)	360
Segment	20 (19/1/0)	2310
Stock Price	10 (10/0/0)	950

Disponible en `http://sci2s.ugr.es/keel/datasets.php`

Tabla 2: Parámetros considerados en la comparación de los métodos

Algoritmos	*Parámetros*
Apriori	*minSop = 0,1, minConf = 0,8*
Alatasetal	N_{eval}=50000, nCromoInicialAleat=12, r = 3, TamTorneo = 10, P_{sel}= 0,25, P_{cru} = 0,7, P_{mut_min} = 0,05, P_{mut_max} = 0,9, $Peso_{sop}$ = 5, $Peso_{conf}$ = 20, $Peso_{amplRule}$ = 0,05, $Peso_{amplInterv}$ = 0,02, $Peso_{cubrimiento}$ = 0,01
GAR	TamPop = 100, nItems = 100, N_{eval} = 50000, P_{sel}= 0,25, P_{cru} = 0,7, P_{mut} = 0,1, ω = 0,4, Ψ= 0,7, μ= 0,5, minSop = 0,1, minConf = 0,8
MODENAR	TamPop = 100, N_{eval}=50000, Umbral= 60, CR = 0,3, $Peso_{sop}$ = 0,8, $Peso_{conf}$ = 0,2, $Peso_{comp}$ = 0,1, $Peso_{amplInterv}$ = 0,4
MOPNAR	N_{eval}=50000, H=13, m=3, TamPop=N^{m-1}_{H+m-1}, T=10, δ=0,9, η_r=2, γ=2, P_{mut}= 0,1, α = 5 %

4. Estudio Experimental

Para analizar el funcionamiento del método propuesto hemos seleccionado 4 BDs de la vida real con distintos tamaños. La Tabla 1 resume las principales características de las 4 BDs y muestra el enlace al repositorio de datos KEEL-dataset [5] del cual podemos descargarlas. Para cada BD se muestra el número de registros ("#Reg") y el número de atributos reales, enteros o nominales ("#Atrib(R/E/N)").

En este estudio comparamos a MOPNAR con un algoritmo clásico para extraer reglas de asociación, Apriori [17], con un algoritmo evolutivo mono-objetivo y otro AEMO para obtener RACs, GAR [13] y MODENAR [4], respectivamente, y con Alatasetal, un algoritmo evolutivo para extraer RACPNs propuesto por Alatas y otros en [3]. Estos métodos están disponibles en KEEL [7].

La Tabla 2 muestra los parámetros de los métodos analizados. Con estos valores para nuestra propuesta, hemos tratado de facilitar las comparaciones, seleccionando parámetros estándar comunes que funcionan bien en la mayoría de los casos, y para el resto de los algoritmos hemos seleccionado los parámetros y los objetivos de acuerdo a las recomendaciones de los autores de cada propuesta. Destacar que Apriori y GAR necesitan un minSop y minConf para extraer RACs, seleccionando para ellos valores estándar que funcionan bien en la

[3] Esta BD fue diseñada sobre la base de los datos proporcionada por la Oficina del Censo de EE.UU. [http://www.census.gov] (Acceso Lookup [http://www.census.gov/cdrom/lookup]: Resumen del archivo 1).

Tabla 3: Resultados obtenidos por los algoritmos analizados

BD	*Algoritmos*	$\#R$	Med_{Sop}	Med_{Conf}	Med_{Lift}	Med_{CF}	$Med_{Netconf}$	Med_{Amp}	$\%Reg$
House16H	Apriori	1749917	0,22	0,97	2,19	0,83	0,45	8,65	**100**
	Alatasetal	90,67	0,19	**0,99**	1,03	0,58	0,03	8,76	98,09
	GAR	105,60	**0,76**	0,90	1,03	0,20	0,17	**2,01**	99,99
	MODENAR	64,40	0,69	**0,99**	1,15	0,72	0,19	7	81
	MOPNAR	84,80	0,34	0,89	**6,18**	**0,86**	**0,70**	2,91	99,37
Movement Libras	Apriori	-	-	-	-	-	-	-	-
	Alatasetal	0	-	-	-	-	-	-	-
	GAR	2,60	**0,42**	0,94	3,73	0,90	**0,90**	**2**	53,28
	MODENAR	23,40	0,01	0,15	**32,31**	0,04	0,15	61,61	3,17
	MOPNAR	59,80	0,31	**0,97**	19,46	**0,95**	0,84	2,63	**99,50**
Segment	Apriori	-	-	-	-	-	-	-	-
	Alatasetal	47,50	**0,53**	0,94	1,03	0,26	0,02	4,14	**100**
	GAR	18,80	0,36	0,89	2,49	0,58	0,47	**2**	97,97
	MODENAR	58,80	0,33	0,97	1,72	0,93	0,58	10,60	56,49
	MOPNAR	83,20	0,29	**0,98**	**16,75**	**0,97**	**0,85**	2,74	99,91
Stock	Apriori	855	0,13	0,91	4,77	0,88	0,76	4,16	99,48
	Alatasetal	14,40	0,08	**1**	**96,54**	0,92	0,72	2,73	21,04
	GAR	2	**0,56**	0,87	1,35	0,62	0,62	**2**	73,30
	MODENAR	63,80	0,48	0,92	1,75	0,61	0,30	3	81,86
	MOPNAR	83	0,22	0,94	15,07	**0,93**	**0,86**	2,92	**99,64**

mayoría de los casos para todas las BDs. Además los resultados obtenidos para los AEMO se refieren al conjunto de reglas no dominadas obtenido.

La Tabla 3 muestra los resultados medios de 5 ejecuciones de los algoritmos analizados, donde $\#R$ representa el número medio de reglas, Med_{Sop}, Med_{Conf}, Med_{Lift}, Med_{FC}, $Med_{Netconf}$ representan los valores medios de soporte, confianza, lift, FC y netconf, respectivamente, Av_{Amp} el número medio de atributos involucrados en las reglas y $\%Reg$ es el tanto por ciento de registros de la BD cubiertos por las reglas generadas. Hay que señalar que Apriori no puede ejecutarse en Movement Libras y Segment porque tienen muchos atributos.

Como podemos observar el método propuesto obtiene mejores valores para casi todas las medidas de interés que el resto de los algoritmos en todas las BDs. Destacar que algunos de los métodos obtienen reglas con valores altos para las medidas de soporte y confianza (incluso en varias BDs mejores que nuestra propuesta) pero ellos presentan peores valores para el resto de las medidas. Esto se debe a que obtienen reglas con conjuntos de ítems con soportes altos en el consecuente o que representan dependencia negativa. Además, el número medio de reglas obtenidas es reducido (menor que 100 en todas las BDs) y con un bajo número de atributos lo que facilita su interpretabilidad desde el punto de vista del usuario. Por otra parte, el cubrimiento de la BD es muy alto (cercano a 100 % en todas las BDs) aportando un conocimiento sobre la totalidad de la BD. Destacar que en la mayoría de las BDs el resto de los algoritmos obtienen menos reglas que MOPNAR pero con valores más bajos de cubrimiento.

La Fig. 1 representa un boxplot de los valores del FC para todas las reglas obtenidas en una de las 5 ejecuciones realizadas en la BD stock seleccionada aleatoriamente. Observe que MOPNAR presenta mejores valores de FC que el resto de los algoritmos analizados y todas sus reglas representan dependencias

positivas, obteniendo valores cercanos al máximo valor que FC puede alcanzar. Destacar que algunas reglas obtenidas por Alatasetal y MODENAR representan independencia o dependencia negativa según los valores de esta medida.

La Fig. 2 representa los frentes de Pareto obtenidos por nuestra propuesta cuando se aplica el proceso de reinicialización en diferentes momentos del proceso evolutivo de las 5 ejecuciones realizadas sobre la BD stock. Observe que hemos modificado los objetivos para mostrarlos como objetivos de minimización. Podemos ver cómo la PE y el proceso de reinicialización nos permiten mejorar los frente Pareto y aumentar el número de soluciones no dominadas para cada proceso de reinicialización.

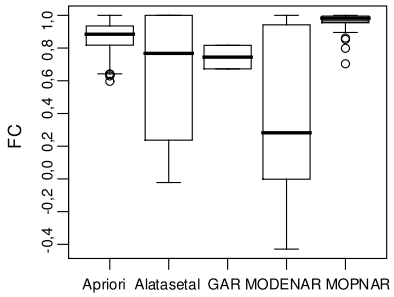


Figura 1: Boxplot de la medida FC para los algoritmos analizados en la BD Stock

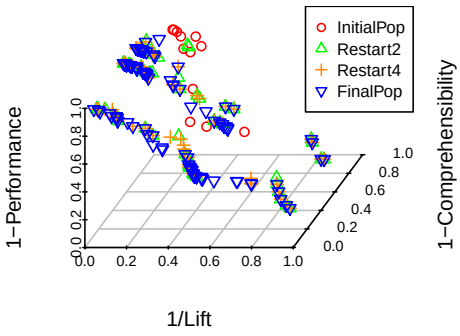


Figura 2: Frentes de Pareto obtenidos en diferentes momentos del proceso evolutivo en la BD Stock

5. Conclusiones

En este trabajo hemos propuesto un nuevo AEMO para extraer RACPNs llamado MOPNAR, que nos permite obtener un conjunto reducido de RACPNs fáciles de comprender, interesantes y con un buen cubrimiento de la BD, maximizando tres objetivos: rendimiento, interés y comprensibilidad. Para hacer esto, esta propuesta extiende el AEMO MOEA/D-DE para realizar un aprendizaje evolutivo de los intervalos de los atributos y una selección de condiciones para cada regla. Además, introduce al modelo evolutivo una PE y un proceso de reinicialización para almacenar todas las reglas no dominadas encontradas, promover su diversidad y mejorar el cubrimiento de la BD.

Si observamos los resultados obtenidos podemos ver como el método propuesto obtiene un conjunto reducido de RACPNs, con un buen equilibrio entre el número de reglas, el soporte y el cubrimiento, presentando altos valores de cubrimiento en todas las BDs. Además, estas reglas tienen pocos atributos, lo que permite una mejor comprensión del usuario y presentan altos valores de las medidas de interés en todas las BDs.

Referencias

1. R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *SIGMOD*, pages 207–216, Washington D.C., 1993.
2. K.-I. Ahn and J.-Y. Kim. Efficient mining of frequent itemsets and a measure of interest for association rule mining. *JIKM*, 3(3):245–257, 2004.
3. B. Alatas and E. Akin. An efficient genetic algorithm for automated mining of both positive and negative quantitative association rules. *Soft. Comput.*, 10(3):230–237, 2006.
4. B. Alatas and E. Akin. MODENAR: Multi-objective diferential evolution algorithm for mining numeric association rules. *Appl. Soft Comput.*, 8(1):646–646, 2008.
5. J. Alcala-Fdez, A. Fernandez, J. Luego, J. Derrac, S. Garcia, L. Sanchez, and F. Herrera. Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *J. Mult-Valued Log. S*, 17(2-3):255–287, 2011.
6. J. Alcala-Fdez, N. Flugy-Pape, A. Bonarini, and F. Herrera. Analysis of the effectiveness of the genetic algorithms based on extraction of association rules. *Fund. Inform.*, 98(1):1–14, 2010.
7. J. Alcala-Fdez, L. Sanchez, S. Garcia, M. del Jesus, S. Ventura, J. O. J. Garrell, C. Romero, J. Bacardit, V. Rivas, J. Fernandez, and F. Herrera. Keel: A software tool to assess evolutionary algorithms to data mining problems. *Soft. Comput.*, 13(3):307–318, 2009.
8. F. Berzal, I. Blanco, D. Sanchez, and M. Vila. Measuring the accuracy and interest of association rules: A new framework. *Intell. Data Anal.*, 6(3):221–235, 2002.
9. M. Fidelis, H. Lopes, and A. Freitas. Discovering comprehensible classification rules with a genetic algorithm. In *Proceedings of the 2000 Congress on Evolutionary Computation*, pages 805–810, CA, USA, 2000.
10. L. Geng and H. Hamilton. Interestingness measures for data mining: A survey. *ACM Comput. Surv.*, 38(3), 2006.
11. J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, second edition edition, 2006.
12. H. Li and Q. Zhang. Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II. *IEEE T Evolut. Comput.*, 13(2):284–302, 2009.
13. J. Mata, J. Alvarez, and J. Riquelme. An evolutionary algorithm to discover numeric association rules. In *ACM Symposium on Applied Computing*, Madrid, Spain, March 2002.
14. K. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer, Norwell, MA, 1999.
15. S. Ramaswamy, S. Mahajan, and A. Silberschatz. On the discovery of interesting patterns in association rules. In *24rd International Conference on Very Large Data Bases*, pages 368–379, San Francisco, CA, USA, 1998.
16. E. Shortliffe and B. Buchanan. A model of inexact reasoning in medicine. *Math. Biosci.*, 23:351–379, 1975.
17. R. Srikant and R. Agrawal. Mining quantitative association rules in large relational tables. In *ACM SIGMOD International Conference on Management of data (SIGMOD96)*, pages 1–12, Montreal, Quebec, Canada, 1996.

Algoritmo de programación evolutiva para identificación de arquitecturas software

Aurora Ramírez, José Raúl Romero y Sebastián Ventura

Dpto. de Informática y Análisis Numérico
Universidad de Córdoba, Campus de Rabanales, 14071 Córdoba
{aramirez, jrromero, sventura}@uco.es

Resumen La aplicación de técnicas metaheurísticas para la resolución de tareas de la Ingeniería del Software se engloba dentro del área denominada SBSE (*Search Based Software Engineering*). Por otra parte, el análisis y diseño del software y, en concreto, el análisis arquitectónico son fases importantes en el desarrollo de proyectos, ya que permiten prevenir errores en el futuro. No obstante, este tipo de tareas no ha sido muy tratada en SBSE a pesar de que la obtención de la arquitectura del software puede ser abordada como un problema de búsqueda y optimización. En este trabajo se presenta un algoritmo de programación evolutiva para la optimización de arquitecturas basadas en componentes. Para ello se ha desarrollado una nueva representación del problema, una función de *fitness* basada en métricas de diseño y un conjunto de operadores genéticos. La propuesta ha sido validada con arquitecturas reales, mostrando resultados interesantes y prometedores.

Keywords: Programación evolutiva, arquitecturas software, *Search Based Software Engineering*

1. Introducción

La utilización de técnicas de optimización y búsqueda en el dominio de la Ingeniería del Software, denominada *Search Based Software Engineering* (SBSE), ha experimentado un gran crecimiento en los últimos años [1], donde las técnicas bio-inspiradas han sido las metaheurísticas más empleadas [2].

En este dominio, los problemas a resolver están fuertemente vinculados a decisiones humanas, especialmente en las fases de análisis, diseño y pruebas. El análisis arquitectónico constituye un ejemplo de un proceso centrado en el experto que, a su vez, tiene una gran repercusión en la calidad y reusabilidad del software. En concreto, la identificación de componentes es el proceso mediante el cual se detectan los bloques constituyentes del software y sus funcionalidades, enfoque útil cuando se trabaja con sistemas complejos [3].

Este proceso puede ser abordado a partir de la información disponible en los modelos de análisis, pues en ellos se detallan las clases y las relaciones entre ellas, representando conjuntamente la estructura global del sistema. Habitualmente, estos modelos son los más utilizados pero, a medida que el sistema crece,

se vuelven demasiado complejos y es necesario realizar un análisis a nivel arquitectónico. De este modo, los componentes son identificados a partir de conjuntos de clases que definen funcionalidades específicas. En este punto, el problema puede ser enfocado como un proceso de optimización de estos componentes a partir de la búsqueda de agrupaciones de clases funcionalmente relacionadas.

En este trabajo se propone el desarrollo de un algoritmo de programación evolutiva para la identificación de arquitecturas software basadas en componentes, el cual es orientado como un método de apoyo a la decisión del ingenierio. El problema es enfocado desde un nivel de abstracción elevado, a diferencia de otras propuestas existentes, más cercanas a la refactorización de código y el mantenimiento del software. Se ha definido una representación cercana al experto, donde se consideran los elementos (componentes, clases, etc.) con los que este trabaja. La función de *fitness* propuesta se basa en criterios de diseño, mientras que los operadores genéticos representan transformaciones arquitectónicas.

El resto del artículo está estructurado de la siguiente manera. En la Sección 2 se describen propuestas relacionadas con este trabajo, centrándose en la aplicación de técnicas bio-inspiradas. El diseño del algoritmo evolutivo se detalla en la Sección 3. A continuación, la Sección 4 incluye la experimentación realizada y los resultados obtenidos. Finalmente, las conclusiones y el trabajo futuro se discuten en la Sección 5.

2. Trabajo relacionado

En el ámbito del análisis y diseño del software existen multitud de aspectos que pueden ser optimizados mediante la utilización de metaheurísticas [4]. En estos casos, el principal objetivo es apoyar al ingeniero en la toma de decisiones, de forma que se ofrecen diferentes alternativas entre las cuales elegir.

En [5] se desarrolla un algoritmo genético interactivo donde las funcionalidades identificadas en los casos de uso son traducidas en un conjunto de clases, atributos y métodos.

La introducción de patrones de diseño en el contexto de la refactorización del software ha sido estudiado en [6], donde se hace uso de la programación genética para manejar grafos y árboles que codifican la arquitectura resultante y el conjunto de transformaciones realizadas, respectivamente. En [7], un algoritmo genético es combinado con técnicas de clustering para extraer arquitecturas basadas en componentes a partir de código fuente.

La presencia de enfoques multi-objetivo también es frecuente en SBSE cuando se deben optimizar conjuntamente varios factores de diseño. La búsqueda de arquitecturas alternativas basándose en aspectos como la disponibilidad y el coste [8] o la modificabilidad y la eficiencia [9] son ejemplos de ello.

Aunque las propuestas anteriores trabajan a nivel de arquitecturas software, abarcan problemas diferentes al aquí considerado. La refactorización y búsqueda de alternativas de diseño requieren la existencia de una arquitectura inicial, aspecto que no es necesario en esta propuesta. Por otro lado, la extracción de componentes a partir de código fuente utiliza otro tipo de información al que

aquí se propone, pues los modelos de análisis están centrados en aspectos de alto nivel, permitiendo abordar el problema en una fase temprana del desarrollo.

3. Diseño del algoritmo

3.1. Genotipo y fenotipo

El problema que aquí se plantea puede ser definido como la búsqueda de la configuración óptima de un conjunto de elementos (componentes, interfaces y conectores) que permitan representar la arquitectura de un sistema software. De forma más precisa, se trata de descubrir la existencia de estos elementos a partir de otro tipo de información, los modelos de análisis del software, de forma que de ellos se abstraiga un modelo arquitectónico basado en componentes. Estos modelos pueden ser procesados directamente desde herramientas de modelado con las que trabajan los ingenieros del software y que, habitualmente, utilizan la notación UML.

En este punto resulta necesario definir adecuadamente algunos de los conceptos relacionados con el alcance del problema propuesto [3]. Los *componentes* son unidades de composición que presentan una funcionalidad bien definida y que es puesta a disposición de otros mediante sus *interfaces*. La implementación del componente es realizada en su interior por un conjunto de clases, que permanece oculto e inaccesible desde el exterior del componente. Las interfaces, por tanto, son el medio de interacción entre los componentes, pues en ellas se especifica el conjunto de operaciones que un componente provee al resto y que pueden identificarse a partir de la interacción entre clases pertenecientes a distintos componentes. Finalmente, los *conectores* representan la unión de varios puntos de interacción entre componentes, habitualmente pares de interfaces proveídas y requeridas. La Figura 1 representa un ejemplo de arquitectura modelada con UML, que constituye el fenotipo del problema.

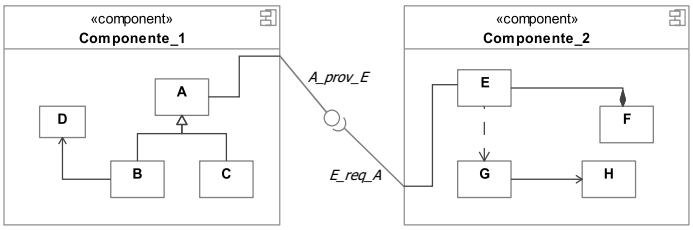


Figura 1. Fenotipo del problema

La selección de una codificación apropiada es un aspecto clave en el desarrollo de cualquier algoritmo de búsqueda, ya que influye en su eficiencia y, a su vez, debe ser interpretable, especialmente cuando el proceso está destinado a usuarios no expertos en computación evolutiva. A diferencia de la mayoría de propuestas

en el área de SBSE, que utilizan codificaciones lineales, se ha optado por una representación en árbol, donde los elementos definidos anteriormente constituyen los nodos del mismo en función de su nivel de composición.

Como se puede apreciar en la Figura 2, que representa el genotipo correspondiente al fenotipo anterior, la raíz del árbol representa la arquitectura completa, compuesta por un conjunto de componentes y conectores. Cada componente, a su vez, es desglosado en los tres elementos prefijados que lo describen (sombreados en la figura): el conjunto de clases que implementa su funcionalidad, las interfaces proveídas y las interfaces requeridas. Finalmente, los conectores se componen de un par de interfaces, una proveída y otra requerida.

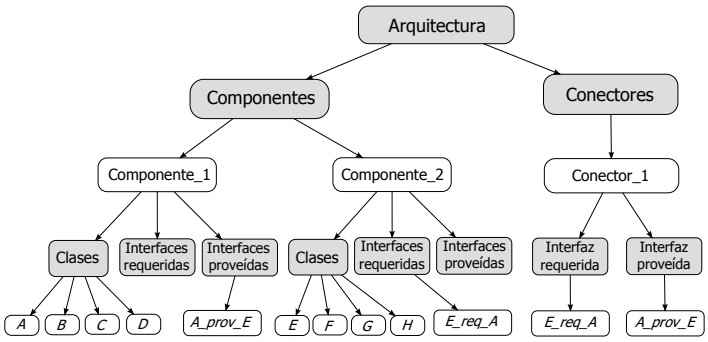


Figura 2. Genotipo del problema

El problema de optimización propuesto presenta varias restricciones que deben ser consideradas durante el proceso evolutivo. En primer lugar, cada una de las clases que componen el modelo inicial deben pertenecer a un único componente, proceso que es controlado en la inicialización del algoritmo.

Otros aspectos relacionados con la composición de los individuos resultantes también deben ser tenidos en cuenta. Por un lado, no pueden existir componentes vacíos, ya que no implementarían ninguna funcionalidad. Por otro lado, todos los componentes deben presentar, al menos, una interfaz que le permita interactuar con el resto de componentes. Finalmente, la presencia de componentes mutuamente dependientes, esto es, donde uno provee y requiere al otro (y viceversa), no son admitidas desde el punto de vista del análisis arquitectónico.

3.2. Inicialización de la población

La población inicial se crea mediante la generación aleatoria de un conjunto de individuos válidos. Cada individuo de la población es construído realizando una distribución aleatoria de las clases del modelo inicial de análisis en un número también aleatorio de componentes, obteniendo así una población diversa respecto al número de componentes. Una vez realizada esta distribución, es posible determinar las interfaces y los conectores establecidos entre estos componentes.

3.3. Función de *fitness*

La evaluación de los individuos está basada en criterios de diseño ampliamente extendidos en el desarrollo basado en componentes. La cohesión hace referencia a la obtención de componentes con funcionalidades bien definidas. Por otro lado, el acoplamiento indica la interrelación existente entre componentes. Finalmente, otro aspecto interesante a considerar es el tamaño de los componentes, directamente relacionado con la complejidad de los mismos. Estos tres conceptos han sido traducidos a tres métricas cuantificables sobre el problema propuesto, tal y como se detalla a continuación.

La cohesión global de la arquitectura se obtiene como la media de la cohesión calculada sobre cada componente (coh_i) mediante la Ecuación 1. Esta expresión se compone de dos términos ponderados que cuantifican dos características: el grado de unión entre las clases presentes en el interior del componente y el número de agrupaciones que lo conforman. El primero, que varía entre 0 y 1, contabiliza el número de relaciones establecidas entre las clases, clasificadas en función de los distintos tipos de relaciones UML y ponderadas en función de su importancia relativa: asociaciones (n_{as}, w_{as}), dependencias (n_{de}, w_{de}), agregaciones y composiciones (n_{ag}, n_{co}, w_{ac}) y generalizaciones (n_{ge}, w_{ge}). El segundo término, definido en el rango [-1,0], representa una penalización por el número de agrupaciones (n_{gr}) en el interior del componente, fundamentado en el concepto de *componentes conexas* en la Teoría de Grafos y bajo la idea de que varias agrupaciones representan dispersión en cuanto a la funcionalidad del componente. n_{cl} representa el número de clases en el interior del componente. Todos los pesos presentes en la Ecuación 1 han sido prefijados experimentalmente.

$$coh_i = \frac{w_{c1}}{(w_{as} + w_{de} + w_{ac} + w_{ge}) \cdot (n_{cl} - 1)} \cdot \Big[w_{as} \cdot n_{as} + w_{de} \cdot n_{de} + \\ + w_{ac} \cdot (n_{ag} + n_{co}) + w_{ge} \cdot n_{ge} \Big] + w_{c2} \cdot \frac{1 - n_{gr}}{n_{cl} - 1} \quad (1)$$

El siguiente término a considerar es el exceso de interrelaciones entre los componentes, lo cual se refleja en la medida de acoplamiento (*acopl*) propuesta en la Ecuación 2. En ella se contabilizan las relaciones existentes entre clases pertenecientes a distintos componentes que no son especificadas a través de las interfaces. Por ejemplo, esta situación puede ocurrir durante el proceso evolutivo ante una relación de generalización en la que la clase padre y las subclases no están ubicadas en el mismo componente. C indica el número de componentes, mientras que R, definido en la Ecuación 3, establece una penalización por la presencia de este tipo de relaciones. Concretamente, R acumula, para cada par de componentes i y j, el peso de la relación más fuerte entre las k relaciones ($k = 0...m$), $r^k_{i,j}$, establecidas entre ellos. La normalización de la métrica se consigue mediante el término R_{max}, pues representa la peor configuración posible (todos los componentes están mutuamente conectados por medio de la relación más fuerte). Por tanto, la medida toma valores entre 0 (no existen relaciones entre los componentes salvo las definidas a través de las interfaces) y 1 (todos los componentes están fuertemente acoplados).

$$acopl = \frac{1}{R_{max}} \cdot \frac{2 \cdot R \cdot (C-2)!}{C!} \tag{2}$$

$$R = \sum_{i=1}^{C} \sum_{j=i+1}^{C} max\ r_{i,j}^{k} \tag{3}$$

El coeficiente de variación, CV, mide la dispersión en el tamaño de los componentes, definido como el número de clases que los conforman. Su cálculo se obtiene por medio de la Ecuación 4, donde μ y σ representan la media y la desviación estándar del tamaño de los componentes, respectivamente. Esta medida puede tomar valores entre 1 y 8.

$$CV = \frac{\sigma}{\mu} \tag{4}$$

Finalmente, la función de *fitness* es construída a partir de la suma ponderada de los tres términos anteriores. Puesto que el acoplamiento y el coeficiente de variación deben ser minimizados y esta última, a su vez, normalizada, se han aplicado las correspondientes transformaciones aritméticas con los límites superiores de cada medida, tal y como se observa en la Ecuación 5.

$$fitness = w_{coh} \cdot \frac{\sum_{i=1}^{n} coh_i}{n} + w_{acopl} \cdot (1 - acopl) + w_{cv} \cdot \left(\frac{8 - CV}{8}\right) \tag{5}$$

3.4. Operadores genéticos

Dada la presencia de varias restricciones en el problema, la recombinación de individuos no ha sido considerada ante la dificultad de generar soluciones válidas. El único operador considerado es la mutación, para el cual se han diseñado diferentes variantes que representan las siguientes transformaciones arquitectónicas:

- *Añadir un componente*: Se crea un nuevo componente, mediante la selección aleatoria de clases e interfaces pertenecientes a otros componentes.
- *Eliminar un componente*: Se selecciona el componente con mayor acoplamiento para ser eliminado de la arquitectura. Una vez identificado, se realojan sus clases e interfaces en el resto de componentes.
- *Unir dos componentes*: Se seleccionan dos componentes, aquel con mayor acoplamiento y otro aleatorio, para ser fusionados en uno solo.
- *Dividir un componente*: Se selecciona aleatoriamente un componente entre aquellos que presentan varios grupos diferenciados de clases, los cuales son distribuidos en dos nuevos componentes.
- *Mover una clase*: Se selecciona de forma aleatoria una clase del modelo para ser transferida desde su componente actual hacia otro elegido aleatoriamente.

La selección de un operador u otro está basada en una ruleta probabilística, la cual es creada para cada padre en función de los operadores que se le pueden aplicar y la probabilidad asociada a cada mutador, configurable por el usuario.

Finalmente, la presencia de restricciones (ver Sección 3.1) debe ser considerada también a la hora de generar nuevos individuos. Los mutadores que extraen clases de unos componentes para ubicarlas en otros comprueban previamente que el movimiento a realizar no implique la ausencia de clases en el componente origen. La creación de individuos con componentes aislados o mutuamente dependientes es también identificada tras la evaluación del individuo, de forma que si el descendiente creado no es válido, se realiza un número máximo de intentos que, si es alcanzado, retorna al individuo inicial.

3.5. Selección y reemplazo

En este trabajo, el método de selección utilizado consiste en la actuación de cada individuo de la población como padre, produciendo un único descendiente mediante la aplicación de un operador de mutación. Por otro lado, la estrategia de competición entre padres e hijos para formar parte de la nueva población consiste en la selección del mejor individuo entre cada padre y su descendiente. La elección de ambos procedimientos se ha llevado a cabo tras la realización de una experimentación previa, de forma que se controla y mantiene la diversidad introducida por los mutadores. A su vez, una selección determinista suele ser el método de selección más habitual en EP.

4. Experimentación

El algoritmo de programación evolutiva propuesto ha sido desarrollado en Java con el *framework* JCLEC [10]. Se han utilizado dos librerías Java públicas para el preprocesado de la información procedente de los modelos de análisis: SDMetrics Open Core [1] y Datapro4j [2].

Tabla 1. Instancias del problema y sus características

	#Clases	#Asoc	#Depen	#Agreg	#Compos	#Gener
NekoHTML	24	16	5	0	0	10
Aqualush	58	69	6	0	0	20
Datapro4j	59	4	18	1	4	50

Para realizar la experimentación del algoritmo se han considerado los modelos de análisis de tres sistemas software, cuyas características se recogen en

[1] http://www.sdmetrics.com/OpenCore.html
[2] http://www.uco.es/grupos/kdis/datapro4j

la Tabla 1. En ella se muestran el número de clases (#Clases) y el número de apariciones de cada tipo de relación UML: asociaciones (#Asoc), dependencias (#Depen), agregaciones (#Agreg), composiciones (#Compos) y generalizaciones (#Gener).

4.1. Configuración del algoritmo

En la Tabla 2 se detallan los parámetros del algoritmo y los valores asociados a cada uno de ellos, obtenidos tras haber efectuado una serie de experimentos previos. Se han realizado 30 ejecuciones con diferentes semillas aleatorias sobre los tres problemas considerados con el objetivo de obtener resultados no sesgados.

Puesto que no existen otras propuestas ante las cuales comparar, se ha realizado también una búsqueda aleatoria (RS, *Random Search*). Dada la configuración anterior, el algoritmo evolutivo genera un máximo de 100 individuos válidos en cada una de las 100 generaciones, por lo que la búsqueda aleatoria debe generar 10.000 soluciones válidas para realizar una comparación justa.

Tabla 2. Configuración de parámetros

Parámetro	**Valor**	**Parámetro**	**Valor**
Tamaño de la población	100	Prob. Mut. *Añadir componente*	0,250
Número de generaciones	100	Prob. Mut. *Eliminar componente*	0,125
Nº máx. de intentos en la mutación	10	Prob. Mut. *Dividir componente*	0,125
W_{coh}	0,300	Prob. Mut. *Unir componentes*	0,250
W_{aco}	0,400	Prob. Mut. *Mover clase*	0,250
W_{cv}	0,300	Número de componentes	Entre 2 y 8

4.2. Resultados experimentales

En la Tabla 3 se recogen los resultados medios de *fitness*, cohesión, acoplamiento y CV obtenidos por los dos algoritmos. Se puede observar cómo el algoritmo de EP consigue mejores soluciones en términos de medias para todas las instancias consideradas. Además, para el primer problema seleccionado se ha alcanzado el menor valor de acoplamiento en todas las ejecuciones.

Aunque la búsqueda aleatoria es capaz de obtener individuos que representan arquitecturas balanceadas en cuanto al tamaño de los componentes, los valores obtenidos para el resto de métricas son muy inferiores a los proporcionados por el algoritmo de EP. Por tanto, es posible alcanzar un compromiso entre cohesión y acoplamiento mediante la evolución de soluciones generadas aleatoriamente.

Más aún, el algoritmo de EP siempre encuentra soluciones con valores de cohesión positivos. Esto resalta la dificultad de generar soluciones donde el número y tipología de las relaciones entre las clases pueda contrarrestar la penalización por la presencia de varias agrupaciones.

Estos resultados también ofrecen información interesante en función de los problemas considerados. Como puede observarse, el valor de *fitness* para la arquitectura de Datapro4j es algo inferior al de los otros problemas debido a la diferencia entre los valores de acoplamiento alcanzados. Este hecho es significativo, pues este sistema presenta un número elevado de relaciones de generalización, las cuales implican la máxima penalización en el cálculo del acoplamiento.

Tabla 3. Resultados del algoritmo de EP y la búsqueda aleatoria

		NekoHTML	AquaLush	Datapro4j
EP	*Fitness*	$0,7301 \pm 0,0014$	$0,6793 \pm 0,0254$	$0,4780 \pm 0,0206$
	Cohesión	$0,1977 \pm 0,0105$	$0,1262 \pm 0,0446$	$0,0767 \pm 0,0991$
	Acopl.	$0,0000 \pm 0,0000$	$0,0600 \pm 0,0899$	$0,9868 \pm 0,2571$
	CV	$0,7806 \pm 0,1208$	$0,9222 \pm 0,0754$	$0,9868 \pm 0,2571$
RS	*Fitness*	$0,6198 \pm 0,0402$	$0,4640 \pm 0,0112$	$0,3316 \pm 0,0102$
	Cohesión	$-0,0390 \pm 0,0828$	$-0,2510 \pm 0,0401$	$0,0541 \pm 0,0980$
	Acopl.	$0,1420 \pm 0,0776$	$0,3694 \pm 0,0346$	$0,9522 \pm 0,1001$
	CV	$0,3129 \pm 0,2576$	$0,3460 \pm 0,1014$	$0,1008 \pm 0,0963$

Considerando la estructura de las soluciones proporcionadas por el algoritmo se pueden apreciar otros aspectos interesantes. En el caso del sistema Aqualush, cuya solución arquitectónica original está basada en capas, se han identificado las agrupaciones de clases que conforman algunas de estas capas. A su vez, cabe destacar la habilidad del algoritmo para generar y mantener diferentes tipos de arquitecturas en cuanto al número de componentes y conectores a lo largo de toda la evolución. Este hecho es importante, pues permite presentar al experto diferentes alternativas, entre las cuales tomará su decisión de diseño.

Finalmente, con el objetivo de comparar estadísticamente los resultados obtenidos por ambos procedimientos, se ha realizado el test de Wilcoxon bajo la hipótesis nula, H_0, de que ambos algoritmos tienen un rendimiento similar. Con un 90 % de confianza, H_0 puede ser rechazada, por lo que existen diferencias significativas entre la propuesta de EP y la búsqueda aleatoria.

5. Conclusiones y trabajo futuro

En este artículo se presenta un algoritmo de programación evolutiva aplicado al ámbito de la identificación de componentes software a partir de modelos de análisis. Se ha definido una codificación en árbol que permite manejar eficientemente las soluciones generadas por el algoritmo, a la vez que ofrece una representación comprensible para usuarios no expertos en técnicas metaheurísticas. La definición de una función de *fitness* basada en criterios de diseño y la creación de operadores genéticos específicos constituyen el resto del modelo.

La aplicación de algoritmos bio-inspirados en el dominio del análisis arquitectónico constituye a su vez una novedad, pues la mayoría de propuestas para

la optimización de arquitecturas en SBSE están orientadas a la refactorización de código o el despliegue.

En este sentido, extender la definición del problema con otros aspectos del análisis y diseño del software constituye una línea de trabajo futura. Finalmente, el algoritmo propuesto puede ser mejorado mediante la inclusión de nuevos operadores que exploren más variantes arquitectónicas. La aplicación de otras metaheurísticas, así como la hibridación de alguna de ellas con EP, constituyen otros retos a considerar en el futuro.

Agradecimientos Este trabajo ha sido apoyado por la Junta de Andalucía y el Ministerio de Ciencia y Tecnología mediante los proyectos P08-TIC-3720 y TIN2011-22408, y fondos FEDER.

Referencias

1. M. Harman, S. A. Mansouri, and Y. Zhang, "Search Based Software Engineering: Trends, Techniques and Applications," *ACM Comput. Surv.*, vol. 45, no. 1, pp. 11:1–11:61, 2012.
2. M. Harman, "Software Engineering Meets Evolutionary Computation," *Computer*, vol. 44, no. 10, pp. 31–39, 2011.
3. C. Szyperski, *Component Software: Beyond Object-Oriented Programming*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2nd ed., 2002.
4. O. Räihä, "A survey on search-based software design," *Computer Science Review*, vol. 4, no. 4, pp. 203 – 249, 2010.
5. C. L. Simons, I. C. Parmee, and R. Gwynllyw, "Interactive, Evolutionary Search in Upstream Object-Oriented Class Design," *IEEE Transactions on Software Engineering*, vol. 36, no. 6, pp. 798–816, 2010.
6. A. C. Jensen and B. H. Cheng, "On the Use of Genetic Programming for Automated Refactoring and the Introduction of Design Patterns," in *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation (GECCO '10)*, (Portland, Oregon, USA), pp. 1341–1348, ACM, 7-11 July 2010.
7. S. Kebir, A.-D. Seriai, A. Chaoui, and S. Chardigny, "Comparing and combining genetic and clustering algorithms for software component identification from object-oriented code," in *Proceedings of the Fifth International C* Conference on Computer Science and Software Engineering*, C3S2E '12, (New York, NY, USA), pp. 1–8, ACM, 2012.
8. A. Martens, D. Ardagna, H. Koziolek, R. Mirandola, and R. Reussner, "A Hybrid Approach for Multi-attribute QoS Optimisation in Component Based Software Systems," in *Proceedings of the 6th International Conference on the Quality of Software Architectures (QoSA '10)*, (Prague, Czech Republic), pp. 84–101, Springer, 23-25 June 2010.
9. S. Vathsavayi, O. Räihä, and K. Koskimies, "Using Quality Farms in Multi-objective Genetic Software Architecture Synthesis," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC '12)*, (Brisbane, Australia), pp. 1–8, IEEE, 10-15 June 2012.
10. S. Ventura, C. Romero, A. Zafra, J. A. Delgado, and C. Hervás, "JCLEC: a java framework for evolutionary computation," *Soft Comput.*, vol. 12, no. 4, pp. 381–392, 2007.

Un algoritmo voraz iterativo para el problema de la planificación sobre máquinas paralelas no relacionadas

Francisco J. Rodríguez[1], Carlos García-Martínez[2], Christian Blum[3], and Manuel Lozano[1]

[1] Departamento de Ciencias de la Computación e Inteligencia Artificial, Universidad de Granada, Granada, España,
[2] Departamento de Informática y Análisis Numérico, Universidad de Córdoba, Córdoba, España,
[3] Departamento de Ciencias de la Computación e Inteligencia Artificial, Universidad del País Vasco, San Sebastián, España,
Ikerbasque, Fundación Vasca para la Ciencia, Bilbao, España
fjrodriguez@decsai.ugr.es, cgarcia@uco.es, christian.blum@ehu.es, lozano@decsai.ugr.es

Resumen En este trabajo, tratamos el problema de la planificación de un conjunto de trabajos que se han de ejecutar en una serie de máquinas paralelas y no relacionadas de forma que se minimice el tiempo de ejecución total ponderado. El algoritmo voraz iterativo genera una secuencia de soluciones mediante la aplicación de un doble proceso de construcción y destrucción sobre soluciones obtenidas mediante una heurística. En los últimos años, los algoritmos voraces iterativos se han empleado para resolver multitud de problemas debido a que se basan en unos principios sencillos, son fáciles de implementar y muestran un rendimiento sobresaliente. Además, la escalabilidad de los algoritmos de optimización actuales se ha convertido en un requerimiento esencial. En este trabajo, proponemos un algoritmo voraz iterativo para instancias grandes del problema mencionado. A través de un estudio experimental presentamos los beneficios de nuestra propuesta con respecto al estado del arte.

Keywords: algoritmos voraces iterativos, máquinas paralelas no relacionadas, planificación, metaheurísticas

1. Introducción

El problema de la *planificación de trabajos sobre máquinas paralelas no relacionadas para minimizar el tiempo de ejecución total ponderado* (PMPnR) considera n trabajos que tienen que ser procesados en m máquinas paralelas no relacionadas. Destacar que una máquina solo puede procesar un trabajo al mismo tiempo. De esta forma, si una máquina i procesa un trabajo j, el tiempo necesario para ello es p_{ij}, siendo este valor arbitrario en el caso de las máquinas no relacionadas. Además, a cada trabajo se asocia un peso w_j. El objetivo

es planificar los trabajos de forma que la suma ponderada de los tiempos de finalización de cada máquina sea mínima:

$$\text{Minimizar} \sum_{j=1}^{n} w_j \cdot C_j,$$

donde C_j es el tiempo en el que el trabajo j termina.

Las *aplicaciones en el mundo real* de este problema son muy variadas, cubriendo campos tan diferentes como los recursos humanos [1], gestión de producción [2], instalaciones de correos [3], sistemas robotizados [4], torneos deportivos [5] y procesos químicos [6].

El problema de minimizar el tiempo total ponderado en solo dos máquinas es NP-duro [7,8]. McNaughton introdujo el problema en 1959 y desde entonces se han presentado múltiples alternativas para resolverlo. En primer lugar, podemos encontrar procedimientos exactos basados en ramificación y poda [9] y programación matemática [10]. Sin embargo, este tipo de aproximaciones presentan limitaciones para resolver instancias grandes. Por lo tanto, los *algoritmos aproximados* se presentan como la alternativa más común para afrontar este problema. Dentro de esta categoría están los algoritmos basados en *metaheurísticas*. Para el PMPnR se han propuesto algoritmos basados en búsqueda local [11], algoritmos genéticos [12], búsqueda tabú [11] y procedimientos de búsqueda adaptativa aleatorizada voraz [13].

Hoy en día, los problemas con *alta dimensionalidad* son uno de los temas más interesantes en lo que se refiere a la investigación sobre algoritmos evolutivos y metaheurísticas. Muchos de los problemas que encontramos en el mundo real presentan alta dimensionalidad. En estos casos, muchos de los algoritmos de optimización existentes ven deteriorado su rendimiento rápidamente conforme aumenta el tamaño del espacio de búsqueda. Por lo tanto, la escalabilidad de las nuevas técnicas de optimización es un requisito imprescindible. Existen trabajos en la literatura que encaran el problema de la alta dimensionalidad para otras variantes del problema estudiado en este trabajo. Sin embargo, para el problema PMPnR, los trabajos anteriores se han centrado en resolver instancias con un máximo de 200 trabajos [11,12,13].

En este trabajo, proponemos un *algoritmo voraz iterativo* (IG, por sus siglas en inglés *iterated greedy*) [14,15,16] que permite alcanzar soluciones de alta calidad especialmente para las instancias grandes, considerando escenarios con hasta 1000 trabajos y 50 máquinas. IG es una metaheurística simple y efectiva para afrontar problemas de optimización combinatoria. Destaca porque se puede implementar fácilmente y presenta un rendimiento sobresaliente; de hecho, se ha aplicado con éxito en un considerable número de problemas [17,18,19,20]. Los algoritmos IG tratan de mejorar una solución de forma iterativa, eliminando primero elementos de esa solución y completando después la solución parcial obtenida mediante un procedimiento de construcción. Además, se suele considerar un procedimiento de mejora que, mediante perturbaciones locales, trata de encontrar una solución mejor cercana a la solución obtenida después del procedimiento destructivo-constructivo.

El resto de este trabajo está estructurado de la siguiente forma. En la Sección 2, detallamos el algoritmo IG propuesto. En la Sección 3, realizamos un análisis experimental del algoritmo propuesto con el objetivo de comparar sus resultados con respecto a otras metaheurísticas para el problema PMPnR. Finalmente, en la Sección 4, recogemos las conclusiones y el trabajo futuro.

2. IG para el problema PMPnR

En la Figura 1, detallamos el esquema general del algoritmo IG propuesto. Éste trata de mejorar de forma iterativa una solución mediante 4 etapas diferentes: destrucción, construcción, mejora y criterio de aceptación. IG comienza generando una solución inicial mediante el procedimiento constructivo (Inicializar(s_0)). A continuación, se repiten de forma iterativa las 4 etapas anteriores hasta que se cumple la condición de parada. En primer lugar, se eliminan algunos elementos de la solución actual (destrucción, Destruir(s, n_d)). Después, la solución parcial se reconstruye usando de nuevo el procedimiento constructivo (construcción, Construir(s_d)). La solución obtenida es refinada mediante un proceso de mejora local (mejora, Mejorar(s_c)). Finalmente, IG decide entre reemplazar la solución actual con la obtenida después del proceso de mejora local o descartar ésta última (criterio de aceptación, Aceptar(s_p, s)).

A continuación, describimos en detalle cada una de las 4 etapas anteriores:

- Fase de destrucción. Durante esta fase se eliminan n_d componentes de la solución actual, dando lugar a una solución parcial s_d. En cada paso, el elemento a eliminar se selecciona de la siguiente forma. En primer lugar, se selecciona una máquina de forma aleatoria y, mediante un torneo binario, se elige un trabajo de esta máquina para ser eliminado. En particular, se seleccionan dos trabajos de forma aleatoria y se elimina aquel con un menor valor de p_{ij}/w_j.
- Fase de construcción. En este caso, se completa la solución parcial s_d utilizando un procedimiento voraz. Este procedimiento, en cada paso, evalúa todos los trabajos no asignados para su planificación en las distintas máquinas de acuerdo a una determinada heurística. Se realiza la asignación trabajo-máquina con mejor valor para dicha heurística. Este proceso se repite hasta que tenemos una solución completa. Existen múltiples heurísticas para evaluar estas asignaciones, en este trabajo hemos utilizado una propuesta de Li et al. [12]. En primer lugar seleccionamos un trabajo no planificado de acuerdo a la siguiente fórmula:

$$\overline{j}_* = \text{argmin}\{t_i + p_{ij}/w_j : i = 1, \ldots, m; j = 1, \ldots, n\},$$

donde t_i es el tiempo en el que comenzaría la ejecución del trabajo j en la máquina i. Una vez seleccionado el trabajo $\overline{j}_*$ que minimiza la expresión anterior, elegimos la máquina en la que se va a procesar de la siguiente forma:

$$i_* = \text{argmin}\{t_i + p_{i\overline{j}_*}/w_{\overline{j}_*} : i = 1, \ldots, m\}.$$

- Fase de mejora. El objetivo de este paso es encontrar mejores soluciones que se encuentran cerca de la solución actual en el espacio de búsqueda. Para ello se utiliza un procedimiento de búsqueda local. En este caso se utiliza la búsqueda local del primer mejor basada en movimientos de inserción de trabajos [11].
- Criterio de aceptación. En esta fase, se decide si la solución obtenida después de la fase de mejora reemplaza a la solución actual. En nuestro caso hemos adoptado un criterio de aceptación que, generando un número aleatorio uniforme p_a entre 0 y 1, acepta s_p como la nueva solución actual si $p_a \leq 0{,}5$. En caso contrario, se descarta s_p.

```
   Input: t_max, n_d
   Output: s
1  J̄ ← J;
2  s_0 ← solución vacía;
3  s ← Inicializa(s_0);
4  while no se alcance t_max do
5  |   s_d ← Destruir(s, n_d);
6  |   s_c ← Construir(s_d);
7  |   s_p ← Mejorar(s_c);
8  |   if Aceptar(s_p, s) then
9  |   |   s ← s_p;
10 |   end
11 end
```

Figura 1. Pseudocódigo del algoritmo IG propuesto

3. Estudio experimental

En esta sección, detallamos los experimentos realizados en este trabajo con el objetivo de analizar el comportamiento del algoritmo IG presentado en la sección anterior. En primer lugar, en la Sección 3.1, perfilamos el marco experimental para el estudio; a continuación, comparamos los resultados del algoritmo propuesto frente a otras metaheurísticas estado del arte para este problema (Sección 3.2).

3.1. Marco experimental

Todos los algoritmos comparados han sido implementados en C++, utilizando gcc 4.6 para compilar el código fuente. Los experimentos se han realizado en un ordenador con un procesador Intel® Core™ i7-930 a 2.8 GHz (8MB de memoria caché, 4 nucleos y 8 hilos de ejecución) con 12 GB de memoria RAM

y sistema operativo Fedora^{TM} Linux V15. Para cada ejecución de un algoritmo se utiliza un único hilo de ejecución. El tiempo máximo de ejecución (t_{max}) es n segundos, donde n es el número de trabajos de la instancia.

En este estudio, hemos considerado instancias para 20 combinaciones diferentes de número de trabajos y número de máquinas, además de 10 instancias para cada combinación. Además, hemos considerado 3 conjuntos de instancias distintos. El primer conjunto está formado por instancias denominadas no correladas [17] (UC). El segundo y tercer conjunto están formados por instancias correladas por trabajos (JC) o por máquinas (MC) [17] en las que los valores de p_{ij} presentan una correlación con el trabajo o la máquina considerada, respectivamente. En total, el marco experimental propuesto está compuesto por un total de 600 instancias diferentes.

3.2. IG frente a otras metaheurísticas para el problema PMPnR

En esta sección, comparamos los resultados de IG frente a otras metaheurísticas encontradas en la literatura para el problema PMPnR. En concreto, hemos considerado los siguientes algoritmos:

- Algoritmo iterativo multiarranque (**MultiLS**) [11].
- Búsqueda tabú (**Tabu**) [11].
- Algoritmo geńetico (**AG**) [12].
- GRASP y encadenamiento de trayectorias (**GRASP**) [13].

Para implementar todos estos algoritmos hemos utilizado C++, empleando las mismas estructuras de datos que en el caso de IG. Los valores de los parámetros para cada uno de estos algoritmos son los recomendados en las publicaciones originales. Con el objetivo de realizar una comparación justa, las condiciones de ejecución son las mismas que para IG, esto es, 1 ejecución sobre cada una de las instancias y el mismo tiempo de CPU que IG. Además, proporcionamos los resultados obtenidos por IBM-ILOG-CPLEX para resolver el modelo de programación entera cuadrática propuesto en [11] con un tiempo máximo de ejecución de 2 horas.

Caso 1: instancias UC. En la Tabla 1, vemos los resultados de todos los algoritmos sobre las instancias UC. Para cada una de las 20 combinaciones diferentes de valores de n y m, mostramos la media sobre 10 instancias diferentes. La Tabla 1 proporciona dos medidas diferentes: (1) el resultado absoluto en media y (2) el porcentaje de desviación medio (PDM), que se calcula de la siguiente forma:

$$PDM = \frac{C_{met}(i) - C_{cplex}(i)}{C_{cplex}(i)} \cdot 100,$$

donde $C_{met}(i)$ representa el valor medio obtenido por una determinada metaheurística para las instancias de tipo i y $C_{cplex}(i)$ es la mejor solución en media obtenida por CPLEX en 2 horas de ejecución. En negrita hemos resaltado el mejor resultado absoluto para cada tipo de instancia.

n	m	CPLEX	AG		GRASP		MultiLS		Tabu		IG	
		Sol.	Sol.	PDM	Sol.	PDM	Sol.	PDM	Sol.	PDM	Sol.	PDM
1000	50	*	82551	–	158604,6	–	1593174	–	1593174	–	**81542**	–
	20	372192,4	373949,1	0,47	576481	55,00	3577016	981,80	3577016	965,26	**372005,4**	-0,05
	10	1286840,3	1291135	0,33	2034004	58,16	8229562	543,43	8229562	539,73	**1286803**	-0,00
	5	4501612,8	4506558	0,11	5280587	17,31	17162120	276,16	17162120	281,35	**4501648**	0,00
500	50	24248,3	23393,3	-3,52	25653,5	5,71	151931,7	484,77	151931,7	526,62	**22854,3**	-5,74
	20	97140,7	98024,1	0,91	97082,9	-0,06	476167,7	340,77	476167,7	390,27	**96985,4**	-0,16
	10	326886	329057,8	0,66	327037,8	0,05	1232319	283,66	1232319	277,55	**326867**	-0,01
	5	1090245	1093097	0,26	1090345	0,01	3170840	190,20	3170840	191,07	**1090260**	0,00
300	50	10848,8	9910,1	-8,63	9678,1	-10,76	11844,8	27,75	11844,8	9,07	**9621,5**	-11,29
	20	36561,5	37179,7	1,693	36524,1	-0,10	38132,4	3,19	38132,4	4,31	**36489,3**	-0,19
	10	117841,1	118791,6	0,81	117856,2	0,01	120852	7,15	120852	2,55	**117822,7**	-0,02
	5	389826,5	391163,3	0,34	389854,5	0,01	444678,4	10,80	444678,4	14,04	**389831,8**	0,00
250	50	7858,5	7258,3	-7,62	7042,6	-10,37	7352,3	-7,09	7352,3	-6,46	**6998,5**	-10,92
	20	27594,4	28026,6	1,56	27577,4	-0,06	27672,7	0,28	27672,7	0,28	**27554,9**	-0,14
	10	80333,9	81287,4	1,187	80341,3	0,01	80408,2	0,09	80408,2	0,09	**80333,3**	-0,00
	5	284705,8	286048,8	0,47	284719,2	0,01	284846,6	0,08	284846,6	0,05	**284711,6**	0,00
200	50	5795,9	5138,5	-11,22	5006,6	-13,49	5062,7	-12,61	5062,7	-12,51	**4989,4**	-13,79
	20	17026,4	17345	1,87	16995,8	-0,18	17066,8	0,06	17066,8	0,24	**16983**	-0,25
	10	52918,1	53676,7	1,437	52925	0,01	53039,2	0,11	53039,2	0,23	**52919,6**	0,00
	5	182100	183400,4	0,71	182113,6	0,01	182166	0,02	182166	0,04	**182108,6**	0,00
PDM				-0,956		5,329		164,770		167,568		**-2,240**

Tabla 1. Resultados de los algoritmos comparados sobre las instancias UC.

A partir de estos resultados, podemos hacer las siguientes observaciones:

- IG obtiene el mejor valor medio global de PDM con respecto a todos los algoritmos comparados en este estudio. Además, merece la pena destacar que IG obtiene, de acuerdo al valor medio global de PDM, mejores resultados en media que CPLEX a pesar de ejecutarse durante un tiempo sensiblemente inferior.
- IG obtiene el mejor resultado en media en todas las instancias. Esto es especialmente importante cuando analizamos los resultados en las instancias con mayor número de trabajos y máquinas. De hecho, CPLEX no es capaz de encontrar ninguna solución factible para las instancias con 1000 trabajos y 50 máquinas. Por el contrario, IG obtiene en las instancias grandes soluciones de gran calidad que mejoran notablemente las obtenidas por CPLEX.
- Aparte de IG, solamente AG es capaz de mejorar (en media) los resultados de CPLEX. El rendimiento de GRASP y, sobretodo, de MultiLS y Tabu se degrada rápidamente cuando afrontan instancias con más de 300 trabajos.

Caso 2: instancias JC y MC. En este apartado, mostramos los resultados obtenidos sobre los conjuntos de instancias JC y MC. En las Tablas 2 y 3 aparecen los resultados en media sobre las 10 instancias para cada uno de los 20 tipos de problemas para las instancias JC y MC, respectivamente.

A partir de la información de las Tablas 2 y 3, podemos hacer las siguientes comentarios:

- IG alcanza los mejores resultados globales medios de PDM para ambos conjuntos. Para las instancias JC, el resultado medio absoluto es mejor que el

n	m	CPLEX	AG		GRASP		MultiLS		Tabu		IG	
		Sol.	Sol.	PDM	Sol.	PDM	Sol.	PDM	Sol.	PDM	Sol.	PDM
1000	50	*	1681082	–	1860562	–	2384658	–	2354086	–	**1676338**	–
	20	2812214,7	4020891	0,03	4397002	6,67	5690978	28,87	5658121	28,38	**3993860**	-0,72
	10	7992969,8	8041254	0,60	8696201	8,80	11322570	41,66	11366880	42,21	**8000019**	0,09
	5	16468641,0	16510720	0,26	17208400	4,49	22368410	35,82	22468270	36,43	**16470900**	0,01
500	50	455696,3	441278,1	-3,11	456530,1	0,24	503967,8	10,66	497768,5	9,29	**440223,3**	-3,34
	20	990823,3	1004954,4	1,43	1001098,5	1,04	1122845	13,33	1153897	16,46	**994813,4**	0,40
	10	1946721,1	1968328	1,11	1953564	0,35	2386413	22,59	2368174	21,64	**1950641**	0,20
	5	3960537,6	3981339	0,53	3962971	0,06	4960288	25,24	4896540	23,63	**3961860**	0,03
300	50	180707,9	179281,2	-0,78	184901	2,33	179148,7	-0,85	179130,7	-0,86	**179103**	-0,88
	20	376898,4	384228,5	1,95	383863,7	1,85	**378403,8**	0,40	378783,3	0,50	380244,1	0,89
	10	721510,6	732777,5	1,56	725082,2	0,49	728089,6	0,91	727883,3	0,88	**724190,2**	0,07
	5	1454399,9	1466592	0,84	1455911	0,10	1489475	2,41	1488205	2,33	**1455348**	0,07
250	50	133957,5	131809,1	-1,60	136706,9	2,06	130900,3	-2,28	**130869,6**	-2,30	131725,6	-1,66
	20	269330,4	275372	2,24	274606	1,96	**269692,7**	0,13	269789,6	0,17	272281,6	1,10
	10	508920,1	518192,9	1,82	511824,3	0,57	**509635,9**	0,14	**509635,9**	0,14	511335,1	0,47
	5	1017577,1	1028106	1,03	**1018244**	0,07	1019075	0,15	1018609	0,10	1018525	0,09
200	50	93210,3	91365,1	-1,97	94318	1,19	**90742,6**	-2,64	90848,7	-2,53	91397,3	-1,94
	20	179713,8	183398,7	2,05	180621,4	0,51	**179780,5**	0,04	180057,3	0,19	181828,6	1,18
	10	333166,1	340022,6	2,06	333605,4	0,13	**333502,1**	0,10	333566,5	0,12	335165,1	0,60
	5	662227,1	670125,7	1,19	**662420,4**	0,03	662448,5	0,03	662454,4	0,03	663023,5	0,12
PDM				0,59		1,73		9,30		9,31		**-0,17**

Tabla 2. Resultados de los algoritmos comparados sobre las instancias JC.

n	m	CPLEX	AG		GRASP		MultiLS		Tabu		IG	
		Sol.	Sol.	PDM	Sol.	PDM	Sol.	PDM	Sol.	PDM	Sol.	PDM
1000	50	*	1059816,4	–	1229773	–	2791200	–	2742746	–	**1016939,3**	–
	20	2427184,9	2775194	23,95	2946319	36,01	6885581	307,66	6847211	305,28	**2565055**	9,56
	10	4837163,3	5567272	15,14	5563880	15,17	14032600	191,16	14075400	192,17	**4985071**	3,01
	5	12205493,2	13417679	10,25	12664642	3,98	29683830	150,67	29829520	151,77	**12243061**	0,34
500	50	256643,2	275730,4	7,50	271501,1	5,93	430804,1	67,87	429214,9	67,47	**250819,9**	-2,24
	20	613633,6	688743,6	12,32	629461,8	0,59	945266,5	54,70	1021257,5	67,07	**617157,2**	0,58
	10	1109898,1	1260246	13,59	1112615,9	0,26	2145814	94,69	2053756	85,90	**1111212,5**	0,12
	5	2316674,1	2556277	10,84	2317241	0,03	4509104	101,77	4475050	99,97	**2316942**	0,01
300	50	155277,3	161737,5	4,17	156473,6	0,77	155540,6	0,17	**154242,6**	-0,68	155080,2	-0,12
	20	326842,2	356135,9	8,99	331031,2	1,29	331163,6	1,38	331663,8	1,48	**329766,9**	0,90
	10	540214,8	597591,7	10,70	541631,4	0,27	582245,9	8,07	573157,1	6,33	**541385,4**	0,22
	5	1045814,3	1125015,4	7,72	1046170,6	0,03	1189685,7	13,97	1235179	19,27	**1046159,3**	0,03
250	50	90587,8	94908,8	4,80	90478,5	-0,10	88473,7	-2,31	**88071,3**	-2,76	89862,1	-0,78
	20	217191,9	239618,3	10,34	219469,5	1,05	217265,2	0,04	**217262,9**	0,03	219497,1	1,06
	10	508912,6	552872,8	9,13	510777,7	0,38	509260	0,07	**509650,6**	0,16	511137,1	0,44
	5	930673,6	1003699,6	8,34	**931087,9**	0,05	960528,1	3,37	954203,7	2,85	931297,6	0,07
200	50	68725,9	72127	4,96	69643,7	1,34	**67531,2**	-1,73	67547,7	-1,70	69196,7	0,69
	20	162849	176181,6	8,23	164387,3	-0,15	**162610,3**	-0,14	162692,7	-0,09	164627,5	1,10
	10	344534,7	373137,2	8,69	344697,6	0,05	**344464,6**	-0,02	344492,1	-0,01	345944,5	0,40
	5	828483,3	862823,1	5,04	**828611,8**	0,01	828666,5	0,02	828723,1	0,03	829178,9	0,08
PDM				9,72		3,52		52,18		52,34		**0,81**

Tabla 3. Resultados de los algoritmos comparados sobre las instancias MC.

de CPLEX. Además, para las instancias de tipo MC el valor medio global de PDM es menor del 1 %.

- Sin tener en cuenta IG, GRASP y AG obtienen los mejores resultados. Este hecho pone de relieve la importancia que tiene considerar información heurística para resolver el problema. Sin embargo, MultiLS y Tabu mejoran su rendimiento significativamente con respecto a las instancias UC, pero no

son competitivos en las instancias más grandes, en las que tener información heurística acerca del espacio de búsqueda es primordial.

- Las instancias en las que IG obtiene los mejores resultados son aquellas con más de 250 trabajos. Es importante en este punto resaltar que la complejidad de una instancia aumenta a medida que se incrementa el número de máquinas. Estas instancias representan un gran reto debido a que el coste asociado a un trabajo no depende solamente de la máquina en la que está alojado si no que depende también del resto de trabajos planificados en la misma máquina. Para minimizar este coste, es necesario por lo tanto considerar la planificación de esos trabajos en otras máquinas, lo que aumenta considerablemente el tamaño del espacio de búsqueda. De hecho CPLEX encuentra problemas para resolver instancias que consideran un número grande de máquinas, hasta tal punto que no es capaz de resolver las instancias con 1000 trabajos y 50 máquinas. Por este motivo, los buenos resultados de IG en estas instancias grandes son especialmente significativos. De hecho, IG obtiene el mejor resultado absoluto medio para 11 de los 12 tipos de instancias con más de 250 trabajos.

4. Conclusiones

En este trabajo, hemos propuesto un algoritmo IG para afrontar el problema PMPnR. El estudio experimental realizado pone de relieve que el algoritmo propuesto es muy competitivo con respecto a otros algoritmos de la literatura para este problema. Particularmente, los resultados obtenidos para las instancias más grandes son especialmente relevantes. Además, es importante destacar que el comportamiento de IG ha sido estudiado sobre un número muy grande de instancias diferentes, lo que resalta que el funcionamiento del algoritmo es muy robusto.

Creemos que el algoritmo IG presentando en este trabajo representa una contribución significativa que es merecedora de más estudios. En este sentido, pretendemos explorar dos líneas de investigación. En primer lugar, incluir mecanismos de memoria en los modelos de IG que permitan ayudar a mejorar el rendimiento en instancias grandes. En segundo lugar, adaptar el modelo de IG propuesto para su aplicación en otras variantes del problema que incluyan restricciones como tiempos de liberación de los trabajos o tiempos de compromiso de los mismos.

5. Agradecimientos

Este trabajo ha sido financiado por los proyectos TIN2012-37930-C02-01/02 y P08-TIC-4173 del Ministerio de Economía y Competitividad y de la Junta de Andalucía, respectivamente.

Referencias

1. Rosenbloom, E., Goertzen, N.: Cyclic nurse scheduling. European Journal of Operational Research **31** (1987) 19–23
2. Pendharkar, P., Rodger, J.: Nonlinear programming and genetic search application for production scheduling in coal mines. Annals of Operations Research **95**(1) (2000) 251–267
3. Jarrah, A.I.Z., Bard, J.F., de Silva, A.H.: A heuristic for machine scheduling at general mail facilities. European Journal of Operational Research **63**(2) (1992) 192–206
4. Rochat, Y.: A genetic approach for solving a scheduling problem in a robotized analytical system. Journal of Heuristics **4** (1998) 245–261
5. Croce, F.D., Tadei, R., Asioli, P.: Scheduling a round robin tennis tournamentunder courts and players availability constraints. Annals of Operations Research **92** (1999) 349–361
6. Brucker, P., Hurink, J.: Solving a chemical batch scheduling problem by local search. Annals of Operations Research **96**(1) (2000) 17–38
7. Bruno, J., Coffman, E., Sethi, R.: Scheduling independent tasks to reduce mean finishing time. Communications ACM **17**(7) (1974) 382–387
8. Lenstra, J., Rinnooy-Kan, A., Brucker, P.: Complexity of machine scheduling problems. In P.L. Hammer, E.L. Johnson, B.K., Nemhauser, G., eds.: Studies in Integer Programming. Volume 1 of Annals of Discrete Mathematics. Elsevier (1977) 343–362
9. Azizoglu, M., Kirca, O.: Scheduling jobs on unrelated parallel machines to minimize regular total cost functions. IIE Transactions **31**(2) (1999) 153–159
10. Chen, Z.L., Powell, W.: Solving parallel machine scheduling problems by column generation. Informs Journal on Computing **11**(1) (January 1999) 78–94
11. Vredeveld, T., Hurkens, C.: Experimental comparison of approximation algorithms for scheduling unrelated parallel machines. Informs Journal on Computing **14**(2) (2002) 175–189
12. Lin, Y., Pfund, M., Fowler, J.: Heuristics for minimizing regular performance measures in unrelated parallel machine scheduling problems. Computers & Operations Research **38**(6) (2011) 901–916
13. Rodriguez, F., Blum, C., García-Martínez, C., Lozano, M.: GRASP with path-relinking for the non-identical parallel machine scheduling problem with minimising total weighted completion times. Annals of Operations Research **201**(1) (2012) 383–401
14. Culberson, J.C., Luo, F.: Exploring the k-colorable landscape with iterated greedy. In: Dimacs Series in Discrete Mathematics and Theoretical Computer Science, American Mathematical Society (1996) 245–284
15. Jacobs, L., Brusco, M.: A local-search heuristic for large set-covering problems. Naval Research Logistics **42** (1995) 1129–1140
16. Ruiz, R., Stützle, T.: A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. European Journal of Operational Research **177**(3) (2007) 2033–2049
17. Fanjul-Peyro, L., Ruiz, R.: Iterated greedy local search methods for unrelated parallel machine scheduling. European Journal of Operational Research **207**(1) (2010) 55–69
18. Framinan, J., Leisten, R.: A multi-objective iterated greedy search for flowshop scheduling with makespan and flowtime criteria. OR Spectrum **30** (2008) 787–804

19. Lin, S.W., Lee, Z.J., Ying, K.C., Lu, C.C.: Minimization of maximum lateness on parallel machines with sequence-dependent setup times and job release dates. Computers & Operations Research **38**(5) (2011) 809–815
20. Urlings, T., Ruiz, R., Stützle, T.: Shifting representation search for hybrid flexible flowline problems. European Journal of Operational Research **207**(2) (2010) 1086 – 1095

17-20 sept 2013
MULTICONFERENCIA
CAEPIA'13
Madrid

IV Simposio sobre Lógica Fuzzy y Soft Computing [LFSC 2013]

Editores asociados:
Luis Martínez
Edurne Barrenechea

Simposio organizado por
European Society for Fuzzy logic and Technology (EUSFLAT)

EUSFLAT
EUROPEAN SOCIETY FOR FUZZY LOGIC AND TECHNOLOGY

Prefacio

En estas actas se recogen todas las contribuciones presentadas en el IV Simposio de Lógica Fuzzy y Soft Computing (LFSC) de la European Society for Fuzzy Logic and Technology (EUSFLAT) organizado en coordinación con la MultiConferencia CAEPIA en el CEDI 2013.

El IV Simposio LFSC consolida y muestra una vez más la importancia de la comunidad fuzzy en nuestro país y su más que notable evolución en la faceta investigadora. Como puede comprobarse en estas actas del Congreso, además de contribuciones sobre los fundamentos de la Lógica Fuzzy y áreas clásicas en nuestra comunidad como el control, la decisión, etc., también podemos encontrar aplicaciones realmente novedosas y actuales de la lógica fuzzy en campos como la recuperación de información, el comercio electrónico, servicios en la web, etc.

En este libro se presentan de forma organizada las distintas comunicaciones del Simposio, de forma que podemos encontrar secciones en el mismo dedicadas a los fundamentos teóricos y técnicas básicas para el tratamiento de la incertidumbre, la adquisición y representación del conocimiento, el razonamiento aproximado. Así como también podemos encontrar secciones de aplicaciones de la lógica difusa y técnicas de soft computing a áreas como las bases de datos, la recuperación de información, comercio electrónico, servicios web, la toma de decisiones, sistemas inteligentes y control.

Con todos estos contenidos se ha desarrollado este libro de actas, que muestra el estado actual de investigación y desarrollo en la Lógica Fuzzy en España, y que esperamos satisfaga las expectativas del lector.

Es evidente que la aparición de un libro de actas como este es debido al esfuerzo y empeño de un gran número de personas. Por lo que, desde aquí expresar a la European Society for Fuzzy Logic and Technology, que nos ha confiado la organización del Simposio. A todos los autores que han respondido de forma diligente a la petición de trabajos, observando y mejorando los niveles de calidad que ha mantenido este Simposio a lo largo de sus ediciones anteriores. A los miembros del Comité científico que han seleccionado y revisado estos trabajos con rapidez y eficiencia. Finalmente y no por ello menos importante quiero felicitar y agradecer el gran trabajo realizado por el comité organizador sin cuyo esfuerzo no hubiera sido posible preparar este Simposio.

Edurne Barranechea Tartas
Luis Martínez López

Organización

Editores del Track

Edurne Barrenechea	Universidad Pública de Navarra
Macarena Espinilla	Universidad de Jaén
Luis Martínez	Univesidad de Jaén
Sebastià Massanet	Universitat de les Illes Balears
Jose A. Olivas	Universidad de Castilla la Mancha
Miguel Pagola	Universidad Pública de Navarra
Rosa Mª Rodríguez	Universidad de Jaén
J. Tinguaro Rodríguez	Universidad Complutense de Madrid

Comité de Programa

Jesús Alcalá, U. Granada
Rafael Alcalá, U. Granada
Senen Barro, U. S. Compostela
Iluminada Baturone, IMSE-CSIC
José Manuel Benítez, U. Granada
Fernando Bobillo, U. Zaragoza
Alberto Bugarín, U. S. Compostela
Humberto Bustince, U. P. Navarra
J. Manuel Cadenas, U. Murcia
Tomasa Calvo, U. Alcalá
Pablo Carmona, U. Extremadura
Jorge Casillas, U. Granada
Juan Luis Castro, U. Granada
Elena Castiñeira, U. P. Madrid
Francisco Chiclana, U. De Montfort
Oscar Cordón, U. Granada
Inés Couso, U. Oviedo
Susana Cubillo, U. P. Madrid
Mª Teresa De Pedro, CSIC
Mª José Del Jesús, U. Jaén
Miguel Delgado, U. Granada
Francesc Esteva, IIIA-CSIC
Javier Fernández, U. P. Navarra
Pere García, IIIA-CSIC
José L. García-Lapresta, U. Valladolid
Luis Garmendia, U. C. Madrid
Mª Ángeles Gil, U. Oviedo
Lluís Godo, IIIA-CSIC
Daniel Gómez, U. C. Madrid
Antonio González, U. Granada
Sergio Guadarrama, ECSC
Francisco Herrera, U. Granada
Enrique Herrera-Viedma, U. Granada
Luis Jiménez, U. Castilla-La Mancha
Mª Teresa Lamata, U. Granada
Vicente Liern, U. Valencia
Enrique López, U. León
Bonifacio Llamazares, U. Valladolid
Luis Magdalena, ECSC
María José Martín, U. Granada
Gaspar Mayor, U Illes Balears
José M. Molina, U. Carlos III Madrid
Javier Montero, U. C. Madrid
Susana Montes, U. Oviedo
Eduard Montseny, U. P. Catalunya
Claudio Moraga, ECSC
Manuel Mucientes, U. S. Compostela
Manuel Ojeda, U. Málaga
José A. Olivas, U. Castilla-La Mancha
Miguel Pagola, U. P. Navarra
José Ignacio Peláez, U. Málaga
Antonio Peregrín, U. Huelva
Raúl Pérez, U. Granada
Ana Pradera, U. Rey Juan Carlos

Jordi Recasens, U. P. Catalunya
Ignacio Rojas, U. Granada
Luciano Sánchez, U. Oviedo
Matilde Santos, U. C. Madrid
Pilar Sobrevilla, U. P. Catalunya
Vicenç Torra, CSIC
Enric Trillas, ECSC
Amparo Vila, U. Granada
Adolfo R. de Soto, U. León
Santiago Sánchez, IMSE-CSIC
Daniel Sánchez, U. Granada
Miguel Angel Sicilia, U. Alcalá
Alejandro Sobrino, U. S. Compostela
Joan Torrens, U. Illes Balears
José Luis Verdegay, U. Granada

Sponsors

- European Society for Fuzzy Logic and Technology (EUSFLAT)

BioSeq: Una librería R para el análisis de secuencias de datos

Jorge Cordero[1], Jorge Martínez[1], Oscar Sánchez[1]

Victoria López[2], Beatriz González[3]

[1] Facultad de Informática, Universidad Complutense de Madrid

jorgecsanchez@gmail.com, jorge.martinez.ladron@googlemail.com
oscarsb1988@gmail.com
[2] Dep. Arquitectura de Computadores y Automática, Universidad Complutense de Madrid

vlopez@fdi.ucm.es
[3] Dep. Estadística e Investigación Operativa, Universidad Complutense de Madrid

beatrizg@mat.ucm.es

Abstract. Este trabajo presenta una librería en lenguaje R que incluye funciones para el análisis de secuencias de datos y el acceso y tratamiento de bases de datos biológicas. El análisis de secuencias, como otros problemas en bioinformática, supone el tratamiento de miles o millones de datos biológicos en los que hay una presencia notable de incertidumbre, imprecisión e incluso omisión de información. Las técnicas del Soft Computing representan una buena alternativa con un coste computacional aceptable y proporcionan soluciones muy aproximadas a la óptima ideal. Este trabajo propone la implementación de funciones están basadas en estas técnicas, desarrolladas en lenguaje R para su libre disposición repositorio del proyecto CRAN.

Keywords. Alineamiento de secuencias, Bioinformática, Lenguaje de programación R, Incertidumbre, Soft Computing.

1 Introducción

La bioinformática es una disciplina científica multidisciplinar que utiliza la computación y las tecnologías de la información para adquirir, procesar, distribuir, analizar, interpretar e integrar información biológica y genética. En los últimos años se ha convertido en un área de investigación con gran potencial donde destaca la investigación en torno a la comparación de estructuras de proteínas, el análisis e interpretación de secuencias de nucleótidos y aminoácidos, el mapeo de cromosomas o la interpretación de expresiones genéticas. La dificultad de resolver estos problemas radica en que casi todos ellos son de tipo 'NP-Hard', es decir, problemas que son al menos tan difíciles de resolver como un problema 'NP'. En este sentido las técnicas 'Soft Computing' ofrecen enfoques alternativos que pueden obtener soluciones

eficientes basadas en heurísticas fiables trabajando en grandes espacios de búsqueda y manejando datos incompletos y con incertidumbre [1].

Por otra parte el proyecto CRAN [2] ofrece un conjunto de librerías y funcionalidades muy adecuadas al análisis estadístico y modelado matemático de problemas entre los cuales se encuentran los que se abordan en este artículo. CRAN proporciona el lenguaje de programación R de paradigma adecuado a nuestros propósitos y con una interfaz basada en cálculo simbólico que facilita su uso por parte de cualquier tipo de investigador, aunque no tenga conocimientos especializados en programación.

Este artículo está focalizado en la creación de una librería para uso docente orientada al tratamiento de bases de datos donde usuarios, profesores e investigadores puedan utilizarla para el análisis de tablas junto con su posterior aplicación de técnicas fuzzy. Se encuentra organizado de la siguiente forma. La sección 2 está dedicada a la aplicación del Soft Computing en Bioinformática. La sección 3 está dedicada al problema principal que nos ocupa en este artículo: el alineamiento de secuencias y su tratamiento con técnicas Soft Computing. La sección 4 está dedicada al tratamiento y gestión de bases de datos relacionadas. En la sección 5 se justifica la creación de la librería BioSeq en el proyecto CRAN, presentando detalles de estructura y mostrando ejemplos. Para finalizar, las conclusiones y el trabajo futuro se describen en la sección 6.

2 Soft Computing en Bioinformática

En bioinformática uno de los problemas más estudiados es la precisión y la incertidumbre de los procesos, ya que al trabajar con miles o millones de datos, los procesos adquieren un coste que puede hacer fracasar los proyectos. Por ello las técnicas de Soft Computing son adecuadas para tratar de solucionar problemas utilizando información incompleta, imprecisa o con incertidumbre. Estas técnicas son apropiadas para el manejo de información ambigua, como la que utilizamos día a día para tomar decisiones. Además, ofrecen soluciones robustas y de bajo coste computacional, similares a las que se llegaría aplicando el razonamiento humano. La falta de datos y la presencia de ruido son una característica común de los datos biológicos. Estas técnicas permiten también gestionar grandes cantidades de datos y ayudan a descubrir las relaciones ocultas y las correlaciones que existen entre ellos. A diferencia de los algoritmos y las técnicas de la informática tradicional, el Soft Computing permite una adaptación sencilla a un entorno donde se producen muchos cambios. En definitiva, el uso de estas técnicas permite extraer conclusiones a partir de la información disponible, de la información oculta en los datos y de los procesos de toma de decisiones, de manera que se puedan diseñar un conjunto de reglas que produzca un comportamiento similar al de un experto para diagnosticar una enfermedad [3].

La lógica difusa, desarrollada por Zadeh [4], es una técnica muy utilizada en Ingeniería de Control. Se trata de una formalización del razonamiento aproximado que imita la lógica humana de toma de decisiones. La idea clave es que utiliza una forma simple para calcular las salidas a partir de las entradas utilizando sentencias del tipo ‘If-Then’. Los más interesante de la lógica difusa es que admite incertidumbre en

los antecedentes de la sentencia 'If' y produce consecuentes precisos o difusos [1,3]. Entre las aplicaciones de la lógica difusa a la bioinformática se incluyen el alineamiento de secuencias basado el remodelado difuso de un algoritmo de programación dinámica, la segmentación de genes utilizando microarrays o el análisis de expresiones genéticas [5].

Por otra parte, las redes neuronales artificiales (ANN) son un modelo de aprendizaje y procesamiento automático capaz de capturar y representar relaciones complejas entre un conjunto de entradas y salidas. Una red neuronal es un sistema formado por una o más capas de neuronas interconectadas que colaboran entre sí. Cada neurona recibe un conjunto de entradas a través de sus interconexiones y emite una salida. La salida está definida por una función de propagación, una función de activación y una función de transferencia. Una red neuronal tiene la capacidad de aprender cómo realizar una tarea basándose en los datos de entrenamiento. Además, posee una elevada tolerancia a fallos y puede realizar cálculos intensivos en tiempo real, manejando datos difusos, con ruido o inconsistencias. Entre las aplicaciones de las redes neuronales a la bioinformática se incluyen la predicción de la estructura de proteínas, la clasificación de secuencias de proteínas o el análisis de patrones de expresiones genéticas [5].

Los algoritmos genéticos (GAs) son algoritmos de búsqueda y optimización que se basan en mecanismos de selección natural y genética. Son eficientes, adaptativos y robustos que producen soluciones que se aproximan mucho a la solución óptima. Los algoritmos genéticos utilizan secuencias de caracteres para representar los cromosomas de los organismos, un mecanismo de supervivencia para seleccionar los organismos con mayor capacidad de adaptación y aplican un criterio aleatorio para controlar la reproducción y evolución de la población. Entre las aplicaciones de los algoritmos genéticos a la Bioinformática se incluyen el alineamiento y la comparación de secuencias de ADN, ARN y de proteínas, la predicción y la segmentación de estructuras de proteínas o la interpretación de expresiones genéticas [5].

3 Alineamiento de secuencias

En Bioinformática, el alineamiento de secuencias se utiliza para buscar, comparar y alinear secuencias por pares o múltiples. Los métodos de alineamiento por pares, que son los utilizados en la librería, se utilizan para identificar la mayor similitud entre subsecuencias o cadenas completas, mientras que los métodos múltiples son usados con más de dos secuencias. Los métodos de alineamiento por pares, tanto el global como el local, son fundamentalmente similares y sólo difieren en la estrategia de optimización utilizada en el alineamiento de residuos similares. Ambos tipos pueden basarse en algunos de los siguientes tres métodos: El método de la matriz de puntos, el método de programación dinámica y el método por palabra. Existe además un tipo de alineamiento de secuencias denominado estructural, y que se basa en la búsqueda de similitudes en las formas y estructuras tanto secundarias como terciarias de proteínas y ARN.

Tipos de alineamientos

El alineamiento de secuencias a nivel local, es utilizado en bioinformática para la búsqueda de subsecuencias que contengan un alto grado de similitud. Se aplica tanto en cadenas de nucleótidos de ADN y ARN, como en cadenas de aminoácidos de las proteínas. Al algoritmo más utilizado para realizar los alineamientos locales es el de Smith-Waterman [6], y está basado en el algoritmo de programación dinámica Needleman-Wunsch [7]. Su principal diferencia, es que el algoritmo Smith-Waterman se aplica a secuencias muy diferenciadas entre sí.

El alineamiento de secuencias a nivel global, es utilizado en bioinformática para la búsqueda de la similitud óptima de secuencias completas, aplicándose a cadenas similares y de tamaño parecido. Se aplica tanto en cadenas de nucleótidos de ADN y ARN, como en cadenas de aminoácidos de las proteínas. Al algoritmo utilizado para realizar los alineamientos globales es el Needleman-Wunsch, y está basado en la programación dinámica y en el principio de optimalidad de Bellman. Su principal diferencia respecto al algoritmo de Smith-Waterman, es que se aplica a secuencias completas y muy similares entre sí.

El alineamiento híbrido surge con la necesidad de solapar parte de una secuencia con la otra. En este caso, se superponen el inicio de una con el final de la otra. Estos métodos semiglobales o glocales, se aplican a secuencias muy diferenciadas utilizando una variante del algoritmo Smith-Waterman, cuya característica primordial es la ausencia de penalizaciones en las zonas superpuestas de las secuencias [8].

Técnicas de modelado

La programación dinámica está implementada dentro de una serie de herramientas para la búsqueda y alineamiento de secuencias donde se contemplan algunas técnicas del Soft Computing, como pueden ser las Redes Bayesianas en los modelos estadísticos de Markov o la sensibilidad y la especificidad probabilística. A continuación se enumeran herramientas relacionadas e implementadas siguiendo alguna técnica soft.

- FASTA es un paquete de alineamiento de secuencias de nucleótidos y aminoácidos que cuenta con herramientas de cálculo estadístico de las similitudes para analizar el azar o la homología existente dentro de un alineamiento [9].
- BLAST es una herramienta que utiliza el algoritmo Smith-Waterman para los alineamientos de secuencia [10].
- Clustal es una aplicación que permite realizar alineamientos globales de nucleótidos y aminoácidos que implementa un algoritmo heurístico progresivo para calcular múltiples alineamientos [11].
- SSEARCH es una implementación de búsqueda basado en el algoritmo de Smith-Waterman. En comparación con FASTA se le considera más lento, pero tiene mayor sensibilidad [12,13].

- HMMER es una herramienta para identificar homologías entre secuencias de nucleótidos y aminoácidos que mejora significativamente la aceleración del algoritmo Smith-Waterman para el alineamiento de dos secuencias [14].
- SAM realiza una búsqueda local y global utilizando el modelo estadístico oculto de Márkov [15].

Además de la Programación Dinámica clásica, en la literatura se pueden encontrar referencias a la programación dinámica difusa, planteada como una extensión del Principio de Optimalidad de Bellman al ámbito difuso donde se tiene en cuenta para ello, el Principio de Optimalidad de Bellman-Zadeh [16]. En el ámbito difuso, este principio es utilizado en problemas donde están presentes mapas de estados difusos con limitaciones y objetivos borrosos [17]. En este ambiente difuso, también podemos considerar técnicas de búsqueda por muestreo donde se utilizan métodos como el de Monte Carlo para solventar problemas de programación dinámica donde los parámetros y variables pueden ser difusas [18].

4 Gestión y pruebas sobre bases de datos biológicas

En Internet podemos acceder a gran cantidad de bases de datos biológicas de organizaciones internacionales. Estas organizaciones proveen aplicaciones web mediante las cuales podemos buscar información, homologías, secuencias iguales y patrones sobre la información tratada. También pueden ejecutarse algoritmos de predicción y de alineamiento de secuencias entre otros. Dichas bases de datos están almacenadas en servidores especializados. Para el manejo de dichos datos, el lenguaje R [2] y sus librerías son muy adecuados. En primer lugar porque R es de uso gratuito y libre, teniendo como única condición respetar los términos de licencia de GNU General Public License vs 2. [19].

Este lenguaje de programación interactivo permite crear funciones propias y reutilizar código programado por otros usuarios. Así, de manera libre se consigue mejorar la productividad y poder acortar los tiempos de desarrollo. R tiene más de dos millones de usuarios activos compartiendo paquetes y datos, lo que hace que cada vez tenga más peso en el ámbito de la investigación y más concretamente en el de la Bioinformática.

Para realizar pruebas sobre este trabajo, se han creado tres tablas distintas en el paquete BioSeq, de las cuales nos interesa especialmente la tabla 'Cabras'. Dicha tabla contiene información de 19 medidas físicas de un total de 531 cabras, la tabla ha sido cedida por la Facultad de Veterinaria de la UCM y en dicha tabla existen datos perdidos, valores NA, que deberán ser estimados. La primera propuesta de estimación se ha realizado mediante correlaciones entre las variables.

Para encontrar correlaciones más allá de la correlación conocida entre el peso y el perímetro torácico [20], hemos propuesto utilizar lógica difusa. La intención no es encontrar una correlación exacta entre una característica u otra, sino que el hecho de que una característica sea considerada pequeña, o mediana o grande, determine el hecho de que otra lo sea, si bien es cierto que los resultados obtenidos no son precisamente positivos creemos que cambiando el enfoque podemos encontrar resultados válidos. Lo primero que hacemos es seleccionar un grupo de referencia lo

más grande posible, para así evitar coincidencias puntuales y que las conclusiones a las que lleguemos sean lo más genéricas posibles y poder refutarlas con el resto de individuos en el caso de llegar a alguna conclusión, si bien es cierto que ciertas propiedades valdrán para casos concretos, esto es que es común que una ecuación calcule perfectamente el valor de una variable en hembras, pero no en machos.

Para ello se ha clasificado el grupo de cabras por sexos y posteriormente por edad, individuos andoscos de 2 a 3 años, trasandoscos de 3 a 4 años y cerrados de 4 años en adelante. El grupo que tenemos con mayor número de representantes son las hembras cerradas con 226 individuos. Una vez seleccionados vemos las correlaciones entre variables, que serán los resultados base y por lo tanto tendremos que tenerlos muy en cuenta a la hora de analizar los resultados obtenidos. Los resultados base los mostramos en la Fig. 1 y se han obtenido utilizando la librería *corrgram* [2].

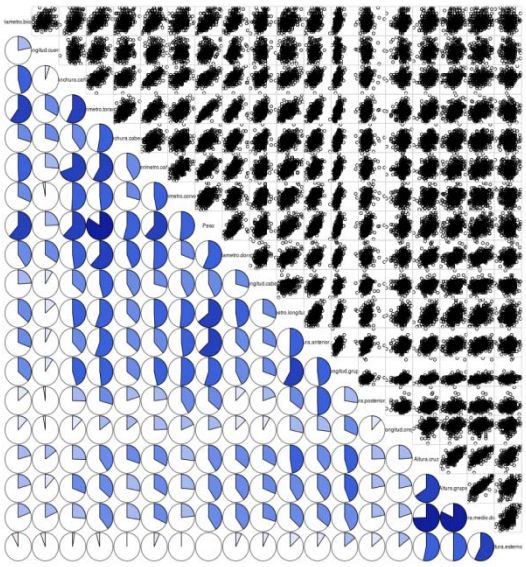


Fig. 1. Correlación de variables en hembras cerradas.

Podemos ver que hay dos correlaciones razonablemente buenas que son la del peso con perímetro torácico, que es la que hemos mencionado anteriormente, y la altura de la grupa con la altura del medio dorso.

Ahora bien, nuestra propuesta es ver si es factible llegar a conclusiones de correlaciones usando lógica difusa, esto es, en vez de tener los datos exactos tener su tamaño relativo, así podemos ver correlaciones de la forma si la variable X es pequeña tenemos que las variables Y y Z también lo son. Por lo tanto vamos a ver ahora las correlaciones conseguidas haciendo uso de lógica difusa. Como queremos hallar un método escalable y que no sea sensible al número de medidas tomadas, ya que en nuestro caso al tener 19 variables perderíamos generalidad si hubiera que realizar la fuzzificación para cada variable, hemos decidido utilizar fuzzy C-Means Clustering, para lo cual utilizaremos la librería *e1071* [22]. Dicho algoritmo devuelve

el grado de pertenencia a cada grupo, o cluster. Vamos a aplicar dicho algoritmo so-sobre cada variable para hallar la pertenencia a los grupos de manera local, en vez sobre la totalidad de la tabla, donde los resultados obtenidos no serían muy esclarecedores.

Después calculamos un número derivado de esos datos. Para lo cual una vez tenemos los datos de pertenencia a cada clúster, los multiplicaremos por unos pesos, dichos pesos se otorgan en función del center de cada grupo, de manera que de menor a mayor center asignaremos números naturales del 1 en adelante, en el caso de haber 5 grupos tendremos 5 pesos (1, 2, 3, 4, 5). Dichas multiplicaciones las sumaremos y tendremos un número derivado del grado de pertenencia. Los resultados de la correlación usando este resultado los podemos ver en la Fig. 2.

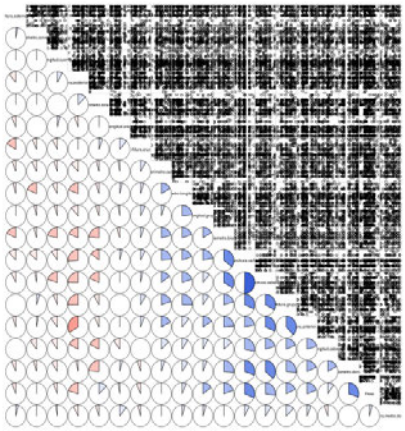


Fig. 2. Correlación aplicando cmeans y utilizando 5 clusters.

La Fig. 2 muestra nubes de puntos bastante difusas, lo cual es adverso a lo esperado. La causa es el algoritmo Fuzzy C-Means ya que, aunque un dato sea extremadamente alto se incluirá en algún grupo en mayor medida, probablemente en el grupo del mayor *center* y con una probabilidad de pertenencia cercana al 1, y en dicho grupo habrá otros datos con un alto porcentaje de pertenencia que lógicamente deberían distar más de ese valor extremadamente alto. A esto hay que sumar algo que sabíamos de antemano y es que los centers iniciales son seleccionados aleatoriamente, por lo que los resultados pueden variar entre ejecuciones distintas y con ello las correlaciones. Aun así podemos apreciar cómo en el caso de la anchura de la cabeza y de la anchura de la caña la correlación es ligeramente mejor que en el caso base. Por lo tanto creemos que Fuzzy C-Means no es el mejor algoritmo en este caso y que habría que fuzzificar los datos de manera directa para comprobar si el hecho de haber conseguido una mejora es un caso puntual o es una alternativa viable para encontrar correlaciones. Para ello, habrá que definir para cada variable un tamaño pequeño, mediano y grande, con estas tres funciones de pertenencia podría darnos buenos resultados, o por lo menos darnos un primer acercamiento para saber si es viable o no.

5 La librería BioSeq

En el ámbito de la investigación en Bioinformática se plantean problemas específicos como la comparación y el alineamiento de secuencias, la comparación de estructuras de proteínas, el análisis de secuencias de nucleótidos y aminoácidos, el mapeo de cromosomas, la interpretación de expresiones genéticas, la construcción de árboles filogenéticos o el análisis de perfiles de expresión génica en microarrays. Entre todos estos problemas, el análisis de secuencias y la comparación de estructuras de proteínas son de especial relevancia.

La librería BioSeq tiene como objetivo aportar técnicas y herramientas orientadas a resolver problemas específicos de esta disciplina, como la comparación y el alineamiento de secuencias y la gestión de bases de datos biológicas. El ámbito de aplicación de esta librería es fundamentalmente académico. Esta primera versión de la librería implementa el algoritmo de Smith-Waterman para alineamiento local y el de Needleman-Wunsch para alineamiento global en el lenguaje R. El objetivo es implementar nuevos algoritmos para ampliar la funcionalidad y las bases de datos de la librería en futuras versiones.

BioSeq ofrece tres bases de datos: Diesel, Gasolina95 y Cabras. Las dos primeras almacenan lecturas de consumos de combustible y la última recoge datos morfológicos de 531 cabras. La base de datos Diesel almacena consumos de combustible desde el año 2003 hasta 2012, incluye los litros repostados, el coste en euros y el kilometraje realizado. Gasolina95 almacena consumos de combustible desde el año 2008 hasta 2012, incluye los litros repostados, el coste en euros y el kilometraje realizado. Por último, la base de datos Cabras almacena datos morfológicos de un rebaño de cabras compuesto por 531 machos y hembras con edades que van desde los 2 hasta los 4 años, clasificados en las categorías Andosco (intervalo [2, 3) años), Trasandosco (intervalo [3, 4) años) y Cerrado (para edades mayores o iguales a 4 años). Estos datos se tomaron en el año 1997 de una población de cabras del Guadarrama compuesta por 7236 hembras y 204 machos. Esta información fue elaborada por profesores del Departamento de Producción Animal de la Facultad de Veterinaria de la Universidad complutense de Madrid. Respecto al alineamiento de secuencias, se dispone de varias funciones implementadas, entre las que destacan las funciones SmithWaterman y NeedelmanWunsch, cuya especificación se describe a continuación.

SmithWaterman

Declaración SmithWaterman(seq1, seq2, gap, MSHeader, MSData, MSCol, MSRow)

Argumentos: seq1: vector de la primera secuencia; seq: vector de la segunda secuencia; gap: penalización por Gap; MSHeader: vector de la cabecera de la matriz de puntuación a utilizar; MSData: vector de los valores de la matriz de puntuación, rellenados por filas; MSCol: número de columnas de la matriz de puntuación; MSRow: número de filas de la matriz de puntuación.

Descripción: Alineamiento local de dos secuencias utilizando el algoritmo Smith-Waterman.

Ejemplo de uso

```
SmithWaterman(c("A","C","A","C","A","C","T","A"),
c("A","G","C","A","C","A","C","A"),  -5, c("A","G","C","T"), c(10,-1,-3,-4,-1,7,-5,-3,-3,-5,9,0,-4,-3,0,8),  4, 4)
```

Resultados

```
[1] "A" "-" "C" "A" "C" "A" "C" "T" "A"
[1] "A" "G" "C" "A" "C" "A" "C" "-" " Nombre A"
```

NeedlemanWunsch

Declaración NeedlemanWunsch(seq1, seq2, gap, MSHeader, MSData, MSCol, MSRow)

Argumentos: seq1: vector de la primera secuencia; seq: vector de la segunda secuencia; gap: penalización por Gap; MSHeader: vector de la cabecera de la matriz de puntuación a utilizar; MSData: vector de los valores de la matriz de puntuación, rellenados por filas; MSCol: número de columnas de la matriz de puntuación; MSRow: número de filas de la matriz de puntuación.

Descripción: Alineamiento global de secuencias utilizando el algoritmo Needleman-Wunsch.

Ejemplo de uso

```
NeedlemanWunsch(c("A","C","A","C","A","C","T","A"),
c("A","G","C","A","C","A","C","A"),  -5,  c("A","G","C","T"),  c(10,-1,-3,-4,-1,7,-5,-3,-3,-5,9,0,-4,-3,0,8),  4,  4)
```

Resultados

```
[1] "A" "-" "C" "A" "C" "A" "C" "T" "A"
[1] "A" "G" "C" "A" "C" "A" "C" "-" "A"
```

6 Conclusiones

En este artículo se presenta la versión preliminar de la librería BioSeq para el proyecto CRAN en lenguaje R. Además, se analiza la utilidad de las técnicas de Soft Computing en el análisis de secuencias y se propone la creación de un conjunto de funciones relacionadas. La librería incluye una pequeña colección de funciones relativas al tratamiento de datos y al análisis de secuencias mediante operaciones clásicas que abordan ya ciertos grados de incertidumbre y tratamiento de información incompleta en los procesos. Las funciones principales se muestran en el artículo. Para las primeras pruebas se han creado tres bases de datos reales, una de ellas de formato adaptado mediante fuzzificación de variables en el formato FASTA y lista para aplicar a los algoritmos desarrollados sobre alineamientos. Como resultado se ha obtenido una vía para incorporar funciones de utilidad a la librería BioSeq que puede utilizarse para la enseñanza tanto de creación de librerías en R como del problema de alineamiento de secuencias con resolución de incertidumbre. Además la ejecución de

estas funciones sobre bases de datos preparadas y la visualización de los resultados en R, permite mejorar las decisiones sobre aceptación o rechazo de técnicas y algoritmos adecuados para un problema concreto.

Referencias

1. Prithivi, P. and Vengadesan, G. P.: Soft Computing Methodologies in Bioinformatics. National Conference on Future Computing. Volume 1, 276-279. (2012)
2. www.cran.r-project.org
3. Mitra, S. and Hayashi, Y.: Bioinformatics with Soft Computing. IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, Vol. 36, No. 5, 616-635 (2006)
4. Zadeh, L. : Fuzzy sets. Information and Control, Vol. 8. 338-353, (1965)
5. Jena, R. K., Aqel, M. M. Srivastava, P. and Mahanti, P. K.: Soft Computing Methodologies in Bioinformatics. European Journal of Scientific Research. Vol.26. No.2, 189-203 (2009).
6. Smith, T.F., Waterman, M.S.: Identification of common molecular subsequences. J Mol Biol. 147 (1) 195-197. (1981)
7. Needleman, S.B. and Wunsch, C.D.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of molecular biology (Elsevier) 48 (3) 443-453 (1970)
8. Brudno, M., Malde, S., Poliakov, A., Do, C.B., Couronne, O., Dubchak, I., Batzoglou, S.: Bioinformatics. 19. Suppl 1 (90001): i54–62. (2003)
9. Lipman, D. J., Pearson, W. R.: Rapid and sensitive protein similarity searches. Science 227 (4693): 1435–41. (1985)
10. Altschul, S. F., Gish, W., Miller, W., Myers, E.W., Lipman, D. J.: Basic local alignment search tool. Journal of Molecular Biology 215 (3): 403–10 (1990)
11. Chenna, R., Sugawara, H., Koike, T., Lopez, R., Gibson, T.J, Higgins, D.G., Thompson, J.D.: Multiple sequence alignment with the Clustal series of programs. Nucleic Acids Res 31 (13) 3497–3500 (2003)
12. Pearson, W.R., Lipman, D.J.: Improved tools for biological sequence comparison. Proc. Natl. Acad. Sci. U S A. 85(8) 2444-2448. (1988)
13. Lipman, D.J., Pearson, W.R.: Rapid and sensitive protein similarity searches. Science. 227(4693) 1435-41 (1985)
14. Durbin, Richard; Eddy, Sean R.; Krogh, Anders et al., eds. (1998). Biological sequence analysis: probalistic models of proteins and nucleic acids. Cambridge, UK: Cambridge University Press.
15. Hughey, R., Karplus, K,. Krogh, A.: SAM: sequence alignment and modeling software system. Technical report UCSC-CRL-99-11 (2003)
16. Bellman, R. and Zadeh, L. Decision-Making in a Fuzzy Environment. Management Science, 17: pp. 141-164. (1970)
17. Journal of mathematical analysis and applications 85, l-23, 1-2. (1982)
18. Buckley, J. J., Jowers, L.J.: Monte Carlo Methods in Fuzzy Optimization Series: Studies in Fuzziness and Soft Computing, Vol. 222: pp. 227.(2008)
19. http://www.gnu.org/licenses/gpl-2.0.html
20. Ignacio García Lara, M. A. Vázquez Ferreño, E. Fernández Calviño, L. Vidal Galego, Mª Teresa de Jesús García Lara. “Ecuaciones de predicción del peso vivo de hembras holstein”

Modelado de Ludus Manager con Lógica Fuzzy

Antonio Dominguez Goñi[1], Juan Íñigo Monjas[1], Miguel Morata Moralejo[1], Victoria López[2], Rubén Fuentes-Fernández[3]

[1] Facultad de Informática, Universidad Complutense de Madrid
Profesor García Santesmases, s/n,
28040 Madrid, Spain

{tutxo7, elpadrino214, mmoratam}@gmail.com
[2] Dep. Arquitectura de Computadores, Universidad Complutense de Madrid
Profesor García Santesmases, s/n,
28040 Madrid, Spain

vlopez@fdi.ucm.es
[3] Dep. Ingeniería del Software e Inteligencia Artificial, Universidad Complutense de Madrid
Profesor García Santesmases, s/n,
28040 Madrid, Spain

ruben@sip.ucm.es

Abstract. Ludus Manager (LM) es un juego de tipo competitivo jugador-máquina para dispositivos móviles desarrollado en el sistema operativo Android. El jugador simula una escuela virtual de gladiadores que evoluciona proporcionándole nuevos retos y experiencias. La aplicación LM cuenta con una interfaz gráfica sencilla de manejar cuyo uso intensivo no repercute desfavorablemente en el rendimiento de la aplicación. Para mejorar la interacción del usuario con la máquina, LM ejecuta un sistema de inferencia basado en lógica Fuzzy que permite calcular el valor de los gladiadores mediante la agregación de ciertas variables que definen las características de cada escuela. El sistema permite entrenar a los gladiadores mediante decisiones tomadas en tiempo real y competir con la escuela de gladiadores de la máquina y evolucionar de acuerdo a variables randomizadas y lógica propia de entrenamiento.

Keywords: Juegos competitivos, Lógica Fuzzy, Mamdani, Agregación, Inferencia, Android.

1 Introducción

Los juegos en smarthphones tienen un recorrido similar a sus antecesores en ordenadores personales, con las diferencias de la evolución de las tecnologías y su repercusión en la sociedad actual. El pequeño tamaño de estos dispositivos y su conexión a Internet, ha permitido la evolución rápida de aplicaciones que los usuarios se descargan y utilizan en su rutina diaria [2]. Dentro de las aplicaciones para smartphones, los juegos competitivos son muy destacables por su popularidad y por

tener un rol protagonista en sistemas de aprendizaje [11-14]. En este artículo se describe el juego competitivo Ludus Manager (LM).

Tras la configuración inicial, el jugador de LM compra una escuela de gladiadores básica con unos recursos económicos fijos iniciales. A continuación, el jugador gestiona la nueva escuela de forma estratégica. Para ello el juego contempla la competición con otras escuelas en un escenario virtual. Esta competición requiere que el jugador, jugando el rol de entrenador del Ludus, planifique el entrenamiento de cada uno de los luchadores. Esto se hace en base al análisis de las características y habilidades de los luchadores, y es decisivo a la hora de que la escuela gane prestigio y competiciones dentro del universo virtual de la aplicación. El jugador también tendrá la posibilidad de ir mejorando las instalaciones de su escuela (hasta un límite definido por el nivel de la ciudad donde se encuentre), para así tener acceso a mejores tipos de entrenamientos, forjar equipamiento mejorado y más variado, o poder alojar más gladiadores y empleados. La aplicación contempla incluso la posibilidad de vender escuelas y comprar otras mejor ubicadas. El progreso del videojuego en modo offline se ejecuta con simulación semanal, lo que permite al jugador observar la evolución de su escuela y facilita la toma de decisiones.

Los combates entre gladiadores se ejecutan de manera automática y aleatoria, mediante una serie de algoritmos que calculan el resultado de cada asalto (unos 5 segundos de combate) teniendo en cuenta un número considerable de variables involucradas. Estas variables sirven para definir las características y habilidades de cada gladiador, su reputación, su equipamiento, sus estilos de combate e incluso la 'tendencia' que adquiere el combate.

Este artículo se estructura de la siguiente forma: La sección 2 describe la justificación de la lógica fuzzy como técnica para el cálculo de la variable valor. En la sección 3 se analiza el modelado del problema haciendo especial énfasis en el cálculo del valor de un gladiador. En esta sección también se incluyen detalles de la implementación realizada. En la sección 4 se muestran los resultados obtenidos comparando para tres personajes con distinto grado de entrenamiento. Finalmente, la sección 5 resume y concluye el trabajo realizado.

2 Lógica difusa

En este artículo se ha escogido esta técnica por la gran utilidad que aporta en sistemas de decisión, resolución y comprensión de datos. La lógica fuzzy es capaz de imitar la forma en la que tomamos decisiones los humanos. Para justificar el uso de esta técnica proponemos el siguiente ejemplo donde se muestra la pérdida de información relevante que ocurre al aplicar lógica convencional en la evaluación de variables cualitativas (e incluso cuantitativas). El ejemplo lo vemos sobre la evaluación de la calidad de un jugador de baloncesto. En la lógica binaria convencional, podríamos desarrollar un sistema que analizase características del jugador de baloncesto, y nos devuelva la calidad de ese jugador mediante un valor numérico comprendido entre 0 y 10. Para ello se puede diseñar un conjunto de reglas como se muestra a continuación:

- Si un jugador mide más de 210 cm, asignar valor 2,

- Si la estatura está comprendida entre 195 cm y 210 cm, asignar valor 1,
- En otro caso, asignar valor 0.
- Si la edad de un jugador está comprendida entre 22 y 28 asignar valor 2,
- Si está comprendida en los intervalos 18-21 y 29-32, asignar valor 1,
- En otro caso asignar valor 0.
- Si el porcentaje de acierto en tiros de 3 puntos es superior al 50% , asignar 2,
- Si es inferior al 20%, asignar 0,
- En cualquier otro caso asignar 1.
- Si el porcentaje de acierto en tiros de 2 puntos es superior al 50% , asignar 2,
- Si es inferior al 20%, asignar 0,
- En cualquier otro caso asignar 1.
- Si el porcentaje de acierto en tiros de 1 puntos es superior al 50% , asignar 2,
- Si es inferior al 20%, asignar 0,
- En cualquier otro caso asignar 1.

El sistema utiliza estas reglas para realizar la comparación entre dos jugadores como se muestra en el siguiente ejemplo. Supongamos los datos de entrada sobre dos jugadores mostrados en la Tabla 1.

Tabla 1. Ejemplo datos jugadores

Jugador	Estatura	Edad	% tiros 3p	% tiros 2p	% tiros 1p
1	194cm	21	48	80	100
2	195cm	28	5	50	60

Tras aplicar las reglas anteriormente descritas a los datos se obtiene una valoración de 6 y 7 puntos respectivamente. Sin embargo, parece evidente que el primer jugador es mejor. El problema radica en que el sistema no es capaz de entender que la diferencia entre alturas es prácticamente irrelevante, que las edades de ambos jugadores están próximas al siguiente intervalo, el porcentaje de tiros de 3 del segundo jugador es desastroso, o los porcentajes del primer jugador son dignos de récord. Para solucionar estas incongruencias, la lógica difusa tiene en consideración y comprende los cuantificadores que usamos los humanos en nuestro lenguaje, por tanto el primer jugador tendría un porcentaje de tiros libres absolutamente perfecto, y el segundo jugador tendría un porcentaje de tiros libres evaluable en término medio.

El proyecto LM se ha servido de la lógica difusa para cualificar a los gladiadores según una serie de variables, tales como sus características físicas, su habilidad en los diferentes estilos de combate, o su fama y porcentaje de victorias. En el siguiente apartado se explican los detalles de la implementación, con un sistema propio de valoración, con el que modelamos la inteligencia del sistema a la hora de tomar decisiones sobre la compra y la venta de gladiadores en Ludus Manager.

3 Modelado e implementación

En esta sección se muestra cómo se ha desarrollado parte de la inteligencia artificial de la máquina, las librerías utilizadas para la parte gráfica y los algoritmos desarrollados para las distintas funcionalidades de la aplicación de forma que sean lo más reales

posibles. Para ello se han utilizado herramientas basadas en Soft Computing [3-7]. Lo más destacado de este apartado es el uso de lógica difusa para recalcular el precio de un gladiador según sus características en el momento de la compra-venta. La variable 'precio del gladiador' es fundamental para el correcto funcionamiento del juego, ya que se define en términos de casi todas las restantes incluyendo cantidades y cualidades. Para su cálculo se ha utilizado el método de inferencia de Mamdani [10,14] con el que se ha realizado la agregación de la información de las variables involucradas para definir el precio. La justificación de la lógica difusa en este tipo de aplicaciones ya ha sido discutida en la literatura [9,11]. Este método de inferencia representa un modelo suficiente y adecuado al problema a tratar gracias a su sencillo sistema de cálculo basado en operaciones básicas y agregación por máximos. Otro sistema alternativo analizado es el de la lógica de Sugeno [16]; sin embargo, como método preliminar en el videojuego resulta más complejo en cuanto a operaciones. La agregación de la información mediante el uso de operadores ponderados y ordenados tipo OWA [17], ha sido analizada y desechada en esta primera versión debido a la necesidad de ponderar de acuerdo con criterios que deben ser extraídos tras el análisis de comportamiento y que por ello deben relegarse a una segunda fase. El esquema desarrollado en este proyecto consta de los siguientes elementos y procesos:

1. **Datos de entrada:** Se trata de tres datos numéricos que provienen de la clase Gladiador, cuyos valores representan la media de las características físicas, habilidades que tenga en la lucha y la fama de un personaje del juego.

2. **Fuzzificación:** Esta es la fase de conversión de los datos de entrada en valores correspondientes a las funciones de pertenencia de acuerdo con el modelo difuso. Se han representado las variables 'características', 'habilidades' y 'fama' que hacen referencia a las propiedades del gladiador. Sus rangos van desde 0 a 150, expresadas en términos lingüísticos 'bajo' y 'alto'. La función de pertenencia, que nos indica el grado de pertenencia de una variable al conjunto correspondiente, viene dada por la ecuación trapezoidal (1). La justificación de uso de este tipo de funciones de pertenencia puede encontrarse en [1]. Como ejemplo de implementación de una variable en el entorno difuso, se muestra la Fig. 1.

$$A(x) = \begin{cases} \mathbf{0} & \boldsymbol{si}\ (\boldsymbol{x \le a}) \vee (\boldsymbol{x \ge d}) \\ \frac{(x-a)}{(b-a)} & \boldsymbol{si}\ x\ \epsilon (a, b] \\ \mathbf{1} & \boldsymbol{si}\ \ x \epsilon (b, c) \\ \frac{(d-a)}{(d-c)} & \boldsymbol{si}\ x \epsilon (b, d) \end{cases} \qquad (1)$$

La variable 'precio' mostrada en la Fig. 2 depende directamente de las variables anteriores y tiene un rango 0-3000 con tres términos lingüísticos asociados ('bajo', 'medio', 'alto').

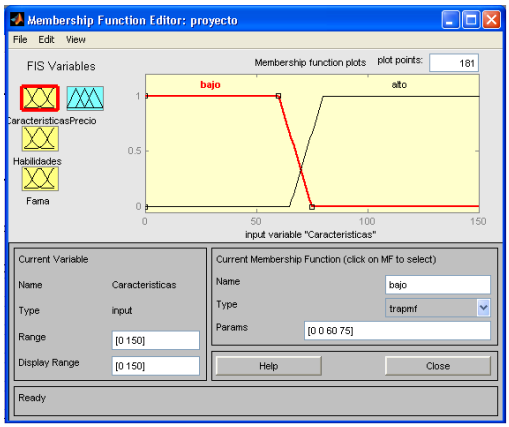


Fig. 1. Implementación de la variable 'características'.

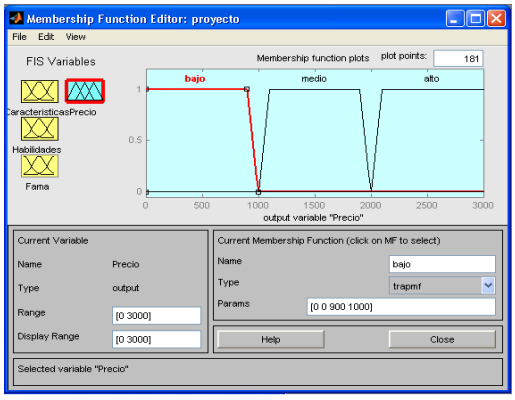


Fig.2. Implementación de la variable 'precio'.

3. **Evaluación de reglas:** Se define un total de ocho reglas, resultado de la evaluación de dos etiquetas (bajo y alto) para tres variables de entrada (características, habilidades y fama). El conjunto de reglas se muestran en la Fig. 3.

```
if caracteristicas is bajo and habilidades is bajo and fama is bajo then precio is bajo
if caracteristicas is bajo and habilidades is bajo and fama is alto then precio is medio
if caracteristicas is bajo and habilidades is alto and fama is bajo then precio is bajo
if caracteristicas is bajo and habilidades is alto and fama is alto then precio is medio
if caracteristicas is alto and habilidades is bajo and fama is bajo then precio is medio
if caracteristicas is alto and habilidades is bajo and fama is alto then precio is alto
if caracteristicas is alto and habilidades is alto and fama is bajo then precio is medio
if caracteristicas is alto and habilidades is alto and fama is alto then precio is alto
```

Fig. 3. Reglas de inferencia asociadas al cálculo del precio.

El operador lógico que se ha utilizado para la conjunción entre las variables es el mínimo [14]:

$$\mu_{A\cap B}(x) = \min[\mu_A(x),\mu_B(x)] \qquad (2)$$

4. **Inferencia**: La inferencia aplicada a las reglas determina el conjunto de salida de cada regla por medio del operador de Mamdani [10] por mínimos:

$$\min(\mu,\mu_W(z)),\forall z \qquad (3)$$

5. **Agregado**: Finalmente, se realiza la función de pertenencia de la variable de salida que en nuestro caso es a partir de la operación MAX entre todos los conjuntos de salida de la etapa de inferencia

6. **Defuzzificación**: Determinamos el dato más representativo mediante la técnica del centroide [8] basado en el cálculo discretizado de la fórmula en rango real (4).

$$C = \frac{\int_a^d f(x)xdx}{\int_a^d f(x)dx} \qquad (4)$$

4 Resultados

En esta sección se muestran los resultados obtenidos tras la realización de tres pruebas (Figs. 4-9). La primera de ellas está realizada sobre un personaje no entrenado previamente. Se trata de un gladiador recién comprado en la escuela y que por lo tanto tendrá unos valores presumiblemente bajos (peores) en comparación con otros gladiadores ya entrenados.

Fig. 4. Resultado LM sobre personaje sin entrenamiento previo.

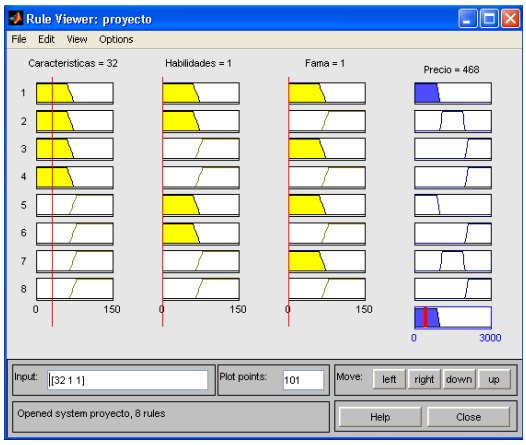


Fig. 5. Resultado MatLab sobre personaje sin entrenamiento previo.

En esta primera prueba podemos comprobar que el personaje comprado inicialmente, cuyas características serán peores que las de los otros personaje debido a que no se le ha entrenado, tendrá un valor bajo. Los valores que le entrarán al FIS (Fuzzy Inference System) tendrán un valor muy bajo. Por lo tanto al hacer la etapa de fuzzificación vemos que todas aquellas reglas que tengan como premisa valor "bajo", tendrán el mayor grado de pertenencia con valor 1, esto es debido a que al fuzzificar el primer término del antecedente de las cuatro primeras reglas (características 'is bajo') vemos a que grado de bajo pertenece características sí tiene un valor de 32. Como podemos ver 32, corresponde a un valor muy bajo ya que el rango máximo es hasta 75 de ahí que obtengamos un valor de pertenencia de 1. Como el operador es AND (mínimo), muchas de las reglas no devolverán un conjunto de salida. Esto se debe a que los otros dos valores a tratar tendrán grado 0, luego el conjunto de salida será el vacío salvo para la primera regla porque es la única regla que al fuzzificar los tres antecedentes de la regla, devuelve un grado de pertenencia mayor que 0 en cada una de ellas, luego el conjunto de salida no será el vacío. Podemos observar al realizar la operación de agregación *max*, que el conjunto de salida es igual al conjunto de salida de la primera regla, luego si calculamos el valor del personaje del juego tendrá un valor de 468.447304 monedas que representa el centro simétrico de la figura geométrica que se muestra en la Fig. 5, luego nuestro FIS devuelve un valor más exacto al que devuelve MatLab ya que se trata de un resultado no truncado.

En una segunda prueba, se ha escogido un personaje entrenado a nivel medio Esto hace que el personaje tenga valores medios en cuanto a características, habilidades y fama. Por ello, muchas de las reglas, por no decir todas, devolverán un conjunto de salida. Agregando esos conjuntos de salida mediante la operación MAX, formaremos un conjunto de salida del cual aplicaremos el método del centroide. El resultado de nuestro sistema devolverá un total de 1639,91150 monedas como precio asignado al jugador.

Fig. 6. Resultado LM sobre personaje con entrenamiento previo y nivel medio.

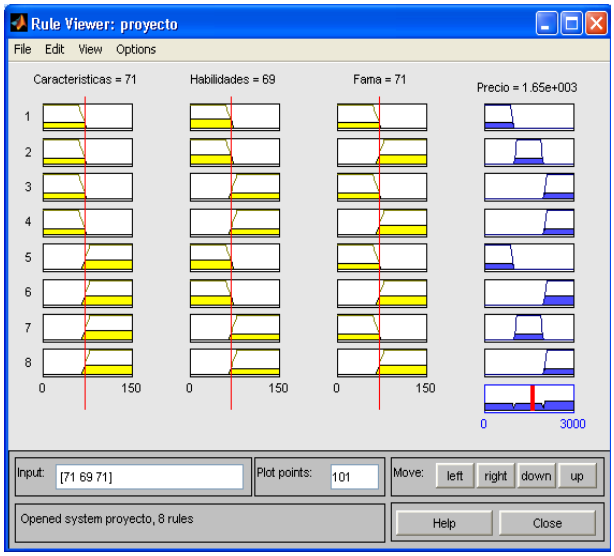


Fig.7. Resultado MatLab sobre personaje con entrenamiento previo y nivel medio.

Por último, en la tercera prueba tenemos un personaje cuyas habilidades y características son altas, debido a que se le ha aplicado un entrenamiento exhaustivo. En este caso nos encontramos con una situación similar a la de la prueba 1, pero la regla que será candidata a ejecutarse será la última. Esta dará un valor en este caso de 2516,53845 monedas.

Fig. 8. Resultado LM sobre personaje con entrenamiento exhaustivo y nivel alto

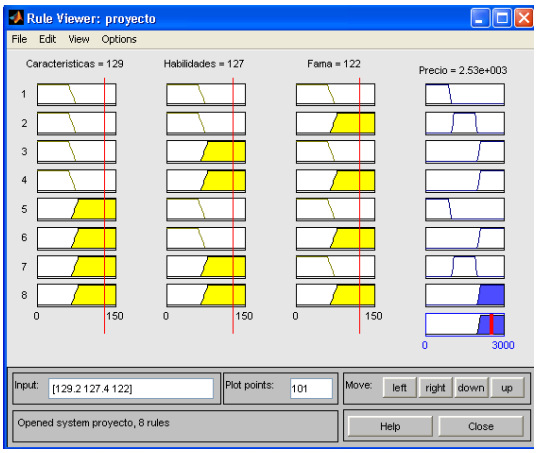


Fig. 9**.** Resultado MatLab sobre personaje con entrenamiento exhaustivo y nivel alto

Los resultados obtenidos en los tres casos analizados ofrecen mayor precisión con más rapidez de cálculo sin depender de ninguna herramienta externa, al realizarse los cálculos directamente en el código ejecutable del Smartphone.

5 Conclusiones

En este trabajo se ha presentado una aplicación de la Lógica Fuzzy y el Soft Computing en la evaluación de una variable determinante en el videojuego LM. LM emplea un procedimiento de evaluación basado en reglas de inferencia y la lógica de Mamdani [10] para obtener los resultados buscados. Como medida de testing se han comparado el comportamiento de tres casos generales y se han obtenido mejores resultados con LM que con la herramienta ToolBoxFuzzyLogic de Matlab. Como consecuencia, se ha optado por la integración en LM del método estudiado.

Como trabajo futuro se contempla el análisis de otras variables que pueden entrar a participar en la mejora del juego, sobre todo las relacionadas con la difusión del juego en las redes sociales en la futura versión online, donde además de competir con la máquina, el jugador tendrá que competir con otros usuarios online en tiempo real.

Referencias

1. Patel, A. V.: Transformation Functions for Trapezoidal MembershipFunctions. International Journal of Computational Cognition 2(3), 115–135 (2004)
2. Charlesworth, A.: The Ascent of Smartphone. Engineering & Technology 4 (3), 32–33 (2009)
3. Fuzzy Engine, http://fuzzyengine.sourceforge.net/index.html
4. Fuzzy Tech, http://www.fuzzytech.com
5. Ying, H.: Fuzzy Control and Modeling – Analytical Foundations and Applications. Wiley-IEEE Press, Series on Biological Engineering (2000)
6. Ying, H.: The Simplest Fuzzy Controllers Using Different Inference Methods are Different Nonlinear Proportional-Integral Controllers with Variable Gains. Automática 29(6), 1579–1589 (1993)
7. Jantzen, J., Verbruggen, H., Ostergaard, J.-J.: Fuzzy Control in the Process Industry – Common Practice and Challenging Perspectives. In: Zimmermann, H.-J. (ed.) Practical Applications of Fuzzy Technologies, vol. 6, pp. 3–56. Springer (1999)
8. Ousslaah, M., Nguyen, H. T., Kreinovich, V.: A New Derivation of Centroidede Fuzzification. In: 10th IEEE International Conference on Fuzzy Systems, vol. 2, pp. 884–887. IEEE Press, New York (2001)
9. Jacques, M. A. P., Pursula, M., Niittymäki, J., Kosonen, I.: The Impact of Different Approximate Reasoning Methods on Fuzzy Signal Controllers. In: 9th Meeting of the Euro Working Group on Transportation, pp. 33 (2002)
10. Mamdani, E. H.: Application of Fuzzy Logic to Approximate Reasoning Using Linguistic Synthesis. IEEE Transactions on Computers C-26(12), 1182–1191 (1977)
11. Sazonov, E. S. et al. (2002) Fuzzy Logic Expert System for Automated Damage Detection from Changes in Strain Energy Mode Shapes. Non-Destructive Testing and Evaluation 18(1), 1–20 (2002)
12. Uzunboylua, H., Cavusb, N., Ercaga, E.: Using Mobile Learning to Increase Environmental Awareness. Computers & Education 52 (2), 381–389 (2009)
13. Luyben, W. L., Luyben, M. L.: Essential of Process Control. McGraw-Hill (1997)
14. Yen, J., Langari, R.: Fuzzy Logic. Prentice-Hall (2004)
15. Li, Y., Musilek, P., Wyard-Scott, L.: Fuzzy Logic in Agent-Based Game Design. In: 2004 IEEE Annual Meeting of the Fuzzy Information, Processing NAFIPS 2004, vol 2, pp 734–739. IEEE Press (2004)
16. Kaur, A., Kaur, A.: Comparison of Mamdani-Type and Sugeno-Type Fuzzy Inference Systems for Air Conditioning System. International Journal of Soft Computing and Engineering 2(2), 323–325 (2012)
17. Yager, R. R., Kacprzyk, J.: The Ordered Weighted Averaging Operators – Theory and Applications. Kluwer Academic Publishers, Norwell, MA (1997)

Método para la Detección de Intrusos basado en la Sinergia de Técnicas de Inteligencia Artificial

César Guevara, Matilde Santos y José Antonio Martín H.

Facultad de Informática, Universidad Complutense de Madrid, C/ Profesor García Santesmases s/n 28040-Madrid, España. Teléfono: +34 91 394 76 20; Fax: +34 91 394 7527

cesar.guevara@gmail.com, msantos@dacya.ucm.es, jamartinh@fdi.ucm.es

Abstract. Las técnicas de clasificación basadas en Inteligencia Artificial son herramientas computacionales que han sido aplicadas en la detección de intrusos (IDS), pues resuelven de manera eficiente muchos de los problemas en el área de la seguridad informática. La detección de intrusos conlleva utilizar gran cantidad de información para una adecuada aproximación a la identificación de intrusos, por eso es necesario emplear técnicas basadas en el conocimiento. En este trabajo se identifica el comportamiento de los usuarios autorizados y de los intrusos mediante la sinergia de varias de las técnicas de IA: redes neuronales, máquinas de vectores soporte y árboles de decisión. El sistema desarrollado permite la aplicación de esas técnicas de clasificación y su combinación para obtener una mayor eficiencia y fiabilidad en la detección de usuarios en las redes o sistemas informáticos. Para probar este modelo se ha diseñado un escenario que sirve para demostrar la validez de la propuesta y para acreditar líneas futuras de trabajo.

Keywords: Inteligencia Artificial, Detección Intrusos, Redes Neuronales, Arboles de Decisión, Máquinas de Vectores Soporte

1 Introducción

En la actualidad los Sistemas de Detección de Intrusos (IDS) son de gran relevancia para la seguridad en las redes y sistemas informáticos ya que cada vez y con mayor frecuencia aparecen nuevas formas de ataques, algunos muy complejos [1].

Los IDS en su mayoría analizan el tráfico de la red e impiden cualquier intrusión previa al ingreso de un usuario al sistema de información. El análisis realizado por el IDS en general es a bajo nivel, generando alertas aisladas y manejando una inmensa cantidad de información. Otros tipos de IDS realizan un filtrado de anomalías de la información recopilada de los equipos host, así como también de las bases de datos de los sistemas con los que interactúan [2].

El campo de detección de intrusos sigue siendo un frente abierto a la investigación para desarrollar metodologías dinámicas que puedan adaptarse a la evolución de

ataques informáticos, cada vez más sofisticados y complejos. Las estrategias que utilizan los IDS se clasifican en dos grupos: detección de uso indebido y detección de anomalías. La metodología de Detección de Intrusos Basada en Anomalías, la cual es utilizada en este trabajo, es extensa y compleja, y ha permitido obtener resultados favorables pero no enteramente adaptables a las necesidades actuales de la tecnología.

La detección de uso indebido de un sistema informático utiliza el conocimiento de las secuencias de actividades que constituyen un ataque, las cuales están almacenadas en una base de datos. La información recopilada es comparada con los patrones de ataque almacenados y, si coinciden, originan una alarma. Esta estrategia es una de las más utilizadas de forma comercial [3]. Sus ventajas son la correspondencia con un patrón de intrusión de base de datos ya almacenado (rapidez), y que el número de falsos positivos sea muy bajo (fiabilidad y precisión). Sin embargo algunas de sus desventajas son, por ejemplo, la incapacidad de detectar nuevos ataques al no ser completamente dinámico, y además el requisito fundamental de mantener actualizada de forma continua la base de datos de los patrones.

Por otro lado, la detección de anomalías utiliza la información de un comportamiento normal del usuario y todo lo que sea diferente de aquella conducta es identificado como un intruso. Sus desventajas son que genera una gran cantidad de falsos positivos y que el comportamiento normal de los usuarios es bastante difícil de modelar, por la necesidad de contar con un periodo amplio de recopilación de información para el aprendizaje del comportamiento [4].

Teniendo en cuenta estas premisas, nuestra propuesta es desarrollar un método dinámico para la detección de intrusos mediante el análisis de anomalías en el tráfico de la red. Como un primer paso para alcanzar el objetivo en esta investigación se utilizan de manera conjunta varias estrategias de clasificación que provienen de la Inteligencia Artificial, en concreto los árboles de decisión, las máquinas de vectores de soporte y las redes neuronales. Se puede probar así que la sinergia de varias técnicas hace más eficiente y fiable el sistema de detección.

Para esta propuesta se han evaluado diferentes metodologías de clasificación con la misma base de datos, para luego aplicar una sinergia de aquellas técnicas que dan mejores resultados y obtener así un resultado mejor en la detección de intrusos que utilizándolas por separado. La metodología propuesta debería ser capaz de procesar la información en tiempo real, filtrando anomalías que pudiesen estar originadas por intentos de intrusión. Debe además generar alertas en caso de encontrar anomalías y almacenarlas de forma dinámica para ser utilizadas en el futuro como aprendizaje para la detección de intrusiones.

La base de datos utilizada es un benchmark en la literatura de seguridad [9], con el fin de probar la validez de esta propuesta para en un futuro aplicarla a bases de datos reales con mayor complejidad.

El artículo está estructurado de la siguiente manera: en la siguiente sección se presenta la evaluación de las técnicas de clasificación que van a ser utilizadas en el estudio; posteriormente se detalla el diseño y la descripción del método propuesto que combina varias de ellas. En la siguiente sección se presentan y se discuten los resultados obtenidos. Y finalmente se comentan las conclusiones con la información relevante del estudio y líneas futuras de investigación.

2 Evaluación de Técnicas de Clasificación

Para el desarrollo de este método de detección de intrusos se ha realizado un análisis previo de un determinado número de técnicas de clasificación, para seleccionar las más adecuadas para nuestro estudio. Las técnicas que se han aplicado son las siguientes:

- Red Neuronal Multicapa [5]
- Arboles de Decisión [6]
- Máquinas de Vectores de Soporte [7] [14]
- Red Neuronal de Hopfield [5]
- Redes de Bayes [5]

Para realizar la evaluación de las técnicas de clasificación se ha empleado el Dataset NSL-KDD [8, 9], el cual contiene información de patrones de comportamiento de usuarios en redes informáticas, tanto intrusos como autorizados. El Dataset de NSL-KDD contiene 40 atributos o variables que proporcionan distinto tipo de información sobre los accesos (protocolos, tiempos, tipos de acceso, etc.). Para este estudio se emplearon 25.192 registros para el entrenamiento (el 20% de la base de datos) y 17.102 registros para las pruebas (el 20% del dataset), como se sugiere en [9].

Todos los atributos que incluye el dataset, que describe el comportamiento completo del usuario, son significativos por lo que no es recomendable dividirlo en pequeños grupos de datos ya que se corre el riesgo de perder información relevante para la clasificación.

Para valorar el desempeño de las técnicas de clasificación se realizó una validación cruzada, sirviéndose de la técnica de K-Fold [10], utilizando diez particiones de los datos (k =10) para cada una de las técnicas utilizadas [11]. Estas pruebas permitieron conocer el porcentaje de casos clasificados correctamente y la precisión del clasificador para cada una de las cinco técnicas de clasificación (Figura 1).

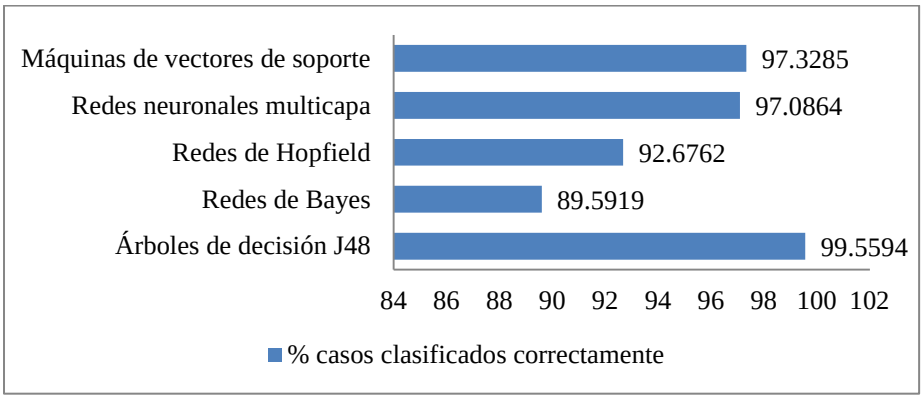


Fig. 1. Porcentaje de casos clasificados correctamente con cada técnica

En la tabla 1 se muestran varios indicadores que facilitan la selección de las técnicas más adecuadas y eficientes para nuestro estudio, así como el porcentaje de casos (registros) clasificados correctamente. Además presenta los indicadores de los comportamientos intrusivo y normal. El error medio absoluto del clasificador permite conocer la diferencia entre el valor de la medida y el valor tomado como exacto del fallo del clasificador. La precisión del clasificador permite conocer la eficiencia del clasificador. La matriz de error muestra la información de las pruebas realizadas a cada uno de los clasificadores, proporcionando información de los verdaderos positivos y los falsos positivos.

				MATRIZ DE ERROR			
Nombre técnica de clasificación	**% casos clasificados correctamente**	**Error medio absoluto**	**Precisión del clasificador**	**Verdaderos positivos Normal**	**Verdaderos positivos Intrusos**	**Falsos Positivos Normal**	**Falsos Positivos Intrusos**
Árboles de decisión J48	99.5594	0.006	0.996	13389 (53.14%)	11692 (46.42%)	51 (0.20%)	60 (0.24%)
Redes de Bayes	89.5919	0.103	0.896	12272 (48.71%)	10298 (40.88)	1445 (5.74%)	1177 (4.67%)
Redes de Hopfield	92.6762	0.123	0.927	12694 (50.39%)	10653 (42.29%)	1090 (4.33%)	755 (2.99%)
Redes neuronales multicapa	97.0864	0.03	0.971	13064 (51.86%)	11394 (45.23%)	349 (1.38%)	385 (1.53%)
Máquinas de vectores de soporte	97.3285	0.027	0.974	13261 (52.64%)	11258 (44.69%)	485 (1.92%)	188 (0.75%)

Tabla 1. Comparación de técnicas de clasificación con el Dataset NSL-KDD

El número de aciertos es elevado en cada una de las técnicas ya que se ha utilizado un número muy alto de ejemplos para el entrenamiento.

Después del resultado obtenido de la evaluación se observa que las tres técnicas de clasificación que han alcanzado un mejor resultado son: Redes Neuronales Multicapa, Máquinas de Vectores de Soporte y Árboles de Decisión.

3 Modelo Propuesto

En este artículo se propone un modelo para la detección de intrusos como el que se muestra en la figura 2. Resulta de la sinergia de las tres técnicas de clasificación anteriormente citadas que daban mejores resultados: Redes Neuronales, Máquinas de Vectores de Soporte y Árboles de Decisión.

La utilización de estas técnicas es la base fundamental para la disminución de detecciones fallidas y para mejorar el rendimiento del filtrado de intrusos. Con este esquema se incrementará porcentualmente la detección de intrusos y disminuirá la tasa de falsos positivos de manera significativa, lo que beneficiará la seguridad de los sistemas.

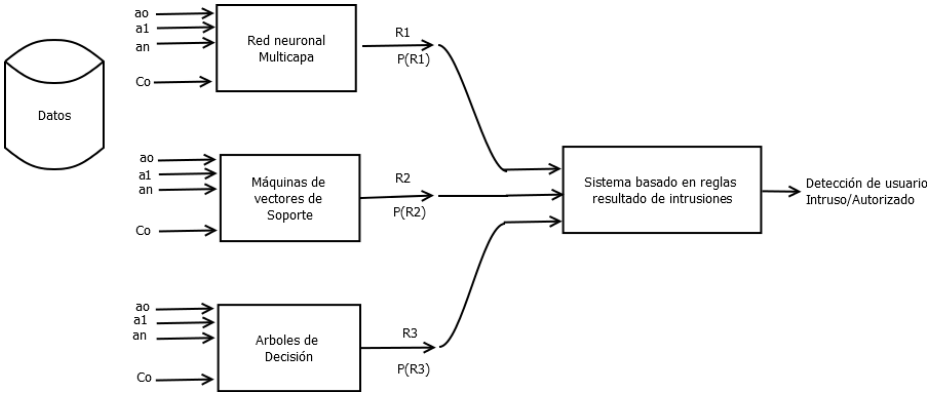


Fig. 2. Diagrama de la sinergia de Técnicas de Clasificación para Detección de Intrusiones

El modelo se basa en un proceso de uso simultáneo de la información con las tres técnicas de clasificación. Cada una de las partes de este proceso realiza tareas distintas con la información recopilada en la tarea anterior, lo que es importante para obtener un resultado eficiente. El flujo de operaciones se muestra en la figura 3.

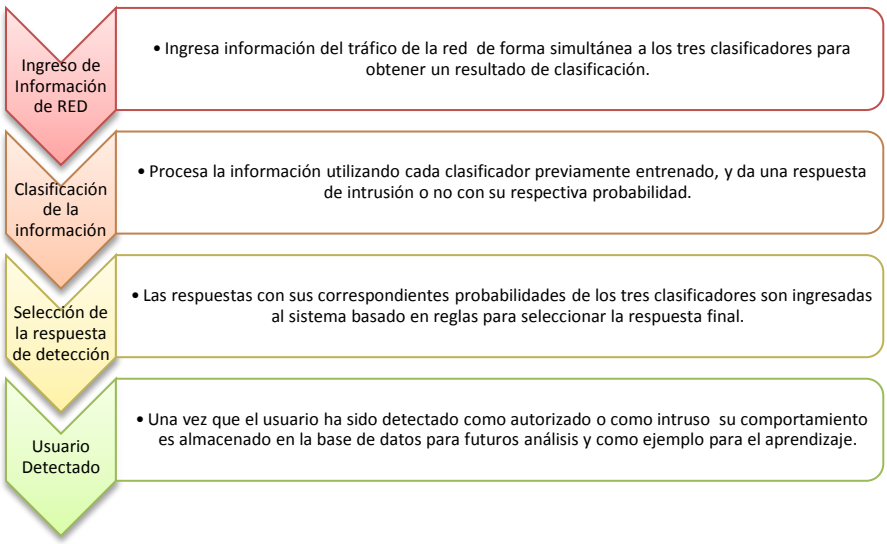


Fig. 3. Flujo de la información del modelo de sinergia de clasificadores para detección de intrusos.

A continuación se detallan cada una de las fases del proceso de detección de intrusos del modelo propuesto.

3.1 Configuración de técnicas de clasificación

Previamente a la aplicación de las técnicas de clasificación es necesario describir la configuración de cada una de ellas.

Red Neuronal Multicapa: se ha construido una red neuronal con 40 neuronas de entrada, tantas como atributos de los usuarios de la red; 2 capas ocultas, la primera con 20 neuronas y la segunda con 16; la salida de esta red neuronal es una sola neurona que da como resultado una de las dos clases posibles, accesos permitido o acceso intrusivo.

Máquinas de vectores de soporte (SVM): se ha utilizado como kernel la función RBF (Radial Basis Function), entrenando las SVM con el algoritmo SMO (Sequential Minimal Optimazer). El entrenamiento fue realizado con la librería LIBSVM [14] utilizando los 40 atributos del trafico de la red.

Árboles de decisión: el árbol de decisión utiliza los 40 atributos del dataset del tráfico de red y aplica el algoritmo J48 para la clasificación.

Todos los clasificadores se entrenaron con 25.912 registros para el entrenamiento y 17102 registros para pruebas.

3.2 Descripción del modelo

Ingreso de información de la red: se ingresa de manera simultánea la información de tráfico de la red a los tres clasificadores, con sus cuarenta atributos por registro, normalizados [12].

Clasificación de la información: se clasifica la información de la red dando como salida si un usuario es un intruso o está autorizado, con un porcentaje de fiabilidad y precisión.

Fase de Selección de resultados mediante sistema basado en reglas: los resultados de los tres clasificadores anteriores se introducen en el sistema de reglas como entradas. Estas entradas se denominan R1, R2 y R3, con sus respectivas probabilidades P(R1), P(R2) y P(R3). Luego el sistema de decisión basado en reglas recibe estos seis valores, y proporciona el resultado final de la clasificación. La figura 4 presenta las 8 reglas que conforman este sistema. La salida será un valor binario, "1" si es intruso y "0" si está autorizado.

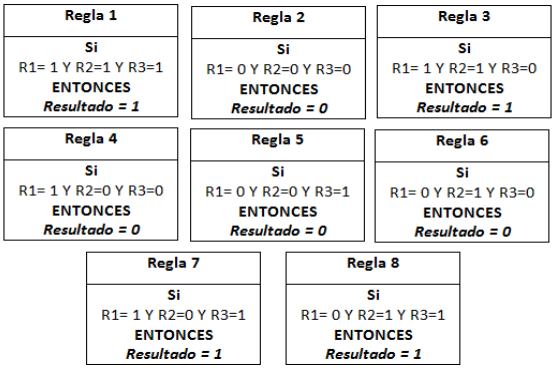


Fig. 4. Reglas para la identificación final del usuario.

Después de obtener el resultado de la detección es necesario conocer la probabilidad de la respuesta. Para ello que se realiza un cálculo matemático utilizando la media de los datos que cumplen las reglas anteriormente descritas. Para las reglas 3, 4, 5, 6, 7 y 8 se calcula la media de las dos probabilidades correspondientes a los resultados que verifican las reglas anteriormente descritas, ya que dos de los clasificadores proporcionan el mismo resultado y el tercero uno distinto (1).

$$\mathbf{Med= P(Ra)+P(Rb)/2} \quad (1)$$

En el caso de las reglas 1 y 2, donde la salida de los tres clasificadores es la misma, la media de probabilidades es:

$$\mathbf{Med=P(R1)+P(R2)+P(R3)/3} \quad (2)$$

Es imprescindible utilizar la media de las probabilidades de los resultados de las distintas técnicas de clasificación para obtener una probabilidad conjunta. Además, esta probabilidad resultante será registrada dentro del histórico del comportamiento del usuario, lo que permite que el modelo sea dinámico al contar con nueva información.

Para validar este sistema de reglas se utilizaron registros de la base de datos [9]. En la tabla 2 se presenta una pequeña muestra de cómo se presentan los datos para las pruebas.

R1	P(R1)	R2	P(R2)	R3	P(R3)	Clasificador
0	0.99995	0	0.99968	0	0.99945	0
0	0.99995	0	0.99968	0	0.99032	0
1	0.99924	1	0.99055	1	0.99032	1
1	0.99924	1	0.99055	1	0.99945	1

Tabla 2. Muestra de información para la decisión final de la respuesta.

Usuario Detectado: finalmente se obtiene un resultado (Intruso, 1, o Autorizado, 0), el cual se almacena en la base de datos para que pueda utilizado para futuros análisis de patrones de comportamiento dentro de la red.

4 Discusión de Resultados

En la tabla 3 se presentan los resultados para los indicadores de eficiencia utilizados y los falsos positivos y falsos negativos obtenidos con la técnica propuesta.

% casos clasificados correctamente	% Error medio absoluto	% Precisión del clasificador	% Verdaderos positivos autorizados	% Verdaderos positivos Intrusos	% Falsos Positivos autorizados	% Falsos Positivos Intrusos
99.9%	0.1	99.9	99.9% (2997)	99.9% (2997)	1% (3)	1% (3)

Tabla 3. Indicadores utilizados para probar la propuesta de sinergia de Técnicas de Clasificación para Detección de Intrusiones.

Como ya se comentó, la precisión de las técnicas de clasificación para esta base de datos es muy elevada, con un valor de 99.9% (**Precisión del Clasificador**) en todos los casos de clasificación. En la aplicación de la sinergia de estrategias inteligentes se obtiene un resultado más elevado que al utilizar las técnicas de manera individual. De hecho se alcanza el 99.9% de aciertos en verdaderos positivos y verdaderos negativos (2.997 casos); por lo tanto hay sólo un 1% de falsos positivos y de falsos negativos, lo que significa que el modelo realiza la clasificación de forma muy eficiente.

La información con la que se ha realizado el estudio es muy completa y extensa, se ha contado con muchos ejemplos para entrenar los sistemas de clasificación, y por eso los resultados obtenidos con las técnicas seleccionadas ya eran buenos.

Lo que se ha probado en este trabajo es cómo la fusión de varias de estas técnicas mejora el resultado, con el fin de poder aplicar esta sinergia a problemas más complejos donde no haya tanta información o ésta esté incompleta [13].

5 Conclusiones y Trabajos Futuros

En este estudio se propuso un método para la detección de intrusos mediante la sinergia de técnicas de clasificación que provienen de la Inteligencia Artificial.

Se ha evaluado la aplicación de varias técnicas de clasificación a un problema bien definido, y se ha probado como la combinación de varias de ellas mejora el porcentaje de aciertos. Se ha utilizado como base de conocimientos el dataset NSL-KDD.

Las líneas de trabajo futuro son principalmente dos. Por un lado, habría que aplicar esta estrategia de fusionar técnicas a problemas más complejos, con menos ejemplos de entrenamiento, o señales ruidosas, con información incompleta, etc. Por otro, estamos trabajando para realizar pruebas con información real de usuarios que acceden a bases de datos de información y realizan una serie de operaciones sobre los registros.

Referencias

1. Debar, H., Viinikka, J.: Introduction to Intrusion Detection and Security Information Management. In: Foundations of Security Analysis and Design III FOSAD 2005. LNCS, 3655, pp. 207--236. Springer (2005).
2. Zurutuza, U.: Revisión del estado actual de la investigación en el uso de data mining para la detección de intrusiones. Escuela Politécnica Superior de Mondragón, Departamento. Informática 2005.
3. Bace, R.G, Mell, P.. Intrusion detection systems. Gaithersburg, MD: U.S. Dept. of Commerce, Technology Administration, National Institute of Standards and Technology, 2001.
4. Wang, W., Gombault, S.: Distance measures for anomaly intrusion detection. In: Security and Management (2007) 17–23
5. Haykin, S. Neural networks: a comprehensive foundation. New York: Macmillan, 2004.
6. Quinlan, J.R.. Induction of decision trees. Broadway, N.S.W., Australia: New South Wales Institute of Technology, School of Computing Sciences, 1986.
7. Abe, S. Support vector machines for pattern classification. London: Springer, 2005.
8. MIT: Mit Lincoln laboratory-Darpa intrusion detection evaluation (retrieved march 2009). http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/index.html (1999)

9. Nsl-kdd data set for network-based intrusion detection systems." Available on: http://nsl.cs.unb.ca/NSL-KDD/, March 2009.
10. Moore, A.W., Lee, M.S. 1993. Efficient Algorithms for Minimizing Cross Validation Error. In Machine Learning: Proceedings of the Eleventh International Conference, Morgan Kaufmann.
11. Esposito, M., Mazzariello, C., Oliviero, F., Romano S.P., Sansone, C. "Evaluating pattern recognition techniques in intrusion detection systems." in Proceedings of the 5th International Workshop on Pattern Recognition in Information Systems (PRIS) 2005, May 2005, pp. 144 -153.
12. Wang W., Zhang X., Gombault S., Knapskog S.J., Attribute Normalization in Network Intrusion Detection, IEEE, 2009 10th International Symposium on Pervasive Systems, Algorithms, and Networks. 978-0-7695-3908-9/09, pp-448-453.
13. Lee, W., Stolfo, S.J. and Mok, K. (1999). Data Mining in work flow environments: Experiments in intrusion detection. In Proceedings of the 1999 Conference on Knowledge Discovery and Data Mining.
14. Chang C.-C., Lin C.-J., LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2-27, 2011. Available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

Diseño de un controlador borroso para el sistema de exposición de una cámara de fotos semi-automática

Jorge Martínez Ladrón de Guevara, Matilde Santos Peñas

Facultad de Informática, Universidad Complutense de Madrid, España

jorgeama@estumail.ucm.es, msantos@ucm.es

Abstract. Este artículo presenta un sistema borroso para el control de exposición de una cámara de fotos. El sistema de control utiliza conjuntos difusos, correspondientes a la naturaleza de las variables de entrada, en sus reglas de inferencia y proporciona resultados precisos para la distancia focal, el valor de exposición, la apertura del diafragma y la velocidad de obturación de una cámara digital automática. Se busca la solución más apropiada para las condiciones de cada fotografía. Los resultados obtenidos son buenos y se asemejan al comportamiento real del usuario. El prototipo del controlador se ha desarrollado con Xfuzzy 3 y Java.

Keywords: Lógica fuzzy, cámara de fotos, valor de exposición, diafragma, velocidad de obturación, Xfuzzy

1 Introducción

En los últimos años, los sistemas de control automático incorporados en las cámaras digitales se han desarrollado considerablemente. En la actualidad es prácticamente imposible cometer errores en la de exposición de una fotografía o en el enfoque y prácticamente todas las fotografías que hacemos tienen una buena calidad técnica. Las cámaras modernas incluyen tecnologías muy avanzadas y sistemas de control para asegurar que la exposición [1-7], el balance de blancos [1, 2, 8], la distancia focal [9] o el enfoque [1, 2] son los más apropiados para las condiciones de la escena que se va a fotografiar. Desde finales de los años 80 se han desarrollado circuitos para el sistema de exposición basadas en lógica borrosa [2, 3, 4, 6], el enfoque automático [2], el balance de blancos [2, 8] o el auto zoom [9]. Los sistemas de control de la exposición se basan fundamentalmente en tecnologías que utilizan la óptica de la cámara [10] para medir la luminancia y el contraste en distintos puntos de la imagen. La mayoría de los sistemas de exposición automática (AE) que existen, algunos de los cuales aplican la lógica borrosa, miden la luz a través de la lente, comparan la luminancia de la imagen, hacen histogramas y deciden la mejor exposición. En nuestro caso se trabaja con una cámara sencilla que no mide la luz y solo recibe un valor de luminosidad dado por el fotógrafo, por lo que en base a estas apreciaciones subjetivas se ha diseñado un sistema menos complejo, que utiliza otras variables para obtener el valor de la exposición.

© Springer-Verlag Berlin Heidelberg 2011

Este trabajo desarrolla un sistema difuso que recibe como entradas la luminosidad de la escena, el tipo de fotografía, la sensibilidad del sensor y la distancia aproximada al sujeto. Da como resultado el valor de exposición (VE), la apertura del diafragma (f), la velocidad de obturación (s) y la distancia focal más apropiada para obtener la imagen que se desea. El controlador es borroso debido a la naturaleza borrosa de las entradas del sistema. En vez de medir la luminosidad a través de la lente, el valor de exposición se calcula a partir de niveles de luminosidad borrosos que dados por el fotógrafo, que dependen de su apreciación personal. El sistema utiliza reglas de inferencia simples y ofrece resultados que, en general, se asemejan al comportamiento de una cámara digital de gama media. Las reglas del controlador se construyen utilizando las tablas 3, 4, y 5, elaboradas expresamente para este sistema. La tabla 3 relaciona los valores borrosos de luminosidad y la sensibilidad del sensor con el valor de exposición (VE). Las tablas 4 y 5 relacionan el valor de exposición (VE) y la combinación de apertura de diafragma y velocidad de obturación con el tipo de fotografía. Es normal que existan distintos criterios sobre cuál es la apertura de diafragma idónea para hacer un retrato o un paisaje, de manera que los valores de esta tabla se deben modificar para personalizar el comportamiento del sistema según las preferencias de cada fotógrafo.

Este artículo se organiza en cinco apartados. El apartado 2 describe el sistema de control de la exposición y muestra la relación entre la apertura del diafragma, la velocidad de obturación y la sensibilidad del sensor digital con las características de la imagen. El apartado 3 describe el controlador borroso, sus entradas, salidas, la representación de los datos en conjuntos borrosos, las reglas de inferencia y su implementación en Xfuzzy. El apartado 4 muestra los resultados obtenidos con un subconjunto de los casos de prueba, con la finalidad que se vea que el comportamiento del sistema es satisfactorio en distintos escenarios. Finalmente, el apartado 5 incluye las conclusiones del trabajo realizado.

2 El control de la exposición

El sistema de control de la exposición de una cámara fotográfica se basa en la combinación de tres elementos: la apertura del diafragma (f), la velocidad de obturación (s) y la sensibilidad del sensor digital. Este sistema utilizan un exposímetro para medir la luz que refleja la escena y, una vez calculado el valor de exposición, recomiendan una combinación de apertura de diafragma (f) y velocidad (s) dependiendo del modo de operación de la cámara: automático, prioridad a la apertura, prioridad a la velocidad o un tipo de fotografía predefinido (retrato, paisaje o deportes). La elección de una combinación de apertura y velocidad determinan a su vez las características de la imagen. Si se quiere obtener una imagen con alto contraste y profundidad de campo reducida, se debe elegir una velocidad alta y un diafragma muy abierto, por contra, si se desea conseguir una imagen con bajo contraste y mucha profundidad de campo, se debe elegir un diafragma cerrado y una velocidad de obturación baja [11].

El objetivo del sistema de control de la exposición es calcular el valor de exposición (VE) para las condiciones de luz y la sensibilidad del sensor que recibe como entradas. Una vez calculado el valor de exposición, se selecciona la combinación de

apertura de diafragma (f) y velocidad de obturación más adecuada para el tipo de fotografía seleccionado (figura 1). La relación entre la apertura de diafragma y la velocidad de obturación es inversamente proporcional; en la medida que uno de ellos aumenta, el otro se reduce para mantener el mismo valor de exposición (VE). Además de la proporcionalidad entre estos valores, hay que tener en cuenta que la elección de un diafragma más abierto o más cerrado determina la profundidad de campo, el contraste y el desenfoque por movimiento. Esto define las características de la imagen y está dado por una de las entradas del controlador: el tipo de fotografía. Este sistema se ha diseñado para tomar fotografías de deportes y acción, retratos o paisajes.

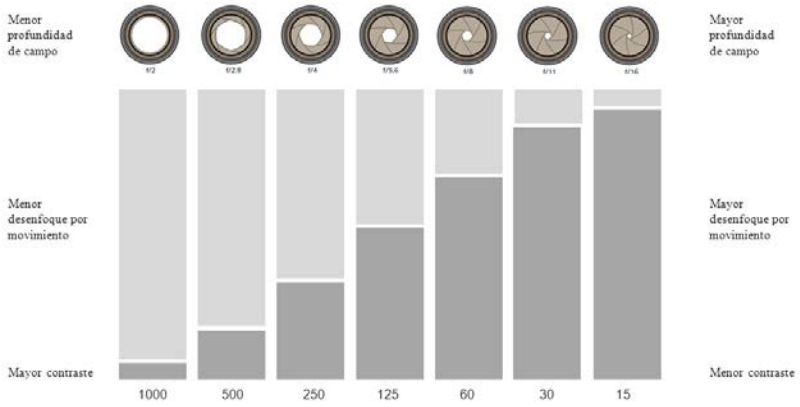


Fig. 1. Relación entre apertura de diafragma y velocidad de obturación

El valor de la exposición determina las distintas combinaciones de la apertura del diafragma (f), la velocidad de obturación (s) para una sensibilidad del sensor determinada y unas condiciones de luminosidad. El valor de exposición se calcula como Log2 (f2/s) [11]. En la escala de valores de exposición adoptada internacionalmente, el valor VE 1 corresponde a una apertura f 1.4 y velocidad de 1 seg. La figura 2 ha sido adaptada de una tabla de valores de exposición VE [12], y muestra la relación entre este valor y la apertura del diafragma (f) y la velocidad de obturación (s).

Por cada unidad de aumento de esta escala, el valor de exposición se reduce a la mitad. Por ejemplo, el valor de exposición VE 2 es la mitad de VE 1, VE 3 es un cuarto de VE 1, VE 4 es un octavo de VE 1 y así sucesivamente. Los valores de exposición (VE) de la figura 2 van desde 1 hasta 20 porque el rango de la velocidad llega hasta 1/200 seg. y f 22. Como se puede ver en la Figura 3, existen distintas combinaciones de apertura de diafragma y velocidad de obturación que tienen el mismo valor de exposición. Por ejemplo, f 8 con velocidad 1/15 seg. y f 1.4 con velocidad 1/500 seg. son equivalentes. Ambas combinaciones tienen VE 10 y dejan pasar la misma cantidad de luz al sensor. Las imágenes que producen tienen la misma exposición pero sus características son completamente diferentes.

s	f 1.0	f 1.4	f 2.0	f 2.8	f 4.0	f 5.6	f 8.0	f 11	f 16	f 22
1	0	1	2	3	4	5	6	7	8	9
2	1	2	3	4	5	6	7	8	9	10
4	2	3	4	5	6	7	8	9	10	11
8	3	4	5	6	7	8	9	10	11	12
15	4	5	6	7	8	9	10	11	12	13
30	5	6	7	8	9	10	11	12	13	14
60	6	7	8	9	10	11	12	13	14	15
125	7	8	9	10	11	12	13	14	15	16
250	8	9	10	11	12	13	14	15	16	17
500	9	10	11	12	13	14	15	16	17	18
1000	10	11	12	13	14	15	16	17	18	19
2000	11	12	13	14	15	16	17	18	19	20

Fig. 2. Velocidad (s), apertura (f) y VE

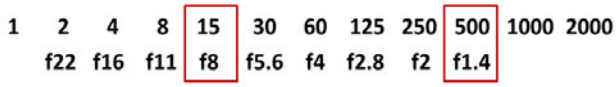


Fig. 3. Combinaciones de f y s para un VE

Si se utiliza f 8 y 1/15 seg. se tiene mayor profundidad de campo y desenfoque por movimiento. Si se utiliza f 1.4 y 1/500 seg., entonces la profundidad de campo es mínima, la imagen se congela y se evita el desenfoque por movimiento.

3 El controlador difuso

Para el desarrollo del prototipo se propone el uso de un controlador difuso debido a la imprecisión de algunas de las entradas del sistema. La luminosidad de la escena es un valor difuso definido por cinco valores distintos (Tabla 1). El tipo de fotografía (Tabla 2) no es un dato borroso, sin embargo da lugar a distintas interpretaciones válidas por lo que su significado sí se puede considerar impreciso. La distancia al sujeto también es un valor aproximado (Tabla 3).

El objetivo de este sistema de control es calcular el valor de exposición (VE), la apertura del diafragma (f), la velocidad de obturación (s) y la distancia focal de la lente que recomienda utilizar. Para ello utiliza como valores de entrada la luminosidad de la escena, la sensibilidad del sensor, el tipo de fotografía que se desea realizar y la distancia aproximada al sujeto (Figura 4).

La sensibilidad del sensor puede tomar los valores 100, 200, 400, 800 y 1600 ISO. La luminosidad es un dato borroso porque no se dispone de un exposímetro para calcularla de forma precisa. Se consideran cinco niveles de luminosidad distintos, desde más alta dada por un sol brillante con cielo despejado, hasta la luminosidad más baja (Tabla 1). El tipo de fotografía está definido por el conjunto de valores: deportes y acción, retrato o paisaje (Tabla 2). La distancia al sujeto también es un

valor borroso que se describe utilizando términos como muy cerca, cerca o lejos (Tabla 3).

Fig. 4. Entradas y salidas del controlador borroso

	Sol brillante y cielo despejado
	Sol brillante y cielo parcialmente cubierto
	Sol velado y sombras poco definidas
	Cielo cubierto y sombras apenas definidas
	Cielo muy nublado

Tabla 1. Luminosidad de la escena

	Deportes y acción
	Retrato
	Paisaje

Tabla 2. Tipos de fotografía

Muy cerca	Menos de un metro
Cerca	Entre uno y dos metros
Intermedia	Entre 2 y tres metros
Lejos	Entre 3 y cuatro metros
Muy lejos	Más de cuatro metros

Table 3. Distancia

El controlador debe utilizar estos datos imprecisos y aplicar reglas borrosas para determinar las salidas del sistema.

3.1 La representación de los datos borrosos

La lógica borrosa, a diferencia de la lógica clásica, permite representar valores imprecisos utilizando funciones de pertenencia que toman valores en un intervalo entre 0 y 1 [9]. Este controlador utiliza funciones de pertenencia trapezoidales y Singleton para representar los datos de entrada y salida. Los niveles de luminosidad se representan con conjuntos difusos definidos por funciones de pertenencia trapezoidales con etiquetas: muy baja, baja, media, alta o muy alta. El tipo de fotografía y la sensibilidad del sensor se representan con funciones Singleton. Por último, la distancia se representa con una función de pertenencia trapezoidal definida con cinco distancias: muy cerca, cerca, intermedia, lejos y muy lejos.

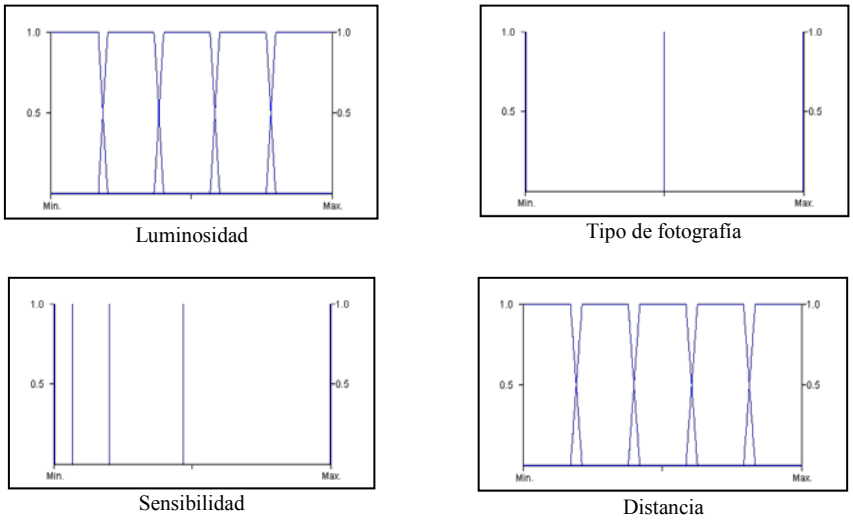
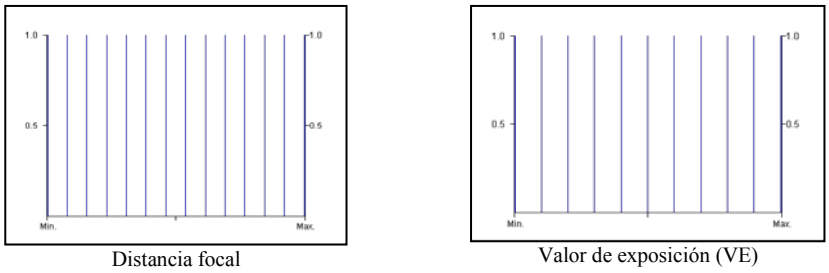


Fig. 5. Funciones de pertenencia de las entradas del controlador

Las salidas del controlador son valores precisos y se representan con funciones de pertenencia Singleton.



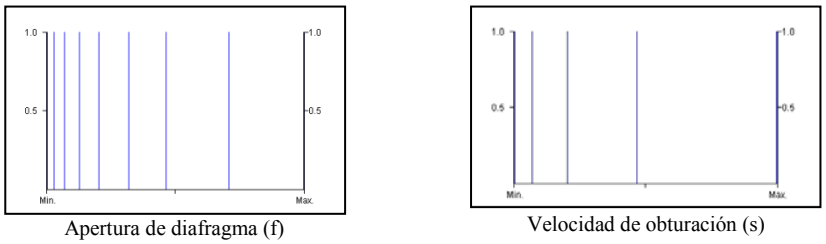


Fig. 6. Funciones de pertenencia de las salidas del controlador

3.2 Las reglas de inferencia

El sistema de inferencia combina el uso de datos borrosos con datos precisos aplicando reglas difusas para determinar la solución más apropiada para cada fotografía. Los valores de entrada son convertidos a valores borrosos mediante un proceso de 'borrosificación', después se aplican las reglas de inferencia del tipo IF - THEN [9] y finalmente se convierten los valores de salida mediante un proceso de 'deborrosificación' [13]. Se ha utilizado el método de implicación de Mamdani por su sencillez [14]. El controlador utiliza tres reglas de inferencia, una de ellas para calcular la distancia focal en base a la distancia aproximada del sujeto (Si la distancia es pequeña, recomienda una distancia focal con un ángulo abierto; a medida que aumenta, distancia focal con un ángulo más cerrado), y dos reglas anidadas para calcular el valor de exposición (VE), la apertura del diafragma (f) y la velocidad de obturación (s). Para calcular el valor de exposición, la apertura de diafragma y la velocidad de obturación se utilizan dos reglas anidadas basadas en las relaciones de las tablas 3, 4, y 5.

ISO					
100	11	12	13	14	15
200	12	13	14	15	16
400	13	14	15	16	17
800	14	15	16	17	18
1600	15	16	17	18	19

Tabla 3. Relación entre luminosidad, sensibilidad y VE

f	125	250	500	1000	2000
f 4.0	11	12	13	14	15
f 5.6	12	13	14	15	16
f 8.0	13	14	15	16	17
f 11.0	14	15	16	17	18
f 16.0	15	16	17	18	19

Tabla 4. Relación entre apertura, velocidad y VE

f	125	250	500	1000	2000
f 4.0					
f 5.6					
f 8.0					
f 11.0					
f 16.0					

Tabla 5. Relación entre apertura, velocidad y tipo de fotografía

Las reglas de inferencia del controlador se basan en la información de las tablas anteriores para seleccionar la combinación de apertura de diafragma y velocidad de obturación más apropiadas a las condiciones de luminosidad y el tipo de fotografía que se desea. Por ejemplo, para determinar el valor de la exposición se utilizan reglas que combinan distintos niveles de luminosidad con cada valor de sensibilidad. A continuación se muestran las reglas codificadas en el lenguaje XFL3 de Xfuzzy [15, 16]

```
if(ISO==iso100 & luminosidad==muyBaja) -> VE=ve11;
if(ISO==iso100 & luminosidad==baja) -> VE=ve12;
if(ISO==iso100 & luminosidad==media) -> VE=ve13;
```

La relación entre la apertura, la velocidad, el valor de exposición y el tipo de fotografía, se recoge en las tablas 4 y 5. A continuación se muestran ejemplos de las reglas de inferencia en lenguaje XFL3 [16] para VE 11, VE 12 y VE 13. Estas reglas utilizan el mínimo para el operador lógico AND. Durante las pruebas del sistema se han comparado los resultados del uso de las funciones MAX y Centroide para el proceso de 'deborrosificación' de las salidas. Finalmente, se ha optado por la función MAX porque da mejores resultados [13, 14].

```
if(VE==ve13 & fotografia==deportes) -> s=v500, f=f4x0;
if(VE==ve13 & fotografia==retrato) -> s=v250, f=f5x6;
if(VE==ve13 & fotografia==paisaje) -> s=v125, f=f8x0;
```

El controlador calcula el valor de exposición a partir de la relación entre la luminosidad y la sensibilidad ISO de la tabla 3. Una vez que se tiene el VE, es necesario encontrar la combinación de apertura de diafragma y velocidad de obturación más adecuada para el tipo de fotografía que se desea. Para ello, se busca el valor de exposición en la tabla 4 y se elige la combinación de apertura de diafragma y velocidad más adecuada utilizando la tabla 5. Por ejemplo, para los valores de entrada luminosidad "sol velado y sombras poco definidas", 100 ISO, fotografía de tipo deportes y distancia muy lejos, el controlador da como resultado VE 13, f 4.0 con s500 y distancia focal 100mm. Para calcular la apertura y la velocidad de obturación, primero se calcula el VE utilizando la tabla 3. Con las condiciones de luz "sol velado y sombras poco definidas" y sensibilidad 100 ISO se obtiene VE 13. En la tabla 4, existen tres valores de exposición VE 13, que corresponden con los valores f 4.0 con s500, f 5.6

con s250 y f 8.0 con s125. Para elegir una de estas combinaciones se busca en la tabla 5 cuál de ellas corresponde con el tipo de fotografía deportes. En este ejemplo se elige f 4.0 con s500. Esta combinación de diafragma abierto y la velocidad de obturación alta, produce una imagen con poca profundidad de campo, apropiada para deportes. La distancia es mayor a cuatro metros y se recomienda utilizar una lente de 100mm.

3.3 El desarrollo con Xfuzzy

La implementación del controlador se ha realizado con Xfuzzy, un software libre para diseño de sistemas borrosos desarrollado por el Instituto de Microelectrónica de Sevilla (CNM-IMSE). El código del controlador borroso se ha desarrollado en XFL3, un lenguaje para la definición de sistemas basados en lógica difusa [16]. El prototipo Java utiliza el código Java generado automáticamente por Xfuzzy [15, 17].

4 Resultados

A continuación se muestran los resultados que se han obtenido con un conjunto significativo de los casos de prueba, que permite ver cómo se comporta el sistema.

Entradas				Salidas			
Luminosidad	Fotografía	ISO	Distancia	VE	f	s	Focal
Muy baja	Retrato	100	Muy lejos	VE 11	f 4.0	s125	100
Muy alta	Deportes	800	Lejos	VE 18	f 16	s1000	80
Media	Retrato	100	Muy cerca	VE 13	f 5.6	s250	35
Baja	Deportes	400	Lejos	VE 14	f 5.6	s500	80
Muy alta	Retrato	800	Cerca	VE 18	f 22.0	s500	50
Media	Retrato	100	Intermedia	VE 13	f 5.6	S250	60
Muy baja	Paisaje	1600	Lejos	VE 15	f 11.0	s250	80
Muy alta	Deportes	400	Muy lejos	VE 17	f 11.0	s1000	100
Baja	Paisaje	100	Cerca	VE 12	f 8.0	s60	50
Media	Retrato	800	Cerca	VE 16	f11.0	s500	50
Alta	Deportes	1600	Intermedia	VE 18	f 16.0	s1000	60
Muy baja	Paisaje	200	Muy lejos	VE 12	f 8.0	s60	100
Media	Retrato	800	Lejos	VE 16	f 11.0	s500	80
Alta	Paisaje	200	Muy cerca	VE 15	f 11.0	s250	35
Muy baja	Retrato	400	Cerca	VE 13	f 5.6	s250	50
Baja	Deportes	1600	Intermedia	VE 16	f 8.0	s1000	60
Muy baja	Paisaje	100	Cerca	VE 11	f 5.6	s60	50
Alta	Retrato	800	Lejos	VE 17	f 16.0	s500	80

5 Conclusiones y Trabajos Futuros

El controlador que se ha presentado en este artículo se basa lógica borrosa debido a la naturaleza de sus valores de entrada. El sistema de control utiliza conjuntos borrosos en sus reglas de inferencia y ofrece resultados precisos para la distancia focal, el

valor de exposición, la apertura del diafragma y la velocidad de obturación. Las reglas del sistema se basan en relaciones existentes entre los datos de entrada y buscan siempre la solución más apropiada para las condiciones de cada fotografía. Los resultados son satisfactorios y se asemejan al comportamiento de una persona con conocimientos de fotografía. Hay que tener en cuenta que el controlador utiliza un nivel de luminosidad impreciso y el valor de exposición calculado no siempre coincide con el de un exposímetro. Además, este controlador está limitado por el rango de velocidades y de aperturas de diafragma de la figura 2. Para utilizar una gama de velocidades mayor, solo se debe ampliar esta tabla y las reglas de inferencia asociadas. Por último, es conveniente decir que basta con modificar las tablas de datos que definen las reglas de inferencia para ajustar el comportamiento del controlador.

Referencias

1. Lee, Y. M. A Fuzzy Control Processor for Automatic Focusing, IEEE Trans. on Consumer Electronics, 40, 2, 138-144, 1994
2. Haruki, T.; Kikuchi, K., Video camera system using fuzzy logic. Consumer Electronics, IEEE Transactions on , vol.38, no.3, pp.624,634, Aug 1992
3. Jiang T.; Kuhnert, K.-D.; Duong Nguyen; Kuhnert, L. Multiple templates auto exposure control based on luminance histogram for onboard camera. Computer Science and Automation Engineering (CSAE), 2011 IEEE International Conference on, On page(s): 237 - 241 Volume: 3, 10-12 June 2011
4. Shimizu S. et al. A new algorithm for exposure control based on fuzzy logic for video cameras IEEE Transactions on Consumer Electronics, 1992, 38(3): 617-623
5. Lee J.S. et al. An advanced video camera system with robust AF, AE and AWB control. IEEE Transactions on Consumer Electronics, 2001, 47(3): 694-699
6. Murakami, M.; Honda, N. An exposure control system of video cameras based on fuzzy logic using color information. Fuzzy Systems, 1996., Proceedings of the Fifth IEEE International Conference on, On page(s): 2181 - 2187 vol.3 Volume: 3, 8-11 Sep 1996
7. Liang J.Y.; Yajie Qin; Zhiliang Hong An Auto-exposure algorithm for detecting high contrast lighting conditions. ASIC, 2007. ASICON '07. Page(s): 725 – 728
8. Liu Y-C.; Wen-Hsin Chan; Ye-Quang Chen, Automatic white balance for digital still camera, Consumer Electronics, IEEE Transactions on , vol.41, no.3, pp.460,466, Aug 1995
9. Elamvazuthi, I., Vasant, P. Webb, J. (2009). "The Application of Mandani Fuzzy model for Auto Zoom Function of a Digital Camera". Applied Bioinformatics. 1, 4, 191-122.
10. Kuno, T., Sugiura, H., Matoba, N. A new automatic exposure system for digital still cameras. IEEE Transactions on Consumer Electronics, Vol. 44, No. 1. 192-199. (1998)
11. Pradera, A. El libro de la fotografía. Alianza Editorial. 1997
12. Weston,C. Domine la exposición digital. Index Book. 2009
13. Jantzen, J. Design of Fuzzy Controllers. (1998). Technical University of Denmark, Department of Automation. Tech. report no 98-E 864 (design), 19 Aug 1998
14. Santos, M. Un enfoque aplicado del control inteligente, RIAI, 8, 283-296 (2011)
15. Xfuzzy Team. Herramientas de CAD para lógica difusa. CNM IMSE (2012) http://www2.imse-cnm.csic.es/Xfuzzy/index_sp.html
16. Xfuzzy Team. XFL3 El lenguaje de especificación de Xfuzzy 3. http://www2.imse-cnm.csic.es/Xfuzzy/Xfuzzy_3.3/xfl/index_sp.html
17. Xfuzzy Team. Entorno de desarrollo Xfuzzy 3. CNM IMSE (2012), http://www2.imse-cnm.csic.es/Xfuzzy/Xfuzzy_3.3/tools/xfuzzy_sp.html

Sistema fuzzy de decisión para la gestión de capital en inversión en bolsa

Rodrigo Naranjo, Javier Arroyo, Albert Meco, Matilde Santos

(rnaranjo.ina@gmail.com)

Abstract. En este trabajo se presenta un sistema de gestión de capital basado en lógica difusa. A diferencia de otros trabajos anteriores que usan un sistema de trading más simple, en este caso se han definido nuevos indicadores borrosos (o *fuzzy*) que permiten no sólo orientar en la decisión de cuándo entrar o salir del mercado, sino también en la cantidad a invertir o a retirar en cada una de las operaciones. También se ha usado otra técnica de soft computing, los algoritmos genéticos para la optimización de los parámetros del sistema. Por lo tanto se ha desarrollado un sistema de ayuda a la toma de decisión en la gestión del capital con técnicas borrosas. Esta propuesta ha sido simulada y comparada con otras estrategias clásicas asentadas en la bolsa con resultados satisfactorios.

Keywords: trading, lógica borrosa, algoritmos genéticos, toma de decisiones, bolsa, f-óptima, gestión de capital.

1 Introducción

Dentro del área financiera, una de las principales preocupaciones ha sido la predicción de los movimientos de los precios en los mercados bursátiles. Diversos estudios muestran que el rendimiento del mercado bursátil es predecible en cierto grado. Por ejemplo en [1], utilizando datos de mercados bursátiles, registraron una correlación positiva en el rendimiento semanal.

Para poder predecir el movimiento de los precios futuros se han utilizado diversas técnicas y métodos, basándose en el análisis del histórico de los mercados. Algunos trabajos, como el de [2], utilizan algoritmos genéticos para encontrar reglas de trading que superen a la estrategia clásica *Buy & Hold*; también se han aplicado redes neuronales, cuyo primer exponente en este campo fue White en 1988 [3], quien utiliza un perceptrón multicapa alimentado de series de precios y volúmenes de la empresa IBM, y obtiene resultados favorables en comparación con un modelo lineal autoregresivo. Se han usado también árboles de regresión [4] y máquinas de vectores soporte [5], comparando el sistema basado en máquinas de vectores soporte con otro basado en árboles predictores aleatorios. En [6] se utiliza un modelo de lógica clásica y otro de lógica borrosa para predecir el signo de la variación del índice bursátil IPSA. En resumen, en el ámbito de la inversión en bolsa se han usado diversas técnicas de Soft Computing, que han resultado muy útiles.

Sin embargo, algunos estudios muestran que no es suficiente predecir el movimiento de los precios del mercado, sino que además es necesario tener una gestión de capital eficiente que indique la cantidad a invertir. Con este objetivo se han aplicado diferentes técnicas, algunas de ellas sencillas, como por ejemplo el modelo de lote fijo, fracción fija, fracción fija periódica, fracción fija del beneficio, etc., donde realmente no existe un ratio equilibrado entre la progresión de los beneficios y el riesgo asumido en cada operación. Otras, más sofisticadas, intentan optimizar dicho ratio, como por ejemplo, la fórmula de Kelly [7], la f-óptima [8] y sus variantes, Fixed Ratio [9], etc.

En definitiva, para implementar un sistema completo de trading, entendiendo como tal el conjunto de señales que identifican alertas en los valores a invertir, con la finalidad de conocer cuándo operar tanto en la entrada como en la salida del mercado, se debe incorporar un sistema de gestión de capital, el cual indica la cantidad de capital a invertir para obtener una progresión de los beneficios, controlando en todo momento el riesgo deseado.

Por tanto, y según la descripción anterior, el sistema completo de trading planteado en el presente trabajo se muestra en la figura 1:

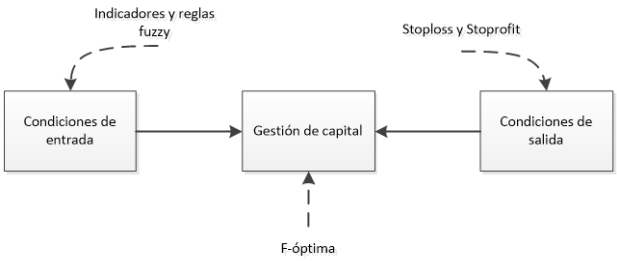


Fig. 1. Componentes del sistema de trading propuesto

Como se observa en la figura 1, las condiciones de entrada al mercado la componen un conjunto de indicadores y reglas borrosas que indican el momento adecuado para la entrada en el mercado, ya bien sea en corto o en largo. A continuación el sistema de gestión de capital implementado, basado en la f-óptima de Ralph Vince, utiliza la información proporcionada por el sistema borroso para calcular la cantidad de capital con la que se debe entrar en el mercado. A su vez establece las condiciones de salida según el riesgo deseado, mediante dos conceptos: *stop-loss* y *stop-profit*.

En este trabajo se han implementado dos sistemas, el primero de ellos basado en indicadores y reglas borrosas, así como en la f-optima. El segundo sistema se ha desarrollado como un sistema de reglas fijas y una estrategia de gestión de capital todo-nada. En ambos casos se han utilizado las mismas condiciones de salida para que no influya este factor en la comparación de los resultados obtenidos.

2 Sistema de trading

En este apartado se detallan los diferentes componentes del sistema de trading que se muestran en la figura 1.

2.1 Indicadores

En este trabajo se ha utilizado un conjunto de 3 indicadores técnicos. Concretamente se trata del indicador RSI de *l* sesiones, ADX de *l* sesiones y un indicador personalizado basado en el cruce de medias de *m* y *n* sesiones con filtrado MACD.

Los parámetros *m* y *n* serán el número de sesiones para el cálculo de las medias móviles, siendo $m < n$ (*m*: media móvil rápida y *n*: media móvil lenta). El parámetro *l* será una media (truncada a entero) de los anteriores parámetros *m* y *n*. Además se utilizará en parámetro *s* que indica el suavizado necesario para el cálculo del indicador MACD.

Dichos parámetros (*m*, *n* y *s*) vendrán determinados por el proceso de optimización realizados para cada valor aplicado.

RSI – Relative Strenght Index

El RSI es un indicador muy popular entre los analistas técnicos. Su valor oscila entre el 0 y el 100. Muestra la fuerza del valor comparando el último cierre del valor con valores anteriores. Una interpretación común del RSI es suponer un estado de sobreventa para valores bajos (< 30), y análogamente, estados de sobrecompra para valores altos (> 70).

ADX – Average Directional Index

El ADX pertenece también a la familia de osciladores. Fluctúa entre 0 y 100. Muestra la fuerza de la tendencia actual del mercado. En general, si el ADX tiene un valor mayor de 20, consideramos que el mercado está en tendencia. Para valores menores que 20, suponemos una situación de lateralidad del valor. Valores mayores de 40 indican fuerte tendencia.

MyMACD – My Moving Average Convergence Divergence

El indicador MyMACD es un indicador personalizado implementado para este trabajo. Dicho indicador se basa en el uso de cruces de 2 medias móviles, una rápida y otra lenta, denominadas así ya que la primera de ellas utiliza un número de sesiones inferior a la segunda y, por tanto, es más sensible a los cambios en el precio del valor. Además, para evitar las falsas señales de entrada en la medida de lo posible, se ha utilizado el indicador MACD (*Moving Average Convergence Divergence*) que proporciona, junto con la diferencia de dos medias móviles de diferente longitud, la diferencia entre una señal de referencia que corresponde al promedio móvil exponencial del indicador MACD y el propio MACD.

El valor proporcionado está acotado entre 0 y 100. Un valor centrado en 25 indica una señal de entrada en largo en el mercado; por el contrario, un valor centrado en 75 indica una entrada en corto al mercado.

2.2 Condiciones de entrada

Como se ha descrito anteriormente, los indicadores usados en el presente trabajo son el indicador RSI, ADX y MyMACD. Con la información proporcionada por estos indicadores, se han creado dos sistemas. El primero de ellos (Sistema Clásico), basado en reglas fijas y el segundo (Sistema Borroso) utiliza un sistema de reglas borrosas basado en la información que proporciona la borrosificación de dichos indicadores.

Para ambos sistemas se han creado dos salidas, que serán las encargadas de indicar la entrada al mercado. La primera de ellas, *Bullish*, indica que el mercando está en alza, y por tanto, se debería entrar en largo. La otra salida, *Bearish*, indica lo contrario, es decir, indica entrar en corto al mercado.

Sistema Clásico

Dado que este sistema se basa en un sistema de reglas fijas, las salidas proporcionadas (*Bullish* y *Bearish*) serán binarias, indicando si entrar o no en el mercado, pero en ningún momento el grado de fuerza con el que entrar. El sistema de reglas elegido es el siguiente (Tabla 1):

	MyMACD		RSI		ADX		Bullish
if	>15 &&<35	&&	>30 &&<85	&&	>25 &&<55	then	100
elseif	>15 &&<35	&&	>25 &&<85	&&	>55	then	100
elseif	>15 &&<35	&&	<25	&&	>25 &&<55	then	100
elseif	>15 &&<35	&&	<25	&&	<25	then	100
elseif	>15 &&<35	&&	<25	&&	>55	then	100
elseif	<15 || >85 || (>35 &&<65)	&&	<25	&&	>25 &&<55	then	100
else						then	0

	MyMACD		RSI		ADX		Bearish
if	>65 &&<85	&&	>85	&&	>25 &&<55	then	100
elseif	>65 &&<85	&&	>85	&&	<55	then	100
elseif	>65 &&<85	&&	>85	&&	>55	then	100
elseif	>65 &&<85	&&	>35 &&<85	&&	>25 &&<55	then	100
elseif	>65 &&<85	&&	>35 &&<85	&&	>55	then	100
elseif	<15 || >85 || (>35 &&<65)	&&	>85	&&	>25 &&<55	then	100
else						then	0

Tabla 1. Reglas para el Sistema Clásico

Sistema Borroso

Se ha usado un sistema tipo Mamdani, con operadores basados en el máximo y mínimo y el método del cálculo del centroide para la desborrosificación [10].

Los indicadores y las salidas se han borrosificado tal y como se indica a continuación (Figura 2). En todas las variables se han usado funciones de pertenencia triangulares no simétricas. Se han definido cinco conjuntos borrosos para la entrada MyMACD y 3 conjuntos borrosos para el resto de las entradas y para las salidas.

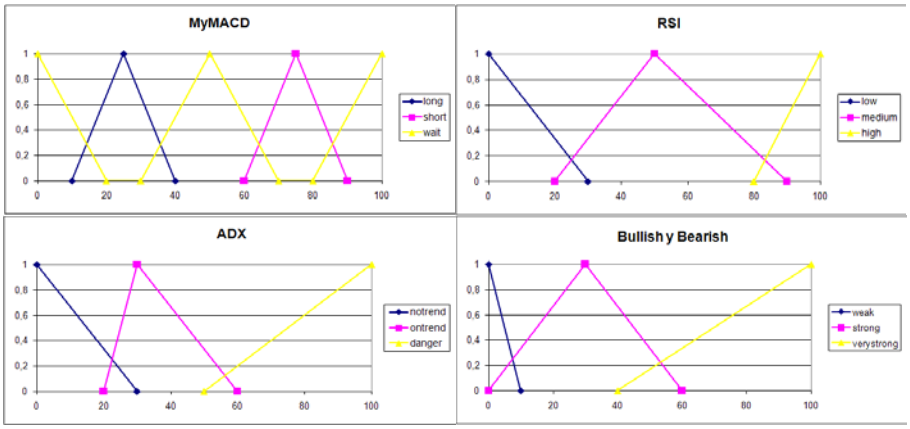


Fig. 2. Borrosificación de indicadores y salidas

Se ha usado un conjunto de reglas tipo *if-then*, en donde cada salida tiene tres posibles estados, *weak* en el cual el sistema no entrará en el mercado, y otros dos (*strong* y *verystrong*) en el que se entrará en el mercado con distinta fuerza (Tabla 2).

MyMACD	RSI	ADX	Bullish	Bearish
LONG	HIGH	ONTREND	WEAK	WEAK
LONG	HIGH	NOTREND	WEAK	WEAK
LONG	HIGH	DANGER	WEAK	WEAK
LONG	MEDIUM	ONTREND	VERYSTRONG	WEAK
LONG	MEDIUM	NOTREND	WEAK	WEAK
LONG	MEDIUM	DANGER	STRONG	WEAK
LONG	LOW	ONTREND	VERYSTRONG	WEAK
LONG	LOW	NOTREND	STRONG	WEAK
LONG	LOW	DANGER	STRONG	WEAK
SHORT	HIGH	ONTREND	WEAK	VERYSTRONG
SHORT	HIGH	NOTREND	WEAK	STRONG
SHORT	HIGH	DANGER	WEAK	STRONG
SHORT	MEDIUM	ONTREND	WEAK	VERYSTRONG
SHORT	MEDIUM	NOTREND	WEAK	WEAK
SHORT	MEDIUM	DANGER	WEAK	STRONG
SHORT	LOW	ONTREND	WEAK	WEAK
SHORT	LOW	NOTREND	WEAK	WEAK
SHORT	LOW	DANGER	WEAK	WEAK
WAIT	HIGH	ONTREND	WEAK	STRONG
WAIT	HIGH	NOTREND	WEAK	WEAK
WAIT	HIGH	DANGER	WEAK	WEAK
WAIT	MEDIUM	ONTREND	WEAK	WEAK
WAIT	MEDIUM	NOTREND	WEAK	WEAK
WAIT	MEDIUM	DANGER	WEAK	WEAK
WAIT	LOW	ONTREND	STRONG	WEAK
WAIT	LOW	NOTREND	WEAK	WEAK
WAIT	LOW	DANGER	WEAK	WEAK

Table 2. Reglas borrosas del sistema de trading

2.3 Condiciones de salida

Para ambos sistemas (clásico y borroso) se han establecido las mismas condiciones de salida (*stop-loss* y *stop-profit*).

Cuando el sistema indica una entrada en el mercado, se establece una parada de protección a partir del cual, si se supera, se establece la salida del mercado. De esta forma se acotan las máximas pérdidas por cada operación y servirá para el cálculo del capital a invertir. Del mismo modo se establece un umbral de ganancias (*stop-profit*) a partir del cual se realizaría la salida del mercado pero, a diferencia del caso anterior, por el hecho de superar las ganancias previstas. De este modo se aseguran unas ganancias prudenciales evitando que el mercado pueda dar un giro inesperado.

Ambos parámetros se calculan tomando como referencia la volatilidad del precio en el momento de la compra, usando el indicador ATR (Average True Range) de *m* sesiones, donde *m* tiene el mismo significado que el descrito en los apartados de los indicadores. Dicho valor proporcionado por el indicador ATR será calculado nuevamente cada cierre de sesión.

Concretamente el *stop-loss* se ha tomado como 2 veces el ATR de *m* sesiones y el *stop-profit* como 1,5 veces.

2.4 Gestión de capital

A diferencia del trabajo [11], en el cual el porcentaje de capital se indicaba directamente mediante la salida (comprendida entre 0 y 100) que proporcionaba el sistema borroso, en el presente trabajo se ha implementado un sistema de gestión de capital basado en la f-óptima que además utiliza la información proporcionada por la salida del sistema borroso.

La f-óptima proporciona la forma de calcular la cantidad exacta a invertir para producir un crecimiento geométrico de los beneficios. Para ello se basa en el resultado obtenido de las anteriores operaciones, incrementando el porcentaje a invertir cuando aumentan las operaciones ganadoras y por ende disminuyendo conforme llegan operaciones perdedoras. El resultado proporcionado es un porcentaje a aplicar sobre el capital a invertir. Sin embargo, a pesar de ser teóricamente el valor óptimo, la mayoría de las veces implica correr un riesgo demasiado elevado debido al alto valor obtenido. Por ello se suele utilizar la llamada f-líquida (o f-fraccional) que se sitúa en un 10% de la f-óptima.

Sin embargo, en ocasiones donde la seguridad de estar ante una operación ganadora es alta, marcada por la salida del sistema borroso planteado (en este caso correspondería a un valor alto, cercano a 100), dicha f-líquida podría ser insuficiente, es decir, podría asumirse correr más riesgo (y por lo tanto la posibilidad de obtener mayores beneficios) ya que el valor de la f-líquida sería demasiado conservador. Por el contrario, ante entradas en el mercado no tan seguras (valores bajos en la salida del sistema borroso), un riesgo conservador como el que ofrece la f-líquida sería idóneo para este caso.

Concretamente, si representamos la curva de la ganancia frente al riesgo asumido (ver figura 3) se observaría como la f-óptima se situaría en el máximo de la curva. La f-líquida, al ser ésta un 10% de la f-óptima, se situaría en la parte izquierda.

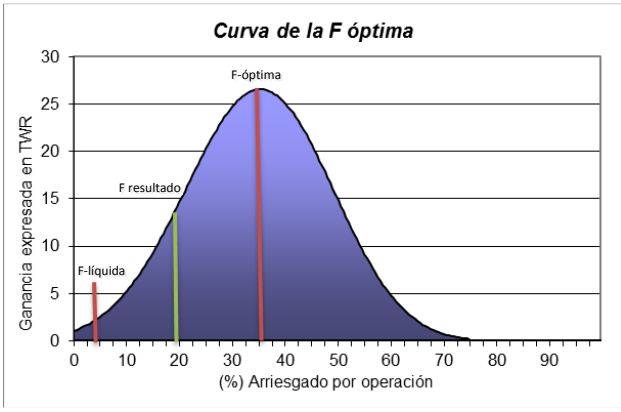


Fig. 3. Representación de f-óptima, f-líquida y f sobre curva de ganancia en función del riesgo asumido por operación

Por tanto se tiene un margen de actuación limitado por un mínimo (f-líquida) y un máximo (f-óptima). El valor de salida del sistema de gestión de capital será un valor comprendido dentro de ese margen debido a una variación proporcionalmente lineal en función del valor obtenido del sistema borroso.

3 Discusión de resultados de las pruebas

Para obtener los resultados que se muestran a continuación, se han implementado ambos sistemas en la plataforma Ninjatrader y se han optimizado utilizando un optimizador propio basado en algoritmos genéticos incorporado en la plataforma sobre los 10 primeros valores del Nasdaq 100. Concretamente se han optimizado los sistemas en el período de un año comprendido entre el 22/12/2010 hasta el 22/12/2011. De dicha optimización se han obtenido los valores óptimos de los parámetros *m*, *n* y *s* anteriormente descritos. A continuación se han validado los sistemas durante el año posterior, obteniéndose los siguientes resultados (Tabla 3).

Si se analizan los resultados obtenidos utilizando el procedimiento de ventana fija (Tabla 3), se observa que de los 10 valores, en 6 de ellos el sistema borroso obtiene mejores resultados que el sistema clásico. Además el sistema borroso tan sólo en 2 de los 10 valores obtiene pérdidas, por el contrario el sistema clásico obtiene pérdidas en la mitad de los casos. Si se observan los resultados obtenidos, se puede deducir que el Sistema borroso obtiene unos resultados suavizados en las ganancias y muy suavizados en las pérdidas con respecto al sistema clásico.

		AAPL	ADBE	ADP	ADSK	AKAM	ALTR	ALXN	AMAT	AMGN	AMZN
Fuzzy	Net Profit	446,99	457,41	-70,55	77,70	-145,00	173,78	91,22	181,77	91,40	593,67
	Max DrawDown	-15,62	-7,81	-1,50	-10,39	-21,28	-5,25	-16,22	-5,60	-11,24	-5,00
	Trades total	21,00	6,00	1,00	3,00	5,00	4,00	16,00	4,00	7,00	7,00
	Trades +	12,00	4,00	0,00	1,00	3,00	2,00	9,00	3,00	3,00	5,00
	Trades -	9,00	2,00	1,00	2,00	2,00	2,00	7,00	1,00	4,00	2,00
	AverageTrade (%)	0,53	2,23	-1,50	0,93	-1,55	1,87	0,30	1,40	0,22	2,31
	AverageWinningTrade (%)	4,35	4,93	0,00	14,25	4,81	8,44	5,24	4,07	5,01	4,10
	AverageLosingTrade (%)	-3,88	-4,00	-1,50	-6,12	-11,37	-4,91	-6,13	-5,60	-3,54	-4,51
	• Ratio avg. Win / avg. Loss	1,12	1,23	0,00	2,33	0,42	1,72	0,85	0,73	1,42	0,91
	Sharpe Ratio	0,23	0,24	1,00	-0,04	-0,66	0,10	0,08	2,71	-0,01	0,34
Clásica	Net Profit	963,74	-222,25	224,82	-137,11	-1117,04	-313,11	1161,14	-898,29	188,25	189,92
	Max DrawDown	-7,50	-3,71	-3,06	-9,78	-17,86	-5,25	-0,47	-15,06	-3,58	-4,10
	Trades total	10,00	1,00	3,00	3,00	3,00	1,00	5,00	4,00	7,00	3,00
	Trades +	6,00	0,00	2,00	1,00	0,00	0,00	4,00	1,00	3,00	2,00
	Trades -	4,00	1,00	1,00	2,00	3,00	1,00	1,00	3,00	4,00	1,00
	AverageTrade	1,79	-3,71	1,33	-0,73	-6,34	-5,25	3,64	-3,66	-0,93	0,76
	AverageWinningTrade	5,20	0,00	3,44	8,03	0,00	0,00	4,88	4,15	2,91	3,37
	AverageLosingTrade	-3,58	-3,71	-3,06	-5,03	-6,34	-5,25	-0,47	-6,58	-3,59	-3,58
	• Ratio avg. Win / avg. Loss	1,45	0,00	1,13	1,60	0,00	0,00	10,46	0,63	0,81	0,94
	Sharpe Ratio	0,29	1,00	2,05	-3,88	-4,76	1,00	0,89	-29,10	-0,21	0,14

Tabla 3. Tabla comparativa de resultados obtenidos con ventana fija

Otra de las pruebas realizadas sobre la misma cartera de valores ha consistido, en vez de utilizar una ventana fija de optimización y validación como en el caso anterior, en utilizar una ventana deslizante de 3 meses. El procedimiento ha consistido en optimizar los sistemas durante un año y validarlo en los 3 meses siguientes. A continuación se optimiza de nuevo los sistemas pero desplazando el período de optimización 3 meses en adelante. Una vez optimizado, se valida en los otros 3 meses siguientes, y así sucesivamente hasta completar el año. Por tanto se realizará 4 veces el ciclo optimización-validación. En la tabla 4 se pueden comparar los resultados obtenidos con este procedimiento y con el anteriormente mostrado:

		AAPL	ADBE	ADP	ADSK	AKAM	ALTR	ALXN	AMAT	AMGN	AMZN
Fuzzy	Optimización 1 año	446,99	457,41	-70,55	77,70	-145,00	173,78	91,22	181,77	91,40	593,67
	Ventana deslizante 3 meses	593,27	405,75	346,50	407,72	98,53	390,50	272,83	390,86	148,30	498,92
	Diferencia ($)	146,28	-51,66	417,05	330,02	243,53	216,72	181,61	209,09	56,90	-94,75
Clásica	Optimización 1 año	963,74	-222,25	224,82	-137,11	-1117,04	-313,11	1161,14	-898,29	188,25	189,92
	Ventana deslizante 3 meses	926,14	39,89	444,02	17,04	-168,82	-267,75	507,03	-769,21	-594,29	206,08
	Diferencia ($)	-37,60	262,14	219,20	154,15	948,22	45,36	-654,11	129,08	-782,54	16,16

Tabla 4. Tabla comparativa resultados obtenidos con ventana fija y deslizante

Al validar ambos sistemas, utilizando el método de la ventana deslizante de 3 meses, las diferencias se hacen más notables. En este caso, el sistema clásico sigue teniendo pérdidas, esta vez en sólo 4 casos, con lo que se ha reducido levemente el efecto del

sistema de validación; además se ha empeorado de forma notable los resultados en 3 casos. Sin embargo, en el sistema borroso han mejorado notablemente, ya que no se han obtenido pérdidas en ninguno de los casos, y tan sólo en 2 de ellos se ha empeorado levemente los resultados anteriormente obtenidos. Ello es debido a que en este sistema, el resultado de las operaciones obtenidas en la optimización se ha introducido como valor de entrada para el cálculo de la f-óptima durante el período de validación. Por tanto, al tener más operaciones, el peso que ofrece el cálculo de dicha f-óptima es mayor que con pocas operaciones, pudiendo gestionar el capital de forma más eficiente.

Un aspecto donde el sistema borroso no sale muy beneficiado es en el del máximo Draw Down, que mide el descenso máximo (en porcentaje) que se produce en la curva de capital a lo largo del periodo considerado. El sistema borroso obtiene en general unos Draw Downs máximos superiores al sistema clásico. Conclusiones

De los resultados obtenidos tras la simulación de ambos sistemas, se observa que la sinergia de un sistema borroso relativamente sencillo con tres indicadores optimizados mediante algoritmos genéticos, y una gestión de capital que acepte como entrada la información que proporciona dicho sistema borroso, permite obtener una mejora notable en comparación con la versión clásica del mismo sistema.

Las pruebas realizadas resultan limitadas porque se centran en diez compañías de un mismo país, en un mismo sector y en un mismo marco temporal donde la mayoría de dichas compañías siguen una senda alcista. Por tanto, una posible ampliación de trabajo consiste en aumentar el marco de pruebas para determinar cómo de efectivo resulta el sistema borroso en otros contextos.

En las pruebas realizadas se intuye que el sistema borroso ofrece una mayor probabilidad de obtener beneficios, y además, una mayor probabilidad de obtener beneficios más altos (en media) durante el período de validación. En el caso de obtener pérdidas, el efecto que produce utilizar el sistema borroso es suavizar dichas pérdidas, frente al sistema clásico en el que las pérdidas son más acentuadas, llegándose a producir pérdidas por encima del 10% del capital inicial. Sin embargo, el alto Draw Down del sistema borroso es un aspecto a intentar corregir antes de usar el sistema para inversión real.

.Una manera sencilla de evitar el alto DrawDown consistiría en dividir el capital disponible en partes iguales entre los diez valores. En las pruebas realizadas hemos podido comprobar que los momentos en los que el sistema decide entrar en cada uno de los valores son distintos y que las pérdidas y las ganancias de cada valor no están correlacionadas, es decir, no se producen necesariamente a la vez. Este hecho, sumado a que nuestro sistema en media proporciona ganancias en el periodo considerado, hace que la gestión de capital que divide el capital en partes iguales aplicada sobre nuestro sistema dé lugar a una curva de capital con una pendiente positiva moderada y con dientes de sierra pequeños. Este tipo de curvas de capital se consideran muy atractivas para los inversores porque exhiben poco riesgo y son susceptibles de obtener mayores beneficios si se utiliza apalancamiento.

Por ello, el trabajo futuro consiste en realizar unas pruebas más amplias para confirmar las propiedades del sistema propuesto: esperanza positiva (es decir, que en media da ganancias) y ausencia de correlación entre ganancias y pérdidas al aplicar el sistema sobre diferentes valores. También sería interesante trabajar en mitigar el Draw Down del sistema borroso en sí, ya que en muchos casos puede ser inasumible para el inversor.

Bibliografía

1. Lo, A. W., & MacKinlay, A. C. Stock market prices do not follow random walks: Evidence from a simple specification test. Review of financial studies, 1(1), 41-66, 1988.
2. Allen, F., Karjalainen, R.. Using genetic algorithms to find technical trading rules. Journal of financial Economics, 1999, vol. 51, no 2, p. 245-271.
3. White, H. “Economic prediction using neural networks: the case of IBM daily stock returns. Neural Networks in Finance and Investing”, Ed. Trippi y Turban, 1988
4. Ghia, K., "Statistical applications for finance: Regression tree and distribution-based models for equity trading", 2007
5. Kumar, M., Thenmozhi, M. “Forecasting Stock Index Movement: A Comparison of Support Vector Machines and Random Forest”. Indian Institute of Capital Markets 9th Capital Markets Conference Paper, 2006
6. Parisi A., Rebolledo J., Cornejo, E., “Modelos de lógica y lógica borrosa en la predicción del IPSA”, 2006.
7. Kelly Jr. J. “A new interpretation of information rate”, Information Theory, IRE Transactions on, 2, 3, 185-189, 1956
8. Vince R., “Portfolio Management Formulas” Willey and Sons, New York, 1990
9. Jones R., ”The Trading Game”, Wiley and Sons, 1999
10. Zadeh, L.A. “Fuzzy Sets”, Information and Control, 8, 3, 338-353, 1965
11. Meco, A. “Sistema de trading fuzzy basado en reglas sobre indicadores técnicos”, Trabajo de la asignatura ”Control Inteligente. Universidad Complutense de Madrid, 2011

Sistema de recomendación colaborativo de la aplicación Hábitat Madrid

Inmaculada Pardines [1], Victoria López[1], Antonio Sanmartin[2], Mar Octavio de Toledo[2], Carlos Fernández[2]

[1] Dep. Arquitectura de Computadores, Universidad Complutense de Madrid
Profesor García Santesmases, s/n,
28040 Madrid, Spain

{inmapl,vlopezlo}@fdi.ucm.es
[2] Facultad de Informática, Universidad Complutense de Madrid
Profesor García Santesmases, s/n,
28040 Madrid, Spain

{antonio.sanmartin9,mar.otr23, carlitos.fv5}@gmail.com

Abstract. En este trabajo se propone la utilización de un sistema de recomendación colaborativo que aplica técnicas lingüísticas difusas sobre una herramienta móvil que gestiona la información e inscripción en el programa de actividades medioambientales del Ayuntamiento de Madrid. La aplicación informa al usuario de todas las actividades: talleres, itinerarios, exposiciones, jardinería, etc., que se realizan en los centros de información y educación ambiental del Ayuntamiento de Madrid. Debido al exceso de información que puede llegar al usuario resulta imprescindible que exista un sistema que permita ofrecer las actividades que más le pueden interesar al usuario. Estas actividades son seleccionadas en base a un perfil de preferencias del propio usuario y a un histórico con la valoración de las actividades previamente realizadas por otros usuarios, sobre todo, por aquellos con un perfil similar.

Keywords. Sistemas de recomendación, filtrado colaborativo, aplicación móvil, actividades medioambientales, lógica difusa

1 Introducción

Las aplicaciones móviles han tenido un gran desarrollo en los últimos años, lo que ha influido en el incremento del uso del teléfono móvil para realizar muchas de las tareas que antes el usuario realizaba en su ordenador o a través de Internet. Esta revolución en el mundo de la tecnología móvil se ha visto acelerada por la aparición de los smartphones [1]. Estos dispositivos no son simples teléfonos, sino que soportan clientes de correo electrónico, GPS, navegador Web, aplicaciones como bloc de notas, organizador, grabador de voz, etc. , propias de un computador.

Esta revolución que ha experimentado el mercado de la telefonía móvil ha modificado los modelos de negocio. Existen tiendas on-line, accesibles desde los propios móviles, como Android Market de Google [2], iTunes Store de Apple [3] o Ovi Store de Nokia [4], de donde los usuarios se pueden descargar una gran variedad de aplicaciones para sus móviles.

En la actualidad muchas áreas de conocimiento tratan de explotar el uso de estos dispositivos. En el campo de la educación aparece el concepto de m-learning [5-6]. Existen estudios que avalan el uso de los teléfonos móviles en el proceso de aprendizaje, debido a que incrementan la motivación de los estudiantes y permiten un ambiente de aprendizaje colaborativo y competitivo. En medicina, muchas especialidades han incorporado el uso de esta tecnología como ayuda en el trabajo clínico [7]. Especialmente útiles resultan las soluciones propuestas para facilitar las tareas de la vida cotidiana, como la realización de la compra en grandes superficies comerciales [8]. Existe otro tipo de aplicaciones que permiten la visualización de mapas y proporcionan al usuario información. Aplicaciones de este tipo suelen utilizarse en el área de ocio y tiempo libre; entre ellas destacan visitas virtuales a ciudades o museos [9-10]. En este contexto se engloba la aplicación móvil Hábitat Madrid que se presenta en este artículo. Se trata de una aplicación que informa de todas las actividades que se realizan en los centros medioambientales del Ayuntamiento de Madrid. A través de esta aplicación el usuario no solo se puede informar de las actividades que se van a realizar sino también puede tramitar su inscripción en las mismas.

La herramienta dispone de información sobre las actividades que se ofertan durante cada trimestre del año. El número de actividades es muy variado, siendo la cantidad de información disponible muy extensa, lo que puede hacer que la búsqueda de información resulte una tarea ardua y poco atractiva. Para resolver este problema proponemos el uso de un sistema de recomendación que permita personalizar la información que reciba cada usuario de acuerdo con sus preferencias.

Los sistemas de recomendación son muy utilizados sobre todo en comercio electrónico [11]. Son una herramienta complementaria dentro de una tienda electrónica que muestra al cliente, cuando realiza una compra, una serie de artículos relacionados con lo que ha comprado, o bien, muestra artículos que han comprado otros usuarios después de comprar el mismo producto que en ese momento está comprando él. Estos productos pueden interesar al cliente pero si tuviese que buscarlos dentro de la tienda, al disponer ésta de muchos productos, puede que no los hubiese encontrado.

En la literatura existen distintos modelos de sistemas de recomendación que difieren en las fuentes de información usadas y en cómo se obtienen las recomendaciones. Se pueden destacar: sistemas de recomendación colaborativos [12], sistemas basados en contenido [13], sistemas basados en información demográfica [14], sistemas basados en conocimiento [15] y sistemas basados en utilidad [16].

En este artículo se propone un sistema de recomendación colaborativo integrado dentro de una aplicación móvil que gestiona la información y reserva de las actividades que desarrollan los centros medioambientales del Ayuntamiento de Madrid. En este sistema las recomendaciones se hacen en base a un perfil de preferencias proporcionado por el propio usuario, un análisis de las similitudes existentes con otros usuarios y la opinión de éstos, sobre todo, de aquellos con perfiles similares tengan de las actividades en las que han participado.

El artículo está organizado de la siguiente forma. En la sección 2 se hace una breve revisión de los modelos de recomendación y de los conceptos de las técnicas de lógica difusa. En la sección 3 se describe el sistema de recomendación propuesto. En la sección 4 se presenta un ejemplo de funcionamiento de la aplicación. Por último, se muestran las principales conclusiones que se pueden extraer de este trabajo.

2 Sistemas de recomendación y lógica difusa

El sistema de recomendación de la aplicación móvil propuesta es básicamente un sistema colaborativo. Este tipo de sistemas recogen valoraciones de los usuarios respecto a un objeto dentro de un dominio dado y generan las recomendaciones en base a las opiniones de otros usuarios de perfil similar. Por lo tanto, su implementación requiere información histórica sobre la valoración que han realizado los usuarios de objetos que han seleccionado en el pasado.

Un sistema colaborativo deberá:

- Guardar un histórico de las valoraciones de los usuarios sobre todos los objetos que han seleccionado.
- Medir el grado de similitud entre los distintos usuarios en base a sus preferencias y valoraciones.
- Generar las recomendaciones que mejor se adaptan a un usuario dado a partir de la información aportada por los demás usuarios del sistema.

Un sistema colaborativo será capaz de realizar una recomendación a un usuario sobre un objeto del dominio que todavía no haya evaluado en base a las evaluaciones de otros usuarios similares a él. Sin embargo, el punto débil de este sistema está en la escasa calidad de las recomendaciones, debida a la falta de información, las primeras veces que un usuario accede al sistema. En la literatura existen distintas aproximaciones para tratar de aliviar este problema [17]. Una solución habitual es utilizar sistemas híbridos que mezclan las características positivas de varios modelos dejando a un lado las características negativas. Se pueden analizar otras soluciones mediante el uso de métodos de recomendación basados en lógica borrosa [18].

Otro problema que surge en los sistemas de recomendación es el de cómo realizar las evaluaciones. Hay dominios en los que la calidad de sus objetos puede ser medida de forma numérica, pero en otros resulta más apropiado realizar esta valoración utilizando palabras, como divertido, largo, breve, bueno, malo,…, en lugar de valores numéricos. Las técnicas de lingüística difusa han resultado adecuadas para modelar este tipo de información [19]. Las variables lingüísticas, introducidas por Zadeh [20], se utilizan para representar esta información. Una variable lingüística se caracteriza por un valor sintáctico (etiqueta) y por un valor semántico (significado). Zadeh en [20] define formalmente una variable lingüística caracterizándola mediante una tupla *(X, U, R(X:u))*, donde *X* es el nombre de la variable; *U* es el universo de discurso; *u* es un nombre genérico para los elementos de *U* y *R(X:u)* es un subconjunto de *U* que representa una restricción en los valores de *u* impuesto por *X*.

3 Sistema de recomendación de la aplicación Hábitat Madrid

La aplicación Hábitat Madrid es una aplicación móvil para sistemas Android [21-22] que muestra al usuario todas las actividades ofertadas por los Centros de Educación Ambiental del Ayuntamiento de Madrid. El número de actividades que proponen estos centros es tan numeroso que para que un usuario busque información sobre las mismas en su móvil de forma rápida y cómoda resulta imprescindible el uso de un sistema de recomendación. Además, para que la información ofrecida se visualice en el dispositivo móvil de forma adecuada el número de actividades recomendadas debe ser pequeño.

De acuerdo con lo dicho y para no limitar la capacidad de elección propia del usuario se ha decidido implementar un sistema de recomendación que ordena las actividades en función de lo atractivas que pueden resultar al usuario. La aplicación, tras pulsar la pestaña de Recomendaciones, muestra al usuario una pantalla con las cuatro actividades que mejor se pueden adaptar a sus gustos y preferencias. Sin embargo, el usuario tendrá la opción de seguir buscando otras actividades que le puedan interesar sin más que pulsar la pestaña Siguiente que aparecerá en esa misma pantalla. De este modo, la aplicación irá mostrando al usuario las aplicaciones de cuatro en cuatro siguiendo un orden, previamente calculado, adaptado al perfil del usuario.

3.1 Descripción general del problema

El sistema de recomendación está caracterizado básicamente por un conjunto de actividades y un conjunto de usuarios. Su objetivo es informar a un usuario determinado de las actividades que más le pueden interesar. De forma general, se puede describir del siguiente modo:
Dado el conjunto de actividades $A = \{a_1, \dots, a_m\}$, cada actividad $a_i \in A$ está definida por un título y un descriptor que resume en qué consiste la actividad y a quién va dirigida.

Además, el sistema tendrá un conjunto de usuarios $U = \{u_1, \dots, u_n\}$, cada usuario $u_j \in U$ deberá rellenar un perfil con sus preferencias cuando se registra en la aplicación. Estas preferencias son seleccionadas entre una serie de palabras clave (itinerarios, exposiciones, jardinería, familiar,…) relacionadas con el tipo de actividades que el usuario desea realizar.

El sistema consta también de un histórico con las valoraciones que los usuarios hacen de las actividades en las que participan. En cada una de las actividades se evalúan tres características:

- **Entorno y recursos**: Mal, regular y bien
- **General**: Mal, regular y bien
- **Personal**: Mal, regular y bien

La característica *Entorno y recursos* pretende valorar tanto el lugar en el que se realiza la actividad como los medios de los que se dispone para poder realizarla. En *General* el usuario deberá evaluar la impresión que le ha causado la actividad en su conjunto. Y la característica *Personal* califica la validez de los monitores de la activi-

dad, para ello el usuario podrá fijarse en su capacidad de comunicación, conocimiento, actitud,…Cada característica se podrá evaluar eligiendo una de estas tres opciones: mal, regular y bien.

A partir del perfil introducido y de los perfiles de otros usuarios ya existentes en la aplicación, el sistema mide el grado de afinidad entre los distintos usuarios. El sistema de recomendación actúa en base a la información sobre los gustos y preferencias introducidas por los distintos usuarios y en base a las valoraciones que éstos hacen de cada una de las actividades en las que han participado. A continuación, se hace una descripción formal del funcionamiento del sistema de recomendación propuesto.

3.2 Modelado matemático

Dado un usuario u, el sistema debe atravesar tres etapas para generar una recomendación sobre las actividades que le puedan resultar más interesantes. A continuación, se realiza una descripción matemática de estas etapas que también se pueden visualizar mediante el diagrama de flujo de la Fig. 1.

- **Etapa 1: Valoración de las actividades**. El sistema de recomendación calcula dos valoraciones de todas las actividades del sistema, ambas valoraciones se realizan en paralelo.

  (a) **Valoración 1**: Se calcula en base a la valoración que otros usuarios han hecho de las actividades en las que han participado.

  $\forall\, a_i \in A, \forall\, u_j \in U$, Calcular:

  (i) $sim\,(u_j, u) = s_j, \quad s_j \epsilon\, [0,1]$ es el grado de similitud del usuario u_j con el usuario u

  (ii) $v\,(u_j, a_i) = \begin{cases} v_{ji} & si\ u_j\ realizó\ y\ valoró\ a_i \\ \perp & e.o.c. \end{cases}$

  v_{ji} es la valoración que el usuario u_j ha hecho de la actividad a_i

  (iii) $v\,(a_i, u) = \sum_{j=1}^{nu} s_j \cdot v_{ji} = v_i$,

  nu es el número de usuarios y v_i es la valoración 1 de la actividad a_i

  (b) **Valoración 2**: Se obtiene a partir del perfil de preferencias del usuario u.

  Dado un usuario $u, \forall\, a_i \in A$, Calcular la valoración 2, v_i', como:

$$v'(a_i, u) = \frac{n_i}{N} = v_i',$$

con

$$n_i = \sum_{k=1}^{N} x_k \,,$$

$$x_k = \begin{cases} 1 & si \;\; p_k \in preferencias\ de\ u\ y\ descriptor\ a_i\ contiene\ k \\ 0 & e.o.c. \end{cases}$$

p_k es una palabra clave, n_i es el número de palabras clave de a_i que coinciden con las preferencias que el usuario u ha marcado en su perfil y N es el número total de palabras clave que el usuario u ha seleccionado como preferencias.

- **Etapa 2: Ordenamiento.** Las actividades se ordenan en función de la valoración de la valoración obtenida en la etapa 1.

  (a) $\forall\ a_i \in A,\ \ u \in U$, calcular la valoración total v_i^* de la actividad a_i en base a las dos valoraciones de esta actividad calculadas para el usuario u en la etapa anterior.

$$v_i^* = v^*(a_i, u) = \ \alpha \cdot v(a_i, u) + (1 - \alpha) \cdot v'(a_i, u), \quad \alpha \in (0,1) \tag{1}$$

  (b) Ordenar el conjunto de actividades $A = \{a_1, \ldots, a_m\}$ de acuerdo con la valoración v_i^* calculada en la etapa 2(a).
  (c) Se seleccionan las *w* actividades mejor valoradas para que sean mostradas en la primera pantalla de la aplicación.

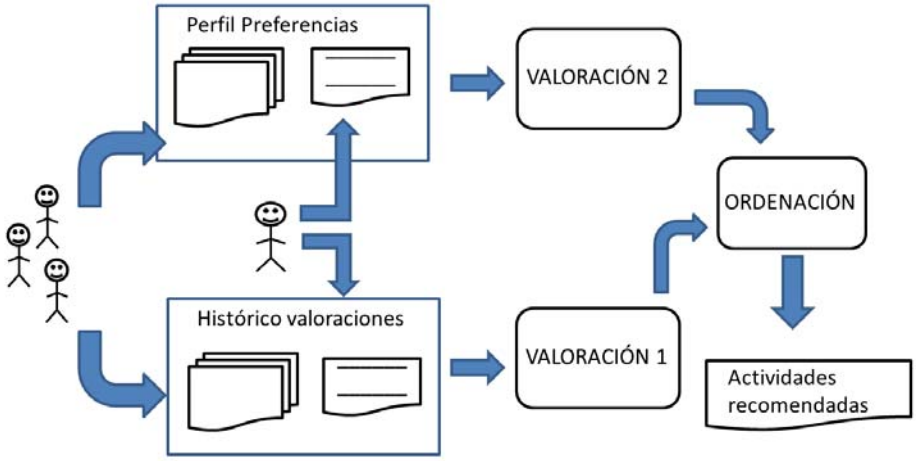


Fig. 1. Diagrama de flujo del sistema de recomendación de Hábitat Madrid

4 Ejemplo de funcionamiento de la aplicación Hábitat Madrid

En esta sección se explica el funcionamiento del sistema de recomendación de la aplicación Hábitat Madrid mediante un ejemplo con datos sintéticos sobre un subconjunto real de las actividades propuestas por los Centros de Educación Ambiental del Ayuntamiento de Madrid para los meses de abril, mayo y junio de 2013 (ver Tabla 2). Supongamos un usuario u que entra por primera vez en el sistema. Desde la pantalla de inicio de la aplicación deberá seleccionar el icono *Configuración* y después la opción *Preferencias.* En la nueva ventana aparecerán una serie de palabras clave relacionadas con la temática de las distintas actividades propuestas, el usuario marcará las palabras clave que hacen referencia al tipo de actividades que más le interesan. De este modo, el sistema dispondrá de información sobre los gustos y preferencias del usuario. El usuario del ejemplo ha marcado tres palabras clave, itinerarios, parques y familiar, como puede verse en la Fig. 2 (a).

El sistema dispondrá también de una base de datos con información sobre la valoración que otros usuarios han hecho de las actividades en las que han participado previamente. Además, dispone de información sobre el grado de similitud s_j de cada usuario u_j con el usuario dado u.

La Tabla 1 contiene información sobre el histórico de valoraciones que los usuarios del sistema han hecho de todas las actividades. Si un usuario no ha participado en una actividad, no puede valorarla, así que en su calificación aparecerá una indeterminación (-). En cada actividad se valoran tres características: Entorno y recursos (E), General (G) y Personal (P). La calificación asignada se puede elegir entre tres posibilidades: Mal (M), Regular (R) Y Bien (B). En esta tabla también se puede observar, junto a cada usuario, el grado de similitud de dicho usuario con el usuario u.

Table 1. Histórico con las valoraciones de las actividades

	u_1 (s_1=1)			u_2 (s_2=0)			u_3 (s_3=0.5)			u_4 (s_4=0.75)		
	E	G	P	E	G	P	E	G	P	E	G	P
a_1	B	R	B	R	R	R	-	-	-	B	B	B
a_2	B	B	B	-	-	-	B	R	R	B	B	B
a_3	-	-	-	B	B	B	B	B	B	R	R	B
a_4	B	R	R	B	B	B	B	B	M	B	B	R
a_5	B	B	B	R	R	R	B	B	B	-	-	-
a_6	B	R	B	B	B	B	R	R	B	B	R	R
a_7	B	R	R	R	R	B	B	B	B	R	M	R

En la Tabla 2 se muestra el título de cada actividad. A partir del título de la actividad y del descriptor de la misma, el sistema busca las palabras claves vinculadas a dicha actividad. Después compara las palabras clave encontradas con las que ha marcado el usuario en su perfil. El cociente n_i/N de la Tabla 2 cuantifica la adecuación de cada actividad desde el punto de vista de los gustos y preferencias del usuario, mediante la búsqueda de palabras clave. Estos índices cuantitativos se combinan en el

algoritmo con la cuantificación de las valoraciones de las actividades de los usuarios que originalmente proceden de variables lingüísticas.

A partir de los datos contenidos en las dos tablas el sistema calcula la valoración total de cada actividad (cuarta columna de la Tabla 2). En este ejemplo hemos dado el mismo peso a las preferencias del usuario que a la valoración que otros usuarios han hecho de las actividades, es decir, hemos utilizado un valor de α=0.5 en la ecuación 1 (sección 3). El sistema, a continuación, ordenará las actividades en función de su valoración total v^* ; en el ejemplo nos quedaría: a_2, a_4, a_6, a_1, a_7, a_5, a_3. De forma que las cuatro primeras actividades mejor valoradas se mostrarán en la primera pantalla de *Recomendaciones*.

Table 2. Valor numérico asignado a cada actividad para el usuario u

	Título	n_i/N	v*
a_1	Itinerario guiado "Despliega tus alas"	0.67	1.13
a_2	Descubre el Retiro en familia	1	1.54
a_3	Taller de Iniciación a la ornitología	0	0.50
a_4	La familia en bicicleta: Ruta especial familias	1	1.31
a_5	Curso de Huerto ecológico	0	0.75
a_6	Pequelandia Viaje a través del parque Juan Carlos I	0.67	1.17
a_7	Exposición "Los árboles del Buen Camino"	0.33	0.79

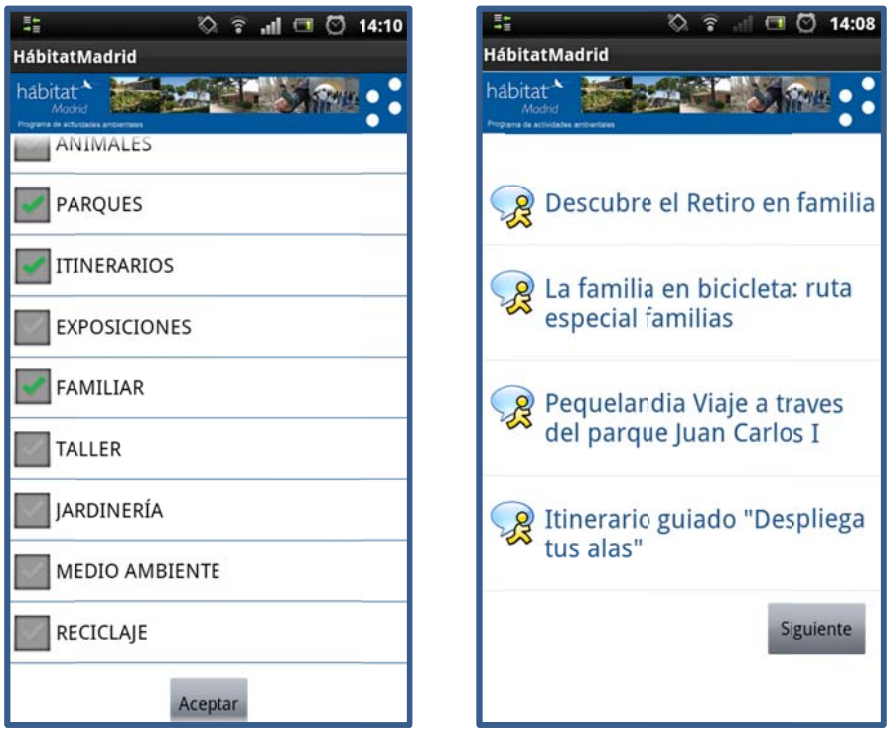
Para poder acceder a las actividades que la aplicación le recomienda el usuario deberá seleccionar la pestaña *Recomendaciones* de la pantalla de inicio. La aplicación mostrará al usuario las cuatro aplicaciones mejor valoradas, tal y como se muestra en la Fig. 2(b). El usuario podrá seleccionar cualquiera de las actividades mostradas para obtener más información o inscribirse en ella. Si estas actividades no le satisfacen y prefiere buscar otras, bastará con que pulse el icono Siguiente de la Fig. 2(b) y la aplicación le mostrará otra pantalla con las siguientes cuatro actividades mejor puntuadas.

5 Conclusiones

En este trabajo se presenta un sistema de recomendación colaborativo integrado en una aplicación móvil desarrollada para el Ayuntamiento de Madrid. El objetivo de esta aplicación es informar a los usuarios del programa de actividades propuesto por las Centros de Información y Educación Ambiental de dicho ayuntamiento. La cantidad de información proporcionada es tan abundante que resulta imprescindible el uso de un sistema de recomendación.

El sistema de recomendación propuesto califica las actividades de acuerdo con el perfil de preferencias de cada usuario y con las valoraciones que otros usuarios (sobre todo, los más afines a él) han hecho de la actividad. El sistema recomendará al usuario las cuatro actividades, que según el modelo aplicado, le resultarán más atractivas. Sin embargo, ofrece la posibilidad de que si ninguna de estas cuatro actividades le satisface pueda seguir informándose de otras. Todas ellas se irán mostrando de mayor a

menor preferencia, según las valoraciones calculadas por el sistema de recomendación propuesto.



(a) (b)

Fig. 2. Pantallas de la aplicación Hábitat Madrid. (a) Perfil de usuario; (b) Actividades recomendadas

Referencias

1. Charlesworth, A.: The Ascent of Smartphone. Engineering & Technology. 4 (3), 32-33 (2009)
2. Ganapati, P.: Independent App Stores Take on Google's Android Market. Wired News (2010). Available at SSRN: http://www.wired.com/gadgetlab/2010/06/independent-app-stores-take-on-googles-android-market/
3. Itunes Store de Apple: http://www.apple.com/es/itunes/
4. Ovi Store de Nokia: http://www.store.ovi.com
5. Clough, G., Jones, A.C., McAndrew, P., Scanlon, E.: Informal learning with PDAs and smartphones. Journal of Computer Assisted Learning. 24 (5), 359-371 (2007)
6. Wankel, L. A., Blessinger, P.: Increasing Student Engagement and Retention using Mobile Applications: Smartphones, Skype and Texting Technologies. Emerald Group Publishing Limited (2013)

7. Tseng, J.: Review of Neurosurgical Smartphone Applications. Journal of Mobile Technology in Medicine. 1 (4) (2012) Available at: http://articles.journalmtm.com/71.pdf
8. Pardines, I., López, V.: Shop&Go: The Routing Problem Applied to the Shopping List on Smartphones. In: F. Sun et al. (eds.) Knowledge Engineering and Management. Springer, Heidelberg (2013).
9. Styliaras, G., Koukopoulos, D.: Educational Scenarios with Smartphones in Cultural Heritage Sites and Environments. Journal of Educational Multimedia and Hypermedia. 21 (3), 285-315 (2012)
10. Kenteris, M., Gavalas, D., Economou, D.: An innovative mobile electronic tourist guide application. Journal Personal and Ubiquitous Computing Archive. 13 (2), 103-118 (2009)
11. Palopoli, L., Rosaci, D., Sarné, G.M.L.: A Multi-tiered Recommender System Architecture for Supporting E-Commerce. Intelligent Distributed Computing VI Studies in Computational Intelligence. 446, 71-81 (2013)
12. Rostami, B., Cremonesi, P., Malucelli, F.: A Graph Optimization Approach to Item-Based Collaborative Filtering. Recent Advances in Computational Optimization Studies in Computational Intelligence. 470, 15-30 (2013)
13. Martínez, L., Pérez, L.G., Barranco, M.: A Multigranular Linguistic Content-Based Recommendation Model. International Journal of Intelligent Systems. 22, 419-434 (2007)
14. Pazzani, M.J.: A framework for collaborative, content-based and demographic filtering. Artificial Intelligence Review. 13 (5-6), 393-408 (1999)
15. Burke, R.D.: Knowledge-based recommender systems. Encyclopedia of Library and Information Science. 69 (32), 2000
16. Burke, R. D.: Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction. 12 (4), 331-370 (2002)
17. Bobadilla, J., Ortega F., Hernando, A., Bernal, J.: A collaborative filtering approach to mitigate the new user cold start problem. Knowledge-Based Systems. 26, 225–238 (2012)
18. Yager, R. R.: Fuzzy logic methods in recommender systems. International Journal Fuzzy Sets and Systems. 136 (2), 133-149 (2003)
19. Delgado, M., Herrera, F., Herrera-Viedma, E., Martín-Bautista, M.J., Vila, M.A.: Combining Linguistic Information in a Distributed Intelligent Agent Model for Information Gathering on the Internet. In: Wang, P.P. (eds.). Computing with Words, pp. 251-276. John Wiley & Sons, Inc. (2001)
20. Zadeh, L.A.: The Concept of a Linguistic Variable and its Application to Approximate Reasoning- I. Information Series. 8, 199-249 (1975)
21. Seltzer L.: Android "Clear Leader" In Smartphone Race Says Report. Byte (2012) Available at: http://www.informationweek.com/byte/personal-tech/smart-phones/android-clear-leader-in-smartphone-race/240143823
22. Gargenta M.: Learning Android. 1st ed. OReally Media, Inc. (2011)

Aplicación de la Soft Computing al Problema de Ruteo de Camiones y Remolques

Isis Torres Pérez[1], Alejandro Rosete Suárez[1], Carlos Alberto Cruz Corona[2], and José Luis Verdegay Galdeano[2]

[1] Instituto Superior Politécnico José Antonio Echeverría, Facultad de Ingeniería Informática, Cuba
{itorres,rosete}@ceis.cujae.edu.cu
[2] Universidad de Granada, E.T.S. de Ingeniería Informática, España
{carloscruz,verdegay}@decsai.ugr.es

Resumen El Problema de Ruteo de Camiones y Remolques (TTRP) es una extensión del Problema de Ruteo de Vehículos (VRP), que tiene como rasgo distintivo el uso de remolques y restricciones de acceso a los clientes. Cada vez es mayor el número de situaciones reales donde se presenta este problema, siendo conveniente mejorar los modelos y métodos que se han utilizado en su resolución. En la literatura este problema se ha abordado sin considerar que el decisor tiene un conocimiento impreciso sobre uno o varios elementos del problema. El presente trabajo se enfoca en modelar y resolver el TTRP con información incierta haciendo uso de las metodologías de la Soft Computing. Además se presentan las principales estrategias de solución y las variantes más estudiadas del problema.

Palabras clave: TTRP, lógica difusa, Soft Computing

1. Introducción

El término TTRP (Problema de Ruteo de Camiones y Remolques) fue propuesto por Chao en el 2002 [1]. Sin embargo existe un conjunto de trabajos de años anteriores que resuelven situaciones de la vida real con características similares al TTRP. El primer trabajo fue presentado en [2] y consistía en una cadena de comestibles en Suiza. Se disponía de una flota compuesta de camiones y remolques y la mayoría de las tiendas estaban localizadas en zonas que no admitían el acceso con remolque. Dos años después Semet considera las restricciones de acceso introducidas en [2] y presenta el *Partial Accesibility Constrained Vehicle Routing Problem* (PACVRP) [3]. Aunque este problema era muy similar al TTRP seguía presentando diferencias. Otro de los trabajos es el estudio realizado en [4] sobre los Problemas de Ruteo de Vehículos con Remolques (VRPT). El VRPT es muy similar al TTRP clásico, aunque el autor lo simplifica a partir de introducir un conjunto de suposiciones que provocan que el modelo del VRPT no se corresponda con el del TTRP. Otro problema relacionado con el TTRP es el *Site-Dependent Vehicle Routing Problem* (SDVRP) [5]. En este problema existe

una flota con diferentes tipos de vehículos y existen relaciones de compatibilidad entre los clientes y los vehículos.

Desde un punto de vista práctico al tratar con problemas de ruteo, las respuestas de los decisores concerniente a los datos que se van a introducir en el modelo en la mayoría de los casos son vagas e inexactas. El TTRP es un problema que por sus características propicia la presencia de vaguedad, imprecisión e inexactitud en la información que se maneja. No obstante, los modelos utilizados en la literatura asumen que los datos disponibles son exactos; consideración que en muchos casos no se corresponde con la realidad. El presente trabajo propone el uso de técnicas de Soft Computing [6], [7] para representar y resolver modelos para el TTRP con información incierta.

Este trabajo está organizado de la siguiente forma: en la sección 2 se describe el TTRP, con sus variantes y métodos de solución. En la sección 3 se introduce un nuevo enfoque para modelar el problema y se presenta un ejemplo experimental. El artículo culmina con las conclusiones en la sección 4.

2. Problema de Ruteo de Camiones y Remolques

El TTRP es un problema de optimización combinatoria que extiende del VRP. Básicamente la diferencia entre estos problemas radica en el uso de remolques como parte de la flota de vehículos y en la presencia de restricciones de acceso a los clientes. El TTRP considera dos tipos de clientes: *truck customer* (TC) y *vehicle customer* (VC). Los TCs son aquellos clientes que solamente pueden ser accedidos por el camión debido a que presentan limitaciones en el acceso. Al contrario, los VCs pueden ser accedidos con o sin el remolque. La flota de vehículos se compone de camiones y remolques con capacidad homógenea. Un camión con su remolque es denominado *vehículo completo* y al camión solo *camión puro*. En este tipo de problema se pueden presentar tres tipos de rutas, como ilustra la Figura. 1.

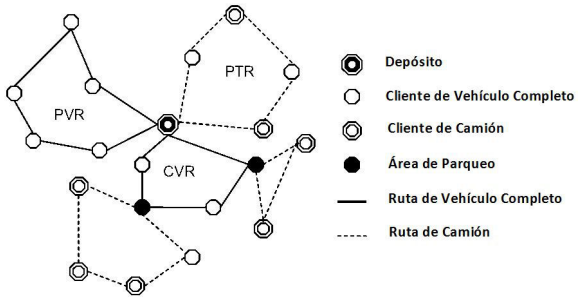


Figura 1. Ejemplo de los diferentes tipos de rutas. Tomado de [8].

En todos los casos las rutas inician y terminan en el depósito central y están limitadas por la capacidad de carga del vehículo utilizado.

- Ruta de Camión Puro (PTR): En este tipo de ruta son visitados ambos clientes y siempre es realizada por el camión sin remolque.
- Ruta de Vehículo Completo (CVR): Esta ruta consiste en un recorrido principal realizado por el vehículo completo y al menos un recorrido secundario (subruta), realizado por el camión.
- Ruta de Vehículo Puro (PVR): Este tipo de ruta es similar a la CVR aunque no contiene subrutas. En su recorrido solamente se visitan VCs.

Las nuevas características impuestas por las situaciones de la vida real, así como las diferentes restricciones operativas dan lugar a diferentes variantes del problema. A continuación presentamos tres de las variantes del TTRP tratadas en la literatura.

RTTRP: El Problema de Ruteo de Camiones y Remolques Relajado es una variante del TTRP que consiste en relajar la restricción asociada al tamaño de la flota de vehículos con el objetivo de reducir el costo total del recorrido [9].

TTRPTW: El Problema de Ruteo de Camiones y Remolques con Ventana de Tiempo considera las restricciones de ventana de tiempo. El modelo del TTRPTW incorpora una ventana de tiempo $[et_i, lt_i]$ y un tiempo de servicio st_i para cada vértice [10].

STTRPSD: El Problema de Ruteo de un Camión y su Remolque con Depósitos Satélites consiste en un único camión con un remolque, donde los clientes solo pueden ser accedidos por el camión puro. Existen un conjunto de áreas de parqueo denominadas depósitos satélites, donde es posible desmontar el remolque y transferir productos entre el camión y el remolque [11].

2.1. Definición formal del problema

El TTRP es un problema que puede ser formulado matemáticamente a partir de un grafo no dirigido o una red $G = (V, A)$, donde V es el conjunto de vértices con n vértices y A es el conjunto de aristas entre los vértices.

El conjunto de los vértices se denota como: $V = \{v_0, v_1, \ldots, v_n\}$, donde v_0 hace referencia al depósito central y los $v_i (i = 1, 2, \ldots, n)$ representan los clientes a ser visitado. Cada vértice tiene asociado una demanda no negativa d_i. El conjunto V puede ser dividido en dos subconjuntos: $V_v \subseteq V$ que contiene a los VCs y $V_c \subseteq V$ que contiene a los TCs. El conjunto de las aristas se denota como: $A = \{(v_i, v_j) : v_i, v_j \in V, i \neq j\}$. Se dispone de una matriz de costos no negativos $C = \{c_{ij}\}$ de tamaño $n \times n$. Cada arista (v_i, v_j) tiene asociado un costo no negativo $c_{v_i v_j}$, que puede ser interpretado como la distancia o el tiempo requerido en la arista para ir de v_i a v_j en el grafo.

Por otra parte, la flota de vehículos se compone de m_c camiones y m_r remolques, donde $m_c \geq m_r$. Los camiones y remolques son considerados idénticos con capacidad de q_c y q_r respectivamente. Un vehículo completo tiene una capacidad igual a $q_c + q_r$. A cada vehículo k se le asigna una ruta R_k. Una ruta en el TTRP se define como la partición del conjunto $V : R_1, \ldots, R_k$ y una permutación δ_k de R_k, para especificar el orden de los clientes en la ruta. Cada ruta inicia y termina en el depósito: $R_k = \{v_0, v_1, \ldots, v_{n+1}\}$, donde $v_0 = v_{n+1}$ denota el depósito.

2.2. Estrategias de solución

Los algoritmos aproximados constituyen la principal alternativa para la resolución del TTRP, aunque existe la propuesta de Drexl [12] basada en un algoritmo "branch and price". A continuación se describen las propuestas de solución más significativas aplicadas en los últimos años a este problema.

Entre los trabajos se encuentra [1] y [13], donde la resolución del TTRP se realiza a partir de una estrategia de dos fases. En el caso de [1] la primera fase se basa en la heurística de asignar primero - rutear segundo (*cluster - first route - second*), mientras que en [13] el autor opta por aplicar dos heurísticas de construcción. La primera T-Cluster es una heurística de inserción que construye las rutas de forma secuencial. La segunda heurística de construcción llamada T-Sweep es una adaptación para el TTRP de la heurística presentada en [14]. Por último en la segunda fase ambos autores aplican un algoritmo basado en Búsqueda Tabú con el propósito de mejorar la solución inicial obtenida en la primera fase.

Otro de los trabajos es el de Lin *et al.* en [8] donde se aplica Recocido Simulado con tres operadores de vecindad. En esta propuesta se utiliza una representación indirecta de la solución usando una permutación de los clientes y adicionando depósitos artificiales (*dummy*) para separar rutas e indicar la terminación de subrutas. Además incorpora un vector binario de tamaño igual a la cantidad de VCs que indica el modo de acceso a estos clientes. Con un procedimiento y la información del vector binario se decodifica la permutación en una solución TTRP. Se aplica un procedimiento de combinación de rutas para reducir el número de vehículos requeridos si se obtiene una solución no factible en cuanto al número de vehículos disponibles. Por último, se agregra a la función objetivo un término de penalización que facilita guiar la búsqueda a regiones factibles.

En [15] los autores diseñan un método basado en programación matemática que emplea una estrategia de asignar primero - rutear segundo. Este método resuelve de forma secuencial dos subproblemas: (1) reducir el tamaño de la flota; y (2) minimizar la longitud del recorrido en cada ruta usando un modelo similar al Problema del Viajante de Comercio (TSP) sin eliminación de subrutas. Los autores utilizan ambos modelos en un mecanismo iterativo que pretende diversificar la búsqueda. Además incluye un mecanismo basado en Búsqueda Tabú que prohíbe la asignación de clientes a rutas que ya fueron exploradas en iteraciones previas.

Entre los trabajos más recientes está el de Villegas *et al.* en [16], este artículo se centra en experimentar con diferentes alternativas híbridas. La mejor alternativa se basa en *Greedy Randomized Adaptive Search Procedure* (GRASP), *Variable Neighborhood Search* (VNS) y *Path Relinking* (PR). Su funcionamiento parte de una fase de construcción donde se emplea un procedimiento basado en rutear primero - asignar segundo (*route - second cluster - first*). Durante la búsqueda la metaheurística explora soluciones no factibles, mientras que el componente VNS juega el papel de mecanismo de reparación y de mejora. El componente

PR es utilizado como un operador de reparación para garantizar la calidad de las soluciones.

Como se puede apreciar existe una amplia variedad de metaheurísticas que han sido aplicadas en la resolución del TTRP. Cada una ha sido probada en problemas de referencias de la literatura [1] obteniéndose en cada caso resultados satisfactorios. Sin embargo, es importante señalar que estos trabajos solamente se centran en los métodos de solución relegando a último plano los modelos.

3. Un enfoque basado en Soft Computing

Entre todos los problemas de optimización, los modelos que han recibido mayor atención son los modelos de Programación Lineal (PL). La formulación más general de un problema PL es:

$$Max\{z = cx/Ax \leq b, x \geq 0\} \tag{1}$$

siendo A una matriz de números reales de dimensión $m \times n$, b un vector en $\mathbb{R}^m$ y c un vector de costos en $\mathbb{R}^n$. Los problemas generales de PL en los que algunas de sus componentes son difusas, se denominan problemas de Programación Lineal Difusa (PLD). Entre los problemas de PLD se distinguen los siguientes:

- **Problemas de programación lineal con restricciones difusas**: considera el caso en el que el decisor asume que puede haber cierta tolerancia en el cumplimiento de las restricciones.
- **Problemas de programación lineal con metas difusas**: en este caso se admite que el valor de la función objetivo sea impreciso o flexible.
- **Problemas de programación lineal con costos difusos**: se caracterizan porque la información que se tiene de los coeficientes de la función objetivo es imprecisa.
- **Problemas de programación lineal con coeficientes difusos en las restricciones**: considera los coeficientes de la matriz tecnológica como números difusos.

Aunque la PLD tiene sus antecedentes teóricos en el trabajo de Bellman y Zadeh de 1970 [17], los problemas de PLD nacen formalmente en 1974. Tanaka *et al* en [18] proponen un modelo para tratar los problemas de PL en un entorno difuso. No obstante, el método de resolución planteado en [18] proporciona una solución constituida por un único punto. En años posteriores se demostró que este método constituía un caso particular de un método general presentado por [19] que permitía obtener una solución difusa, contexto - dependiente y que englobaba la que se había propuesto previamente.

Los problemas de ruteo como el TTRP contienen parámetros y decisiones caracterizados por la incertidumbre. Es usual que los decisores modelen el problema usando conocimiento subjetivo o información lingüística para tomar decisiones, describir parámetros o establecer restricciones. En este tipo de situación lo que se acostumbra es simplificar el problema mediante la supresión del contenido

impreciso. De esta manera se deforma la naturaleza del problema, al pasar de un problema de datos imprecisos a un problema de datos exactos. La estrategia utilizada en este trabajo para resolver el problema se basa en una idea paramétrica general con el fin de transformar problemas difusos en un conjunto de problemas clásicos [19], [20]. Este enfoque paramétrico se divide en dos fases (véase Figura. 2).

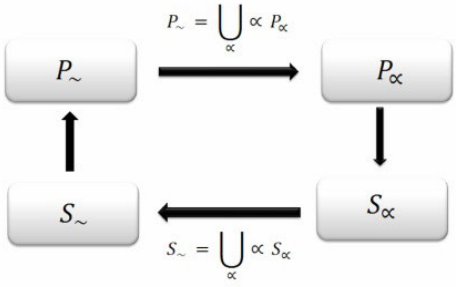


Figura 2. Idea paramétrica.

La primera fase consiste en la transformación de un problema difuso en un problema paramétrico clásico con un parámetro α que representa el nivel de satisfacción del decisor en el intervalo [0, 1]. En la segunda fase, el problema de programación paramétrico se resuelve para cada uno de los diferentes α utilizando técnicas convencionales de programación lineal. Desde 1990, Delgado *et al.* [21] demostraron que el conjunto de soluciones obtenidos por este método contiene las soluciones alcanzadas por otros enfoques diferentes que resuelven problemas de programación lineal difusos.

Un caso de la aplicación de este enfoque es considerando que las restricciones del problema tienen un carácter difuso, es decir, que se admite cierta violación en el cumplimiento de tales restricciones. Para cada resricción esta suposición se puede representar de la siguiente forma:

$$a_i x \leq_f b_i, \qquad i \in M = \{1, 2, \ldots, m\} \tag{2}$$

y modelarla por medio de una función de pertenencia $\mu_i : \mathbb{R} \to [0, 1]$

$$\mu_i(a_i x) = \begin{cases} 1 & \text{si } a_i x \leq b_i \\ f_i(a_i x) & \text{si } b_i < a_i x \leq b_i + \tau_i \\ 0 & \text{si } a_i x > b_i + \tau_i \end{cases}$$

Estas funciones expresan que el decisor tolera violaciones en cada restricción hasta un valor $b_i + t_i, i \in M$, donde τ es denominado el nivel de tolerancia. Por otra parte las funciones f_i se asumen no decrecientes y continuas para estas restricciones. La función μ_i se define para cada $x \in X$ y da el grado de cumplimiento de la i-ésima restricción para $x \in X$. El problema asociado se presenta de la siguiente forma:

$$Max\{z = cx/Ax \leq_f b, x \geq 0\} \tag{3}$$

De esta forma las restricciones pueden ser expresadas como un conjunto difuso y para cada una de ellas se define una función de pertenencia. Para resolver (3) existen diferentes aproximaciones. En particular, [19] utilizando el Teorema de Representación de conjuntos difusos, demuestra que las soluciones se pueden obtener del siguiente modelo auxiliar:

$$Max\{z = cx/Ax \leq b + \tau(1-\alpha), x \geq 0, \alpha \in [0,1]\} \tag{4}$$

con $\tau = (\tau_1, \ldots, \tau_m) \in \mathbb{R}^m$.

3.1. TTRP con restricciones difusas

Con el objetivo de aplicar la estrategia descrita anteriormente al TTRP, se modelo el problema considerando difusas las restricciones asociadas a la capacidad. Estas restricciones imponen que la demanda total de la ruta no puede superar la capacidad del vehículo utilizado para realizar dicha ruta y pueden ser representadas como:

$$\sum d_i x_i^k \leq q_c \qquad \forall R_{cp} \tag{5}$$

$$\sum d_i x_i^k \leq q_c + q_r \qquad \forall R_{vc} \tag{6}$$

donde R_{vc} es el número de rutas de vehículos completo y R_{cp} es el número de rutas de camión puro. El modelo propuesto permite considerar las restricciones como difusas y admite la violación de cada una de las restricciones de acuerdo con un nivel de tolerancia τ. Haciendo uso del modelo auxiliar presentado anteriormente (ver 4) se obtienen las siguientes restricciones:

$$\sum d_i x_i^k \leq_f q_c \Longleftrightarrow \sum d_i x_i^k \leq q_c + \tau(1-\alpha) \tag{7}$$

$$\sum d_i x_i^k \leq_f q_c + q_r \Longleftrightarrow \sum d_i x_i^k \leq (q_c + q_r) + \tau(1-\alpha) \tag{8}$$

Para este ejemplo experimental se implementó la propuesta de Lin *et al* en [8] descrita anteriormente. La implementación se codificó en Java con Eclipse 4.2.1. Se ejecutó en una PC Intel Core Duo a 1.80 HGz con 4GB de memoria RAM y sistema operativo Windows 7.

Los datos del problema utilizado en este trabajo se corresponden con un *benchmark* de los 21 reportados en [1]. Estos *benchmark* se obtuvieron de 7 problemas VRP de prueba [22], generando por cada uno tres problemas TTRP. Las características de este problema se muestran en la Tabla 1.

Para este ejemplo se realizaron 30 ejecuciones de 100000 iteraciones, utilizando diferentes niveles de tolerancia $\tau = \{10, 30, 50\}$ y variando los valores de $\alpha = \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$. Los resultados que se obtienen

Problema # 1						
Clientes	**VC**	**TC**	**Camiones**	**Capac. Cam**	**Remolques**	**Capac. Rem**
50	38	12	5	100	3	100

Tabla 1. Datos del problema de prueba

se presentan en la Tabla 2, por cada valor de α se muestra el mejor valor de la función objetivo y el comportamiento promedio de las soluciones obtenidas. Los valores encerrados entre paréntisis están asociados a la cantidad de rutas de cada tipo (R_{vc}, R_{cp}) obtenidas en la solución.

	$\tau = 10$		$\tau = 30$		$\tau = 50$	
α	Min	Avg	Min	Avg	Min	Avg
0.0	$583.58_{(4,2)}$	655.87	$551.26_{(3,1)}$	651.92	$572.20_{(3,1)}$	626.00
0.1	$583.58_{(4,2)}$	651.39	$556.35_{(4,1)}$	641.70	$566.15_{(3,2)}$	618.57
0.2	$583.58_{(4,2)}$	648.80	$556.35_{(4,1)}$	636.61	$546.68_{(3,1)}$	616.08
0.3	$583.20_{(4,2)}$	647.90	$556.35_{(4,1)}$	630.67	$554.78_{(3,1)}$	614.15
0.4	$567.67_{(3,2)}$	643.05	$559.57_{(4,2)}$	627.38	$559.90_{(3,1)}$	613.78
0.5	$567.67_{(3,2)}$	636.26	$551.84_{(4,1)}$	624.50	$557.61_{(3,2)}$	614.44
0.6	$567.67_{(3,2)}$	636.61	$544.99_{(4,1)}$	621.31	$559.34_{(3,2)}$	609.65
0.7	$567.67_{(3,2)}$	636.25	$548.22_{(4,2)}$	619.04	$559.34_{(3,2)}$	605.69
0.8	$575.57_{(4,1)}$	636.92	$539.81_{(4,1)}$	619.59	$566.91_{(3,2)}$	608.37
0.9	$557.71_{(4,1)}$	635.48	$539.81_{(4,1)}$	620.61	$566.43_{(4,1)}$	611.78
1.0	$558.65_{(4,2)}$	633.48	$539.81_{(4,1)}$	622.02	$543.91_{(4,1)}$	611.01

Tabla 2. TTRP con restricciones de capacidad difusas y costes expresados en distancias

De los resultados obtenidos en la Tabla 2 la conclusión principal apunta a que existe una tendencia a obtener mejores soluciones a medida que el nivel de tolerancia en la restricción aumenta. Es importante destacar que cada variación en la restricción ya sea el valor de α o de τ implican un cambio en el problema original. Otra aspecto es que para los $\alpha - corte$ entre 0.7 y 1 no hay diferencias significativas en los valores que se obtiene. Sin embargo, estos resultados son iniciales y no se pueden extrapolar a valores similares en otras insancias del problema.

Esta manera de modelar el TTRP aún no ha sido abordada en la literatura. Con este trabajo los autores amplian una línea de investigación que potencia el esfuerzo hacia la definición de modelos cada vez más cercanos a la realidad. Por este motivo no se considera la comparación de nuestros resultados con los de otros autores de la literatura, ya que se están tratando problemas totalmente diferentes. No obstante, presentamos en la Tabla 3 una comparación entre los resultados del estado del arte y los valores obtenidos en nuestra propuesta cuando $\alpha = 1$, que equivale al problema original.

Como se puede observar en la Tabla 3 la estrategia propuesta en este trabajo para el TTRP produce resultados comparables con los obtenidos por otras estrategias de solución. Tal es el caso de la mejor solución encontrada que pertenece a la estrategia propuesta cuando $\tau = 50$, aunque en los otros dos casos la mejor

Problema # 1						
Estrategias	**Lin *et al.*** ([8])	**Caramia *et al.*** ([15])	**Villegas *et al.*** ([16])	**Torres *et al.*** ($\tau = 10$)	**Torres *et al.*** ($\tau = 30$)	**Torres *et al.*** ($\tau = 50$)
Min	566.82	566.80	564.68	558.65	539.81	543.91
Avg	568.86	566.80	565.99	729.38	622.02	611.01

Tabla 3. Comparación de los resultados obtenidos para el Problema 1 con estrategias de la literatura

solución supera las reportadas por [8], [15], [16]. Sin embargo, el comportamiento promedio de las soluciones encontradas es mucho mejor en las estrategias de solución del estado del arte. Más allá de los resultados que se presentan lo más significativo de la aplicación del método parámetrico es la obtención de una solución difusa. Así, el decisor dispone de un conjunto de soluciones y no sólo una única solución. Esto le va a permitir seleccionar la solución más adecuada de acuerdo a sus necesidades reales, al conocimiento de la información de que disponga, y a su forma de decidir.

4. Conclusiones

El TTRP es uno de los problemas de ruteo más interesante que surge en estos tiempos, sin embargo aún es poca la atención que se le ha dirigido. En el caso de las soluciones que se han brindado, el modo de modelar el problema parte de que los datos disponibles son exactos; cuando en la realidad la imprecisión en la información esta presente de manera natural. Por esta razón, es aconsejable orientar la investigación hacia la definición de modelos TTRP que incorporen la incertidumbre presente en los datos, y por lo tanto tiene mucho sentido pensar en técnicas de Soft Computing. Los autores apuntan a ampliar la línea de investigación con cada uno de los modelos de PLD para tratar de resolver los problemas prácticos de la vida real, facilitando la construcción de sistemas de apoyo en la toma de decisiones.

Agradecimientos. Este trabajo está parcialmente financiado por los projectos TIN2011-27696-C02-01 del Ministerio de Economía y Competitividad, P11-TIC-8001 de la Junta de Andalucía, y la Asociación Universitaria Iberoamericana de Postgrado (AUIP).

Referencias

1. Chao, I-M.: A tabu search method for the truck and trailer routing problem. Computers & Operation Research. 29(1), 33–51 (2002)
2. Semet, F., Taillard, E.: Solving real life vehicle routing problems efficiently using tabu search. Annals of Operations Research. 41(4), 469–488 (1993)
3. Semet, F.: A two phases algorithm for tha partial accessibility constrained vehicle routing problem. Annals of Operations Research. 61(1), 45–65 (1995)

4. Gerdessen, J.C.: Vehicle routing problem with trailers. European Journal of Operational Research. 93(1), 135–147 (1996)
5. Chao, I-M., Golden, B., Wasil, E.: A computacional study of a new heuristic for the site dependent vehicle routing problem. Information Systems and Research Operations. 37(3), 319–336 (1999)
6. Zadeh, L.A.: Soft computing and fuzzy logic. IEEE Software. 11(6), 48–56 (1994)
7. Verdegay, J.L., Yager, R.R., Bonissone, P.P.: On heuristics as a fundamental constituent of soft computing. Fuzzy Sets and Systems. 159(7), 846–855 (2008)
8. Lin, S-W., Yu, V.F., Chou, S-Y.: Solving the truck and trailer routing problem based on a simulated annealing heuristic. Computers & Operation Research. 36(5), 1683–1692 (2009)
9. Lin, S-W., Yu, V.F., Chou, S-Y.: A note on the truck and trailer routing problem. Expert Systems with Application. 37(1), 899–903 (2010)
10. Lin, S-W., Yu, V.F., Lu, Ch-Ch.: A simulated annealing heuristic for the truck and trailer routing problem with time windows. Expert Systems with Application. 38(12), 15244–15252 (2011)
11. Villegas, J.G., Prins, C., Prodhon, C., Medaglia, A.L, Velasco, N.: GRASP/VND and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots. Engineering Applications of Artificial Intelligence. 23(5), 780–794 (2010)
12. Drexl, M.: A branch and price algorithm for the truck and trailer routing problem. Technical report, RWTH Aachen University, N. 27/2006 Deutsche Post Endowed Chair of Optimization of Distribution Networks. Germany (2007)
13. Scheuerer, S.: A tabu search heuristic for the truck and trailer routing problem. Computers & Operation Research. 33(4), 894–909 (2006)
14. Gillett, B.E., Miller, L.R.: A heuristic for the vehicle dispatching problem. Operation Research. 22(2), 340–349 (1974)
15. Caramia, M., Guerriero, F.: A heuristic approach for the truck and trailer routing problem. Journal of the Operational Research Society. 61(7), 1168–1180 (2010)
16. Villegas, J.G., Prins, C., Prodhon, C., Medaglia, A.L, Velasco, N.: A GRASP with evolutionary path relinking for the truck and trailer routing problem. Computers & Operation Research. 38(9), 1319–1334 (2011)
17. Bellman, R.E., Zadeh, L.A.: Decision-making in a fuzzy environment. Management Science, Application Series. 17(4), B141–B164 (1970)
18. Tanaka, H., Okuda, T., Asai, K.: On fuzzy mathematical programming. Journal Cybernet. 3(4), 37–46 (1974)
19. Verdegay, J.L.: Fuzzy mathematical programming. Fuzzy Information and Decision Processes 231–237 (1982)
20. Delgado, M., Verdegay, J.L., Vila, M.A.: A general model for fuzzy linear programming. Fuzzy Sets and Systems. 21(1), 21–29 (1989)
21. Delgado, M., Verdegay, J.L., Vila, M.A.: Relating different approaches to solve linear programming problems with imprecise costs. Fuzzy Sets and Systems. 37, 33–42 (1990)
22. Christofides, N., Mingozzi, A., Toth, P.: The vehicle routing problem. Combinatorial Optimization. Chichester, UK: Wiley (1979)

Control Inteligente de un Robot Manipulador de Tres Grados de Libertad

Ildefonso Ayala de Diego[1], Matilde Santos Peñas[2]

Dpto. Informática y Automática, ETSI. Informática, UNED, Madrid, Spain
Dpto. Arquitectura de Computadores y Automática, Facultad de Informática, Universidad Complutense de Madrid, Madrid, Spain
iayala1@alumno.uned.es, msantos@ucm.es

Abstract. Este trabajo describe la utilización de dos técnicas de Soft Computing para el control de la posición de un robot manipulador de tres grados de libertad. La estructura de control del robot está formada por un lazo de realimentación lineal con controladores borrosos de tipo proporcional-derivativo y un lazo de precompensación que aproxima la dinámica inversa del robot utilizando redes neuronales. Los resultados de simulación prueban como el sistema controlado por estas técnicas inteligentes es capaz de seguir diferentes trayectorias y responder ante perturbaciones.

Keywords: Robot manipulador, sistema difuso, redes neuronales, control inteligente.

1 Introducción

Se puede considerar un robot manipulador como un conjunto de elementos o eslabones unidos mediante articulaciones que permiten el movimiento relativo entre cada dos eslabones consecutivos. El movimiento de cada articulación puede ser de desplazamiento, de giro o una combinación de ambos. Cada movimiento independiente que puede realizar una articulación con respecto a la anterior se denomina grado de libertad. Existen diferentes tipos de articulaciones pero en robótica generalmente solo se usan articulaciones de rotación (giro) y prismáticas (desplazamiento). Cada una de éstas tiene un grado de libertad [1].

Un robot manipulador de varios grados de libertad es un sistema multivariable, no lineal y fuertemente acoplado. En general no es posible el control descentralizado de este tipo de sistemas por lo que a lo largo del tiempo se han desarrollado diversas técnicas para resolver el problema del control de los robots manipuladores [2]. Una conocida técnica de control combina un lazo de realimentación lineal más un lazo de precompensación que incluye la dinámica inversa del robot, cuyo objetivo es independizar dinámicamente el movimiento de las articulaciones del sistema. Los controladores utilizados habitualmente son de tipo proporcional-derivativo. En el presente trabajo estos controladores se han implementado usando la lógica fuzzy. Esta metodología del *soft computing* facilita el diseño de controladores no lineales como el que requiere esta aplicación, y a su vez permite conseguir las especificaciones de control deseadas. Por otro lado, para identificar la dinámica inversa del robot se han utilizado redes neuronales, una estrategia que ha probado ser muy eficiente para aproximar funciones complejas [10].

Este trabajo está organizado como sigue: en el segundo apartado se presenta el modelo del robot manipulador a controlar y su simulación; en la sección 3 se introducen las técnicas de control inteligente que se utilizarán para el diseño de los controladores y se aplican las metodologías de control mencionadas anteriormente; en el cuarto apartado se muestran algunas de las pruebas realizadas y se comentan los resultados obtenidos. Finalmente en el quinto apartado se presentan las conclusiones de este trabajo.

2 Modelo y Simulación del Robot

El robot elegido en este trabajo es un robot esférico o polar de 3 grados de libertad como el de la figura1, que consta de dos articulaciones giratorias y una articulación prismática. Se muestran también las dimensiones y las variables asociadas a cada articulación.

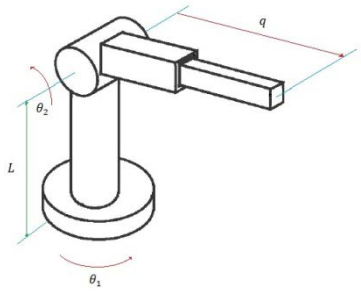


Figura 1. Representación del robot esférico.

El modelo cinemático directo del robot se ha obtenido utilizando el algoritmo de Denavit-Hartenberg [2]. Dicho algoritmo se aplica al robot situado en su posición de partida, vertical ($\theta_1 = \theta_2 = 0$). Los parámetros D-H resultantes son los siguientes:

Tabla 1. Parámetros D-H del robot esférico

Articulación	θ	d	a	α
1	θ_1	L	0	$\pi/2$
2	θ_2	0	0	$-\pi/2$
3	0	q	0	0

Una vez obtenidos los parámetros D-H se calculan las matrices homogéneas de transformación entre los sistemas de referencia ${}^{i-1}A_i$. Llamando C_1, C_2, S_1, S_2 a $cos(\theta_1)$, $cos(\theta_2)$, $sen(\theta_1)$, $sen(\theta_2)$ respectivamente, se obtiene la matriz T que expresa la posición y orientación del extremo del robot en función de sus coordenadas articulares respecto al sistema de referencia fijo situado en la base del robot.

$$T = {}^0A_1\,{}^1A_2\,{}^2A_3 = \begin{bmatrix} C_1C_2 & -S_1 & -C_1S_2 & -qC_1S_2 \\ S_1C_2 & C_1 & -S_1S_2 & -qS_1S_2 \\ S_2 & 0 & C_2 & L+qC_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

El modelo dinámico para este robot se ha deducido utilizando la formulación de Lagrange. Para ello se considera que la masa de cada eslabón está concentrada en su centro de masas. Para poder aplicar las ecuaciones de Lagrange debemos obtener las velocidades de los eslabones derivando con respecto al tiempo los vectores de posi-

ción de los centros de masas. Una vez obtenidas estas velocidades estamos en condiciones de calcular las energías cinéticas T_1, T_2, T_3 y energías potenciales U_1, U_2, U_3 de los eslabones.

Calculamos entonces el Lagrangiano del robot y aplicamos las ecuaciones de Lagrange a cada una de sus coordenadas articulares para obtener las ecuaciones del modelo dinámico inverso del robot.

$$\tau = M(q)\ddot{q} + V(q,\dot{q}) + G(q) + \tau_e \tag{2}$$

Donde τ es el vector de fuerzas y pares aplicados a cada articulación; $q, \dot{q}, \ddot{q}$ son los vectores de posición, velocidad y aceleración articulares, M(q) es la matriz de inercia, $V(q,\dot{q})$ es el vector de fuerzas centrípetas y de Coriolis y $G(q)$ es el vector de fuerzas gravitatorias. Las fuerzas y pares exteriores aplicadas al robot están dadas por τ_e.

El modelo dinámico directo de robot expresa la evolución temporal de las coordenadas articulares en función de las fuerzas y pares que intervienen, y se obtiene despejando las aceleraciones angulares y lineales. En forma matricial, tras la aplicación de las ecuaciones de Lagrange, se obtienen los distintos elementos que conforman el modelo dinámico:

$$M(q)\ddot{q} = \begin{bmatrix} (m_2c_2^2 + m_3d^2)S_2^2 & 0 & 0 \\ 0 & m_2c_2^2 + m_3d^2 & 0 \\ 0 & 0 & m_3 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{d} \end{bmatrix} \tag{3}$$

$$V(q,\dot{q}) = \begin{bmatrix} 0 & 2(m_2c_2^2 + m_3d^2)S_2C_2\dot{\theta}_1 & 2m_3S_2^2d\dot{\theta}_1 \\ -(m_2c_2^2 + m_3d^2)S_2C_2\dot{\theta}_1 & 0 & 2m_3d\dot{\theta}_2 \\ -m_3S_2^2d\dot{\theta}_1 & -m_3d\dot{\theta}_2 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_1 \\ \dot{d} \end{bmatrix} \tag{4}$$

$$G(q) = \begin{bmatrix} 0 \\ -(m_2gc_2 + m_3gd)S_2 \\ m_3gC_2 \end{bmatrix} \tag{5}$$

2.1 Simulación del Robot

La simulación del modelo dinámico directo del robot se llevó a cabo en el entorno MATLAB/Simulink. Se obtienen las trayectorias articulares a partir de las fuerzas y pares aplicados. Los valores numéricos utilizados son: masa del segundo eslabón m_2 = 3 kg, Masa del tercer eslabón m_2 = 2 kg, Centro de masas del segundo eslabón c_2 = 0,2 m.

El diagrama de bloques Simulink, donde se ha considerado el producto de la matriz inversa de la matriz de inercia y un vector de entrada, el vector de fuerzas centrípetas y de Coriolis, y el vector de fuerzas gravitatorias, se muestra en la figura 2.

3 Control del Robot

El método de control utilizado es el del control por par calculado con precompensación del sistema con la dinámica inversa del robot en bucle externo, y control de tipo proporcional-derivativo PD. Se ha considerado la existencia de pares y fuerzas exteriores al robot τ_e. La posición y la velocidad medida por los sensores del robo son utilizadas por los bloques de precompensación para producir el desacoplamiento dinámico de las articulaciones. La señal producida por el controlador y el lazo de

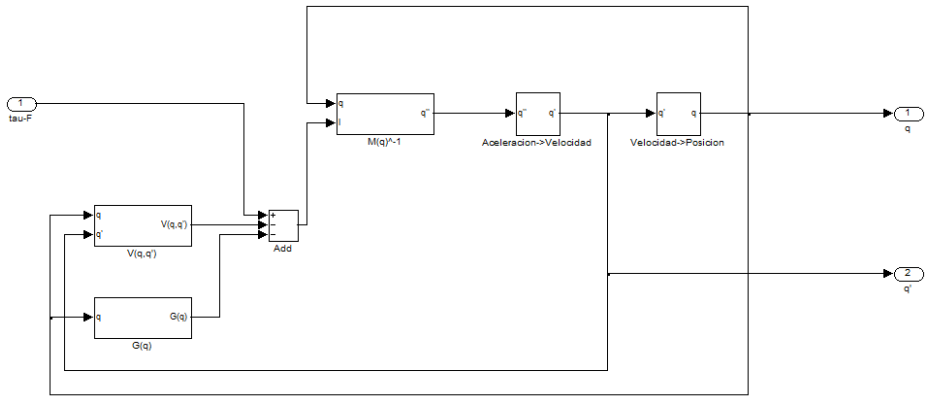


Figura 2. Diagrama de bloques Simulink.

compensación u es entregada a los motores de las articulaciones que producen las fuerzas y pares necesarios para que la trayectoria real de cada variable articular coincida con la trayectoria deseada.

Se han aplicado dos técnicas de Soft Computing para conseguir controlar este robot esférico. El esquema de control propuesto es el de la figura 3, en el que se utilizan redes neuronales para precompensar el sistema con la dinámica inversa del robot y controladores borrosos para el control individual de cada articulación. Según la figura 3 el valor de τ es:

$$\tau = \big(M(q)\ddot{q}_d K^{-1} + \big(V(q,\dot{q}) + G(q)\big)K^{-1}\big)K + K_p e + K_v \dot{e} \tag{6}$$

Es fácil comprobar el efecto de los bloques de precompensación. Supongamos que los controladores no existen (K_v y K_p son matrices nulas). La fórmula anterior se transforma en:

$$\tau = \big(M(q)\ddot{q}_d K^{-1} + \big(V(q,\dot{q}) + G(q)\big)K^{-1}\big)K \tag{7}$$

Igualando esta expresión a (1) se obtiene (8), que se puede expresar como (9):

$$M(q)\ddot{q}_d = M(q)\ddot{q} + \tau_e \tag{8}$$

$$\ddot{q}_d = \ddot{q} + M(q)^{-1}\tau_e \tag{9}$$

Si τ_e es nulo entonces las trayectorias reales y deseadas coinciden en todo momento quedando el modelo desacoplado. Una vez que se consigue el desacoplamiento se utilizan controladores de tipo fuzzy PD en cada articulación de manera individual, como se indica en la figura 3 con el objetivo de compensar perturbaciones no modeladas dadas por el término τ_e. Si el modelo del robot no es bien conocido se puede utilizar una estimación de las matrices $M(q)\ddot{q}_d$, $V(q,\dot{q})$ y $G(q)$ para compensar la dinámica. En este caso los controladores deben compensar las discrepancias entre los valores reales de las matrices $M(q)\ddot{q}_d$, $V(q,\dot{q})$ y $G(q)$ y los estimados.

3.1 Redes Neuronales para compensar la Dinámica del Robot

Las redes neuronales elegidas para el desarrollo de este trabajo son de tipo perceptrón multicapa [11]. La estructura utilizada para este sistema es un perceptrón multicapa con 4 neuronas en la capa de entrada, 3 neuronas en la capa oculta y 3 neuronas en la

capa de salida. Las funciones de activación de las neuronas son de tipo sigmoidal excepto las de la última capa que es de tipo lineal.

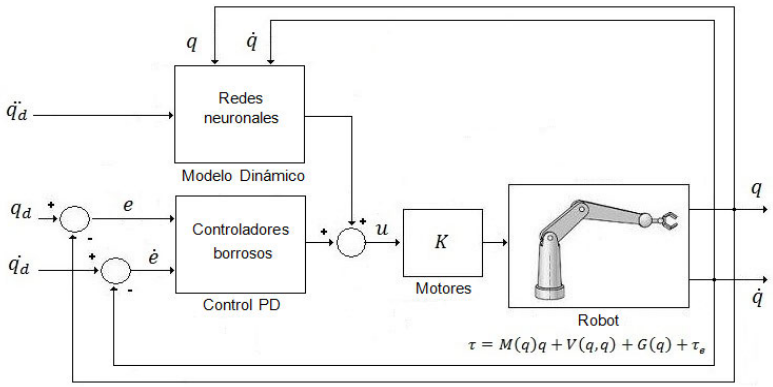


Figura 3. Método de control propuesto utilizando control borroso y redes neuronales.

Para modelar la dinámica del robot se implementaron tres redes neuronales de tipo perceptrón multicapa para aproximar cada uno de los términos del modelo dinámico del robot dados por las matrices $M(q)\ddot{q}_d$, $V(q,\dot{q})$ y $G(q)$.

La matriz $G(q)$ está dada por la siguiente ecuación, por lo que la red neuronal elegida debe aproximar la siguiente función vectorial (10):

$$\begin{bmatrix} 0 \\ -(m_2 g c_2 + m_3 g d) S_2 \\ m_3 g C_2 \end{bmatrix} \qquad \begin{bmatrix} \tau_{g1} \\ \tau_{g2} \\ F_{g3} \end{bmatrix} = \begin{bmatrix} 0 \\ -(m_2 g c_2 + m_3 g d) S_2 \\ m_3 g C_2 \end{bmatrix} \tag{10}$$

El vector de entrada a la red neuronal es el vector de las coordenadas articulares: $[\theta_1, \theta_2, d]$; el vector de salida de la red es: $[\tau_{g1}, \tau_{g2}, F_{g3}]$

La matriz $\mathrm{V(q,\dot{q})}$ está dada por:

$$\begin{bmatrix} 0 & 2(m_2 c_2^2 + m_3 d^2) S_2 C_2 \dot{\theta}_1 & 2 m_3 S_2^2 d \dot{\theta}_1 \\ -(m_2 c_2^2 + m_3 d^2) S_2 C_2 \dot{\theta}_1 & 0 & 2 m_3 d \dot{\theta}_2 \\ -m_3 S_2^2 d \dot{\theta}_1 & -m_3 d \dot{\theta}_2 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_1 \\ \dot{d} \end{bmatrix}$$

Por lo que la red neuronal elegida debe aproximar la siguiente función vectorial:

$$\begin{bmatrix} \tau_{v1} \\ \tau_{v2} \\ F_{v3} \end{bmatrix} = \begin{bmatrix} 0 & 2(m_2 c_2^2 + m_3 d^2) S_2 C_2 \dot{\theta}_1 & 2 m_3 S_2^2 d \dot{\theta}_1 \\ -(m_2 c_2^2 + m_3 d^2) S_2 C_2 \dot{\theta}_1 & 0 & 2 m_3 d \dot{\theta}_2 \\ -m_3 S_2^2 d \dot{\theta}_1 & -m_3 d \dot{\theta}_2 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_1 \\ \dot{d} \end{bmatrix} \tag{11}$$

El vector de entrada a la red neuronal es el vector de las coordenadas y velocidades articulares es: $[\theta_1, \theta_2, d, \dot{\theta}_1, \dot{\theta}_2, \dot{d}]$. El vector de salida de la red es: $[\tau_{v1}, \tau_{v2}, F_{v3}]$

La matriz $\mathrm{M(q)\ddot{q}}$ está dada por:

$$\begin{bmatrix} (m_2 c_2^2 + m_3 d^2) S_2^2 & 0 & 0 \\ 0 & m_2 c_2^2 + m_3 d^2 & 0 \\ 0 & 0 & m_3 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{d} \end{bmatrix}$$

Por lo que la red neuronal elegida debe aproximar la siguiente función vectorial:

$$\begin{bmatrix}\tau_{m1}\\ \tau_{m2}\\ F_{m3}\end{bmatrix}=\begin{bmatrix}(m_2c_2^2+m_3d^2)S_2^2 & 0 & 0\\ 0 & m_2c_2^2+m_3d^2 & 0\\ 0 & 0 & m_3\end{bmatrix}\begin{bmatrix}\ddot{\theta}_1\\ \ddot{\theta}_2\\ \ddot{d}\end{bmatrix} \quad (12)$$

El vector de entrada a la red neuronal es el vector de las coordenadas y aceleraciones articulares es: $[\theta_1, \theta_2, d, \ddot{\theta}_1, \ddot{\theta}_2, \ddot{d}]$. El vector de salida de la red es:

$$[\tau_{m1}, \tau_{m2}, F_{m3}]$$

Se entrenaron distintas redes para cada término del modelo dinámico con 4000 valores aleatorios de sus vectores de entrada y los valores obtenidos a partir de las ecuaciones (10-12) del modelo. Posteriormente se estimó el error de generalización de cada red con otros 1000 valores aleatorios de los vectores de entrada. Finalmente se escogió la red que produjo el menor valor del error de generalización estimado. Las características de cada red neuronal implementada se muestran en la tabla 2.

Tabla 2. Características de las redes neuronales empleadas para la estimación del modelo dinámico

Término del modelo dinámico	Matriz $G(q)$	Matriz $V(q,\dot{q})$	Matriz $M(q)\ddot{q}_d$
Número de entradas	3	6	6
Número de salidas	3	3	3
Número de capas ocultas	1	4	2
Nº de neuronas por cada capa oculta	10	50,30,20,10	30,15

3.2 Control Fuzzy del Sistema Desacoplado

Una vez que se ha conseguido la independencia dinámica de las articulaciones precompensando el sistema con la dinámica inversa del robot, se utilizarán controladores borrosos para el control de la posición de cada una de las articulaciones. Los controladores borrosos son de tipo proporcional-derivativo, donde las variables de entrada son $E = k_p e$ y $DE = k_v \dot{e}$, siendo e y $\dot{e}$ el error y la derivada del error de la señal respectivamente; la variable de salida es U. La señal de control que el controlador entrega al sistema es $u = k_u U$.

Los controladores difusos son de tipo Mamdani. El intervalo de valores de E, DE y U es *[-1,1]*. Para las variables E, DE y U se definieron los siguientes conjuntos difusos: NG: Negativo grande, NP: Negativo pequeño, C: Cero, PP: Positivo pequeño, PG: Positivo grande. Los conjuntos difusos están definidos por funciones triangulares y se muestran en la figura 4 para las entradas, y de igual forma se han asignado a la salida.

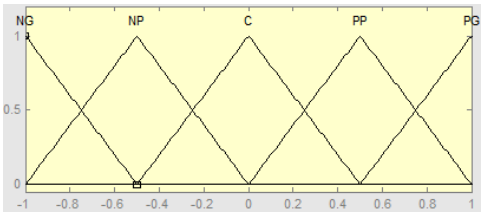


Figura 4. Conjuntos difusos para E y DE.

Con 5 conjuntos difusos para E y DE se definieron un total de 25 reglas (tabla 3):

Tabla 3. Base de reglas del control fuzzy PD

E/DE	NG	NP	C	PP	PG
NG	NG	NG	NG	NP	C
NP	NG	NG	NP	C	PP
C	NG	NP	C	PP	PG
PP	NP	C	PP	PG	PG
PG	C	PP	PG	PG	PG

El método de defuzificación utilizado en los controladores es el método del centroide.

4 Simulación del sistema y discusión de resultados

El sistema de control completo del robot se muestra en la figura 5. El diagrama de bloques principal incluye un subsistema de precompensación y control.

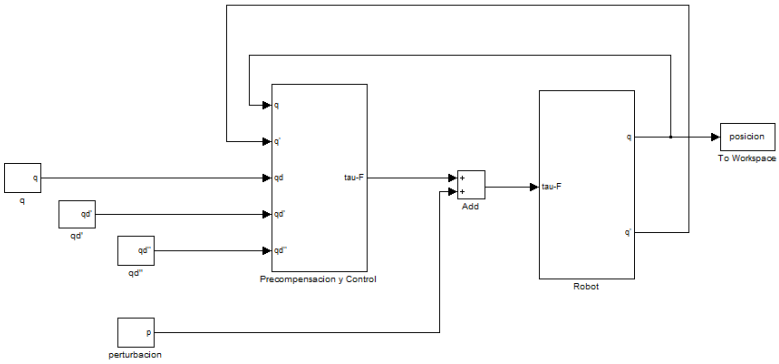


Figura 5. Diagrama de bloques principal del sistema de control del robot.

Se realizaron una serie de pruebas para probar la validez del diseño de estos controladores inteligentes. En primer lugar se realizó una simulación sin tener en cuenta las perturbaciones. Las trayectorias de referencia para cada una de las articulaciones fueron las siguientes: $\theta_1(t) = sen(0.1t)$, $\theta_2(t) = sen(0.1t)$, $d(t) = 0.2 + 0.06t$. Es decir, las trayectorias de las articulaciones giratorias son de tipo senoidal y la tercera articulación se debe mover con velocidad uniforme. El tiempo de simulación fue de t = 30 s. Las posiciones iniciales de las articulaciones fueron: $\theta_1(0) = 0$, $\theta_2(0) = {}^{\pi}/_{2}$, $\theta_3(0) = 0.2$

Es necesario mencionar que θ_2 no debe tomar el valor 0 ya que para dicho valor la matriz de inercia es singular y se podrían producir problemas de estabilidad en el sistema de control.

Las matrices de los parámetros de los controladores borrosos fueron:

$$K_p = \begin{bmatrix} 0.85 & 0 & 0 \\ 0 & 0.72 & 0 \\ 0 & 0 & 0.85 \end{bmatrix} \quad K_v = \begin{bmatrix} 0.62 & 0 & 0 \\ 0 & 0.56 & 0 \\ 0 & 0 & 0.65 \end{bmatrix} \quad K_u = \begin{bmatrix} 15.5 & 0 & 0 \\ 0 & 15.5 & 0 \\ 0 & 0 & 15.5 \end{bmatrix}$$

Finalmente la matriz K fue:

$$K = \begin{bmatrix} 20.5 & 0 & 0 \\ 0 & 20.5 & 0 \\ 0 & 0 & 20.5 \end{bmatrix}$$

Los resultados obtenidos de la simulación para el movimiento de las tres articulaciones del robot se muestran en la figura 6. En los tres casos la línea discontinua indica la trayectoria deseada y la línea continua la trayectoria real seguida por la articulación.

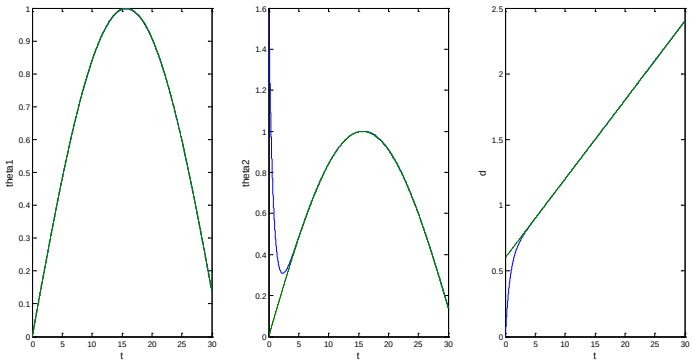


Figura 6. Trayectoria real (línea continua) y deseada (línea discontinua) de la primera, segunda y tercera articulación (de izquierda a derecha)

A continuación se repitió la simulación pero sometiendo al sistema a una perturbación en la señal de entrada al robot consistente en: $[-0.8H(x-8), 0.8H(x-16), 0.5H(x-20)]$. Es decir, tres funciones escalón en los instantes 8, 16 y 20 que perturban el movimiento de cada una de las articulaciones. Además, como la estimación de la dinámica de las redes neuronales no es exacta, podemos considerar que ya existe una perturbación asociada a esta falta de exactitud. Los resultados obtenidos son similares a los mostrados anteriormente. El sistema de control es capaz de hacer frente a ese tipo de perturbación.

En la figura 7 se representan las señales de salida del subsistema de precompensación y control. Se deben observar las variaciones de la señal en los instantes de tiempo en los que se produce cada una de las perturbaciones que producen la compensación de dichas perturbaciones y el control efectivo de la trayectoria de las articulaciones.

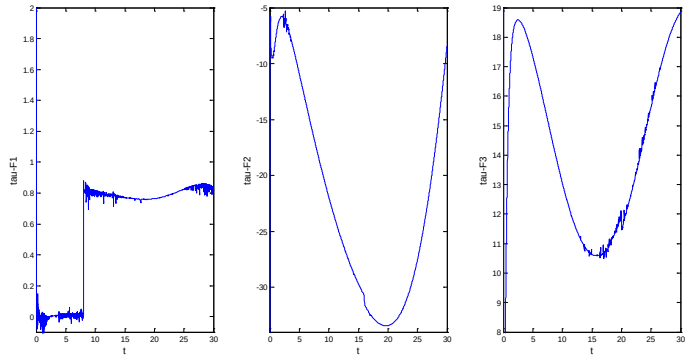


Figura. 7. Primera, segunda y tercera componente de la señal de salida del subsistema de precompensación y control (de izquierda a derecha)

Finalmente se evaluó la aproximación que realizan las redes neuronales de la dinámica inversa del robot, comparando en las figuras 8-10 las estimaciones neuronales (línea continua) de las matrices $M(q)\ddot{q}_d$, $V(q,\dot{q})$ y $G(q)$ y las de la dinámica exacta del robot (línea discontinua).

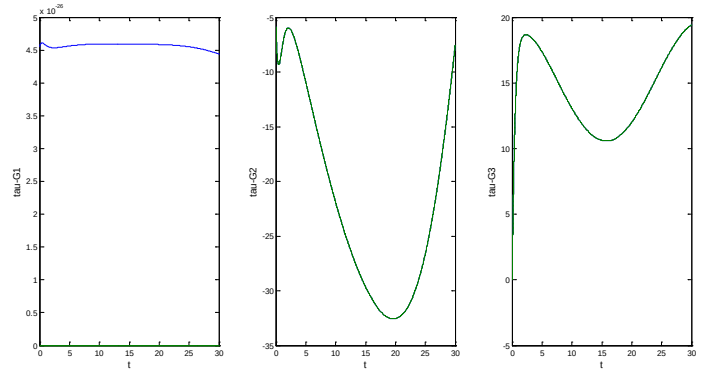


Figura 8. Componentes de $G(q)$.

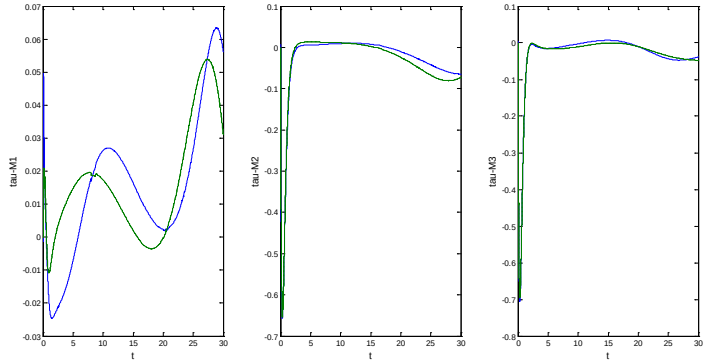


Figura 9. Componentes de $V(q,\dot{q})$.

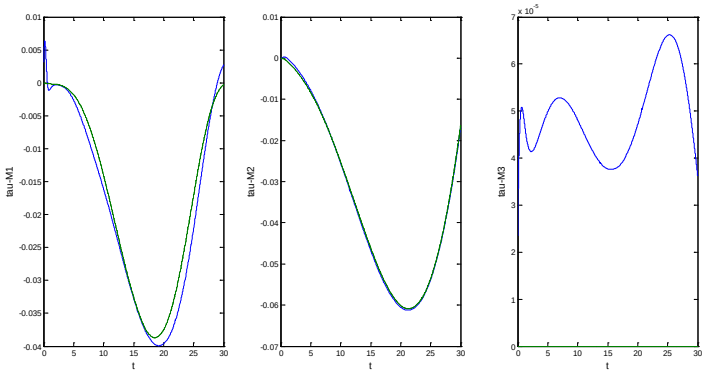


Figura 10. Componentes de $M(q)\ddot{q}_d$.

5 Conclusiones y Trabajos Futuros

Las pruebas realizadas en el apartado anterior muestran que el sistema de control propuesto con redes neuronales y controladores borrosos da resultados satisfactorios en las dos simulaciones realizadas, con el robot sometido a perturbaciones y sin estar sometido a ellas. Es decir, utilizando una estrategia de control de robots implementada habitualmente con controladores clásicos, se ha introducido la utilización de dos técnicas de control inteligente y éstas han demostrado ser eficaces en la resolución del problema de control propuesto.

Aunque en este trabajo se ha utilizado el propio modelo dinámico del robot para mostrar las capacidades de las redes neuronales, este modelo podría no ser conocido. En este caso se podrían utilizar las redes neuronales como método de identificación del sistema.

Referencias

1. Ollero, A. (2001) “Manipuladores y robots móviles”, Marcombo S.A. Boixaeru Ed., 204-210.
2. Barrientos A., Peñín L., Balaguer C., Aracil R. (1997) “Fundamentos de Robótica”, McGraw-Hill, 138-142, 322-325.
3. Artola G., Apóstoli R (2000) “Controlador de robots basado en redes neuronales”, 9º Congreso Chileno de Ingeniería Mecánica, Valparaíso, Chile.
4. ChunHua, Z. (2011) “A Class of Fuzzy Sliding-mode Control Simulation for Two-link Robot Manipulators”, School of Railway Tracks and Transportation, East China, Jiaotong University, Nanchang. 4th Int. Conf. Digital Image Processing (ICDIP 2012) Kuala Lumpur, Malaysia.
5. Heredia, J.A. (1999) “Control Neuronal para un Robot Manipulador con Compensación de Dinámicas no Modeladas”, México, DF.
6. Krabes, M., Döschner. C (1999) “Continous Modelling of Robot Dynamics Using a Multi-dimensional RBF-like Neural Network”, Institute of Automation, University of Magdeburg. Int. Conf. Information, Intelligence, and Systems.
7. Stoian, V., Ivanescu, M. (2008) “Robot Control by Fuzzy Logic”, University of Craiova, Romania. Frontiers in Robotics, Automation and Control, Alexander Zemliak (Ed.), InTech.
8. Sumathi, P. (2005) “Precise Tracking Control of Robot Manipulator Using Fuzzy Logic”, St.Joseph’s College of Engineering, Chennai, India. Mechatronics. IEEE Int. Conf. ICM '05.
9. Yin, J. (1998) “Decentralized Adaptive Fuzzy Control of Robot Manipulators”, IEEE Trans. Systems, Man, and Cybernetics Part B: Cybernetics, 28, 1, 47–57.
10. Hornik, K., Stinchcombe, M., White, H. (1989) “Multilayer feedforward networks are universal approximators”, Neural Networks, 2, 5, 359–366.
11. Isasi Viñuela, P., Galván León, I. M. (2004) “Redes de Neuronas Artificiales. Un enfoque práctico” Pearson Educación, Madrid.

Utilización de Técnicas de Soft Computing para el Control de un Robot Móvil Autónomo de 2 Ruedas

Joseba Menoyo Larrazabal[1], Matilde Santos Peñas[2]

[1]Universidad Nacional de Educación a Distancia, España
jmenoyo5@alumno.uned.es
[2]Universidad Complutense de Madrid, Madrid, España
msantos@dacya.ucm.es

Abstract. Este trabajo presenta la utilización de dos técnicas de Soft Computing para el control de un robot móvil de dos ruedas. La principal aportación es el diseño de un controlador basado en lógica fuzzy que regula su desplazamiento en un entorno con obstáculos. Por otro lado, para hacer más eficiente el sistema, se ha optimizado el control clásico que regula la velocidad de los motores de las ruedas mediante algoritmos genéticos. Se ha desarrollado una plataforma de simulación para mostrar los resultados.

Keywords: Robot autónomo diferencial, Soft Computing, Lógica Difusa, Algoritmos genéticos, PID

1 Introducción

Los robots autónomos tienen múltiples aplicaciones, tanto a nivel industrial como para otros ámbitos como el de la domótica. Sin embargo son sistemas poco estables y difíciles de controlar. Esta complejidad se debe, en primer lugar, a que son sistemas con una dinámica no lineal, multivariables y con acoplos entre sus variables de estado. Este hecho dificulta la obtención de un modelo matemático que represente con precisión su dinámica, y hay que recurrir a aproximaciones que crean incertidumbre respecto a su comportamiento [1]. Además deben ser controlados en tiempo real, por lo que la computación de los algoritmos de control no debe exigir mucho tiempo.

En este trabajo se ha analizado el diseño de un controlador basado en la lógica difusa para el desplazamiento del robot en un entorno con obstáculos. El desarrollo permite estudiar cómo afectan algunas variaciones del sistema borroso en la respuesta del sistema. El control difuso se aplica a innumerables tipos de aplicaciones, principalmente aquellas en los que los modelos matemáticos son muy complejos. De este modo, a partir de técnicas de razonamiento aproximado es factible realizar un control adecuado incluso cuando el entorno no se conoce de forma precisa [2].

Además, para hacer el sistema más eficiente se han optimizado, mediante la aplicación de algoritmos genéticos, los reguladores que controlan los motores de las ruedas del robot.

Para comprobar los resultados se ha generado una plataforma de simulación que genera mapas con obstáculos, donde el robot debe recorrer una trayectoria evitando a éstos.

El artículo se estructura como sigue. En primer lugar se realiza una breve descripción del robot que permitirá definir las variables a controlar. A continuación se diseña el control basado en lógica difusa para este sistema. En la siguiente sección se aplican los algoritmos genéticos a la optimización de los parámetros de sintonía de los controladores PID de las ruedas. Por último se muestran los resultados de la simulación del robot en sus desplazamientos en entornos con obstáculos.

2 Descripción del Sistema y Variables

Se ha desarrollado un modelo del robot que va a permitir simular su comportamiento y definir las especificaciones de respuesta del mismo para optimizar los controladores y validarlos.

El vehículo autónomo que se utiliza en este trabajo es un robot de 2 ruedas que pueden moverse independientemente (modelo diferencial). La figura 1 muestra un esquema del robot. Cada una de las ruedas dispone de un encoder óptico que permitirá calcular y controlar el movimiento realizado por cada una de ellas.

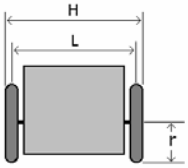


Fig. 1. Esquema del robot diferencial simulado

Sus principales características físicas son:

- Radio de las ruedas: r = 5cm
- Distancia entre ruedas: L = 10cm
- Dimensiones del robot: 10x20cm
- El encoder de cada rueda tiene una resolución de 100 muestras por vuelta. Ambos encoders son de tipo incremental y están acoplados a los ejes de giro de las ruedas, permitiendo obtener mediante odometría la posición y orientación del robot en todo momento [4].

Se van a aproximar tanto el comportamiento dinámico del robot como los sensores y actuadores de los que dispone.

2.1 Posición y Orientación del Robot

A partir de las características del robot se pueden convertir los pulsos leídos por el encoder en el desplazamiento de la rueda:

$$C_m = \frac{\pi D_n}{n C_e} \tag{1}$$

donde, C_m se define como el factor de conversión que convierte los pulsos del encoder en una descomposición lineal de la rueda; D_n es el diámetro nominal de la rueda, y C_e es la resolución del encoder en pulsos por revolución. En caso de utilizar reduc-

tores en los actuadores, "n" sería la relación de engranajes entre el motor y la rueda de tracción. Sin embargo, en este caso supondremos n=1 (accionamiento directo del motor DC).

El incremento lineal de la distancia recorrida por cualquiera de las ruedas se podrá calcular multiplicando el factor de conversión por el nº de pulsos leídos por el encoder (N_i). De este modo, el desplazamiento observado para cada una de las ruedas será:

$$\Delta U_{der} = N_{der} \cdot C_m = N_{der} \cdot \pi \cdot \frac{10}{100}$$

(2)

$$\Delta U_{izq} = N_{izq} \cdot C_m = N_{izq} \cdot \pi \cdot \frac{10}{100}$$

(3)

El incremento lineal de la distancia recorrida por el centro del robot (ΔU), se puede calcular como:

$$\Delta U = \frac{\Delta U_{der} + \Delta U_{izq}}{2}$$

(4)

Finalmente, el incremento del cambio de orientación se calcula de la siguiente forma (siendo "L" la distancia entre ruedas):

$$\Delta\theta = \frac{\Delta U_{der} - \Delta U_{izq}}{L}$$

(5)

De este modo, con las siguientes expresiones se podría calcular la posición (x, y) y orientación (θ) del robot después de cada desplazamiento:

$$\theta_i = \theta_{i-1} + \Delta\theta_i$$
$$x_i = x_{i-1} + \Delta U_i \cdot \cos(\theta_i)$$
$$y_i = y_{i-1} + \Delta U_i \cdot \sin(\theta_i)$$

(6)

2.2 Sensores de Ultrasonidos del Robot

Para detectar los obstáculos que pueda encontrarse el robot a su paso, éste dispone de tres sensores de ultrasonidos, orientados hacia la parte delantera, la izquierda y la derecha. Los valores medidos por estos sensores son las entradas del controlador.

Este tipo de sensores funcionan emitiendo pulsos ultrasónicos. Si estos pulsos reflejan en un objeto, el sensor recibe el eco producido y lo convierte en señales eléctricas. De esta forma pueden detectar objetos con diferentes formas, superficies y materiales. Un sensor de ultrasonidos tiene un alcance máximo relacionado principalmente con el periodo de activación entre pulsos del transductor y la necesidad de un tiempo de supresión de eco. El alcance de los sensores del robot es de 100 cm, con un ángulo de apertura aproximado de 60º (Figura 2).

Situando el sensor delantero orientado hacia el ángulo de referencia al que apunta la parte delantera del robot (θ=0º), éste cubrirá desde +30º a -30º. Si los sensores

laterales se direccionan a +/-60º respecto a la orientación del sensor delantero, se consigue cubrir toda la superficie desde +90º a -90º.

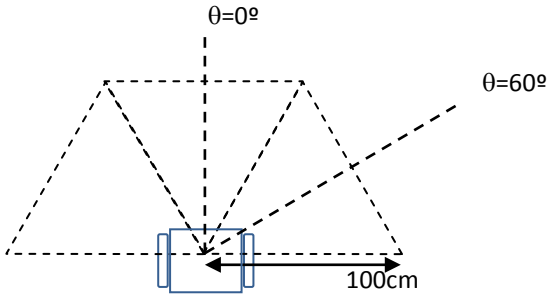


Fig. 2. Disposición de los sensores y aproximación de la superficie de alcance

2.3 Actuadores de Accionamiento de las Ruedas

Cada rueda dispone de su propio motor, controlado por un PID. Una característica de un robot diferencial es que ambas ruedas se muevan de forma independiente. El modelo del actuador de las ruedas es un motor DC con las siguientes características:

- Momento de inercia del rotor: J=0.01 kgm^2
- Fricción: b=0.1 Nms
- Constante de la fuerza contraelectromotriz: K=0.01Nm/A
- Resistencia: R=1.6 ohm
- Inductancia: L=0.48 H

Considerando que la fuerza contraelectromotriz equivale a:

$$e_b = K\frac{d\theta}{dt} \tag{7}$$

Se obtienen las siguientes expresiones:

$$L\frac{di}{dt} + Ri = V - K\frac{d\theta}{dt}$$
$$LsI(s) + RI(s) = V(s) - K(s)s\theta(s)$$
$$I(s) = \frac{V(s) - K(s)s\theta(s)}{R + Ls} \tag{8}$$

A partir de la expresión anterior de I(s) se puede llegar a una ecuación que relaciona el ángulo de giro θ(s) y la tensión aplicada V(s):

$$Js^2\theta(s) + bs\theta(s) = KI(s)$$
$$Js^2\theta(s) + bs\theta(s) = K\frac{V(s) - K(s)s\theta(s)}{R + Ls} \tag{9}$$

Lo que permite obtener la siguiente función de transferencia que será simulada para la optimización del controlador de las ruedas:

$$\frac{\theta(s)}{V(s)} = \frac{K}{s\left((R+Ls)(Js+B)+K^2\right)} \tag{10}$$

3 Control Difuso para el Desplazamiento del Robot

Los sistemas de control difuso o borroso utilizan expresiones lingüísticas para formular las reglas que controlan el sistema. En este tipo de sistemas debe tenerse en cuenta el conocimiento de expertos ya que la toma de decisiones se fundamenta en lo que se defina en ellas [6].

En esta aplicación, el objetivo del sistema del control es conseguir que el robot se desplace por un mapa sin colisionar con ninguno de los obstáculos situados en esa superficie. Para ello se ha diseñado e implementado, usando la toolbox de Lógica Fuzzy de Matlab, [7], un controlador que se encargará de determinar el movimiento a aplicar en cada una de las dos ruedas del robot, en función de la disposición de los obstáculos que detecta cada uno de los tres sensores de los que dispone el sistema.

Por lo tanto, para el sistema de control difuso se han definido 3 entradas, correspondientes a las lecturas de cada uno de los sensores (delante, izquierda, derecha), y 2 salidas, correspondientes a la cantidad de segmentos del encoder que deberá girar cada rueda (Figura 3). Se ha optado por un modelo tipo Mamdani para el controlador borroso.

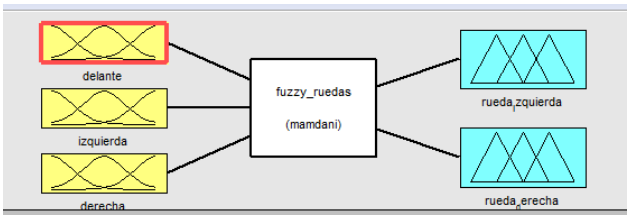


Fig. 3. Esquema del controlador

Variables de entrada.
El alcance máximo de los sensores es de 100 cm, por lo que se ha considerado para las 3 señales de entrada el dominio de números reales [0, 100]. Sobre cada uno de estos universos de discurso se definen 4 conjuntos borrosos, con funciones de pertenencia triangulares, en función de la distancia a la que el sensor detecta el obstáculo más cercano en esa dirección: MC (Muy Cerca), C (Cerca), L (Lejos) y ML (Muy Lejos). La figura 4 muestra la configuración de una de las entradas, y de igual forma se han definido las otras dos.

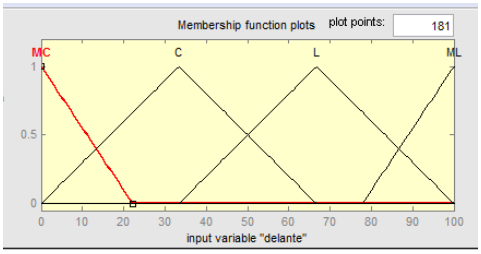


Fig. 4. Conjuntos borrosos para los sensores (entradas)

Variables de salidas.
El modelo de robot de esta aplicación es del tipo diferencial de dos ruedas, por lo que éstas pueden girar de forma independiente. En función de la velocidad y sentido de giro de cada una de ellas, el robot se desplazará más o menos rápido, girará en un sentido u otro, etc.

Se ha definido que durante la simulación, por cada unidad de tiempo, el robot pueda girar cada rueda como máximo el equivalente a la lectura de 30 segmentos del encoder. De este modo, se considera para las 2 señales de salida el dominio de números reales [-30, 30] (los valores negativos implican el giro de la rueda en sentido contrario).

Sobre cada uno de estos universos de discurso se definen 5 conjuntos borrosos en función de la velocidad a la que girará la rueda: MN (Muy Negativo), N (Negativo), Z (Cero), P (Positivo) y MP (Muy Positivo). Tiene asociados funciones de pertenencia triangulares. La figura 5 muestra una de las salidas.

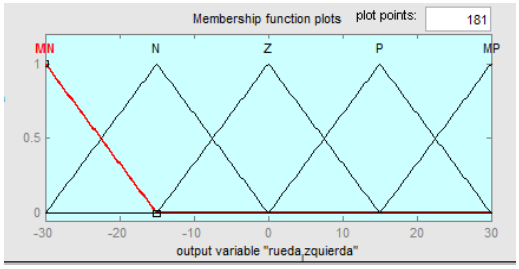


Fig. 5. Conjuntos borrosos para el accionamiento de giro de ruedas (salidas)

Reglas.
Las reglas tienen como antecedentes las tres variables de entrada, y como consecuentes las acciones sobre las dos ruedas, como en la expresión siguiente,

```
SI (delante) es ML Y (izquierda) es C Y (derecha) es MC
ENTONES (rueda_izquierda) es Z, (rueda_derecha) es MP
```

Por ejemplo, si el sensor delantero detecta un objeto relativamente lejos (80cm), el sensor izquierdo detecta que tiene un obstáculo muy cerca (10cm), y el sensor derecho da una lectura de 100 (lo que significa que, o bien existe un obstáculo justo en el

límite de detección del sensor, o que no hay ningún obstáculo en esa dirección), el resultado ante esta situación es un giro de la rueda izquierda de +24.5 segmentos del encoder, y de +5.69 para la rueda derecha. De este modo, al girar la rueda izquierda más rápido que la derecha, el robot se sigue desplazando hacia delante, pero girando hacia la derecha, alejándose así del obstáculo más cercano.

Aplicando las ecuaciones definidas en la sección 2.1 se obtiene que con estos datos el robot girará 33.85º hacia la derecha, y que su centro se desplazará 3,93cm en el eje X y 2.64cm en el eje Y.

También se ha probado a cambiar las funciones triangulares de las señales de entrada y salida por funciones gaussianas. La diferencia en el resultado del desplazamiento no es significativa. Por ejemplo, para los valores de entrada anteriores la salida cambia de: rueda izquierda:+24.5, rueda derecha: +5.69 con funciones triangulares a rueda izquierda:+24.3, rueda derecha: +4.88, con función gaussiana. Se observa que la diferencia no es significativa.

4 Optimización mediante AG del Controlador de las Ruedas

Para implementar la salida obtenida con el controlador fuzzy el sistema cuenta con reguladores PID que accionan los motores de las ruedas. Estos controladores clásicos son optimizados mediante algoritmos genéticos para conseguir una respuesta rápida y que cumpla unas especificaciones de control, ya que es importante que el robot reaccione rápidamente ante la consigna que se le da. En otro caso, podría chocarse con algún obstáculo. También es importante conseguir una buena respuesta ante perturbaciones, que deben ser corregidas de forma eficaz y en el mínimo tiempo posible.

Los Algoritmos Genéticos (AG) reciben ese nombre por inspirarse en la base genética de la evolución biológica. En el caso particular que estamos tratando, se aplican los AG a los tres parámetros de sintonía del PID, las ganancias proporcional Kp, integral Ki, y derivativa Kd. Estos valores se calculan fuera de línea, y se aplican a los controladores para su funcionamiento durante todo el recorrido realizado por el robot.

La configuración del AG aplicado en este problema es la siguiente: cada individuo está formado por tres cromosomas (16 bits), inicializados de forma aleatoria. La función objetivo es el Error Cuadrático Medio (MSE) o la Integral del Tiempo por el Error Absoluto (ITAE) [3] [5] de la respuesta a una señal escalón. Se ha usado el método de la ruleta para seleccionar a los progenitores, diversos tipos de reinserción, y distintos valores para el tamaño de la población inicial, probabilidad de cruce y de mutación.

Al finalizar las simulaciones se seleccionará el individuo que proporcione el menor error en el sistema, determinando sus valores como los parámetros definitivos del controlador PID.

A modo de ejemplo en la Figura 6 se muestran los resultados obtenidos al aplicar los dos tipos de error en la función objetivo con los siguientes parámetros del AG: Nº individuos: 100; Generaciones: 40; Probabilidad de cruce: 0.9; Probabilidad de mutación: 0.01; Tipo de reinserción: Tipo 0 (mejores individuos de cada generación).

Al aplicar el error ITAE en la función objetivo, los parámetros PID obtenidos son los siguientes:

K_p: 99.9832 K_i: 0.0100 K_d: 34.7574

Al aplicar el error MSE como función objetivo, los parámetros PID obtenidos son los siguientes:

K_p: 99.9939 K_i: 8.8502 K_d: 100.00

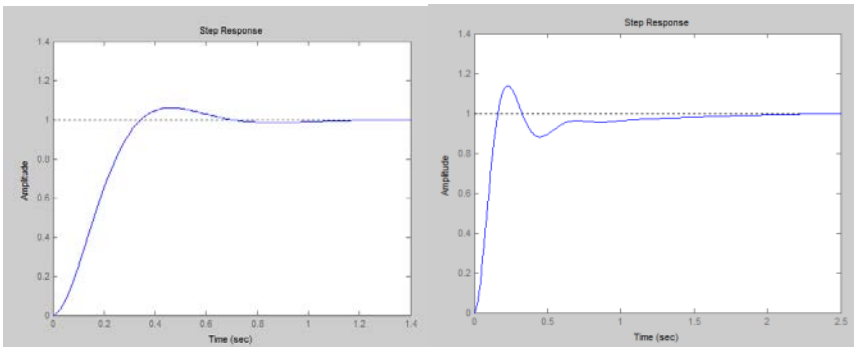


Fig. 6. Respuesta a una señal escalón con cálculo del error ITAE (izquierda) y MSE (derecha)

Tras diversas simulaciones se ha observado que los resultados obtenidos al utilizar el error ITAE en la función objetivo proporcionan una respuesta más suave y con menores oscilaciones. Las respuestas al aplicar como función objetivo el error del tipo MSE son más rápidas, aunque se ha considerado que la diferencia no es significativa ya que lo importante es cuándo alcanza el valor estacionario, y la precisión del motor no es suficiente para notar esta diferencia.

A modo de ejemplo, en la tabla 1 se muestran los resultados obtenidos al repetir varias simulaciones con los mismos parámetros (Población: 100 individuos; 40 generaciones; P_{cruce}=0.9; $P_{mutación}$=0.01). Se puede comprobar que las diferencias entre las diferentes soluciones obtenidas para estas condiciones concretas de simulación son muy pequeñas.

Tabla 1. Resultados de varias simulaciones utilizando los mismos parámetros del AG

Nº Simulación	K_p	K_i	K_d
1	99.9832	0.0100	34.7574
2	99.9924	0.0111	34.9146
3	99.9497	0.0100	35.0000
4	99.9969	0.0103	34.9726
5	99.6170	0.0115	34.3272
6	99.5850	0.0131	34.1486

4.1 Análisis de los Parámetros de Configuración del Algoritmo Genético

En las aplicaciones de sistemas reales, donde es necesaria una respuesta rápida, es importante disminuir el tiempo computacional de los algoritmos de control. Con este objetivo se ha analizado cómo influyen algunos parámetros de configuración del algoritmo genético en el tiempo de cálculo de los parámetros de sintonía del PID.

Se ha analizado el tiempo de convergencia estudiando la evolución del error medio por individuo en cada una de las generaciones, así como la del error del mejor individuo de cada generación. A partir de ese análisis se ha deducido que, para los valores anteriores, el algoritmo converge para 25 generaciones aproximadamente, y las iteraciones posteriores no aportan una mejora sustancial.

También el número de individuos que constituyen la población puede tener un efecto significativo en los resultados del algoritmo y sobre todo en su aplicación al sistema en tiempo real. Al disminuir el número de individuos la carga computacional será menor. Por otro lado, un aumento en el número de individuos incrementa las posibilidades de obtener una mejor solución al sistema. En la Tabla 2 se muestran los resultados obtenidos en función del número de individuos definidos para la población (de 150 a 40) para cinco simulaciones del mismo algoritmo (30 generaciones; P_{cruce}=0.9; $P_{mutación}$=0.01).

Tabla 2. Resultados de simulaciones (criterio ITAE) para distinto tamaño de la población

Nº	150	100	80	60	40
1	1.5035	1.5090	1.5051	1.5116	1.5773
2	1.5018	1.5077	1.5755	1.5475	2.1397
3	1.5046	1.5183	1.5044	1.5279	1.8678
4	1.5027	1.5037	1.5127	1.5084	2.3175
5	1.5074	1.5077	1.5609	2.0202	1.5180

Se observa que a medida que disminuye el nº de individuos, el error del individuo seleccionado como solución tiende a aumentar. Del mismo modo, la diferencia entre las soluciones obtenidas en las diferentes repeticiones también aumenta.

También se ha estudiado la influencia del tipo de reinserción que se aplique para generar la siguiente población. Se ha comprobado que, para este problema, el elitismo (mejor individuo de cada generación) proporciona una convergencia más rápida y mejores soluciones.

5 Resultados de Simulación

Se ha desarrollado una plataforma programada íntegramente en Matlab para simular tanto el controlador difuso como la dinámica del robot y diversos escenarios, con el fin de validar el diseño del controlador inteligente.

Se genera un mapa como una matriz binaria donde el valor de 1 equivale a la existencia de un obstáculo en esa posición. Se han creado 4 tipos de mapas para realizar diferentes pruebas. Se inicializa la posición/orientación inicial del robot, nº de pasos a simular y tipo de mapa a cargar. Se leen las medidas de los sensores de ultrasonidos que se pasan como entrada al controlador difuso. La salida del sistema de control borroso se sustituye en las ecuaciones (6) para calcular el desplazamiento y cambio de orientación realizados por el robot. De este modo, en cada iteración se va actualizando la posición del robot, para finalmente mostrar en pantalla el recorrido competo realizado en el mapa.

En la Figura 7 se muestran dos ejemplos del funcionamiento del robot, en los cuales se pueden observar los cambios de dirección que toma el robot en función de la disposición de obstáculos más cercanos a lo largo de su trayectoria.

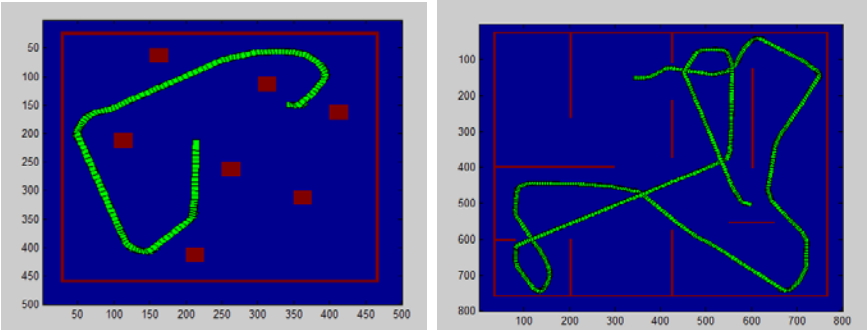


Fig. 7. Simulación del robot móvil en dos escenarios con obstáculos

6 Conclusiones y trabajos futuros

En este trabajo se ha diseñado un controlador difuso para el desplazamiento de un robot móvil diferencial autónomo. Las entradas del controlador provienen de tres sensores, y como salida se obtiene la posición y orientación del mismo. Para conseguir ese desplazamiento se utilizan motores en las dos ruedas, que a su vez son controlados por sendos PID. Estos reguladores clásicos han sido sintonizados mediante algoritmos genéticos.

El desarrollo ha sido validado mediante simulación en una plataforma que se ha creado para ello, con buenos resultados, consiguiendo que el robot evite colisiones con obstáculos situados en el mapa de su trayectoria.

Se ha analizado el funcionamiento del sistema difuso y del algoritmo genético, viendo cómo la variación de sus parámetros influye en la respuesta final.

Como trabajo futuro sería interesante incluir más sensores en la zona delantera reduciendo su ángulo de apertura para conseguir mejores resultados.

Referencias

1. Antsaklis, P.J. (1994), “Defining intelligent control”. IEEE Control Systems.
2. Santos, M.: Un enfoque aplicado del control inteligente, RIAI, 8, 283-296 **(**2011)
3. Andri Mirzal, Shinichiro Yoshii, Masashi Furukawa, "PID Parameters Optimization by Using Genetic Algorithm". Hokkaido University.
4. Borenstein, Johann (1996), "Measurement and Correction of Systematic Odometry Errors in Mobile Robots". IEEE Transactions on Robotics and Automation.
5. Deepyaman Maiti, Ayan Acharya, Mithun Chakraborty, Amit Konar, "Tuning PID and $PI^{\lambda}D^{\delta}$ Controllers using the Integral Time Absolute Error Criterion". Jadavpur University.
6. Zadeh, L.A. (1965), “Fuzzy Sets”. Inform. Contr., vol. 8, 338-352.
7. MathWorks (2011), “Fuzzy Logic Toolbox Toolbox - User’s Guide - R2011b” The MathWorks.

ANMV: Augmented Navigation and Virtual Mapping

Ricardo Pragnell[1], Miguel Gutierrez[1], Victor Ortiz[1], Alejandro Peñalver[1], Victoria Lopez[2]

[1] Faculty of Informatics, Complutense University, Madrid, Spain
Profesor García Santesmases, s/n,
28040 Madrid, Spain
{rpragnellvalentin, vicortizgarcia, apsantorio, chendo89}@gmail.com

[2] Dep. Computer Architecture and Automatics, Complutense University, Madrid, Spain
Profesor García Santesmases, s/n,
28040 Madrid, Spain
vlopez@fdi.ucm.es

Abstract. Indoor localization and mapping has been a largely unresolved problem. Current state-of-the-art approaches use 3D laser scanning (LIDAR) and radio wave positioning. A major drawback is that they require very expensive equipment and are time consuming to set up. This paper presents a complete software and hardware solution: ANVM (Augmented Navigation & Virtual Mapping). This solution provides an inexpensive and accurate mapping device, map editing software and a smartphone application based on Android. ANVM is an innovative project engaging in the fields of robotics, visual computing, and mobile application development. The main goal is to provide a solution to the problem of indoor localization and navigation according to this very simple process: 1. Firstly virtual reconstruction of the space to obtain a 2D map; 2. Secondly the process of gathering information on items to be shown on this map. To this end, this project is further divided into smaller projects that perform each of these tasks: ANVM - Mobile App and ANVM - Virtual Mapping. The localization is carried out using a smartphone device currently available to most users. As result, this tool offers a set of features useful for the user within the interior space.

Keywords: SLAM, particle filter, Kinect, Arduino, reconstruction, 3D Maps.

1 Introduction

ANVM arises with the main idea of capitalizing on a device that offers great possibilities: Microsoft Kinect [6]. This device is able to provide the necessary data to virtualize 3D space and recognize objects inside it. This project aims to apply the device to indoor localization. Initially this is to virtualize the interior space three-

dimensionally with the Kinect and then to use such virtual reconstruction to locate elements within it, including the user of the application. With the increasing use of smartphones in our society, localization via these devices is a good idea. Once the user is located within the interior space, the application aims to guide the user through the reconstruction of the space to the user's destination.

ANMV offers virtualization in two dimensions (a plane or map) based on Bayesian map learning [7] instead of a three-dimensional reconstruction. Furthermore, ANMV offers the possibility to navigate the reconstruction in a smartphone application by using panoramic images. The first approach using image based localization was replaced by a system based on QR codes placed at strategic points in the interior space.

This project is divided into two sub-projects: ANVM - Mobile App and ANVM - Virtual Mapping, that perform such tasks. This paper is devoted mostly to the second of these.

ANVM - Virtual Mapping is responsible for preprocessing the interior and obtaining a map of the area on which the virtualization process is performed.

The scanner device is comprised of a Microsoft Kinect [6], an onboard Arduino Uno microprocessor [1] and a custom mobile platform which utilizes digital encoders for position tracking. By combining the Kinect point cloud data and position information we are able to virtualize indoor environments utilizing the Rao-Blackwellized Particle Filter SLAM technique [2, 5].

Finally through extensive experiments, we have achieved good results succeeding in our goal to create an inexpensive mapping device, costing approximately 90 euros.

This paper is organized as follows: Section 2 is devoted to particle filters and a brief explanation of the related algorithm. In section 3 the mobile platform and its constructions is shown. Section 4 gives details about the mobile app software developed as an interface for the ANMV project and provides data on the architecture. Finally, our conclusions are given in section 5.

2 Particle filters

Particle filters, also known as a sequential Monte Carlo methods (SMC) [4], are a sophisticated model estimation technique based on simulation. Particle filters are usually used to estimate Bayesian models [7] in which the latent variables are connected in a Markov chain - similar to a hidden Markov model (HMM), but typically where the state space of the latent variables is continuous rather than discrete, and not sufficiently restricted to make exact inference. (Definition as seen in [10])

Recent studies have generated several optimizations, in particular we chose Rao-Blackwellised Particle Filtering (RBPF) for Dynamic Bayesian Networks by Arnaud Doucet, Nando de Freitas, Kevin Murphy, and Stuart Russel [2]. The key concept behind this optimization is to decrease the number of particles necessary to achieve same accuracy as with regular particle filtering. This is achieved by partitioning the state nodes, due to the fact that some partitions require a smaller amount of particles we are able to reduce the global number of particles. The particle filter iteration loop and RBPF equations are defined in [2] as follows.

INPUT: N particles (samples) $\left\{r_{0:t-1}^{(i)}, x_{0:t-1}^{(i)}\right\}$ at time *t-1* approximately distributed according to $p\left(r_{o:t-1}^{(i)}, x_{0:t-1}^{(i)} | Y_{1:t-1}\right)$

OUTPUT: N particles $\left(r_{0:t}^{(i)}, x_{0:t}^{(i)}\right)$ at time *t*.

ALGORITHM:
Step 1: Sequential importance sampling step
For i=1,…, N, the sample

$$\left(\hat{r}_t^{(i)}\right) \sim q\left(r_t | r_{0:t-1}^{(i)}, \quad Y_{1:t}\right)$$

And set

$$\left(\hat{r}_{0:t}^{(i)}\right) \triangleq \left(\hat{r}_t^{(i)}, \quad r_{0:t-1}^{(i)}\right)$$

For i=1,…, N evaluate the importance weights up to a normalizing constant:

$$w_t^{(i)} = \frac{p\left(\hat{r}_{0:t}^{(i)} | Y_{1:t}\right)}{q\left(\hat{r}_t^{(i)} | r_{0:t-1}^{(i)}, Y_{1:t}\right) \cdot p\left(\hat{r}_{o:t-1}^{(i)} | Y_{1:t-1}\right)}$$

For i=1,…, N normalize the importance weights by:

$$\widetilde{w}_t^{(i)} = w_t^{(i)} \left[\sum_{j=1}^{N} w_t^{(j)}\right]^{-1}$$

Step 2: Sample selection
Multiply/suppress samples $\left(\hat{r}_{0:t}^{(i)}\right)$ with high/low importance weights $\widetilde{w}_t^{(i)}$ to obtain N random samples $\left(\tilde{r}_{0:t}^{(i)}\right)$ approximately distributed according to $p\left(\tilde{r}_{o:t}^{(i)} | Y_{1:t}\right)$.

Step 3: MCMC step.
Apply a Markov chain Monte Carlo transition kernel with invariant distribution given by $p\left(r_{o:t}^{(i)} | Y_{1:t}\right)$ to obtain $\left(r_{0:t}^{(i)}\right)$.

As an example, Fig.1a shows N samples obtained using the mobile platform, each one from a different particle, which are then combined into a single map, as it is shown in Fig 1b by using RBPF SLAM.

In Fig. 2a –c all results were taken moving the mobile platform in a straight line. Red dots represent the mobile platform's position and blue dots show the Kinect point cloud captures. Fig. 2a corresponds to an initial test where the encoders are not calibrated correctly, as a result lots of deviation is obtained. In Fig. 2b the encoders are calibrated separately. In this case we achieved better results but still much deviation exists due to encoder mechanism slipping. Finally, Fig. 2c thanks to integrated adjustable pressure into the encoder system we almost achieved correct rectilinear motion.

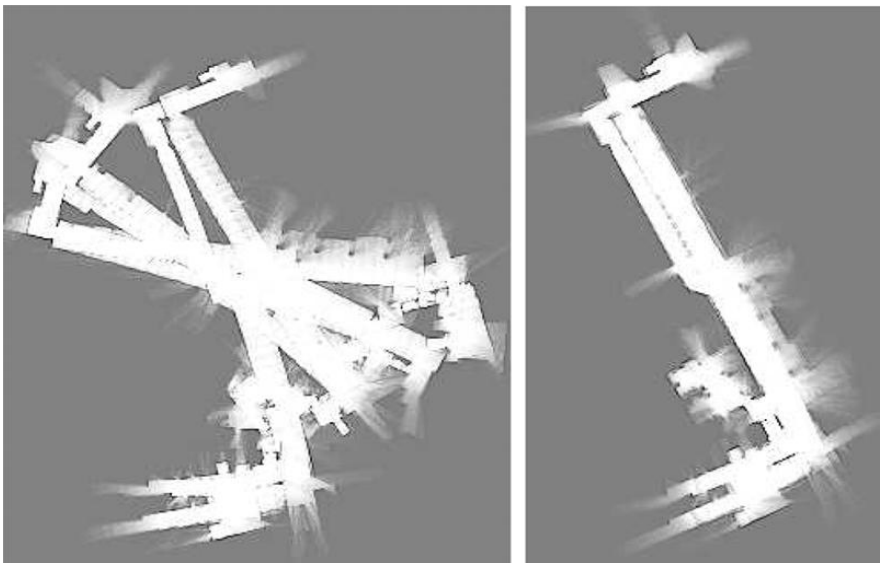


Fig. 1ab. Before and after RBPF SLAM

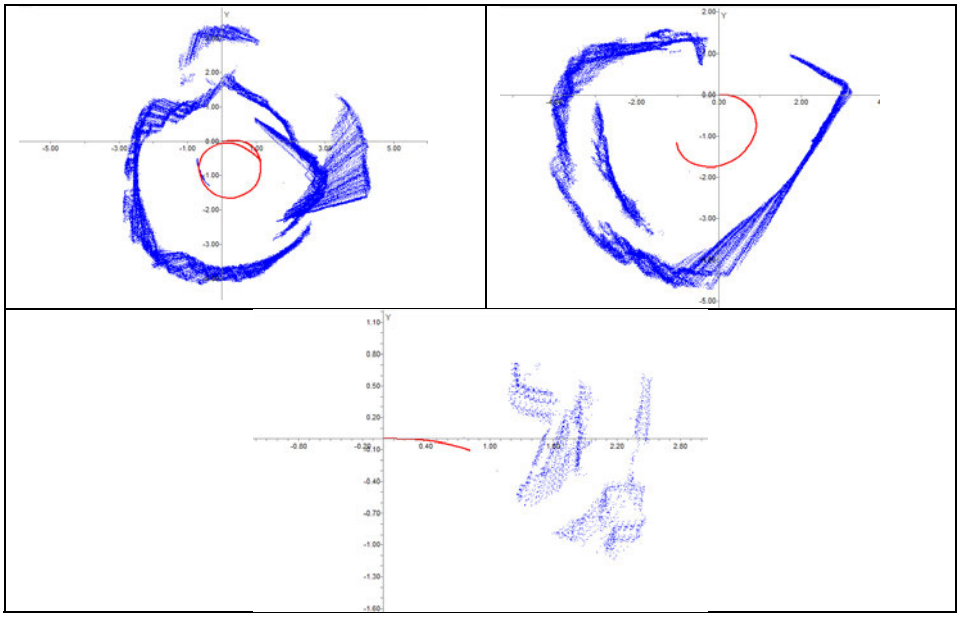


Fig. 2abc. Results according to different encoder settings.

3 Mobile platform

The inputs for the algorithm are raw laser-like scans, captured with Microsoft's Kinect, and positional odometry data provided by our own mobile platform. The mobile platform has a digital encoder coupled into each wheel, providing rotary information. Said encoders are attached using inexpensive spare LEGO pieces, which make it very easy to integrate into other platforms. Figures 3-5 show several parts of the digital encoder attached.

Encoder data is processed by an onboard Arduino Uno chip and sent via serial communication to the workstation running the SLAM algorithm.

Once both sources of information are received they are combined producing different possible time dependent locations for the robot, which are stored into particles. Finally the particles are filtered and the most accurate position is stored. This information is then used to progressively map the platform's surroundings.

Fig. 3. Mobile platform

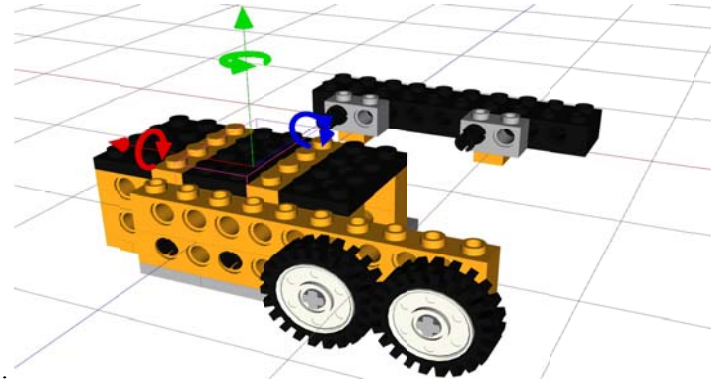


Fig. 4. Encoder design

Fig. 5. Digital encoder attachment

4 Mobile app software and architecture.

This application developed for Android smartphones offers all the functionality to the end user of this project. From ANVM generated map - Virtual Mapping, and after receiving the necessary information from the ANVM - Map Editor allows the user to browse that map. It also offers a set of additional utilities for location and exploration, guided by panoramic images (similar to Google Street View) allowing the user to discover the attractions of the place virtualized. This is a general application that works for any space reconstructed as the parameters simply require the site map and points of interest (restaurants, restrooms, offices or outlets for example). It is designed to automatically configure the application depending on the place rebuilt, without having to create a mobile application for each particular location. For example, in the reconstruction of an airport, the necessary files will be generated using the Map Editor. When set up correctly both the map and airport data will be shown. If later you decide to rebuild a mall for example the same process will be followed, without having to modify the mobile application beyond a change in visual appearance and labels.

Panoramic images used in the smartphone app take up considerable space as a large number of them are required (one for each map node). Therefore, to minimize the space taken up by the app in the phone memory, images are loaded directly from the Internet.

When the user decides to visualize the 3D view of a point of interest (POI), the system launches a thread that downloads the image from a server using a unique URL. Dropbox was used as the storage server. In order to reduce the namespace load as far as possible, these directions differ from each other only in the number of the file in question, and each file is named with the ID of each map node. When an InfoNode is selected in the 3D view, the application will access the map node to which it is connected and download the image file. Fig. 6 shows this process. Downloading said image files must be performed in a different thread from the main one, this being a requirement of the Android platform. Otherwise the app is locked and it is not possible to download. Furthermore, carrying out this action in a separate thread allows smooth user interaction with the displayed panoramas.

5 Conclusions

ANVM (Augmented Virtual Navigation and Mapping) is an innovative project that involves the fields of robotics, visual computing, and mobile application development. The project provides a solution to the problem of indoor locatization by means of a tool capable of providing (in shopping malls, airports, museums, etc.) a better orientation for users, according to two main processes: 1. Virtual reconstruction space to obtain a 2D map. 2. Process of collecting information on the items you want to display on the map. To do this, ANVM is subdivided into two sub-projects that perform each of these tasks: ANVM -Mobile App and ANVM - Virtual Mapping.

The system allows users to be located and to find points of interest within a certain interior space. The localization is carried out using a smartphone, a device currently available to the majority of users. It also offers a set of useful features for the user

within the interior space. ANVM - Mobile App is the sub-project dedicated to fulfilling those tasks.

For the correct operation of this project on your smartphone, you need the interior space preprocessed by the ANVM - Virtual Mapping system, which relies on the Rao-Blackwellized Particle Filter SLAM technique.

This project also includes a program developed in Java, named ANVM - Map Editor that acts as a bridge between the sub-projects. This desktop application enables data generated by the virtual mapper to be interpreted later by the mobile application. It also allows the inclusion of points of interest that the project super-user wishes to include within the mobile application.

Together these tools (ANVM - Mobile App, ANVM - Virtual Mapping and ANVM - Map Editor) comprise the ANVM complete feature set.

Finally through extensive experiments, we have achieved good results succeeding in our goal to create an inexpensive mapping device.

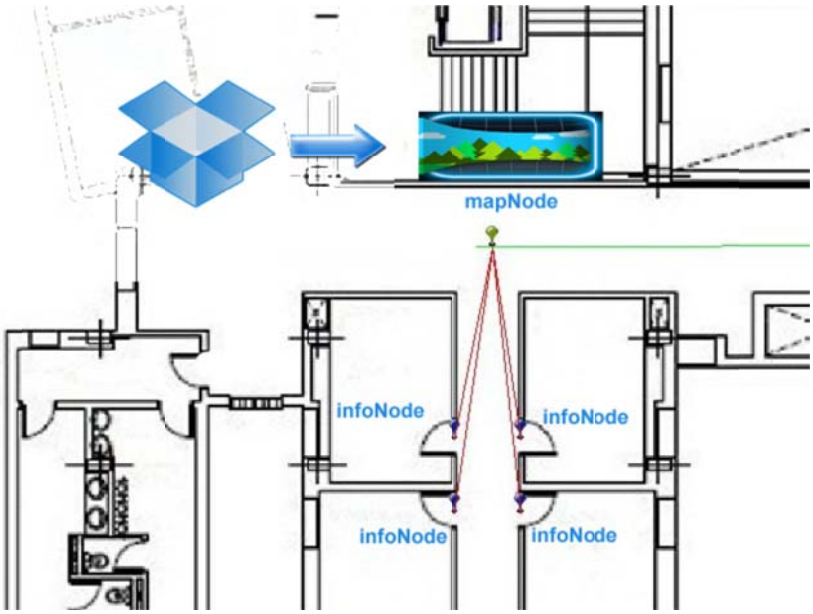


Fig. 6. Corresponding between mapNodes and infoNodes

Referencies

1. Arduino Uno, http://arduino.cc/en/Main/arduinoBoardUno (2005)
2. Arnaud Doucet, Nando de Freitas, Kevin Murphy, Stuart Russel : Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks. UAI '00 Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann Publishers Inc San Francisco, CA, USA (2000)
3. Candy, J. V. (2009). Bayesian Signal Processing: Classical, unscented and Particle filtering methods. Adaptive and learning systems for Signal Processing, Communications and Control Series, John Wiley & Sons.
4. Capp'e, O., Godsill, S. and Moulines, E. (2007) An overview of existing methods and recent advances in sequential Monte Carlo, Proceedings of the IEEE, vol 95 (5), 899-924.
5. Casella, G., Robert, C.P. (1996). Rao-Blackwellisation of sampling schemes, Biometrika, Vol. 83 (1), 81-94.
6. Microsoft Kinect, http://www.microsoft.com/en-us/kinectforwindows/ (2010).
7. Murphy, K. P. (2000). Bayesian map learning in dynamic environments, Advances in Neural Information Processing Systems, 12, MIT Press, 1015-1021.
8. Hu, X.L. , Shön, T. and Ljung, L. (2008). A basic convergence result for particle filtering, IEEE Transactions on Signal Processing, Vol. 56 (4), 1337-1348.
9. Hu, X.L. , Shön, T. and Ljung, L. (2011). A general convergence result for particle filtering, IEEE Transactions on Signal Processing, Vol 59 (7), 3424-3429.
10. Wikipedia, http://www.wikipedia.org/

Sistema de Control Difuso con Reglas Sintonizadas mediante Algoritmos Genéticos para un Proceso de Fermentación Alcohólica

Rafael Rendón Gómez, Matilde Santos Peñas

Universidad Complutense de Madrid
rafaelrendon01@yahoo.es, msantos@ucm.es

Abstract. En este trabajo se describe la combinación de dos técnicas de Soft Computing aplicadas a un problema real de control, el proceso de fermentación alcohólica de mosto de uva para la elaboración de vino. Se ha diseñado un sistema de control incremental basado en lógica fuzzy y las matrices de reglas de dicho controlador difuso se han sintonizado a través de algoritmos genéticos. Este sistema inteligente de control se ha diseñado para mantener la temperatura adecuada de la masa de mosto en fermentación en función de la etapa del proceso, siendo valorado por la concentración del mismo. Los resultados de simulación del controlador inteligente regulando este proceso altamente no lineal demuestran la estabilidad del sistema y la idoneidad del modelo obtenido.

Keywords: Soft Computing, control inteligente, lógica difusa, algoritmos genéticos, fermentación alcohólica, mosto, vino.

1 Introducción

Los sistemas clásicos de control han generado una gran variedad de soluciones prácticas en diversos sectores como la industria, navegación, aeronáutica o el doméstico [1]. Pero a pesar del éxito de estos sistemas de control basados en modelos matemáticos al abordar sistemas lineales, incluso con cierta complejidad, cuando es necesario resolver sistemas no lineales o más complejos, difíciles de modelizar mediante técnicas analíticas o con variables no parametrizables, con incertidumbre, etc., el diseño de los mismos se dificulta significativamente o su aplicación se hace en la práctica imposible. En algunos procesos aparecen características como puntos de equilibrios múltiples, ciclos límite, bifurcaciones, corrimiento de frecuencias y caos, que restringen la aplicación de las técnicas de control lineal a un rango de operación de pequeña señal o alrededor de un punto de equilibrio específico, pero no de forma general.

Una de las alternativas que se han probado más eficaces para controlar este tipo de sistemas complejos haciendo uso del conocimiento y razonamiento humanos, son los controladores inteligentes [2]. En concreto, los sistemas inteligentes que utilizan algunas de las técnicas del Soft Computing, por ejemplo, la lógica difusa o borrosa (Fuzzy Controllers). Su funcionamiento no depende del punto de operación, sino de sencillas reglas lingüísticas que se pueden construir a partir de la experiencia de al-

© Springer-Verlag Berlin Heidelberg 2011

gún experto. Además, el diseño no requiere de un modelo matemático que tan sólo será necesario si se desean realizar simulaciones para validar el controlador inteligente antes de aplicarlo al sistema real.

En este trabajo se describe el diseño e implementación de un controlador difuso incremental para un proceso de fermentación alcohólica. Dicho proceso, ubicado dentro de la industria química y alimentaria, ha sido controlado tradicionalmente mediante sistemas de control clásicos, cuyo objetivo ha sido el mantenimiento de la temperatura de fermentación del mosto en un valor fijo durante todo el proceso, para evitar una parada de las reacciones enzimáticas que generan la fermentación y garantizar así la conclusión completa del proceso en condiciones óptimas de calidad del producto [3].

El control de dichos sistemas mediante controladores clásicos genera principalmente tres grandes inconvenientes:

1. Modelización compleja del sistema, sin tener en cuenta variables no controladas que tienen un impacto importante en el proceso (temperatura exterior, aislamiento, cinética de la reacción, concentración de azúcar, …)
2. Elevada inercia térmica de la masa de mosto en fermentación que dificulta la parametrización del sistema PID de control.
3. No se tiene en cuenta la concentración de azúcar durante el proceso completo de fermentación pudiendo originar paradas al inicio o al final de la fermentación, dejando restos de azúcares residuales que necesitan ser reprocesados o incluso imposibilitando su uso posterior en la bodega.

El uso de un controlador difuso para este proceso simplifica el diseño y optimización del regulador y es capaz de considerar el impacto de las variables no controladas, así como tener en cuenta el nivel de concentración de azúcar de una manera sencilla, sin complicados cálculos ni ecuaciones complejas, y sin hacerlo depender del punto de operación.

Para encontrar la matriz de reglas del controlador difuso se han realizado simulaciones aplicando algoritmos genéticos para obtener los valores óptimos.

El artículo se estructura como sigue. En la sección 2 se describe el proceso de fermentación, haciendo hincapié en las variables que se quieren controlar. En la sección 3 se diseña el controlador difuso, especificando las funciones de pertenencia y las reglas. La sección 4 presenta la aplicación de los algoritmos genéticos para optimizar las reglas del sistema fuzzy. Se muestran por último los resultados del sistema de control aplicado al proceso. El artículo termina con las conclusiones.

2 Descripción y Modelo del Proceso

Es importante entender el funcionamiento del proceso de fermentación alcohólica para adquirir conocimiento sobre su comportamiento que luego será representado en el sistema difuso. El modelo desarrollado se utilizará para la simulación y validación del controlador propuesto.

La fermentación alcohólica es un proceso bioquímico que transforma el azúcar en alcohol, utilizado desde hace seis o siete mil años para producir cerveza y vino. Prácticamente, la fermentación alcohólica se puede presentar como una transformación

química por medio de enzimas en el interior de microorganismos. Por lo tanto, realmente es un proceso complejo que engloba las disciplinas científicas de química, enzimología y microbiología [3].

La fermentación es el proceso básico para la elaboración de vinos, por el cual se transforma el azúcar contenido en el mosto de uva en alcohol vínico. Es un proceso anaeróbico y catabólico llevado a cabo por levaduras, gracias a la acción catalizadora de las enzimas presentes en el mosto [4].

En este estudio presentaremos el caso concreto del control de la fermentación alcohólica del Sherry o Vino de Jerez. Al igual que en el resto de procesos fermentativos, se genera una reacción exotérmica con un ratio medio de emisión de calor de 14,1 kCal por cada 100 g de azúcar reductor fermentado. Esto contribuye a elevar la temperatura del mosto en fermentación que puede llegar a parar la reacción enzimática en caso de alcanzar valores altos, dejando azúcares residuales en el mosto que obligan a costosos reprocesos, o aún más grave, a imposibilitar su uso esperado en la bodega. Por ello, es necesario el control de la temperatura durante el proceso de fermentación mediante sistemas adecuados de refrigeración.

Es importante vigilar la temperatura del proceso de fermentación no sólo en valores altos sino también bajos, ya que valores excesivamente bajos, principalmente al inicio y final de la fermentación, pueden provocar una inhibición del proceso evitando el inicio del mismo o la parada antes de completar el consumo de azúcar [5, 6].

2.1 Control de la temperatura de fermentación

Realmente las necesidades de frío son diferentes en cada fase del proceso y las condiciones de control de temperatura tendrían que tenerlo en cuenta. La emisión de calor es distinta durante las tres fases de la fermentación, ya que la tasa de transformación alcohólica varía en el proceso [7].

Es decir, la temperatura óptima de control depende de la fase fermentativa. Se requiere un valor intermedio de 24ºC en la fase de inducción para permitir el inicio de la fermentación, pero con un valor no demasiado alto que permita bajarlo rápidamente en la siguiente etapa. En la fase tumultuosa el valor idóneo pasa a 22ºC donde la reacción exotérmica es más vigorosa, no debiendo bajar de valores de 20ºC que podrían inhibir el metabolismo de las enzimas que hacen posible la fermentación. Finalmente, durante la fase final o lenta, se requiere un valor mayor de 26ºC que asegure una completa fermentación del mosto para evitar la generación de restos residuales de azúcar (Tabla 1).

Fase	Temperatura
Fase de inducción	24ºC
Fase tumultuosa	22ºC
Fase lenta	26ºC

Tabla 1. Temperatura idónea de fermentación.

Tradicionalmente los procesos de fermentación han sido controlados por controladores PID clásicos programados con un setpoint fijo de temperatura. Estos sistemas tan sólo requieren de un termopar o sensor de temperatura provistos con salida analógica instalado en el depósito de fermentación y conectado eléctricamente al contro-

lador PID. Los inconvenientes de dichos sistemas son que sólo permiten un valor de control para todo el proceso y, además, no tiene en cuenta la fase de fermentación [8].

Para mejorar este sistema es necesario que el controlador conozca en todo momento la fase del proceso de fermentación, lo que podría ser proporcionado a través de la medida directa del grado alcohólico o la concentración de azúcar. Sin embargo, no existen en el mercado muchos instrumentos que puedan suministrar directamente de forma continua algunas de las medidas anteriores. Los más adecuados son los que determinan a tiempo real la densidad de un líquido o incluso la concentración de azúcar. Estos equipos están provistos de fábrica con sensores internos de temperatura para realizar compensación de temperatura [9].

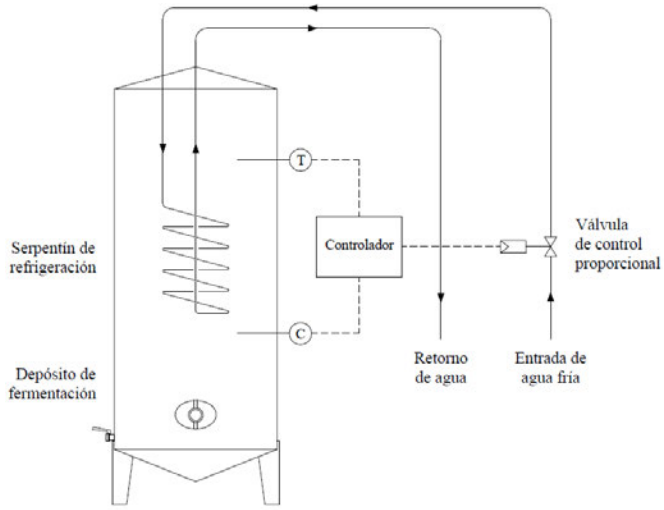


Fig. 1. Sistema de control dual de la temperatura de fermentación.

En la figura 1 se puede observar un sistema dual de control de temperatura de fermentación, donde el controlador del proceso tiene en cuenta tanto la temperatura T como la concentración C expresada en gramos/litro.

La modelización del sistema de control teniendo en cuenta todas sus variables y haciendo uso de un sistema de control clásico es muy compleja y en muchos casos es incapaz de adaptarse a las variables o parámetros fuera del algoritmo de control, además de la dificultad de operar a tiempo real.

Actualmente, no se encuentra disponible en la literatura científica ningún modelo matemático que describa analíticamente el proceso, aunque sin embargo existen numerosos estudios experimentales que detallan su comportamiento. Por ello, este proceso de fermentación es potencialmente abordable desde un sistema de control difuso.

3 Controlador Difuso de la Temperatura

La lógica difusa es una disciplina que usa razonamientos con incertidumbre, expresiones heurísticas, con vaguedad en las definiciones y lógica multivariable, similar a

los térmicos lingüísticos utilizados por el ser humano. Esta lógica es útil en procesos complejos, con dificultad de estimar parámetros, con dificultad para obtener medidas precisas de los sensores de medida, con falta de fiabilidad en los sensores o en ambientes ruidosos [10]. Esta técnica se ha aplicado con éxito al control de temperatura de otros procesos complejos [11, 12].

El objetivo del controlador difuso para este proceso es mantener la temperatura óptima del mosto en fermentación en cada fase del mismo, evitando la sobreoscilación del sistema y la variación de la temperatura en cada fase, haciendo además que la temperatura de equilibrio del mosto sea igual a la salida del líquido de refrigeración.

Puesto que el parámetro que define las diferentes fases del proceso es la concentración de azúcar, la función que determina la temperatura óptima de fermentación puede ser representada por el gráfico mostrado en la figura 2.

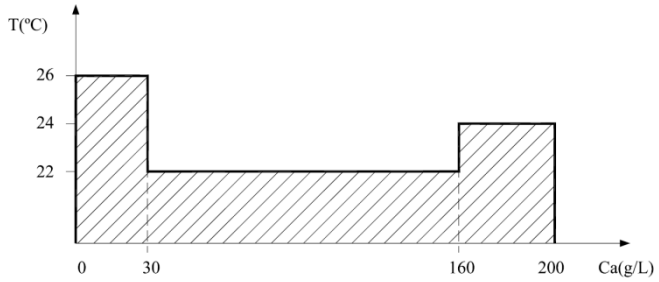


Fig. 2. Temperatura óptima de fermentación en función de la concentración de azúcar C_a*(g/L).*

Esta función puede expresarse mediante reglas determinando la temperatura adecuada de fermentación Ta en función de la fase del proceso midiendo la concentración de azúcar Ca:

```
Si Ca< 30 entonces Ta=26 (Lenta)
Si Ca ≥ 30 y Ca<160 entonces Ta =22 (Tumultuosa)
Si Ca ≥ 160 entonces Ta =24 (Inducción)
```

La particularidad de este proceso es que tiene tres condiciones ideales de funcionamiento en función de la fase del proceso, pero puesto que el error e es común para cada una de ellas, es suficiente definir un solo perfil de funciones de pertenencia para el control de todo el proceso, considerando en todo momento:

$$e = T_a - T \tag{1}$$

Siendo Ta la temperatura adecuada en función de la fase y T la temperatura real del mosto en fermentación.

El controlador fuzzy diseñado es tipo PD incremental, o lo que es equivalente, un PI. Por lo tanto sus variables lingüísticas son la señal de error e y su derivada de/dt. A partir de los valores de estas dos señales, el controlador difuso decide el valor apropiado del incremento del caudal (Δqr) como variable de control del sistema.

Como el error e en cada fase puede estar comprendido entre -10ºC y 2ºC, las funciones de pertenencia deben cubrir todo este rango. Dichas funciones son de tipo trapezoidal y triangular, siendo definidas como Negativo Grande (NG), Negativo Medio (GM), Negativo (N), Muy Pequeño (MP) y Positivo (P) (Figura 3). Por otro

lado, las funciones de pertenencia para la derivada del error Δe también son del tipo trapezoidal y triangular, y están comprendidas entre -0,2 y 0,2 siendo definidas como Negativo (N), Muy Pequeño (MP) y Positivo (P) (Figura 4).

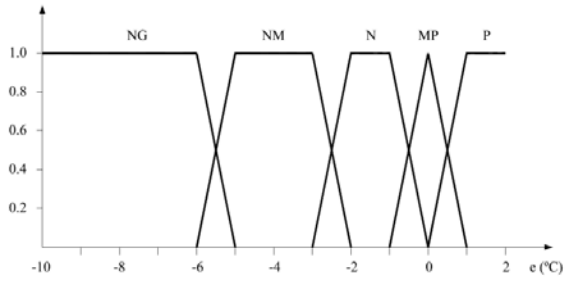


Fig. 3. Funciones de pertenencia para el error *e* en ºC.

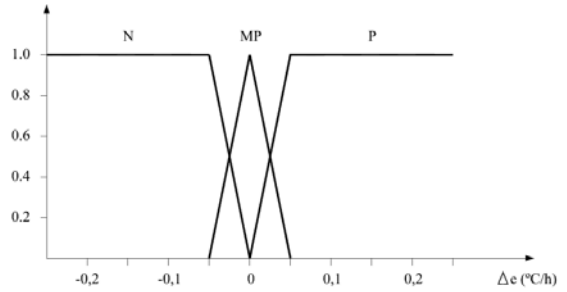


Fig. 4. Funciones de pertenencia para el cambio en el error *Δe* en ºC/h.

El controlador difuso genera un valor de salida Δqr que corresponde físicamente a una señal eléctrica de 0-10V ó 4-20mA conectada a la válvula proporcional que controla el caudal de refrigeración qr.

La matriz de reglas del proceso a controlar se resume en la tabla 2, donde se especifica la acción de control para las 3 funciones de pertenencia de Δe y las 5 del error e.

		e				
		NG	NM	N	MP	P
	N	Δq1	Δq2	Δq3	Δq4	Δq5
Δe	MP	Δq6	Δq7	Δq8	Δq9	Δq10
	P	Δq11	Δq12	Δq13	Δq14	Δq15

Tabla 2. Matriz de reglas del controlador difuso.

En ausencia de valores facilitados por un experto humano que haya controlado previamente el proceso, en el siguiente apartado se sintonizarán los mismos a través de algoritmos genéticos.

El motor de inferencias del controlador se encargará de asociar el estado del proceso a través de las funciones de pertenencia y de realizar un barrido de la matriz de reglas para asignar una salida final del sistema difuso dada por Δq_i.

En este caso aplicaremos como método de defuzzificación el del centro de gravedad. El valor de salida qr, al tratarse de un actuador eléctrico, es convertido a una señal analógica de 0-10V ó 4-20mA, que dirige la apertura-cierre proporcional de la válvula de refrigeración para proporcionar ese caudal.

4 Aplicación de Algoritmos Genéticos para obtener las Reglas del Sistema Difuso

Los algoritmos genéticos forman parte de los algoritmos evolutivos como una disciplina de la inteligencia artificial y Soft Computing. Esta técnica de computación evolutiva se basa en mecanismos de la genética y selección natural. En concreto, los algoritmos genéticos representan una aproximación para la solución de problemas de optimización donde la búsqueda en el espacio de soluciones se realiza de manera evolutiva, reteniendo las soluciones más adecuadas en un proceso análogo a la evolución biológica, mediante la reproducción, selección natural, cruce y mutación [13].

Los algoritmos genéticos trabajan con una población inicial de individuos o cromosomas generados de forma aleatoria, cada uno de los cuales podría representar una posible solución al problema. A través de la función de ajuste o fitness se evalúa la idoneidad de cada uno de dichos individuos para resolver el problema y se clasifican los mejores, que pueden preservados para la siguiente generación (elitismo), denominados padres. Posteriormente se cruzan para generar nuevos individuos, denominados hijos, que también pueden modificarse por mutación alterando aleatoriamente parte de su material genético. Finalmente, los individuos o cromosomas generados se evalúan de nuevo mediante la función de ajuste, repitiendo el proceso n veces.

Los algoritmos genéticos se diferencian de los algoritmos clásicos en que trabajan con un conjunto de soluciones por cada iteración ejecutada, seleccionando el conjunto de la siguiente iteración a través de cómputos heurísticos, en lugar de funciones deterministas. Esta técnica es apropiada para buscar datos u optimizar otros sistemas de control, permitiendo alcanzar óptimos resultados en cortos periodos de computación.

En este trabajo se han usado los algoritmos genéticos para sintonizar la matriz de reglas del controlador difuso (Tabla 2), hallando los valores de mejor desempeño mediante la simulación del proceso y fijando unos datos iniciales para poder ser usado por el motor de inferencia de dicho controlador.

Para ello se ha usado el siguiente modelo matemático linealizado del proceso, donde se expresa la temperatura interna T del mosto en fermentación en función del caudal de refrigeración q_r, $T = T(q_r)$

$$\tau \frac{dT}{dt} + T = k_1 q_r + k_2 Q_f(\mathrm{C}a) \tag{2}$$

$$G_1(s) = \frac{k_1}{\tau s+1} \quad ; \quad G_2(s) = \frac{k_2}{\tau s+1} \tag{3}$$

Los valores de las funciones de transferencia vienen dados por las siguientes variables del proceso:

$$\tau = \frac{m_m c_m}{K\ \rho_r\ q_0\ c_r} \quad ; \quad k_1 = \frac{(T_r - T_0)}{q_0} \quad ; \quad k_2 = \frac{1}{K\ \rho_r\ q_0\ c_r} \tag{4}$$

m_m	Masa de mosto	K	Coeficiente de intercambio
c_m	Calor específico del mosto	ρ_r	Densidad del líquido refrigerante
T	Temperatura del mosto	q_r	Caudal del líquido refrigerante
T_r	Temperatura del refrigerante	c_r	Calor específico del refrigerante

Y los valores T_0 y q_0 corresponden al punto de equilibrio del proceso. Se establece inicialmente como temperatura de equilibrio la temperatura de consigna durante la fase tumultuosa, es decir, $T_0=T_a=22$ºC. Finalmente,

$$T' = q_r' \, G_1(s) + Q_f' \, G_2(s) \tag{5}$$

Donde T', q_r' y Q_f' representan las variaciones de las variables temperatura interna del mosto T, del caudal de refrigeración qr, y del calor producido por la fermentación Qf, respectivamente, en el dominio de Laplace.

La figura 5 muestra el diagrama de bloques del proceso. En la ecuación que representa el modelo del sistema físico del proceso de fermentación provisto de refrigeración interna, T es la variable de estado, qr la variable de control y Qf se considera una perturbación del proceso.

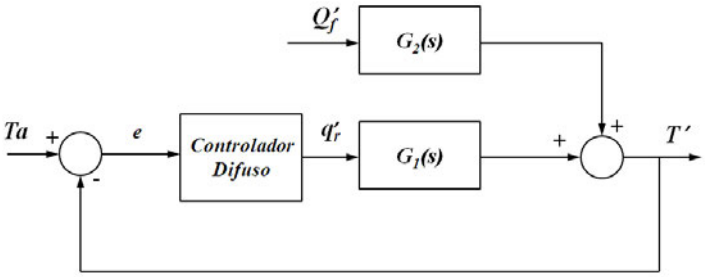


Fig. 5. Diagrama de bloques del proceso.

Será necesario establecer unos valores iniciales correspondientes a las características físicas del proceso para poder simularlo numéricamente. Para ello, se consideran los siguientes valores de una planta real de fermentación [14]:

V = 30.000 L	ρ_r = 1.000 kg/m^3
ρ_m = 1.100 kg/m^3	c_r = 1,0 kCal/kgºC
m_m = 33.000 kg	q_0 = 8 m^3/h
c_m = 1,01 kCal/kgºC	T_0 = 22ºC
K = 0,75	T_r = 20ºC

Con estos datos se obtienen los siguientes valores del modelo para las condiciones descritas anteriormente y obteniendo las ecuaciones del sistema:

$$\tau = 5{,}56 \;\; ; \;\; k_1 = -0{,}25 \;\; ; \;\; k_2 = 0{,}17 \tag{6}$$

$$5{,}56 \;\; \frac{dT}{dt} + T = -0{,}25 q_r + 0{,}17 Q_f \tag{7}$$

$$G_1(s) = \frac{-0{,}25}{5{,}56s+1} \;\; ; \;\; G_2(s) = \frac{0{,}17}{5{,}56s+1} \tag{8}$$

En la aplicación del algoritmo genético, cada individuo de la población tiene un tamaño de 15 cromosomas, es decir, los 15 valores de la matriz de reglas, que han sido inicializados aleatoriamente entre -0,01 y 0,01. Para cada individuo se simula la dinámica del sistema en lazo cerrado, según la figura 5, aplicando una función escalón a la entrada del lazo directo G1(s) y una función en rampa en el lazo de la perturbación G2(s), durante 25s, con un tiempo de muestreo de 25ms. Por cada individuo se obtiene el tiempo de asentamiento ts, el sobreimpulso Mp y el error estacionario, ess, considerando la función de ajuste del sistema a minimizar la dada por fa (10):

$$f_a = 0{,}5t_s + 0{,}3M_p + 0{,}2ess \tag{9}$$

La simulación se realiza con una población de 25 individuos durante 200 generaciones. De acuerdo con el resultado de la función de ajuste, se seleccionan los 4 individuos con mejor desempeño (padres), 12 de los individuos de la población son cruzados (hijos) los cuales son el resultado de promediar componente a componente dos hijos diferentes. Los 12 hijos mutados son el resultado de sumar un número aleatorio entre -0,001 y 0,001 a cada elemento de un hijo escogido aleatoriamente, de tal forma que un individuo no se modifique dos veces. Una vez aplicado el algoritmo genético, el mejor individuo se indica en la tabla 3, que da el mejor valor para la matriz de reglas del controlador incremental difuso del proceso de fermentación alcohólica. Par estos valores, el proceso presenta la siguiente respuesta en el tiempo (Figura 6): $t_s = 17s \ ; \ M_p = 0 \ ; \ ess = 0{,}19\%$

		e				
		NG	NM	N	MP	P
	N	0,0083	0,0062	0,0021	0,0010	-0,0012
Δe	MP	0,0043	0,0028	0,0009	-0,0002	-0,0004
	P	0,0012	0,0003	-0,0001	-0,0003	-0,0008

Tabla 3. Matriz de reglas sintonizadas para Δqr.

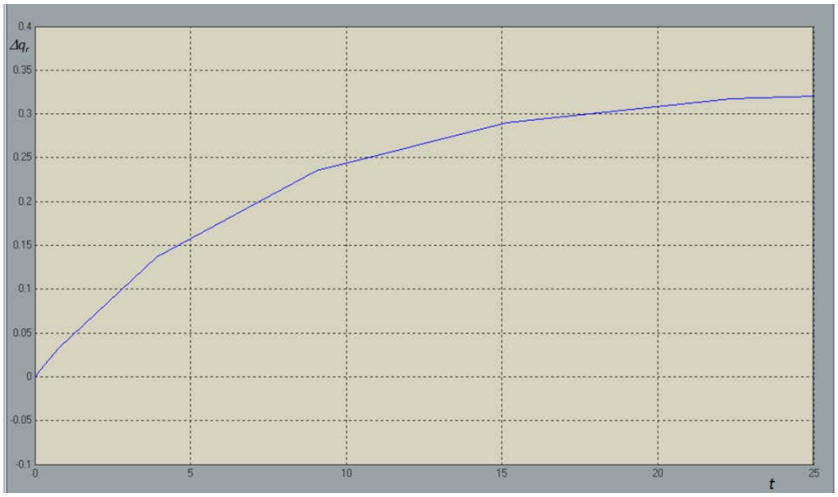


Fig. 6. Respuesta dinámica del sistema para los valores sintonizados de la matriz de reglas.

5 Conclusiones

En este trabajo se han aplicado dos técnicas de Soft Computing a un proceso real, la fermentación alcohólica del vino.

Se ha diseñado un controlador difuso que regula la temperatura del proceso según las distintas fases del mismo, obteniendo el caudal necesario para mantener el valor deseado. Los valores de los consecuentes de la reglas de este controlador han sido obtenidos mediante la aplicación de algoritmos genéticos, optimizando la respuesta del proceso según las especificaciones de control.

Los resultados de simulación, basados en valores reales del proceso, son muy satisfactorios.

Agradecimientos. Al departamento técnico de las bodegas Beam Spain en Jerez (antiguamente Bodegas Pedro Domecq) por facilitar los datos experimentales del proceso de fermentación y permitir la consulta de material bibliográfico [14].

Referencias

1. Dorfand, R.C., Bishop, R.H.: Modern Control Systems, Prentice-Hall (2010)
2. Santos, M.: Un enfoque aplicado del control inteligente, RIAI, 8, 283-296 **(**2011)
3. Blouin, J., Peynaud, E.: Enología práctica: conocimiento y elaboración del vino. Mundi-Prensa (2003)
4. Colombié S., Malherbe S.; Sablayrolles J.: Modeling alcoholic fermentation in enological conditions: feasibility and interest, en: Am. J. Enology & Viticulture 56, 3, 238-245 (2005)
5. Caro Pina, I.: Desarrollo de un modelo cinético aplicado a la fermentación alcohólica y estudio del efecto global de la temperatura, PhD, Universidad de Cádiz, España (1989)
6. Santamaría, P., López, R., Gutiérrez, A., García-Escudero, E.: Influencia de la temperatura en la fermentación alcohólica, Zubía 7, 137-149 (1995)
7. Palacios, C.A., Udaquiola, S.M., Rodríguez, R.A.: Modelo matemático para la predicción de las necesidades de frío durante la producción de vino, Ciencia, docencia y tecnología, 38 (2009)
8. Yair, M.: Winery Technology & Operations: A Handbook for Small Wineries, San Francisco (1996)
9. Ingenieros de Control Asociados S.L.: Transmisor inteligente de densidad y concentración DT301, Internal report, Madrid. España (2012)
10. Ross, T.J.: Fuzzy Logic with Engineering Applications, Wiley (2010)
11. Santos, M., Dexter, A.L.: Control of a cryogenic process using a Fuzzy PID Scheduler, Control Engineering Practice 10, 1147-1152 (2002)
12. Santos, M., Dexter, A.L.: Temperature control in a liquid helium cryostat using a self-learning neurofuzzy controller, IET Proc.-Control Theory Appl. 148, 3, 233-238, (2001)
13. Davis L.: Handbook of Genetic Algorithms, Van Nostrand Reinhold (1991)
14. Datos históricos de la planta de fermentación de Beam Jerez (Bodegas Pedro Domecq)

Estudio preliminar sobre medidas de similitud para la detección de puntos singulares en imágenes de huellas dactilares

Juan Cerrón, Mikel Galar, Carlos Lopez-Molina, José Sanz, Miguel Pagola

Departamento de Automática y Computación,
Universidad Pública de Navarra, 31006, Pamplona, España
{juan.cerron,mikel.galar,carlos.lopez,joseantonio.sanz,miguel.pagola}@unavarra.es

Resumen La detección de puntos singulares es un proceso importante en los sistemas de reconocimiento de huellas. Permiten acelerar el proceso de clasificación además de mejorar la precisión de los algoritmos de correspondencia (*matching*). En este trabajo presentamos un nuevo método para detectar puntos singulares basado en medidas de similitud. Este método se basa en las orientaciones de un mapa de orientaciones frente a una serie de filtros predefinidos que representan las orientaciones ideales de dichos puntos. Para comprobar la validez del método llevamos a cabo un conjunto de pruebas preliminares que muestran la correcta detección de los puntos en una amplia variedad de imágenes.

Palabras clave: Reconocimiento de huellas, puntos singulares, medidas de similaridad, funciones de restricción de equivalencias

1. Introducción

El reconocimiento de huellas dactilares es el sistema de autenticación biométrica más extendido. Las huellas dactilares son las imágenes visibles que producen las crestas y valles papilares de un dedo de la mano. Las características locales de las crestas, así como la relación entre ellas, es lo que determina la individualidad de cada huella [1]. Las características denominadas minucias son las más utilizadas para comparar huellas [2,3,4], es decir, para decidir si dos huellas corresponden al mismo individuo o no. El mayor problema de este proceso de comparación es su coste computacional cuando se trabaja con bases de datos (BBDD) grandes. En la actualidad, el tamaño de las bases de datos incrementa exponencialmente igual que el número de individuos de la sociedad, por lo que una forma de reducir este tiempo de comparación es creando BBDD de menor tamaño con huellas que compartan ciertas características [5]. Se pueden identificar hasta cinco grupos de huellas con las mismas características (clases) denominados: *left-loop, right-loop, whorl, arch y tented-arch* [5]. Poder identificar la clase o el grupo al que pertenece una huella de forma precisa y eficiente es muy importante, ya que una buena identificación afecta directamente a los

procesos posteriores de matching. Una forma simple de clasificar es mediante el número y la posición de los puntos singulares (PSs) de una huella. Los puntos singulares son las zonas de la huella en las que las orientaciones de las crestas varían más rápidamente. Además, los PSs pueden utilizarse también en algoritmos de correspondencia (*matching*) para registrar las huellas, o para mejorar la precisión de los mismos con métodos híbridos [6]. Los dos puntos singulares principales que podemos encontrar se denominan *cores* y *deltas*: siendo el core la zona de la huella en la que las crestas producen un mayor giro; y el delta la zona de la huella donde varias crestas forman la apariencia de un triángulo.

Los algoritmos de detección de puntos singulares utilizan los mapas de orientaciones obtenidos a partir de las huellas originales para encontrar los puntos. Este hecho hace que la eficiencia de estos algoritmos dependa en gran medida de la precisión de dichos mapas. Uno de los métodos más utilizados para calcular este mapa es el método del gradiente [7]. Sin embargo, debido al ruido, manchas, cortes, etc., comunmente presentes en las huellas dactilares, la estimación del mapa de orientaciones puede no ser fiable. Como consecuencia, la estimación de los PSs se ve directamente comprometida.

El método más común para detectar puntos singulares es el método de Poincarè [5], que evalua la curva formada por los vecinos de cada punto del mapa de orientaciones, para etiquetar dicho punto como *core* o *delta* en caso de ser un PS. Sin embargo, también hay otros métodos basados en filtros complejos [8] o en el modelo de *zero-pole* [9].

En este trabajo presentamos una nueva forma de detectar puntos singulares basada en medidas de similitud. De este modo tratamos de reducir la dependencia de Poincarè sobre el mapa de orientaciones, además de ser un método más intuitivo y simple que los otros métodos basados en filtros complejos o el modelo *zero-pole*. El método consiste en establecer una máscara de orientaciones para cada tipo de PS, y posteriormente buscar las zonas del mapa de orientaciones con mayor parecido a dichas máscaras. El nivel de parecido se establece haciendo uso de las medidas de similitud. Dependiendo del nivel del mismo cada zona se etiqueta como PS o no. Una de las ventajas de este método es que resulta más flexible que el resto debido a la gran variedad de medidas de similitud diferentes que existen.

Con el fin de comprobar la validez del proceso de detección de puntos singulares, hemos llevado a cabo una serie de experimentos preliminares sobre las bases de datos del NIST (National Institute of Standards and Technology) y una base de datos creada con la herramienta SFinGe [1] [10,5], que nos permite construir huellas sintéticas similares a las reales. La precisión de los resultados obtenidos en el experimento y la fácil comprensión del método nos hacen creer en un futuro prometedor.

El resto del trabajo está organizado de la siguiente manera: En la Sección 2 explicamos algunos conceptos necesarios para desarrollar el trabajo. Nuestra propuesta para detectar puntos singulares se presenta en la Sección 3. La Sección 4

[1] Synthetic Fingerprint Generator:`http://biolab.csr.unibo.it/research.asp?organize=Activities&select=&selObj=12&pathSubj=111||12&`

muestra los resultados obtenidos con el nuevo método. Finalmente, la Sección 5 indica las conclusiones del trabajo.

2. Preliminares

Esta sección recuerda algunos conceptos necesarios para desarrollar el resto del trabajo. Presentamos las nociones básicas de la teoría difusa que permiten la introducción de las medidas de similitud, y algunos conceptos sobre reconocimiento de huellas dactilares necesarios para entender el método propuesto.

2.1. Conjuntos difusos

Zadeh [11] introdujo la teoría difusa en 1965 con el fin de representar los conceptos abstractos e imprecisos que utiliza el ser humano. Comenzó con la definición de conjunto difuso de la siguiente manera: sea X un conjunto no vacío y finito con N elementos llamado conjunto referencial, un conjunto difuso A sobre X se representa como

$$A = \{(x, \mu_A(x)) \mid x \in X\}$$

donde $\mu_A : X \to [0,1]$ es la función de pertenencia. El conjunto de todos los conjuntos difusos definidos sobre X se denota como $\mathcal{FS}(X)$.

La negación es una de las operaciones básicas de la teoría de conjuntos difusos que permite modelar el concepto de opuesto.

Definición 1 *Una función $n : [0,1] \to [0,1]$ con $n(0) = 1$, $n(1) = 0$, estrictamente decreciente y continua se denomina negación estricta. Además, si n es involutiva, es decir, si $n(n(x)) = x$ para todo $x \in [0,1]$, entonces n es una negación fuerte.*

Una negación fuerte n permite definir el conjuto complementario A_n de un conjunto difuso A: $A_n = \{(x, \mu_{A_n}(x))|\ \mu_{A_n}(x) = n(\mu_A(x))\ \forall x \in X\}$.

En este trabajo usamos el concepto de función de equivalencia restringida (*Resctricted Equivalence Function*, *REF*) [12] para construir medidas de similitud con el fin de comparar los ángulos de los filtros predefinidos (máscaras) y los del mapa de orientaciones. Estas funciones fueron introducidas en [12,13] para medir el grado de equivalencia (cercanía) entre dos puntos.

Definición 2 *[12,13] Una función $REF : [0,1]^2 \to [0,1]$ se llama función de equivalencia restringida asociada a una negación fuerte n, si satisface las siguientes condiciones*

1. *$REF(x,y) = REF(y,x)$ para todo $x, y \in [0,1]$;*
2. *$REF(x,y) = 1$ si y solo si $x = y$;*
3. *$REF(x,y) = 0$ si y solo si $x = 1$ e $y = 0$ o $x = 0$ e $y = 1$;*
4. *$REF(x,y) = REF(n(x), n(y))$ para todo $x, y \in [0,1]$;*

5. *Para todo* $x, y, z \in [0,1]$, *si* $x \leq y \leq z$, *entonces* $REF(x,y) \geq REF(x,z)$ *y* $REF(y,z) \geq REF(x,z)$.

Una forma de construir REFs es utilizando automorfismos del intervalo unidad [12]. Empezamos recordando el concepto de automorfismo.

Definición 3 *Una función continua y estrictamente creciente* $\varphi : [a,b] \to [a,b]$ *tal que* $\varphi(a) = a$ *y* $\varphi(b) = b$ *se denomina automorfismo del intervalo* $[a,b] \subset \mathbb{R}$.

Proposición 1 *[12] Si* φ_1, φ_2 *son dos automorfismos del intervalo* $[0,1]$. *Entonces* $REF(x,y) = \varphi_1^{-1}(1 - |\varphi_2(x) - \varphi_2(y)|)$ *es una función de equivalencia restringida asociada a una negación fuerte* $n(x) = \varphi_2^{-1}(1 - \varphi_2(x))$.

Ejemplo 1 *Si* $\varphi_1(x) = x$ *y* $\varphi_2(x) = \sqrt{x}$, *entonces* $REF(x,y) = 1 - |\sqrt{x} - \sqrt{y}|$ *es una función de equivalencia restringida asociada a* $n(x) = (1 - \sqrt{x})^2$.

A continuación introducimos una forma de comparar dos conjuntos difusos (que en nuestro caso representarán los ángulos, y más concretamente, sus componentes horizontal y vertical), las llamadas medidas de similitud.

Definición 4 *[12] Una función* $SM : \mathcal{FS}(X) \times \mathcal{FS}(X) \to [0,1]$ *se denomina medida de similitud asociada a una negación fuerte* n *si satisface las siguientes propiedades:*

1. $SM(A,B) = SM(B,A)$, *para todo,* $A, B \in FS(X)$;
2. $SM(A,A_n) = 0$ *si y solo si* A *es un conjunto crisp.*
3. $SM(A,B) = 1$ *si y solo si* $A = B$;
4. *If* $A \leq B \leq C$, *entonces* $SM(A,B) \geq SM(A,C)$ *y* $SM(C,B) \geq SM(C,A)$;
5. $SM(A_n,B_n) = SM(A,B)$.

Debemos recordar que una función de agregación proporciona una regla para combinar diferentes valores de entrada en un único valor de salida. Nos resultan útiles porque nos permiten agregar REFs para obtener medidas de similitud.

Definición 5 *Una función de agregación es una función:* $M : [0,1]^N \to [0,1]$ *tal que:* $M(0,0,\ldots,0) = 0$, $M(1,1,\ldots,1) = 1$ *y* M *es no decreciente.*

La Proposición 2 hace uso de las REFs y las funciones de agregación para construir medidas de similitud.

Proposición 2 *[12] Dada una función de agregación* $M : [0,1]^N \to [0,1]$ *tal que:* $M(x_1,\ldots,x_N) = 0$ *si y solo si* $x_1 = \cdots = x_N = 0$; *y* $M(x_1,\ldots,x_N) = 1$ *si y solo si* $x_1 = \cdots = x_N = 1$; *y una función de equivalencia restringida* $REF : [0,1]^2 \to [0,1]$ *asociada a una negación fuerte* n, *se puede definir:*

$$SM : \mathcal{FS}(X) \times \mathcal{FS}(X) \to [0,1] \tag{1}$$

$$SM(A,B) = \underset{i=1}{\overset{N}{M}} REF(\mu_A(x_i), \mu_B(x_i)) \tag{2}$$

siendo SM *una medida de similitud asociada a la negación fuerte* n.

Ejemplo 2 *Siendo M la media aritmética y $REF(x,y) = \sqrt{1-|x-y|}$ una función de equivalencia restringida asociada a una negación fuerte $n(x) = 1-x$, entonces*

$$SM(A,B) = \frac{1}{N}\sum_{i=1}^{N}\sqrt{1-|\mu_A(x_i)-\mu_B(x_i)|} \tag{3}$$

es una medida de similitud asociada a $n(x) = 1-x$.

2.2. Extracción del mapa de orientaciones

Un mapa de orientaciones es la representación en ángulos, normalmente en el rango $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right]$, del flujo de las crestas de una huella dactilar. El método más común para obtener dicho mapa de orientaciones es el conocido como método del grandiente [7]. El gradiente de cada píxel lo estimamos con las máscaras de Sobel para calcular las componentes vertical (G_x) y horizontal (G_y). Debido a que el resultado de un único píxel puede no ser lo suficientemente fiable como para definir correctamente la orientación, se lleva a cabo una media de los gradientes en bloques de $W_1 \times W_1$, de forma que la estimación de la orientación se obtiene teniendo en cuenta los gradientes en dicho bloque.

$$G_{xx} = \sum_{i=1}^{W_1}\sum_{j=1}^{W_1} G_x^2(i,j),\ G_{yy} = \sum_{i=1}^{W_1}\sum_{j=1}^{W_1} G_y^2(i,j) \tag{4}$$

$$G_{xy} = \sum_{i=1}^{W_1}\sum_{j=1}^{W_1} G_x(i,j)G_y(i,j) \tag{5}$$

La dirección media del gradiente Φ, con $-\frac{1}{2}\pi < \Phi \leq \frac{1}{2}\pi$, viene dada por:

$$\Phi = \frac{1}{2}\angle(G_{xx} - G_{yy}, 2G_{xy}), \tag{6}$$

donde $\angle(x,y)$ se define como:

$$\angle(x,y) = \begin{cases} tan^{-1}(x/y) & \text{si } x \geq 0, \\ tan^{-1}(y/x) + \pi & \text{si } x < 0 \wedge y \geq 0, \\ tan^{-1}(y/x) - \pi & \text{en caso contrario} \end{cases} \tag{7}$$

y la orientación final θ, con $-\frac{1}{2}\pi < \theta \leq \frac{1}{2}\pi$, es perpendicular a Φ:

$$\theta = \begin{cases} \Phi + \frac{1}{2}\pi & \text{si } \Phi \leq 0 \\ \Phi - \frac{1}{2}\pi & \text{en caso contrario} \end{cases} \tag{8}$$

Después de obtener la primera estimación del mapa de orientaciones, se lleva a cabo una fase de suavizado para obtener un mapa de orientaciones más preciso. Este suavizado utiliza una ventana $W_2 \times W_2$ uniforme.

2.3. Definición de los filtros

El método que proponemos utiliza tres filtros diferentes de $W_3 \times W_3$ para identificar cores concavos, cores convexos y deltas. Utilizamos dos filtros para detectar cores porque, con el fin de obtener un algoritmo invariante ante rotación, las orientaciones del mapa de orientaciones se multiplican por dos (cada valor θ se transforma a $2 \cdot \theta$). En este nuevo mapa denominado *mapa de orientaciones cuadrado* (O_2), las viejas orientaciones comprendidas en el rango $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right]$ pasan a ser direcciones en el rango $(-\pi, \pi]$, lo que produce que existan cores con direcciones opuestas. Un ejemplo del mapa de orientaciones cuadrado y de los filtros puede verse en la Figura 1. Los tres filtros mencionados, centrados en el eje de coordenadas se modelan de la siguiente forma:

$$F_1(x, y) = tan^{-1}(x, -y) \tag{9}$$

$$F_2(x, y) = tan^{-1}(-x, y) \tag{10}$$

$$F_3(x, y) = tan^{-1}(-y, -x) \tag{11}$$

donde F_1 y F_2 representan los dos posibles filtros para detectar cores, y F_3 representa el filtro para detectar deltas.

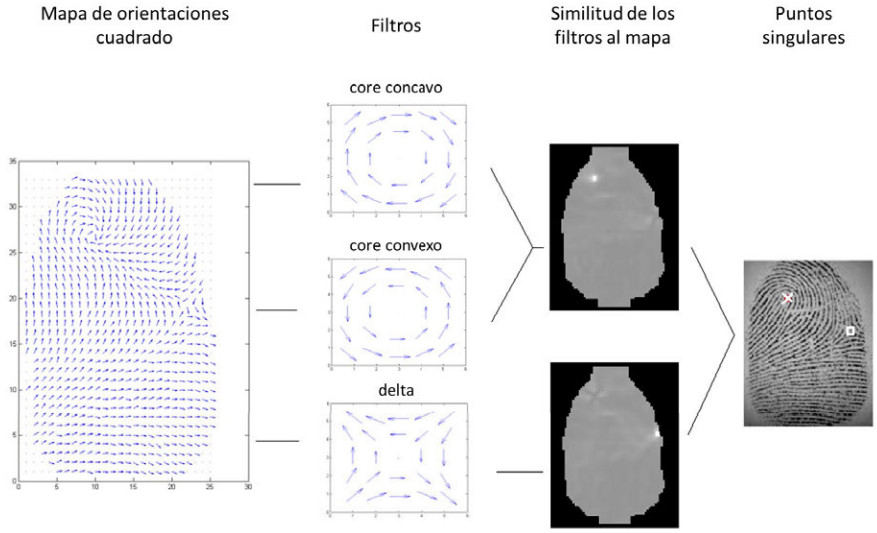


Figura 1. Esquema del método propuesto. En la imagen final la cruz representa la detección de un core y el cuadrado la detección de un delta.

3. Propuesta de detección de puntos singulares

Intuitivamente podemos apreciar el parecido entre las zonas del mapa de orientaciones cuadrado que corresponden a los puntos singulares y los filtros (ver Figura 1). En estas condiciones, la medida de similitud se convierte en una buena candidata para detectar dichos puntos. Como se indica en la Proposición 2, una medida de similitud compara dos conjuntos difusos mediante REFs y funciones de agregación. En nuestro caso, los conjuntos difusos utilizados son la verticalidad (V) y horizontalidad (H) de un bloque o filtro. Dichos conjuntos se utilizan cuando queremos comparar cada bloque $W_3 \times W_3$ de O_2 con cada filtro F_k, $k = 1, 2, 3$. Para ello obtenemos el grado de pertenencia de cada uno a los conjuntos difusos y finalmente comparamos con las medidas de similitud las pertenencias obtenidas para un mismo conjunto difuso. El grado de pertenencia de B (que puede ser tanto un bloque $W_3 \times W_3$ como un filtro F_k), a un conjunto difuso, V o H , para cada posición i, j, se calcula de la siguiente forma:

$$\mu_{H_B}(i,j) = \frac{cos(B(i,j)) + 1}{2} \tag{12}$$

$$\mu_{V_B}(i,j) = \frac{sin(B(i,j)) + 1}{2} \tag{13}$$

A continuación introducimos nuestra propuesta de detección de puntos singulares basada en medidas de similitud:

1. Recorrer el mapa de orientaciones cuadrado (O_2) en bloques solapados de $W_3 \times W_3$.
2. Para cada bloque B en la posición i, j:
   2.1 Para cada filtro filtro F_k, $k = 1, 2, 3$:
   - Calcular la similitud entre las componentes verticales del bloque y del filtro.
   $$SimMap_k^V(i,j) = SM(V_{F_k}, V_B). \tag{14}$$
   - Calcular la similitud entre las componentes horizontales del bloque y del filtro.
   $$SimMap_k^H(i,j) = SM(H_{F_k}, H_B). \tag{15}$$
   - Agregar mediante la media geométrica la similitud de las componentes vertical y horizontal. Nótese que se podría haber usado cualquier otra función de agregación.
   $$SimMap_k(i,j) = \sqrt{SimMap_k^V(i,j) \cdot SimMap_k^H(i,j)}. \tag{16}$$

   2.2 Para cada posición i, j calcular la similitud total del core tomando el máximo valor entre cada punto de $SimMap_1$ y $SimMap_2$. De esta forma, unimos en un mismo mapa de similitudes las respuestas obtenidas por el filtro de cores convexos y por el filtro de cores concavos.
   $$SimMap_{12}(i,j) = max(SimMap_1(i,j), SimMap_2(i,j)). \tag{17}$$

3. Encontrar los máximos locales en cada mapa de similitud ($SimMap_{12}$ y $SimMap_3$). Los dos máximos locales con valores más altos de cada mapa de similitud son considerados puntos singulares si superan un umbral (T). Debemos hacer notar que como mucho pueden encontrarse cuatro puntos singulares: dos cores y dos deltas (huellas de tipo *whorl*).

4. Experimentos

Para estudiar la validez del método hemos llevado a cabo un estudio preliminar testeando nuestro método con dos bases de datos diferentes. La primera es la base de datos más utilizada en experimentos con huellas dactilares, la NIST (National Institute of Standards and Technology). La segunda consiste en una base de datos creada con la herramienta SFinGe [10,5], que permite la creación de huellas sintéticas muy realistas.

Los parámetros usados en los experimentos para el cálculo del mapa de orientaciones, han sido elegidos de acuerdo a los valores estándares estudiados y propuestos en la mayoría de trabajos sobre reconocimiento de huellas dactilares [1,8]. Por otro lado, los valores de los parámetros de nuestro método (tamaño de filtro y umbral para aceptar puntos singulares) han sido obtenidos mediante experimentos preliminares. Todos los parámetros considerados en el estudio pueden verse en el Cuadro 1.

Definición	Var	NIST	SFinGe
Tamaño de bloque del gradiente	W_1	8	4
Tamaño de bloque del suavizado	W_2	8	4
Tamaño del filtro	W_3	5	5
Ubral de PSs	T	0.8	0.8

Tabla 1. Variables utilizadas en el experimento.

Además de estos parámetros, debemos definir la medida de similitud utilizada para comparar los conjuntos difusos construidos. Los automorfismos utilizados para la construcción de la REF son $\varphi_1(x) = x$ y $\varphi_2(x) = x$, y consideramos la media aritmética como función de agregación.

Los resultados obtenidos para una serie de huellas dactilares pueden observarse en la Figura 2 y la Figura 3. Ambas figuras muestran resultados de la base de datos NIST y de las huellas generadas por SFinGe, respectivamente. En estas imágenes representamos los cores detectados con cruces, y los deltas detectados con cuadrados. Debemos destacar que hemos utilizado clases de todos los tipos (*arch, whorl, left-loop* y *right-loop*). En el caso de los arcos, el comportamiento deseado es que no se detecten puntos singulares (cosa que se consigue para las huellas del estudio). Los resultados presentados muestran información interesante, ya que hay puntos singulares bien detectados incluso en zonas de ruido y cortes (Figura 2 y Figura 3), lo que permite confíar en la bondad del método para detectar puntos singulares incluso en huellas con mala calidad. En cualquier caso, también podemos observar que en la huella de tipo *whorl* de los ejemplos

de NIST sólo se ha logrado detectar un único core en lugar de dos, aunque en este caso, de cara a la clasificación no importaría dado que la detección de los dos deltas permitirían la clasificación correcta de la misma.

Figura 2. Puntos singulares detectados en las huellas NIST. De arriba-izquierda a abajo-derecha: *arch, whorl, left-loop* y *right-loop*.

Figura 3. Puntos singulares detectados en las huellas SFinGe. De arriba a abajo: *arch, whorl, left-loop* y *right-loop*.

5. Conclusiones o líneas futuras

En esta contribución hemos propuesto un nuevo algoritmo de detección de puntos singulares basado en medidas de similitud. Estas medidas simplifican y facilitan la visión de los actuales procesos de extracción de puntos singulares debido a su razonamiento intuitivo. A pesar de no haber testeado grandes bases de datos, los resultados obtenidos con distintos tipos de huellas parecen prometedores de cara a futuras pruebas, hecho que resulta indispensable para demostrar su competitividad. Por último, como futuros trabajos tenemos la intención de probar con diferentes medidas de similitud, a parte de comparar nuestro algoritmo con otros como Poincarè, filtros complejos o el modelo *zero-pole*.

Referencias

1. L. Hong, Y. Wan, and A. Jain. Fingerprint image enhancement: algorithm and performance evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):777–789, 1998.
2. X. Jiang and W.Y. Yau. Fingerprint minutiae matching based on the local and global structures. In *Proceedings of the 15th International Conference on Pattern Recognition*, volume 2, pages 1038–1041. IEEE, 2000.
3. X. Chen, J. Tian, and X. Yang. A new algorithm for distorted fingerprints matching based on normalized fuzzy similarity measure. *IEEE Transactions on Image Processing*, 15(3):767–776, 2006.
4. R. Cappelli, M. Ferrara, and D. Maltoni. Minutia cylinder-code: A new representation and matching technique for fingerprint recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2128–2141, 2010.
5. D. Maltoni, D. Maio, A.K. Jain, and S. Prabhakar. *Handbook of fingerprint recognition*. Springer-Verlag New York Inc, 2009.
6. R. Cappelli and M. Ferrara. A fingerprint retrieval system based on level-1 and level-2 features. *Expert Systems with Applications*, 39(12):10465–10478, 2012.
7. A. M. Bazen and S. H. Gerez. Systematic methods for the computation of the directional fields and singular points of fingerprints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):905–919, 2002.
8. M. Liu. Fingerprint classification based on adaboost learning from singularity features. *Pattern Recognition*, 43(3):1062–1070, 2010.
9. L. Fan, S. Wang, H. Wang, and T. Guo. Singular points detection based on zero-pole model in fingerprint images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(6):929–940, 2008.
10. R. Cappelli, D. Maio, and D. Maltoni. Sfinge: an approach to synthetic fingerprint generation. In *Eighth International Conference on Control, Automation, Robotics and Vision (ICARCV2004)*, Kunming, China, December 2004.
11. L. A. Zadeh. Fuzzy sets. *Inform. Control*, 8:338–353, 1965.
12. H. Bustince, E. Barrenechea, and M. Pagola. Restricted equivalence functions. *Fuzzy Sets Syst.*, 157(17):2333–2346, 2006.
13. H. Bustince, E. Barrenechea, and M. Pagola. Image thresholding using restricted equivalence functions and maximizing the measures of similarity. *Fuzzy Sets Syst.*, 158(5):496–516, 2007.

Fuzzy Sets for Texture Contrast Modelling Applied to Pattern Recognition

J. Chamorro-Martínez[1], P. Martínez-Jiménez[1] *, and J.M. Soto-Hidalgo[2]

[1] Department of Computer Science and Artificial Intelligence, University of Granada, {jesus,pedromartinez}@decsai.ugr.es

[2] Department of Computer Architecture, Electronics and Electronic Technology, University of Córdoba, jmsoto@uco.es

Abstract. In this paper, we propose to model the contrast of visual texture by means of a perceptual-based fuzzy approach. For this modelling, fuzzy sets defined on the domain of some of the most representative measures of the contrast property are employed. In order to obtain these fuzzy sets, a functional relationship between the computational values given by the measures and the human perception of contrast is learned. The goodness of each model is analyzed and tested with the human assessments, allowing us to identify the most suitable one to represent the contrast of visual texture. Finally, several experiments are performed in order to show the application of the proposed fuzzy model for pattern recognition.

1 Introduction

Texture is, together with color and shape, one of the most used features for image analysis and, in addition, one of the most difficult to characterize due to its imprecision. In fact, there is not an accurate definition for the concept of texture but some intuitive ideas, as local changes in the intensity patterns or as a set of basic items arranged in a certain way [7]. However, for humans, the most common way to describe texture is by using vague textural properties, like *contrast*, *coarseness* or *directionality* [14,17], that give them an informal way to represent their perception about the texture.

Neurological experiments have shown that texture contrast has a strong influence on visual attention in natural images [13], and, moreover, it is closely related to the depth perception of the textured regions present in the image [8,16]. In this sense, the *contrast* property is considered as one of the most important properties in texture analysis [10,11,12], playing a fundamental role in human visual interpretation [17]. By considering the importance of the texture contrast, in this paper we will focus our study on this perceptual property.

There are many measures in the literature that, given an image, capture the contrast presence in the sense that the greater the value given by the measure,

* This work has been partially supported by the Government of Spain under Research Program *Consolider Ingenio 2010: Multimodal Interaction in Pattern Recognition and Computer Vision* (CSD2007-00018) and under the TIN2009-08296 project.

the greater (lower) the presence of this property [3,17,7]. However, there is no perceptual relationship between the value given by these measures and the degree in which the humans perceive the texture. Thus, given a certain value calculated by applying a measure to an image, there is not an immediate way to decide whether there is a contrasted texture, a non-contrasted texture or something intermediate (i.e. there is not a textural interpretation).

The imprecision associated to these contrast measures suggests the use of representation models that incorporate the uncertainty. Nevertheless, the majority of the approaches that can be found in the literature are crisp proposals [5,7,15] which do not model any kind of imprecision. To face this problem, some proposals arise from the fuzzy set field, and more specifically from the content-based image retrieval area [2,10,6,9,18]. In these proposals, a mapping from low-level statistical features (the crisp measures described above) to high level textural concepts is performed by defining membership functions for each textural feature.

However, in all these fuzzy approaches the membership functions are adjusted manually or by using a fuzzy clustering, but without considering the relationship between the measure values and the human perception of the property. This implies that the obtained membership degrees do not necessarily match what a human would expect. In addition, all these fuzzy approaches do not propose a global modelling of the textural concept, but a fuzzy partition providing a set of linguistic terms associated to this concept. This type of solution is unsuitable for some classical tasks, like pattern recognition, because a single presence degree of the textural property cannot be obtained, but one membership degree for each linguistic term in the partition.

In this paper, we propose a perception-based fuzzy approach for texture contrast modelling in order to solve all these problems. In this approach, the contrast presence is modelled by means of a unique fuzzy set defined on the domain of a representative measure of this property. Thus, the obtained fuzzy set will directly represent the presence degree of the property, allowing its use in pattern recognition problems, as it will be shown in section 5. In order to obtain the membership function, a functional relationship between the computational values given by the measures and the human perception of contrast is learned. This way, the presence degree given by the obtained fuzzy set will match what a human would expect.

The rest of the paper is organized as follows. In section 2 a general overview introducing our methodology is presented. After that, some elements of the model are describe in detail in the following sections; concretely, the way to obtain human assessments about contrast perception is faced in section 3, while section 4 describes the procedure employed to obtain the membership functions of the proposed fuzzy sets. In section 5 the results of applying the models are shown, and the main conclusions are summarized in section 6.

2 Fuzzy Modelling of Contrast: an Overview of the Proposal

In this paper we propose to model the contrast property of visual texture as a fuzzy set $\mathcal{T}_k$ defined on the domain of a representative contrast measure (our reference set). The membership function[3] of this fuzzy set will be defined as

$$\mathcal{T}_k : \mathbb{R} \rightarrow [0, 1] \tag{1}$$

For this modelling, two questions need to be faced: (i) what reference set should be used for the fuzzy set, and (ii) how to obtain the related membership function. Concerning to the reference set, we will define the fuzzy set on the domain of a given contrast measure. From now on, let $\mathcal{P} = \{P_1, \ldots, P_K\}$ be a set of contrast measures, with $P_k \in \mathcal{P}$ being the measure used to define $\mathcal{T}_k$. The measures analyzed in this paper are summarized in section 2.1. All of them are automatically computed from the texture images.

With regard to the membership function, we propose to obtain it by using a perceptually-based approach that relates the contrast measures with the human perception of the property. For this purpose, two questions need to be faced: firstly, how to obtain the data about the "human perception" of contrast and, secondly, how to fit this data with the measures in order to obtain the membership function. To get information about the human perception of contrast, a set of images covering different presence degrees of this property has been gathered. These images are used to collect, by means of a poll, human assessments about the perceived contrast presence. From now on, let $\mathcal{I} = \{I_1, \ldots, I_N\}$ be the set of N images representing contrast examples, and let $\Gamma = \{v^1, \ldots, v^N\}$ be the set of contrast values associated to $\mathcal{I}$, with v^i being the value representing the degree of contrast perceived by humans in the image $I_i \in \mathcal{I}$. The description of the texture image set and the way to obtain Γ are detailed in section 3.

To obtain the membership function $\mathcal{T}_k$ for a given measure $P_k \in \mathcal{P}$, a robust fitting method is employed in order to obtain suitable functions relating the values of the measure calculated for each image with the degree of contrast perceived by humans. This fitting method is described in section 4.

2.1 Contrast Measures

In this paper, we propose to use 4 of the most used contrast measures in the literature. Two of them try to estimate directly the contrast between texels by analyzing the pixels of the image. The first one is the measure defined by Tamura *et al.* in [17], which takes into account both the dynamic range of gray levels in the image and the kurtosis of their distribution. The second one is the contrast measure defined by Abbadeni in [1], which is based on the autocovariance function.

[3] To simplify the notation, as it is usual in the scope of fuzzy sets, we will use the same notation $\mathcal{T}_k$ for the fuzzy set and for the membership function that defines it.

(a) (b) (c) (d) (e)

Fig. 1. Some examples of texture images with different presence degrees of contrast.

The other two measures are obtained by applying statistics over matrices that collect information about the relationships between the gray level of each pixel and their neighbours. The first one is the contrast statistic proposed by Haralick *et al.* in [7], which is obtained from GLCM matrices. The second one is the contrast measure proposed by Amadasun and King in [3], which takes into account both global statistics (as the dynamic range of gray levels in the image) and local statistics calculated from the Neighbourhood Gray-Tone Difference Matrix.

3 Assessment Collection

In this section, the way to obtain the set of values $\Gamma = \{v^1, \ldots, v^N\}$, that represents the presence degree of contrast perceived by humans in the images $I_i \in \mathcal{I}$, will be described. For this purpose, first the image set $\mathcal{I}$ will be selected (section 3.1). After that, a poll for getting assessments about the perception of contrast will be designed (section 3.2). Finally, for a given image, the assessments of the different subjects will be aggregated (section 3.3).

3.1 The Texture Image Set

A set $\mathcal{I} = \{I_1, \ldots, I_N\}$ of $N = 80$ images representing examples of the contrast property has been selected. Figure 1 shows some images extracted from the set $\mathcal{I}$. Such set has been selected satisfying the following conditions:

- It covers the different presence degrees of contrast.
- The number of images for each presence degree is representative enough.
- Each image shows, as far as possible, just one presence degree of contrast.

Due to the third condition, each image can be viewed as "homogeneous" with respect to the presence degree of contrast, i.e., if we select two random windows (with a dimension which does not "break" the original texture primitives and structure), the perceived contrast presence will be similar for each window (and also with respect to the original image). In other words, we can see each image $I_i \in \mathcal{I}$ as a set of lower dimension images (sub-images) with the same presence degree as the original one. This will be very useful for the fitting process, because

we can have a larger number of fitting points without extending the number of images used in the poll.

3.2 The Poll

Given the image set $\mathcal{I}$, the next step is to obtain assessments about the perception of contrast from a set of subjects. From now on we shall note as $\Theta^i = [o_1^i, \ldots, o_L^i]$ the vector of assessments obtained from L subjects for the image I_i. To get Θ^i, subjects are asked to assign images to classes, so that each class has associated a perception degree of contrast.

In particular, five different classes have been considered in the poll. The five texture images shown in Figure 1 are the representative images for these classes. It should be noticed that these images are in decreasing order according to the presence degree of contrast. The first class (Figure 1(a)) represents the presence degree of 1, while the last one (Figure 1(e)) represents the presence degree of 0. The rest of classes represent presence degrees of contrast between 0 and 1, i.e. texture primitives with gradual variations in contrast between the previous ones.

In our approach, 20 subjects have participated in the poll. As result, a vector of 20 assessments $\Theta^i = [o_1^i, \ldots, o_{20}^i]$ is obtained for each image $I_i \in \mathcal{I}$. The degree o_j^i associated to the assessment given by the subject S_j to the image I_i is computed as $o_j^i = (9 - k) * 0.125$, where $k \in \{1, \ldots, 9\}$ is the index of the class to which the image is assigned by the subject.

3.3 Assessment Aggregation

Our aim at this point is to obtain, for each image in the set $\mathcal{I}$, one assessment v^i that summarizes the assessments Θ^i given by the different subjects about the presence degree of contrast. To aggregate opinions we have used an OWA operator guided by a quantifier [19]. Concretely, the quantifier "the most" has been employed, which allows us to represent the opinion of the majority of the subjects. This quantifier is defined as

$$Q(r) = \begin{cases} 0 & \text{if } r < a, \\ \frac{r-a}{b-a} & \text{if } a \leq r \leq b, \\ 1 & \text{if } r > b \end{cases} \tag{2}$$

$\forall r \in [0, 1]$, with $a = 0.3$ and $b = 0.8$. Once the quantifier Q has been chosen, the weighting vector of the OWA operator can be obtained following Yager [19] as $w_j = Q(j/L) - Q((j-1)/L), j = 1, 2, ..., L$. According to this, for each image $I_i \in \mathcal{I}$, the vector Θ^i obtained from L subjects will be aggregated into one assessment $v^i = w_1\hat{o}_1^i + w_2\hat{o}_2^i + ... + w_L\hat{o}_L^i$, where $[\hat{o}_1^i, \ldots, \hat{o}_L^i]$ is a vector obtained by ranking in nonincreasing order the values of the vector Θ^i.

4 Fitting the Membership Function

At this point, the aim is to obtain, for a given measure $P_k \in \mathcal{P}$, the corresponding membership function $\mathcal{T}_k$. In this paper, we propose to find a function that associates the values of the contrast measures with the human assessments about this property. As it was pointed out in section 3.1, thanks to the "homogeneity" in the presence degree of contrast, each image $I_i \in \mathcal{I}$ can be seen as a set of sub-images with the same contrast degree v_i of the original one. From now on, we will denote by $\mathcal{I}_\mathcal{W} = \{I_{i,w}, i = 1, \ldots, N; w = 1, \ldots, W\}$ the set of sub-images extracted from $\mathcal{I}$, where $I_{i,w}$ is the w-th sub-image of I_i and W is the number of sub-images considered for each image; on the other hand we will denote by $m_k^{i,w}$ the result of applying the measure P_k to the sub-image $I_{i,w}$. According to this notation, let $\mathcal{I}_\mathcal{W}^{\text{fit}} \subset \mathcal{I}_\mathcal{W}$ and $\mathcal{I}_\mathcal{W}^{\text{test}} = \mathcal{I}_\mathcal{W} \backslash \mathcal{I}_\mathcal{W}^{\text{fit}}$ be two complementary subsets of $\mathcal{I}_\mathcal{W}$, that will be used for fitting the membership function and testing the obtained model, respectively.

Thus, in order to estimate the membership function that associates the measure values ($m_k^{i,w}$) and the human assessments of contrast (v^i), we propose to fit a suitable function of the form given in Eq. (1) to the subset of points:

$$\Psi_k^{\text{fit}} = \{(m_k^{i,w}, v^i); \forall I_{i,w} \in \mathcal{I}_\mathcal{W}^{\text{fit}}\} \tag{3}$$

In this paper, for each image $I_i \in \mathcal{I}$, $W = 200$ sub-images of size 32×32 have been considered, so $\mathcal{I}_\mathcal{W}$ is formed by 16000 sub-images. We propose to randomly select 75% of them for the fitting, so that 12000 points are contained within Ψ_k^{fit}.

The measure values can be affected by some factors, like brightness, contrast or noise, which typically causes outliers in the fitting points. For this reason, in our approach the membership function is calculated by means of a robust fitting of the multiset Ψ_k^{fit}. In this modelling, the robust fitting based on M-estimators (a generalization of the least squares fitting) is used [4]. In addition, to define $\mathcal{T}_k$, the following considerations are taken into account:

- $\mathcal{T}_k$ should be a monotonic function.
- The values $\mathcal{T}_k(x) = 0$ and $\mathcal{T}_k(x) = 1$ should be reached.

Regarding the above properties, we propose to define $\mathcal{T}_k$ as a function of the form[4]

$$\mathcal{T}_k(x; a_n \ldots a_0, \alpha, \beta) = \begin{cases} 1 & x < \alpha, \\ poly(x; a_n \ldots a_0) & \alpha \leq x \leq \beta, \\ 0 & x > \beta \end{cases} \tag{4}$$

with $poly(x; a_n \ldots a_0)$ being a polynomial function

$$poly(x; a_n \ldots a_0) = a_n x^n + \ldots + a_1 x^1 + a_0 \tag{5}$$

[4] Note that this function is defined for measures that decrease according to the perception of contrast. For those that increase, the function needs to be changed appropriately, i.e. it takes the value 0 for $x < \beta$, it takes the value 1 for $x > \alpha$, and the polynomial function is computed for $\beta \leq x \leq \alpha$.

Table 1. Fitting error, test error and parameters of the membership function corresponding to each contrast measure.

Contrast measure	Fitting error	Test error	a_3	a_2	a_1	a_0	α	β
Tamura	0.0340	0.0649	1.6877	-3.9536	3.8763	-0.5728	0.1775	0.9620
Amadasun	0.0780	0.1108	0.2803	-1.1488	1.8254	-0.3839	0.2462	2.1288
Abbadeni	0.1003	0.1393	0.3977	-1.8684	2.9594	-1.0859	0.5172	2.6235
Haralick	0.1157	0.1416	0.6232	-1.8714	2.0679	-0.4100	0.2502	1.8773

In our proposal, the parameters $a_n, \ldots, a_0$, α and β of the function $\mathcal{T}_k$ are calculated by carrying out a robust fitting on Ψ_k^{fit}, with the constraint to obtain a strictly monotonic function between α and β. For the polynomial function, the cases of $n = 1, 2, 3, 4$ (i.e. linear, quadratic, cubic and quartic functions) have been considered.

The second column of Table 1 shows for each measure $P_k \in \mathcal{P}$ the least fitting error obtained. Note that this value can be viewed as the goodness of each measure to represent the perception of contrast. Table 1 has been ranked in increasing order of these fitting errors. In all the cases, the least error has been obtained for a polynomial function of order $n = 3$ (the use of higher order functions does not provide better fits).

In addition, the test error for each measure has been calculated by using the subset of points Ψ_k^{test} and it is shown in the third column of Table 1. In our approach, this error is calculated as the mean absolute difference between the values v_i and the degrees obtained by applying the function $\mathcal{T}_k$ to the values $m_k^{i,w}$, for all the points $(m_k^{i,w}, v^i) \in \Psi_k^{\text{test}}$, i.e.

$$E^{\text{test}} = \frac{\sum_{(m_k^{i,w}, v_i) \in \Psi_k^{\text{test}}} \left| \mathcal{T}_k(m_k^{i,w}) - v_i \right|}{card(\Psi_k^{\text{test}})} \tag{6}$$

with $card(\Psi_k^{\text{test}})$ being the cardinality of Ψ_k^{test}. The parameters of the membership function corresponding to each contrast measure are shown in the rest of the columns of Table 1.

5 Results

In this section, the fuzzy sets proposed in this paper for texture contrast modelling will be applied to several examples in order to analyze their performance. In particular, the model with least fitting error and least test error (corresponding to the measure of Tamura according to Table 1) will be employed for pattern recognition.

For the first experiment, we have considered Figures 2(a) and 2(b), corresponding to two natural images where several textures with different perception degrees of contrast are present. Figures 2(c) and 2(d) show a mapping from these

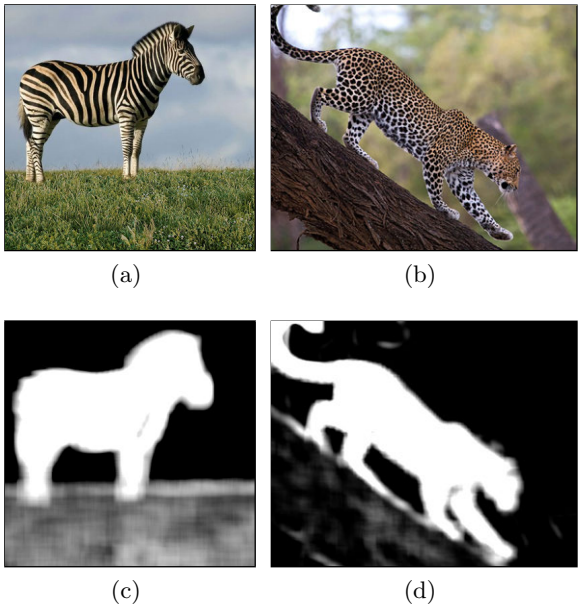
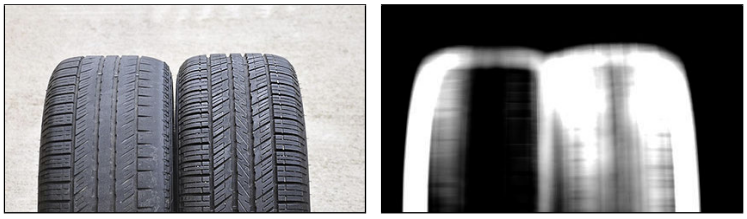


Fig. 2. Results for two natural images. (a)(b) Original images. (b) Mapping from the original images to their contrast values using the proposed model.

natural images to their contrast values using the proposed model. For each pixel in the original images, a centered window of size 32×32 has been analyzed and its contrast membership degree has been calculated using the proposed model. This degree has been mapped into a gray level from 0 to 255. It can be noticed that, as it was expected, three different degrees of contrast are shown in each mapping: a high contrasted texture (pixels in white), corresponding to the zebra in the first case and the leopard in the second one; a low contrasted texture (pixels in black), corresponding to the background of each image; and a half-contrasted texture (pixels with an intermediate gray level) corresponding to the grass and the branch, respectively. Thus, we can see that our model captures the evolution of the perception degrees of contrast, and the obtained mappings, that represent the estimated presence degree of this property, can be directly interpreted by humans.

As it was commented in Section 1, the contrast of a visual texture is related to the depth perception and the relief of the corresponding physical texture: textures with high relief in nature may appear with high contrast in the image (due to the illumination effect). Thus, similar physical textures with different relief, can be distinguished in an image by analyzing the contrast of the corresponding visual textures. Figure 3 shows an example where the proposed contrast model is used to identify defective patterns.



(a) (b)

Fig. 3. Identification of defective patterns. (a) Original image. (b) Mapping from the original image to its contrast values using the proposed model.

Lets consider the image shown in Figure 3(a), where we can see two tires with different wear levels. The tire on the right has deep grooves, while the one on the left has an irregular wear in the center and on one side. Figure 3(b) shows a mapping from this image to its contrast values using the proposed model. It can be noticed that the worn parts correspond to areas with low contrast degrees, so they can be identified if only the pixels with contrast degree lower than 0.1 are selected.

6 Conclusions

In this paper, the contrast property of visual texture has been modelled by means of fuzzy sets defined on the domain of computational measures of this property. In order to define these models, parametric functions have been employed, where the corresponding parameters have been calculated by taking into account the relationship between the computational measures and the human perception of contrast. This way, the shape of the membership function has been adjusted to represent this relationship, and the obtained membership degrees match what a human would expect. We have concluded that the model obtained by using the measure of Tamura has the best ability to represent the perception of contrast. Moreover, the use of a unique fuzzy set to model the texture contrast as a whole has allowed its application to pattern recognition problems, as has been shown in the experiments of section 5

References

1. N. Abbadeni, N. Ziou, and D.S. Wang. Autocovariance-based perceptual textural features corresponding to human visual perception. In *Proc. 15th International Conference on Pattern Recognition*, volume 3, pages 901–904, 2000.
2. H. Aboulmagd, N. El-Gayar, and H. Onsi. A new approach in content-based image retrieval using fuzzy. *Telecommunication Systems*, 40(1):55–66, 2008.
3. M. Amadasun and R. King. Textural features corresponding to textural properties. *IEEE Transactions on Systems, Man and Cybernetics*, 19(5):1264–1274, 1989.

4. A.E. Beaton and J.W. Tukey. The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data. *Technometrics*, 16:147–185, 1974.
5. K.I. Chang, K.W. Bowyer, and M. Sivagurunath. Evaluation of texture segmentation algorithms. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 294–299, 1999.
6. C.Y. Chiu, H.C. Lin, and S.N. Yang. A fuzzy logic CBIR system. In *Proc. 12th IEEE International Conference on Fuzzy Systems*, volume 2, pages 1171–1176, May 2003.
7. R.M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804, 1979.
8. S. Ichihara, N. Kitagawa, and H. Akutsu. Contrast and depth perception: effects of texture contrast and area contrast. *Perception*, 36(5):686–95, 2007.
9. S. Kulkarni and B. Verma. Fuzzy logic based texture queries for CBIR. In *Proc. 5th International Conference on Computational Intelligence and Multimedia Applications*, pages 223–228, 2003.
10. H.C. Lin, C.Y. Chiu, and S.N. Yang. Finding textures by textual descriptions, visual examples, and relevance feedbacks. *Pattern Recognition Letters*, 24(14):2255–2267, 2003.
11. W.T. Lin, C.H. Lin, T.H. Wu, , and Y.K. Chan. Image segmentation using the k-means algorithm for texture features. In *Proc. International Conference on Computer, Electrical, and Systems Science, and Engineering (ICCESSE)*, pages 26–28, 2010.
12. W. Niblack, R. Barber, W. Equitz, M. Flickner, E.H. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin. The QBIC project: Querying images by content, using color, texture, and shape. In *Proc. SPIE 1908, Storage and Retrieval for Image and Video Databases*, pages 173–187, 1993.
13. D.J. Parkhurst and E. Niebur. Texture contrast attracts overt visual attention in natural scenes. *European Journal of Neuroscience*, 19(3):783–789, 2004.
14. A.R. Rao and G.L. Lohse. Identifying high level features of texture perception. *CVGIP: Graphical Models and Image Processing*, 55(3):218–233, May 1993.
15. T.R. Reed and J.M.H. Dubuf. A review of recent texture segmentation and feature extraction techniques. *CVGIP: Image Understanding*, 57(3):359–372, 1993.
16. A.G. Rempel, W. Heidrich, and R. Mantiuk. The role of contrast in the perceived depth of monocular imagery. In *Proc. ACM SIGGRAPH Symposium on Applied Perception in Graphics and Visualization*, APGV '11, pages 115–115, 2011.
17. H. Tamura, S. Mori, and T. Yamawaki. Textural features corresponding to visual perception. *IEEE Trans. on Systems, Man and Cybernetics*, 8:460–473, 1978.
18. B. Verma and S. Kulkarni. A fuzzy-neural approach for interpretation and fusion of colour and texture features for CBIR systems. *Applied Soft Computing*, 5(1):119–130, 2004.
19. R.R. Yager. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transactions on Systems, Man and Cybernetics*, 18(1):183–190, 1988.

The WAT Algorithm, an Algorithm to see both Woods and Trees

MATTIOLI ARAMBURU, Gabriel

Mathematics and Computer Science Department
ETSAV - Universitat Politecnica de Catalunya
C/ Pere Serra 1-15, 08190, Sant Cugat del Valles, Spain
gabriel.mattioli@upc.edu

Abstract. Several pattern recognition and clustering algorithms such as Fuzzy c-Means require to know a priori the number of clusters to be found. This problem is non-trivial in several cases. In this work we provide a new algorithm to solve this problem based on the idea of "'unfocusing"' the data. We will study the performance of this algorithm upon biomedical images and analyze the results obtained.

Keywords: Clustering, WAT algorithm, unfocus, spot

1 Introduction

Finding the number of clusters in an image is a fundamental issue in the field of clustering. In order to find the different objects in an image, many algorithms such as Fuzzy c-Means and Expectation-Maximization require to know previously how many clusters should be found. In the literature several algorithms have been proposed for this purpose. To recall some of them we refer the reader to [7] and [10]. However, most of these methods are based on an iterative application of the clustering algorithm, for instance and from now on on this work Fuzzy c-Means (FCM), and study the variance, bias and error committed considering different amounts of output clusters in order to find the best figure. This approach has a fundamental problem. Given that clustering algorithms are in general computationally costly, it may become unaffordable to explore all the different possibilities to reach the best one. For instance, if the initial image has 300 real objects, according to these methods FCM should be ran 300 times in order to find the "true" figure.

To overcome this problem in this work we propose a new algorithm named WAT (Woods And Trees) Algorithm. The idea behind this method is very different from the ones cited above as it is not based on a variational study of an iterative application of FCM but on a direct estimation of the number of clusters in the image.

In this work we will formally expose this new algorithm and analyze briefly either its performance in a practical case and its theoretical consistency. As will

be pointed in the concluding remarks of this work, despite the method proposed has been initially focused on solving the estimation of number of clusters problem, the viewpoint sustaining this algorithm can be tuned to solve further problems in the field of clustering as well as being used as a clustering method by itself.

This work is structured as follows:

In Section 2 the WAT algorithm will be proposed. The idea underlying the algorithm will be explained in depth as well as the parameters governing the algorithm.

Section 3 will be concerned with some technical issues of the WAT algorithm. Some mathematical calculations will be detailed and some theoretical ideas that can sustain and link some steps of the algorithms will be remarked.

In Section 4 we will study the performance of the algorithm on two real biomedical images of breast tumours.

Finally, the concluding remarks of this work will be given.

2 The WAT Algorithm

In this section we will introduce the WAT algorithm. First of all we will explain the seed of the idea as well as where the name comes from. Following we will expose the algoritm. In further sections we will discuss more deeply some technical issues related to the algorithm.

The keystone of this work is the premise that sometimes it is necessary to "step back" in order to better understand an image. In other words, sometimes an unfocused image provides more structural information about an image than the original focused one because of an excess of details. It is well known in the field of clustering that noise is in the details [2], but the essential fact we are pointing is that sometimes it is the fact of having details what brings noise to the system because it may distract or overfeed the algorithm.

This viewpoint is not only not new but recorded in the popular expressions of many languages. In English this expression is:

You can't see the wood for the trees.

This expression gives name to the algorithm we propose: Woods And Trees (WAT) Algorithm.

The main actor of the WAT algorithm is the concept of spot we define below.

Definition 21. *Let p be a pixel of an image I of dimension $N \times M$, $E : I \times I \to [0,1]$ a fuzzy relation evaluating similarity of gray level between pixels, d a distance measure, $h \in [0,1]$ and $r \in [1,N]$. We will say that p defines a spot if and only if:*

$$\forall p' \quad d(p,p') \leq r \Leftrightarrow E(p,p') \geq h$$

.

The definition of spot is very intuitive and straightforward from the mathematical concept of ball. If a pixel p is inside a cluster then there must be some

homogeneity in the gray-level within a neighborhood of p and hence for adequate values of radius r, p defines a spot.

This definition has two main problems. The first one is that it does not take into account the noise practical images always have. The second one is that apply this definition to all the pixels p' of a neighborhood of a pixel p and iterate for all the pixels p in the image is computationally very costy, opposite to one of the initial aims of this work.

In order to overcome these problems the algorithm introduces the following parameters:

2.1 Parameters of the WAT algorithm

The WAT algorithm will have the following parameters

- $r \in \mathbb{Z}^+;\ r << N, M$: Radius of granularity
- $h \in \mathbb{R};\ h \in [0, 1]$: Similarity of gray-level to suppose common cluster
- $k \in \mathbb{Z}^+;\ k \in [1, (2r)^2]$: Number of pixels to evaluate in each neighborhood
- $l \in \mathbb{Z}^+;\ l \in [0, k]$: Threshold of similar pixels within a neighborhood to assume spot is defined
- $\epsilon \in \mathbb{R};\ \epsilon \in (0, 1)$: Exhaustivity of search in the image

Let us briefly explain the sense of these parameters.

The parameters r and h measure the accuracy and granularity to define spots (Definition 21).

r is the radius of the squared-balls considered around each pixel p and it is remarkable that should be significantly smaller than the dimensions of the image.

The gray-level resemblence of pixels is evaluated through the fuzzy relation E. The parameter h stands for the lower threshold ($E(p, p') \geq h$) we assume in order to suppose common cluster between pixels p' and the centroid p of the ball.

ϵ and k are stadistical parameters. In order to reduce complexity, the WAT algorithm will not evaluate which pixels define spots among all the $N \times M$ pixels of the image but just among a random number n of them. Choosing random numbers has the problem that it is not ensured that all the image will be covered by balls. However for the purpose of the algorithm it is not necessary to study all the image but a significant part of it. The ϵ parameter represents thus the upper bound of the percentage of the image which will not be studied. Let us remark that this value is an estimation as random numbers are being considered. A further problem which will be studied in Section 3.1 is how to decide the number n of pixels given ϵ.

Chosen pixels $p_1, ..., p_n$ and squared balls of radius r around them it is still costy to evaluate the resemblance between p_i and all the $4r^2 - 1$ pixels of its neighborhood. Besides the intutive concept of spot is intrinsically fuzzy rather than so exhaustive. For that reason the k parameter is introduced: the gray-level will not be compared with all the pixels of the ball but with k randomly chosen.

Finally, l to avoid outsiders and noisy pixels. If a pixel p is compared with k neighbors and it is very similar to almost all of them except, intuitively we should consider that still we are observing a spot. Hence, to assume a spot is defined it will be required that p is similar to l pixels, being l a number less than but close to k.

2.2 The WAT Algorith

The WAT algorithm runs as follows:

Algorithm 22. *INPUT: A $N \times M$ image and values of the parameters r, h, k, l, ϵ*

1. *Calculate the number n of pixels*
2. *Choose randomly n pixels $p_1, ..., p_n$ in the image*
3. *For each p_i*
   (a) *Choose k random pixels $p'_1, ..., p'_k$ in a squared neighborhood of radius r around p_i*
   (b) *Compare the gray-level between p_i and p'_j for $j = 1, ..., k$ using the fuzzy relation E*
   (c) *If l pixels among $p'_1, ..., p'_k$ verify $E(p_i, p'_j) \geq h$ then p_i defines a spot*
4. *Denote $s_1, ..., s_m$ the pixels defining spots*
5. *For each s_i*
   (a) *For j=1,...,n if $d(s_i, s_j) < 2r$ and $E(s_i, s_j) \geq h$ then s_i and s_j belong to the same cluster*
6. *Denote $c_1, ..., c_p$ spots defining different clusters*

OUTPUT: $c_1, ..., c_p$

Let us remark some issues about the algorithm proposed.

First of all, despite we begin talking about pixels, further about spots and finally about clusters; at a practical level we are all the time working with pixels. Step three of the WAT algorithm labels as "spots" those pixels veryfing the conditions exposed and step five identifies thses spots if they come from a common source (cluster).

The output of the algorithm is a collection of pixels which can be considered as seeds for any seeded image segmentation algorithm. The amount of output pixels provides the amount of objects in the image.

Another interesting point is that step five can be regarded as a filtered single linkage. This viewpoint will be discussed in Section 3.2. The condition of restricting the comparison of spots to those at distance less or equal to $2r$ is because balls have radius two, hence two overlapping balls must have their centers at most at distance $2r$.

This condition forces the algorithm to find objects. If this condition was removed then the algorithm would compare all spots and the output would not be the amount of different objects but the amount of different classes of objects.

The difference is that in the same case an object which is found twice disjonctly in the image is counted twice and in the second case its identified as one.

It is an interesting point of WAT to have this flexibility because it allows to face either class-finding and object-finding problems withous adding new spatial distances or changing the code or core of the method.

3 Some technical comments about the WAT algorithm

In this section we will discuss some technical issues related with the WAT algorithm provided in Section 2. Firstly we will specify how the parameter n that regulates the number of pixels to be studied can be defined given a radius r i an exhaustivity level ϵ. Besides we will discuss how the 4th step of the algorithm can be seen as a filtered Single Linkage.

3.1 How to decide the initial number of pixels

One of the aims of the WAT algorithm is to be accurate "enough" at a minimal computational cost. The ϵ parameters regulates the search space of the algorithm. For instance, for $\epsilon = 0.05$ the algorithm should statistically ensure that the 95% of the image is covered by n squared-balls of radius r chosen randomly. The question is how to define the figure n.

This is a very interesting mathematical problem. In this work we propose the following formula for an image of dimension $N \times M$:

$$n \geq \frac{log(\epsilon)}{log(1 - \frac{(2r)^2}{N \cdot M})}$$

Taking n the smallest integer satisfying the previous inequality. Let us explain where this formula comes from.

We denote A_i the uncovered area after placing i random squared balls. Obviously

$$A_0 = N \cdot M$$

If $r << N, M$

$$A_1 = A_0 - (2r^2) = A_0 - (2r)^2 \frac{A_0}{A_0}$$

And we can derive the iterative formula

$$A_{i+1} = A_i - (2r)^2 \frac{A_i}{A_0}$$

Rewriting the previously formula we can see it as a geometric serie

$$A_{i+1} = A_i(1 - (2r^2)/NM) = NM(1 - (2r^2)/NM)^i$$

We are looking for the value of i such that $A_{i+1} \leq NM \cdot \epsilon$.

$$NM(1-(2r^2)/NM)^i < NM \cdot \epsilon$$

Simplifying and taking logarithms we have the final inequality

$$i \geq \frac{log(\epsilon)}{log(1-\frac{(2r)^2}{N \cdot M})}$$

An analysis of the method described to find n shows that the figure found is statistically sound and consistent. The crucial point is how to estimate how much area does a new random ball cover after i balls have been placed. Statistically, if $r << N, M$ we can despise the cases where the new ball intersects the limits of the image and then the expected value of "new" area the new ball covers is exactly $(2r)^2 \frac{A_i}{NM}$. Hence, at each step we are taking off the most likelihood value.

	$\epsilon = 0.1$	$\epsilon = 0.05$	$\epsilon = 0.01$
$r = 3$	122804	159771	245607
$r = 5$	44209	57517	88416
$r = 10$	11052	14379	22103

Table 1. Values of n for different values of r and ϵ on an image of 1.92 megapixels

3.2 Joining spots step as a filtered Single Linkage

The WAT algortihm has two core steps. The first one is to identify pixels contained in spots, which are taken by the algorithm as "seeds" of a spot. The second one is to identify which of these seeds are defining the same spot.

Due to how we have chosen the amount of pixels n (Section 3.1) we can suppose (with statistical consisteny) that the randomly selected squared balls cover the $(1-\epsilon) \times 100\%$ of the image. Identifying which balls define the same spot is hence a matter of joining the similar ones.

Condition of Step 5 in the WAT algorithm (Algorithm 22) states that two pixels will be labelled into the same spot if the balls of radius r and centered in p and p' have non-void intersection and are similar in gray-level.

If an object contains several balls the identification is concatenated so all of them are equally labelled.

Figure 1 illustrates how different balls are identified into the same spot.

It can be observed that this approach is equivalent to a filtered Single Linkage.

One of the problems of Single Linkage (SL) is that concatenating transitive chains may eventually lead to a global connection that trascends the objective clusters [3] [9]. In our approach this problem is overcame, with an adequate value of the values of the parameters, because SL is not executed over all the pixels but just over the ones that define spots. This filter avoids the problematic transitive chains of a simple SL.

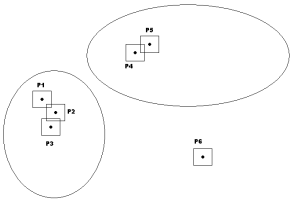


Fig. 1. Three clusters. It is shown how different spots are linked

4 Example of an application of WAT

In this section we will provide an example of an application of WAT on real biomedical images.

Figures 2 and 3 shows two mammographies with breast tumours.

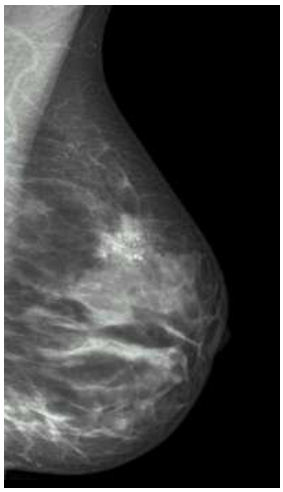


Fig. 2. Mamography 1

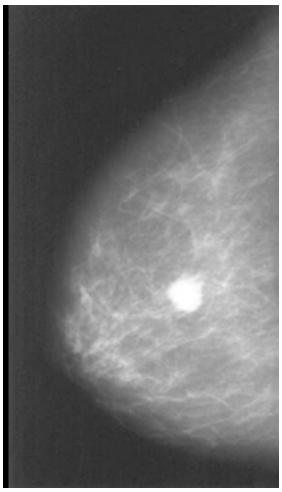


Fig. 3. Mamography 2

The tumours are pointed out in figures 4 and 5.

Our aim is to automatically detect these tumours. In order to do that we have applied the proposed WAT algorithm and following a FCM algorithm to find the number of clusters proposed by what.

The final values of the parameters have been set as:

- $r = 4$
- $h = 0.7$
- $k = 5$

Fig. 4. Mamography 1. Circled we find the calcification

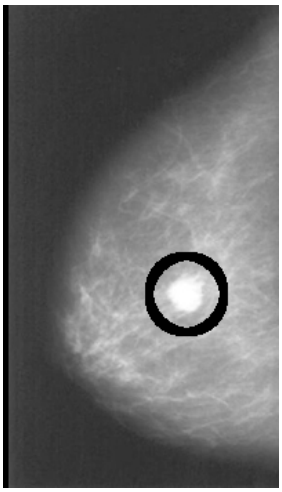


Fig. 5. Mamography 1. Circled we find the calcification

- $l = 4$
- $\epsilon = 0.05$

The WAT algorithm has proposed in both images that 4 clusters should be found. Figures 6 and 7 show the output clusters given by FCM for images 1 and 2 respectively with the seeds given by WAT.

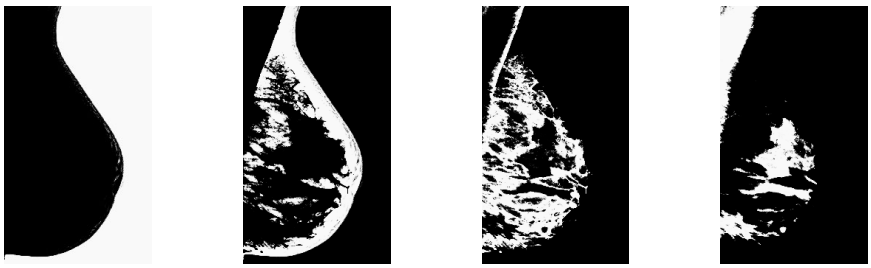


Fig. 6. Mammography 1: Background, tissue, fibers and tumours clusters

As it can be observed, in Mammography 1 the process does not provide exactly the aimed tumours but a good approximation of them. The image should be cleaned in a postprocess in order to obtain the tumours exactly.

In Mammography 2 the output obtained is not so clear as there is no cluster which corresponds clearly to the tumour. However, it can be observed that in figures 4 and 4 there is a slightly deeper gray colour where the tumour is. This

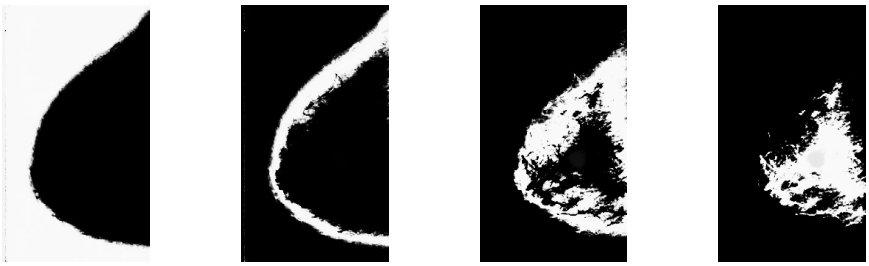


Fig. 7. Mammography 1: Background, tissue, fibers and tumours clusters

happens because FCM identifies the tumour in two classes with a certain non-zero membership degree. With this observation the tumour can be detected clearly.

It is a must to observe that, in spite of not having an exact output in neither of both cases, the inaccuracy is located in the FCM stage. WAT algorithm provides in both cases the exact number of clusters: four; which correspond to the background, the tissue, the fibers and the tumours. An analysis of the seeds for FCM provided by WAT shows that these are exactly the classes observed by WAT.

A final comment to be made is that one of the goals of the new algorithm developed was to be computationally cheap. Executing the WAT algorithm with these images has taken 2.64 seconds of computation time while the FCM stage has taken 9.81 seconds. Hence WAT algorithm is clearly superior in the computational dimension to the methods proposed in [7] and [10] which were based on a multiple execution of FCM.

Concluding Remarks

In this work we have provided a new algorithm to estimate the number of clusters in an image or set of data. This is an initial problem when a clustering or image segmentation process has to be done with some of the most popular clustering methods such as Fuzzy c-Means.

The new algorithm proposed is named WAT and has proved to provide the correct number of clusters in the examples shown in Section 4.

Three final remarks should be made to this algorithm. The first one is that its computational cost is very low, specially in comparison with the methods proposed in the literature. On the other hand, though in the example provided the number proposed is the exact one, in general there is no certainty that the number provided by WAT is the best but a good approximation of it. Finally, though the algorithm is based on the idea of unfocusing there is no image unfocusing in this method. Actually, it is the "eye" that regards the image thas unfocuses its sight through the parameter adjustement.

Future work has to be done in order to ensure that the new algorithm proposed is useful in other real cases and to state that the promising results suggested in this work are actually that good. Another interesting study is to analyze the variational behaviour of the algorithm with respect to the parameters.

An interesting issued related to the work done is that further than the WAT algorithm proposed, the unfocusing-based approach is interesting and can lead to other applications in the field of image segmentation and clustering and even to whole new clustering algorithms.

Finally, the author is currently working on giving a formal theoretical background to this unfocusing process to understand in depth what happens at the relational level with the similarity of pixels of a same object when an unfocussing process is triggered.

Acknowledgement

The author is supported by the Universitat Politècnica de Catalunya.

References

1. Duin R.P.W., Pekalska E.: *The dissimilarity representation for pattern recognition. Foundations and applications* World Scientific (2005) .
2. A. B. Hamza, P. L. Luque-Escamilla, J. Martínez-Aroza, R. Román-Roldán: *Removing noise and preserving details with relaxed median filters*. Journal of Mathematical Imaging and Vision. Vol 11, Iss 2, pp 161-177 (1999).
3. Hartigan, J. A.: *Consistency of Single Linkage for High-Density Clusters. Journal of the American Statistical Association*. Vol. 76, No. 374, pp. 388-394 (1981).
4. Lindeberg, T.: *Feature detection with automatic scale selection*. International journal of computer vision. Vol 30(2), pp 79-116 (1998).
5. Mattioli, G., Recasens, J.: *Isomorphisms between Indistinguishability Operators and related Concepts*. Information Sciences. Vol 241, pp 85-100 (2013).
6. Mattioli, G.: *Application of Fuzzy Techniques to Biomedical Images*. UPC publications. Master Thesis (2012).
7. Pelleg, D., Moore, A.: X-means: *Extending k-means with efficient estimation of the number of clusters*. Proceedings of the seventeenth international conference on machine learning. Vol. 1, pp. 727-734 (2000).
8. Recasens, J.: *Indistinguishability Operators. Modelling Fuzzy Equalities and Fuzzy Equivalence Relations*. Series: Studies in Fuzziness and Soft Computing, Vol. 260 (2011)
9. Stuetzle, W., Nugent, R.: *A Generalized Single Linkage Method for Estimating the Cluster Tree of a Density*. Journal of Computational and Graphical Statistics, Vol. 19, Iss. 2 (2010).
10. Tibshirani, R., Walther, G., Hastie, T.: *Estimating the number of clusters in a data set via the gap statistic*. Journal of the Royal Statistical Society: Series B Statistical Methodology. Vol 63, Iss 2 pp 411-423 (2001).
11. Witkin, A.: *Scale-space filtering: A new approach to multi-scale description*. Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'84. Vol. 9, pp. 150-153 (1983).

Medidas de similitud difusas para comparar imágenes segmentadas en color

Daniel Paternain, Mikel Galar, Aranzazu Jurio, Edurne Barrenechea

Departamento de Automática y Computación,
Universidad Pública de Navarra, 31006, Pamplona, España

Resumen En este trabajo presentamos un método que nos permite evaluar la calidad de las imágenes que se obtienen después de realizar su segmentación mediante diferentes aproximaciones. Para ello, consideramos cada componente de una imagen en color segmentada como un conjunto difuso. Posteriormente, proponemos utilizar medidas de similitud (entre conjuntos difusos) que nos permiten comparar dos imágenes segmentadas en color: una de ellas es la *imagen ideal* (imagen segmentada a mano) y la otra es la imagen que se obtiene después de aplicar un método de segmentación de imágenes en color. Realizamos un experimento preliminar con tres algoritmos de segmentación (FCM [1], MAP-ML [7], 2-Tuplas [12]) utilizando las imágenes en color de la base de datos de Berkeley [9] y evaluamos los resultados que obtenemos con nuestro método de evaluación.

Palabras clave: Medidas de similitud, Segmentación de imágenes, Comparación de imágenes

1. Introducción

Uno de los problemas más importantes en los Sistemas de Visión es la identificación de los objetos que aparecen en una imagen. Evidentemente, esta fase de identificación es muy sencilla para las personas pero muy compleja para las máquinas [13]. La división de una imagen en regiones se denomina *segmentación*. En realidad, la segmentación de las imágenes digitales es el proceso de dividir una imagen en regiones disjuntas , partes o clases de tal forma que cada una de ellas tiene propiedades o atributos muy concretos. Cada una de estas regiones representa un objeto de la imagen.

Para comparar la bondad de un algoritmo de segmentación existen diferentes medidas [9,15]. Un mecanismo muy utilizado es la comparación de dos imágenes segmentadas, una obtenida mediante la ejecución de un método de segmentación de imágenes, y la otra hecha a mano por un experto (imagen denotada en la literatura especializada como *imagen ideal*).

En [9] existen diferentes imágenes ideales proporcionadas por un conjunto de expertos. Uno de los factores que afecta de manera crucial a la segmentación de imágenes es el número de clases (regiones) en las que se segmenta, ya que depende de los criterios de evaluación tomados, como por ejemplo el color, la forma o

la textura. Además, en función del contexto o de la aplicación, los expertos pueden realizar diferentes segmentaciones. En este trabajo partimos de la hipótesis de que los píxeles que pertenecen a un objeto de la imagen tienen un color similar y proponemos un algoritmo de comparación que nos permite calcular la similitud de dos segmentaciones de una imagen en color dada. Obviamente, si una de estas dos segmentaciones es la imagen ideal, entonces nuestro algoritmo de comparación nos permite medir objetivamente la bondad de la segmentación obtenida por cualquier método de segmentar imágenes. La principal ventaja de nuestra propuesta es que no penaliza si un píxel se clasifica en diferentes objetos en las dos segmentaciones, siempre y cuando el color medio (promedio) de dichos objetos sea similares.

Las imágenes digitales en sí mismas contienen un importante grado de incertidumbre debido a los procesos de muestreo y cuantificación. Por esta razón, la Teoría de Conjuntos Difusos [16] ha sido ampliamente utilizada como una herramienta para intentar resolver problemas en el área de procesamiento de imagen [2,4,6]. Principalmente, dicha incertidumbre es debida a las limitaciones existentes en las escalas discretas de niveles de gris o de color que son utilizadas en la codificación y en el proceso de muestreo para transformar una imagen analógica en una matriz de píxeles. Por lo tanto, parte de la información que captamos del mundo real en una imagen siempre se pierde en el proceso de digitalización (muestreo y cuantificación). Por estas razones, entre otras, en nuestra propuesta utilizamos la Teoría de Conjuntos Difusos.

El algoritmo que proponemos en este trabajo está basado en: (a) la representación de una imagen en color RBG por medio de conjuntos difusos, un conjunto difuso para cada componente de color, y (b) medidas de similitud difusa.

El resto del trabajo está organizado de la siguiente manera: en la Sección 2 recordamos algunos conceptos básicos para el desarrollo del trabajo. Posteriormente, en la Sección 3, presentamos el algoritmo para comparar imágenes de color segmentadas. Los resultados obtenidos con nuestra propuesta y la comparación que realizamos con otras aproximaciones de la literatura los mostramos en la Sección 4. Finalizamos con algunas conclusiones y líneas de trabajo futuro en la Sección 5.

2. Preliminares

En esta sección recordamos algunos conceptos y resultados de la teoría difusa que son las bases de nuestra propuesta para comparar imágenes de color segmentadas.

Definición 1 *[16] Un conjunto difuso A sobre un universo finito U es una función $U \to [0,1]$.*

Denotamos con $\mathcal{FS}(U)$ al conjunto de todos los conjuntos difusos sobre U y $Card(A)$ al cardinal del conjunto difuso $A \in \mathcal{FS}(U)$.

Sabemos que en la teoría de conjuntos difusos una función $c : [0,1] \to [0,1]$ tal que $c(0) = 1$, $c(1) = 0$, estrictamente decreciente y continua se denota como

negación estricta. Si además, dicha negación c es involutiva, entonces se dice que es una *negación fuerte*.

Bustince et *al.* [3] definen el concepto de Función de Equivalencia Restringida (*Resctricted Equivalence Function, REF*). En dicho trabajo, los autores también presentan diferentes métodos de construcción de *REFs* a partir de automorfismos y de operadores de implicación. En [4] el concepto de *REF* es aplicado para el cálculo de umbral de una imagen dada en escala de grises.

Orduna et *al.* [12] abordan el problema de la segmentación de imágenes en color transformándolo en un problema de *Toma de Decisión*. Los autores consideran un conjunto de expertos de modo que cada uno de ellos, mediante *REFs*, asigna un grado de preferencia de cada píxel a cada objeto de la imagen. Además, también tienen en cuenta la ignorancia asociada al experto al realizar tales asignaciones [5]. Por último, representan los objetos por medio de etiquetas lingüísticas difusas y utilizando el modelo de toma de decisión basado en 2-tuplas [8], clasifican cada píxel.

Definición 2 *[3] Una función* $REF : [0,1]^2 \to [0,1]$ *se llama función de equivalencia restringida asociada a una negación fuerte* c*, si satisface las siguientes condiciones*
1) $REF(x,y) = REF(y,x)$ *para todo* $x, y \in [0,1]$*;*
2) $REF(x,y) = 1$ *si y solo si* $x = y$*;*
3) $REF(x,y) = 0$ *si y solo si* $x = 1$ *e* $y = 0$ *o* $x = 0$ *e* $y = 1$*;*
4) $REF(x,y) = REF(c(x), c(y))$ *para todo* $x, y \in [0,1]$*;*
5) *Para todo* $x, y, z \in [0,1]$*, si* $x \leq y \leq z$*, entonces* $REF(x,y) \geq REF(x,z)$ *y* $REF(y,z) \geq REF(x,z)$*.*

A continuación, recordamos un método de construcción de *REFs* a partir de automorfismos en $[0,1]$, para ello previamente damos la definición de automorfismo.

Definición 3 *Una función continua y estrictamente creciente* $\varphi : [a,b] \to [a,b]$ *tal que* $\varphi(a) = a$ *y* $\varphi(b) = b$ *se denomina automorfismo en* $[a,b] \in \mathbb{R}$*.*

Proposición 1 *[3] Si* φ_1, φ_2 *son dos automorfismos del intervalo unidad entonces,* $REF(x,y) = \varphi_1^{-1}(1 - |\varphi_2(x) - \varphi_2(y)|)$ *con* $c(x) = \varphi_2^{-1}(1 - \varphi_2(x))$ *es una función de equivalencia restringida.*

Ejemplo 1 *Sea* $\varphi_1(x) = \varphi_2(x) = x$*, entonces* $REF(x,y) = 1 - |x - y|$ *es una función de equivalencia restringida.*

A continuación, recordamos el método de construcción de las medidas de similitud descrito en [3].

Proposición 2 *[3] Sea* $\mathcal{M} : [0,1]^n \to [0,1]$ *tal que satisface:*
(A1) $\mathcal{M}(x_1, \ldots, x_n) = 0$ *si y solo si* $x_1 = \cdots = x_n = 0$*,*
(A2) $\mathcal{M}(x_1, \ldots, x_n) = 1$ *si y solo si* $x_1 = \cdots = x_n = 1$*,*
(A3) $\mathcal{M}$ *es no-decreciente;*

Sea $REF : [0,1]^2 \to [0,1]$ *una función de equivalencia restringida. Bajo estas condiciones* $SM : FS(U) \times FS(U) \to [0,1]$*, dada por*

$$SM(A,B) = \underset{i=1}{\overset{n}{\mathcal{M}}} REF(A(u_i), B(u_i)), \tag{1}$$

satisface los siguientes items:
(i) $SM(A,B) = SM(B,A)$*, para todo* $A, B \in \mathcal{F}S(U)$*;*
(ii) $SM(A, A_c) = 0$*, para todo conjunto no difuso* A*;*
(iii) $SM(A,B) = 1$*, si y solo si* $A = B$*;*
(iv) Si $A \leq B \leq C$ *entonces,* $SM(A,B) \geq SM(A,C)$ *y* $SM(C,B) \geq SM(C,A)$*.*
(iv) $SM(A_c, B_c) = SM(A,B)$

Ejemplo 2 *Si tomamos la agregación* $\mathcal{M}$ *la media aritmética y* $REF(x,y) = 1 - |x-y|$ *entonces por la Proposición 2 tenemos que para* $A, B \in FS(U)$

$$SM(A,B) = \frac{1}{n}\sum_{i=1}^{n} 1 - |A(u_i) - B(u_i)|. \tag{2}$$

Cuando la Ec. (2) es utilizada para realizar la comparación global de dos imágenes se denomina *medida de similitud basada en la mejora del contraste* [6].

3. Un nuevo enfoque para comparar imágenes de color segmentadas

En la literatura existen diferentes métodos para segmentar imágenes en color ([1,7,10,11]). Evidentemente, un factor clave es establecer una medida que permita evaluar la calidad de los resultados obtenidos mediante los diferentes enfoques. Además de una comparación visual es necesario llevar a cabo una comparación cuantitativa. Los métodos más intuitivos para la evaluación de imágenes segmentadas y, *a priori* los más objetivos, son los que comparan dichas imágenes con una segmentación ideal.

Esta segmentación ideal se realiza a mano por un experto pero, dependiendo del contexto de la aplicación, los expertos pueden realizar diferentes segmentaciones. Uno de los criterios que dependen de la finalidad de la segmentación es el número de clases, pero también los resultados varían si los expertos consideran características como el color, la forma o la textura de los objetos.

En este trabajo presentamos un nuevo algoritmo para la comparación de imágenes en color segmentadas. Dicho algoritmo sólo se basa en el color de las zonas segmentadas. En este sentido, supongamos una imagen con dos objetos de color muy similar (ver los dos caballos de la Fig.1), de tal forma que en la segmentación ideal dada por un experto los etiqueta como dos objetos diferentes (dos caballos) y un algoritmo de segmentación los etiqueta como un único objeto (caballo). En nuestra propuesta no penalizamos el hecho de que ambos objetos no estén clasificados como dos zonas distintas, ya que el color de ambos objetos

Figura 1. Dos caballos: segmentación ideal son dos objetos y algoritmo de segmentación es un objeto

es muy similar. Por tanto, como ambos objetos tienen una característica (color) similar, pueden ser considerados como una misma zona de la imagen.

Nuestra propuesta para comparar imágenes segmentadas está basada en el concepto de *Color Representativo* de cada objeto. Este color lo calculamos como la media en cada componente (R, G, B) de todos los pixeles etiquetados como dicho objeto. De esta forma, asignamos a cada píxel su correspondiente *Color Representativo*. Proponemos un nuevo índice llamado *Índice de Comparación Total*, IC_{Total}, que mide cómo de similares son dos imágenes segmentadas. Este índice se calcula como la diferencia del *Color Representativo* de cada píxel en ambas imágenes.

3.1. Algoritmo de Comparación

Sea Q una imagen en el espacio de color RGB; de tal forma que $Q = (Q_R, Q_G, Q_B)$ donde Q_R, Q_G, Q_B son las componentes roja, verde y azul, respectivamente. De este modo, una imagen en color de $N \times M$ píxeles es una colección de $N \times M \times 3$ elementos dispuestos en filas y columnas. Para cada componente, cada píxel tiene asignado un valor numérico en $\{0, 1, \cdots, L-1\}$, que representa su intensidad. Para nosotros, cada componente Q_i, con $i \in \{R, G, B\}$, está representado mediante un conjunto difuso, donde el grado de pertenecia de cada elemento es la intensidad del pixel normalizada. Por tanto, $Q \in \mathcal{F}S(U)^3$.

El esquema de nuestra propuesta de algoritmo de comparación viene dado mediante el Algoritmo 1. Está dividido en tres pasos principales. El primer paso consiste en la creación de dos nuevas imágenes asociadas con las dos imágenes segmentadas. En ambas imágenes, los píxeles de cada zona se etiquetan con su *Color Representativo*. Basándonos en estas nuevas imágenes calculamos, para cada componente de color, su *Índice de Comparación*, que mide la similitud entre ambas imágenes en cada componente. Por último, agregamos con la media aritmética los tres índices de comparación, obteniendo el *Índice de Comparación Total*, que es el valor final devuelto por el Algoritmo 1. Si ambas imágenes son iguales, el índice toma valor 1, es decir, el valor máximo. Es necesario señalar que el número de clases de ambas imágenes segmentadas para comparar puede ser diferente. Incluso la imagen original puede ser interpretada como un caso especial de imagen segmentada.

Algorithm 1 Algoritmo de Comparación

Entrada: Una imagen original, $OI \in \mathcal{F}S(U)^3$, y dos imágenes segmentadas de la original $SI_i \in \mathcal{F}S(U)^3$ con $i \in \{1, 2\}$.

Salida: El Índice de Comparación Total, IC_{Total}, entre SI_1 e SI_2.

1: **Para** cada SI_i **hacer**
2: **Para** cada *clase* $z_j \in SI_i$ **hacer**
3: Calcular CR_j (*Color Representativo*):

$$CR_j(z_j) = \left(\frac{\sum\limits_{x \in z_j} q_R^{OI}(x)}{Card(x)}, \frac{\sum\limits_{x \in z_j} q_G^{OI}(x)}{Card(x)}, \frac{\sum\limits_{x \in z_j} q_B^{OI}(x)}{Card(x)} \right)$$

siendo $q_C^{OI}(x)$ la intensidad de los píxeles, x, en OI con $C \in \{R, G, B\}$.
4: **fin Para**
5: Construir una *nueva imagen*, NI_i, asignando a cada pixel el color representativo de la clase a la cual pertenece.
6: **fin Para**
7: Calcular el IC (*Índice de Comparación*) de NI_1 and NI_2, para cada componente, utilizando una medida de similitud (Eq. (1) siendo $\mathcal{M}$ la media aritmética).

$$IC_R(NI_{1R}, NI_{2R}) = \frac{\sum\limits_{l} REF(q_R^{NI_1}(x), q_R^{NI_2}(x))}{Card(x)}, \tag{3}$$

$$IC_G(NI_{1G}, NI_{2G}) = \frac{\sum\limits_{l} REF(q_G^{NI_1}(x), q_G^{NI_2}(x))}{Card(x)}, \tag{4}$$

$$IC_B(NI_{1B}, NI_{2B}) = \frac{\sum\limits_{l} REF(q_B^{NI_1}(x), q_B^{NI_2}(x))}{Card(x)}, \tag{5}$$

siendo $q_C^{NI_i}(x)$ la intensidad de los píxeles, x, de $NI_i \in \{NI_1, NI_2\}$ con $C \in \{R, G, B\}$.
8: Calcular IC_{Total} (*Índice de Comparación Total*) agregando el índice de comparación de cada componente

$$IC_{Total} = \frac{1}{3} \cdot (IC_R(NI_{1R}, NI_{2R}) + IC_G(NI_{1G}, NI_{2G}) + IC_B(NI_{1B}, NI_{2B})). \tag{6}$$

4. Resultados experimentales

En este experimento hemos utilizado 20 imágenes de la base de datos de Berkeley (BSDS) [9]. Todas estas imágenes tienen una resolución de 321×481 o 434×291 píxeles y están en el espacio de color RGB. Esta base de datos también contiene diferentes segmentaciones ideales para cada imagen.

Para obtener las imágenes segmentadas hemos utilizado tres aproximaciones diferentes: el clásico Fuzzy C-Means (FCM) [1], la conocida estimación máxima a posteriori, (MAP) y la estimación máxima de probabilidad (ML) [7] y el modelo basado en 2-tuplas (2-TUP) [12]. También hemos fijado el número de zonas en las que segmentamos las imágenes con los tres algoritmos. En el modelo de 2-TUP los píxeles son siempre clasificados en seis zonas. Por tanto, hemos ajustado los parámetros del algoritmo FCM también para seis zonas, y el MAP-ML comienza por seis zonas, en lugar de diez, como indica en su ejecución por defecto. Las imágenes ideales han sido seleccionadas con un número de zonas mayor que dos y menor que seis (si es posible). De esta manera, nos aseguramos de que los métodos de segmentación estudiados no se encuentran en desventaja en comparación con la segmentación realizada por los seres humanos.

En la Fig. 2 mostramos las segmentaciones obtenidas considerando los tres enfoques indicados anteriormente de 10 imágenes. En la primera columna incluimos las imágenes originales con sus identificadores en BSDS, en la segunda columna observamos las imágenes ideales y en la tercera, cuarta y quinta columna mostramos las segmentaciones obtenidas con los métodos de 2-TUP, FCM y MAP-ML, respectivamente.

En este experimento pretendemos cuantificar el rendimiento de nuestro enfoque (Algoritmo 1), utilizando $REF(x, y) = 1 - |x - y|$ en las expresiones (3)-(5), para la comparación de imágenes segmentadas en color. Para ello, comparamos nuestros resultados con los obtenidos por uno de los indicadores más utilizados, el índice probabilístico de Rand (Probabilistic Rand Index).

4.1. Índice probabilístico de Rand

El índice Rand [14] (ver también [9]) es comúnmente utilizado para la evaluación de imágenes segmentadas cuando se comparan con una imagen ideal (segmentación hecha a mano). La idea principal de esta medida consiste en contar el número de pares de píxeles que tienen un etiquetado coherente en los dos segmentaciones. Es decir, el número de pares de píxeles que en la imagen ideal y en las imágenes a evaluar están etiquetados como el mismo objeto más el número de pares de píxeles que en las imágenes ideales y en las imágenes a evaluar están etiquetados como objetos diferentes, dividido por el número total de pares de píxeles de la imagen. El índice Rand, IR, viene dado por la siguiente expresión:

$$IR(Ev, Id) = \frac{1}{\binom{N \times M}{2}} \sum_{\substack{i,j \\ i \neq j}} [I(l_{ev}(i) = l_{ev}(j) \wedge l_{id}(i) = l_{id}(j)) + I(l_{ev}(i) \neq l_{ev}(j) \wedge l_{id}(i) \neq l_{id}(j))], \quad (7)$$

donde Ev es la imagen evaluada, Id es la imagen ideal, I es la función identidad, $M \times N$ es el número de píxeles en la imagen, $l_{ev}(i)$ representa el pixel etiquetado en Ev y $l_{id}(i)$ representa el pixel etiquetado en Id.

El IR toma valores desde 0 (cuando las dos segmentaciones no tienen similitudes) hasta 1 (cuando las dos segmentaciones son idénticas).

4.2. Resultados obtenidos

Comparamos las imágenes segmentadas, que hemos obtenido con los métodos de 2-TUP, FCM y MAP-ML, con respecto a las imágenes ideales (imágenes segmentadas por humanos) utilizando el Algoritmo 1 (ver el Cuadro 1 (a)), y el índice Rand (ver el Cuadro 1 (b)). En el Algoritmo 1 hemos tomado la Ec. (2) para calcular el Índice de Comparación de cada componente. En ambos cuadros, resaltamos en negrita el mejor resultado para cada imagen.

Como podemos ver, nuestra propuesta y el IR obtienen resultados muy diferentes cuando comparamos las mismas imágenes segmentadas. Como hemos comentado previamente, nuestro Algoritmo 1 está basado exclusivamente en el color de las zonas creadas, mientras que el IR está basado en si cada par de

Imagen	2-TUP	FCM	MAP-ML
(124084)	0.9251	0.9058	**0.9522**
(260058)	**0.9625**	0.9189	0.9496
(295087)	0.9389	0.9035	**0.9405**
(299086)	**0.9770**	0.9493	**0.9770**
(161062)	**0.9719**	0.9613	0.9627
(207056)	0.9692	0.9526	**0.9724**
(374067)	0.9662	0.9514	**0.9782**
(67079)	0.9429	0.9015	**0.9594**
(100075)	**0.9575**	0.8930	0.9374
(58060)	0.9460	0.9043	**0.9606**
(216066)	0.9275	0.8931	**0.9359**
(291000)	**0.9391**	0.8743	0.9374
(80099)	0.9868	0.9749	**0.9930**
(253036)	0.9464	0.9377	**0.9601**
(100080)	0.9268	**0.9529**	0.9266
(197017)	0.8503	0.9285	**0.9594**
(46076)	0.9239	0.9344	**0.9443**
(118035)	**0.9386**	0.9244	0.93774
(126007)	**0.9434**	0.9160	0.9310
(69015)	0.9333	0.9046	**0.9436**

(a)

Imagen	2-TUP	FCM	MAP-ML
(124084)	0.7152	0.7134	**0.7600**
(260058)	**0.8063**	0.6551	0.7461
(295087)	**0.8269**	0.7775	0.7995
(299086)	0.8389	0.7718	**0.866**
(161062)	0.8836	0.7197	**0.9162**
(207056)	**0.6453**	0.5656	0.6343
(374067)	**0.8633**	0.8011	0.7559
(67079)	0.7918	**0.8109**	0.7646
(100075)	0.7048	0.7446	**0.7727**
(58060)	0.8112	0.7319	**0.8333**
(216066)	0.7630	0.7127	**0.7787**
(291000)	0.7675	0.6981	**0.7713**
(80099)	**0.6192**	0.5864	0.4855
(253036)	**0.7754**	0.7561	0.7481
(100080)	0.7598	**0.7768**	0.775
(197017)	0.7115	0.7627	**0.8518**
(46076)	0.8430	**0.8578**	0.8305
(118035)	0.8807	0.8155	**0.8985**
(126007)	0.7873	**0.8621**	0.8025
(69015)	**0.7667**	0.6997	0.6798

(b)

Cuadro 1. (a) Índice de Comparación Total (b) Índice Rand entre la imagen ideal y las segmentaciones obtenidas con los métodos de 2-TUP (segundas columnas), FCM (terceras columnas) y MAP-ML(cuartas columnas).

píxeles están situados en la misma zona o en diferente zona en ambas imágenes segmentadas.

Por ejemplo, si analizamos la imagen (374067), observamos que nuestro algoritmo asegura que la mejor segmentación es la realizada por MAP-ML mientras el IR asegura que la mejor segmentación se obtiene con el método de 2-TUP. La imagen ideal de (374067) está dividida en más de diez áreas, separando zonas de la hierba, grupos de árboles, cada persona del camino, etc. La segmentación obtenida con 2-TUP distingue diferentes áreas en la hierba pero no separa el área de los árboles. Por otro lado, la segmentación obtenida con MAP-ML separa solo grandes áreas: toda la hierba es una misma área. Si analizamos solo el color, la hierba tiene un color muy similar, por lo que el error de segmentar toda la hierba en una zona única no es muy grande. En el mismo sentido, aunque el camino se divide en tres partes, todos sus colores representativos son bastante similares, por lo que la diferencia entre el color de la carretera y el color en la imagen ideal son muy bajos.

5. Conclusiones y líneas futuras

En este trabajo hemos presentado un algoritmo para la evaluación de imágenes en color segmentadas. La metodología propuesta consiste en comparar la segmentación de imágenes obtenidas por cualquier algoritmo con la segmentación ideal (hecha a mano). A diferencia de los métodos de evaluación existentes en la literatura que comparan si los objetos en ambas imágenes se componen de los mismos píxeles, nuestra propuesta mide la similitud entre el color promedio de los objetos de dichas imágenes. Este trabajo es el comienzo de utilizar nuevas medidas de comparación. En un futuro próximo consideramos necesario

analizar el impacto de aplicar estas medidas pero considerando otros espacios de color así como otras funciones de agregación y de equivalencia restringida para el cálculo del índice de comparación total.

Agradecimientos

Este trabajo ha sido subvencionado por el Servicio de Investigación de la UPNa y por los proyectos TIN2010-15055 y TIN2011-29520.

Referencias

1. J.C. Bezdek, R. Ehrlich and W. Full, FCM - The Fuzzy C-Means Clustering-Algorithm, *Computers and Geosciences*, 10:191–203, 1984.
2. J. C. Bezdek, J. Keller, R. Krisnapuram and N.R. Pal, *Fuzzy Models and algorithms for pattern recognition and image processing*, Kluwer, Boston, 1999.
3. H. Bustince, E. Barrenechea and M. Pagola, Restricted equivalence functions, *Fuzzy Sets and Systems*, 157:2333–2346, 2006.
4. H. Bustince, E. Barrenechea and M. Pagola, Image thresholding using restricted equivalence functions and maximizing the measures of similarity, *Fuzzy Sets and Systems*, 158:496–516, 2007.
   to edge detection, *Fuzzy Sets and Systems*, 160:1819–1840, 2009.
5. H. Bustince, M. Pagola, E. Barrenechea, J. Fernandez, P. Melo-Pinto, P. Couto, H. R. Tizhoosh, and J. Montero, Ignorance functions. An application to the calculation of the threshold in prostate ultrasound images, *Fuzzy Sets and Systems*, 161:20–36, 2010.
6. T. Chaira, A.K. Ray, Fuzzy measures for color image retrieval, *Fuzzy Sets and Systems*, 150:545–560, 2005.
7. S. Chen, L. Cao, Y. Wang, J. Liu and X. Tang, Image segmentation by MAP-ML estimations, *IEEE Transactions on Image Processing*, 19:2254–2264, 2010.
8. F. Herrera and L. Martínez, A 2-tuple fuzzy linguistic representation model for computing with words, *IEEE Transactions on Fuzzy Systems*, 8:746–752, 2000.
9. D. Martin, C. Fowlkes, D. Tal and J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, *proceedings of the 8th International Conference on Computer Vision* (Iccv 2001), pages 416–423, Vanvouver (Canada) 2001.
10. E. Navon, O. Miller and A. Averbuch, Color image segmentation based on adaptive local thresholds, *Image and Vision Computing*, 23:69–85, 2005.
11. M. Ozden, E. Polat, A color image segmentation approach for content-based image retrieval, *Pattern Analysis*, 43:3219–3232, 2007.
12. R. Orduna, Overlap functions and their application to image segmentation, *Ph. D. Thesis*, Universidad Publica de Navarra, 2010.
13. M.S. Prewitt, Object enhancement and extraction, A. Rosenfeld, B.S. Lipkin (Editors), *Picture Processing and Psychophysics*, Academic Press, NewYork, 1970.
14. W. M. Rand, Objective criteria for the evaluation of clustering methods, *Journal of the American Statistical Association*, 66:846–850, 1971.
15. R. Unnikrishnan, C. Pantofaru, and M. Hebert, Toward objective evaluation of image segmentation algorithms, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29: 929–944, 2007.
16. L.A. Zadeh, Fuzzy sets, *Information Control*, 8:338–353, 1965.

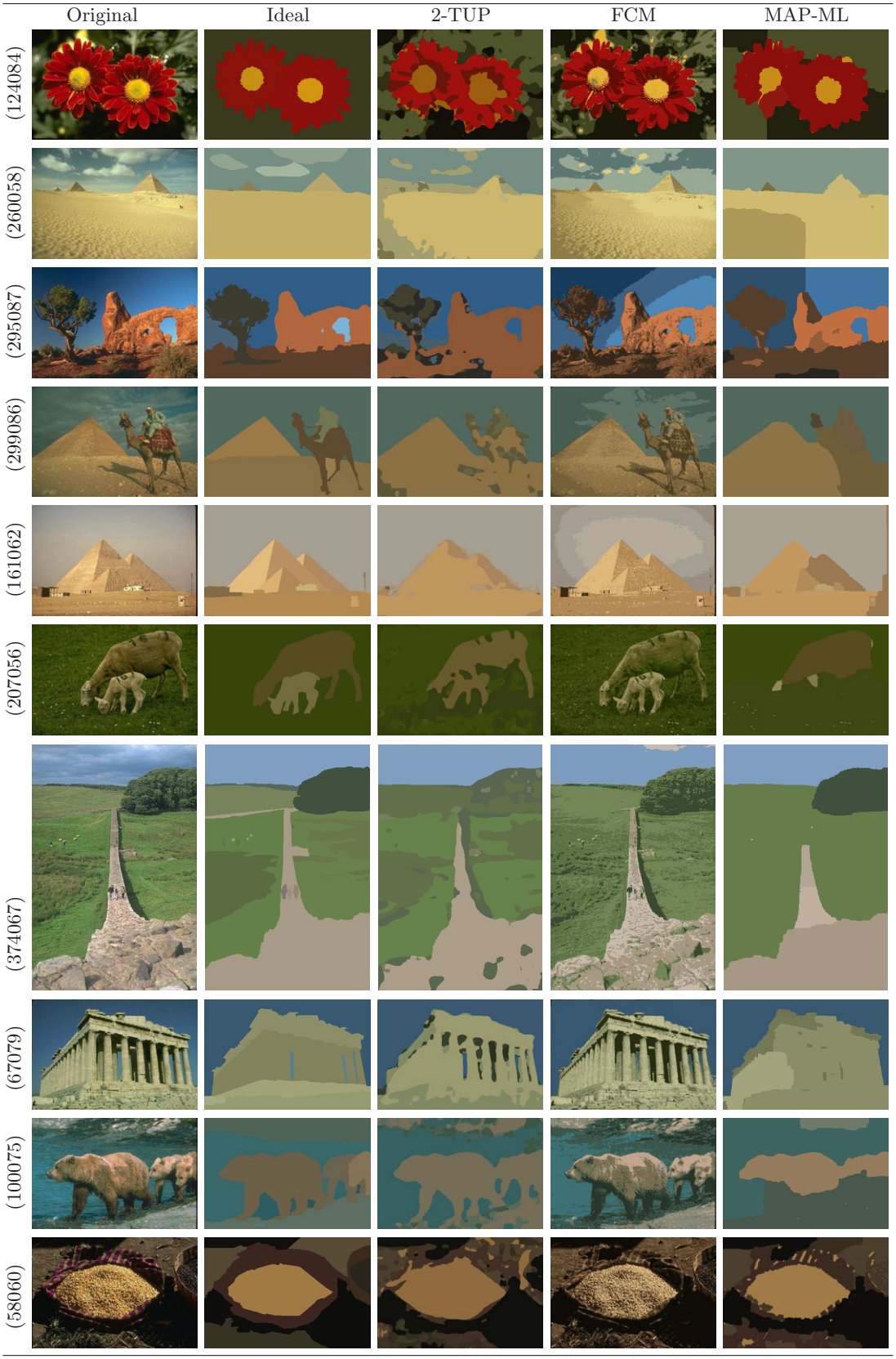


Figura 2. Imágenes originales (primera columna), imágenes ideales (segunda columna) y las imágenes segmentadas con tres métodos diferentes, 2-TUP (tercera columna), FCM (cuarta columna) y MAP-ML (quinta columna).

Fuzzy Delphi method and fuzzy Spearman´s rank correlation coefficient

Raquel Caro[1], Luis Garmendia[2], Ramón González del Campo[2]
[1] Universidad Pontificia de Comillas, Madrid
rcaro@doi.icai.upcomillas.es
[2] Universidad Complutense de Madrid
lgarmend@fdi.ucm.es, rgonzale@estad.ucm.es

Abstract. The Delphi method is an experts consulting method using tests that can be used many times until most of the experts agree. The Spearman´s rank correlation coefficient is a measure of the degree of agreement between referees. This paper defines a fuzzy Delphi method and a fuzzy generalization of the Spearman´s rank correlation coefficient that can be used to make the halting decision to stop the iterative process in a forecasting exercise with the fuzzy Delphi method based on expert opinions.

Keywords: Delphi Method, the Spearman´s rank correlation coefficient, experts, fuzzy sets, strong ranking, fuzzy assignments.

1 Introduction

In decision making sometimes it is needed to ask experts to choose between a number of alternatives or to rank them in a preference relation. Forecasting is designed to help decision making and planning in the present enabling to modify variables now to alter (or be prepared for) the future.

One kinds of questions that frequently are asked to the experts in the applications of the Delphi method in forecasting is ranking, in order of importance, a group of ***n*** items concerning to the future evolution of a system or phenomenon and that they can be of a very different type (options, factors, alternatives, etc.). The experts are required to provide a *strong* ranking, where different items must be associated to different ranks, so the n items are ranked from 1 to n, in order of priority of importance (1 for the highest priority, n for the lowest priority) in the opinion of every reviewer.

The inspiration for developing a fuzzy Delphi method arose when the authors were developing a prospective analysis, based on the use of the Delphi Method, related to the implementation of The Bologna Declaration[1], whose objective is the complete homologation of the university education programs and its assessment across all EU countries. When some colleagues considered as experts on this subject were asked to prioritize a group of alternatives, sometimes they had difficulties to

[1] The Bologna Declaration is a common initiative, approved in June 19th, 1999 by the Ministers of Education of the different countries integrated in the European Union, to implement a homogeneous program in the higher education area, to homologate across the EU countries similar university degrees.

present a *strong* hierarchy. In many cases, the experts do not wish, not know, or simply, they cannot rank the *n* items in such a *strong* way.

The fuzzy Delphi method approach proposed in this paper allows the experts to make a more flexible ranking of the items through fuzzy assignments. However, under these circumstances, it is necessary to adapt and generalize to the fuzzy case, generalizing the way the level of consensus between different referees is measured, to allow a method to decide when the Delphi method tests process must be stopped. This must be done when the results are stabilized and the issue is to have an effective way to evaluate the level of stabilization, which can be done using an a extension of the Spearman´s rank correlation coefficient defined to measure the "proximity" between two fuzzy assignments of experts questioned in a multi-stage survey to decide to continue or to stop the iterative process of the Delphi method.

The decisions made by the experts rely on their individual experience and the need of fuzzy techniques are incorporated in this paper since conventional analytic hierarchy process is unable to reflect human thinking in cases with uncertainty, imprecision or lack of information or opinions.

2 Preliminaries

Fuzzy sets theory (Zadeh, 1965; Zadeh, 1987), are an extension of the classical set theory. A fuzzy set A: X→[0;1] on a universe *X* is defined as:

$$A = \{x, \mu_A(x)) \mid x \in X, \mu_A(x) \in [0,1]\}$$

where $\mu_A(x)$ is the membership degree of an element x of the universe.

2.1 The Delphi method

The classical Delphi is introduced by Kaufman and Gupta (1988) as an experts consulting method using tests that can be used many times until the experts agree.
The Delphi method is perhaps the most-known group judgment technique. In 1960, Helmer and Rescher published the classic paper on "The Epistemology of the Inexact Sciences", which argued that in fields that have not yet developed to the point of having scientific laws, the testimony of experts is permissible. The problem however was how to use this testimony and, specifically, how to combine the testimony of a number of experts into a single useful statement. The basic notion of the Delphi method, theoretical assumptions and methodological procedures developed in the 1950s and 1960s at the RAND Corporation, at Santa Monica, California (Dalkey and Helmer, 1962; Dalkey, 1969).

The Delphi method is actually a structured process for the systematic collection and collation of judgments from a group of experts on a particular issue, by means of a series of questionnaires interspersed with controlled opinion feedback (Adler and Ziglio, 1996). The experts are requested to give their opinion separately and independently about the variables in question. The results of the first round are analyzed statistically by finding their average and are then interspersed to the participants, who are asked if they wish to revise their earlier estimates. This process is followed again and again until the outcome converges to a reasonable solution

from the point of view of the decision maker. More specific, Fowles (1978) describes the following steps for the Delphi method.

There have been several studies supporting the Delphi method, which indicated high agreement between the Delphi estimates and the real numbers (e.g. Ament, 1970; Milkovich et al., 1972; Basu & Schroeder, 1977). On the other hand, the Delphi method has been criticized (Sackman, 1974; Makridakis and Wheelwright, 1978) for dependency of forecasts on the particular judges selected, the sensitivity of results to ambiguity in the questionnaire and the difficulty in assessing the degree of expertise incorporated into the forecast.

Definition 2.1

A strong hierarchy test asks an expert or reviewer to rank a number of items from 1 to n depending on their preference or priority of importance (1 for the highest priority, n for the lowest priority).

The Table 1 shows an example of a strong hierarchy test answered by two reviewers that give a rank of a set of items.

Referee (***h*** = 1)						
	r_i^1		Rank ***k***			
			k = 1	***k*** = 2	***k*** = 3	***k*** = 4
	Item ***i***	***i*** = 1		*item* **1**		
		i = 2	*item* **2**			
		i = 3				*item* **3**
		i = 4			*item* **4**	
Referee (***h*** = 2)						
	r_i^2		Rank ***k***			
			k = 1	***k*** = 2	***k*** = 3	***k*** = 4
	Item ***i***	***i*** = 1	*item* **1**			
		i = 2				*item* **2**
		i = 3		*item* **3**		
		i = 4			*item* **4**	

Table 1- Strong ranking of 4 items made by 2 referees

The ranks or positions that the two referees have assigned to the i^{th} item are denoted as r_i^1 and r_i^2, respectively.

2.1.2 A strong hierarchy test as a crisp relation

A strong hierarchy test can be also defined as a crisp relation $R:I\times K\rightarrow\{0, 1\}$ where I is a set of n items, $I = \{i_1, .., i_n\}$ and K is a set of n ranks, $K = \{r_1, r_n\}$, and having the restriction of having only one '1' in every row and column.

Example 2.1

Let R^h be the strong hierarchy test filled by an expert h.

The crisp relations containing the strong hierarchy tests of table 1 are the following:

$$R^1 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \qquad R^2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

The Spearman´s rank correlation coefficient is a measure of the degree of agreement between referees that can be used to make the stop decision of the Delphi method.

2.2 The classical Spearman´s coefficient

The Spearman´s rank correlation coefficient, denoted as ***SC***, allows to analyze the degree of consensus or agreement reached in the opinions of two referees, who must classify ***n*** items in order of priority, by assigning them ranks between **1** and ***n*** (see Table 1).

Definition 2.1

Given a strong hierarchy test filled by two reviewers, the Spearman´s rank correlation coefficient is the following expression

$$[1] \qquad SC = 1 - \frac{6\sum_{i=1}^{n}\left(r_i^1 - r_i^2\right)^2}{n\left(n^2-1\right)} = 12\frac{\sum_{i=1}^{n}\left(r_i^1 - \frac{n+1}{2}\right)\left(r_i^2 - \frac{n+1}{2}\right)}{n\left(n^2-1\right)}$$

The minimum value -1 is reached if the referee 1 assigns to an item the rank ***r*** and the referee 2 assigns to the same item the rank $\boldsymbol{n-r+1}$, it is, when the two referees assign to each item symmetric ranks respect to the intermediate rank $\frac{\boldsymbol{(n+1)}}{\boldsymbol{2}}$, or in other words, when the summatory $\sum_{i=1}^{n}\left(r_i^1 - r_i^2\right)^2$ is equal to:

$$[2] \qquad \sum_{\boldsymbol{i=1}}^{\left[\frac{\boldsymbol{n}}{\boldsymbol{2}}\right]}\boldsymbol{(n-i)^2} = \frac{\boldsymbol{n\left(n^2-1\right)}}{\boldsymbol{3}}$$

where the function $[\ ]$ is the *integer part* function.

3 The fuzzy Delphi method

The fuzzy Delphi method is a generalization of the Delphi method. The difference is that the ***h*** referees now assign a membership degree in the interval [0, 1] to the association of every item with every rank.

Definition 3.1

Let I be a set of n items, $I = \{i_1, ..., i_n\}$.

Let K be a set of n ranks, $K = \{r_1, \ldots, r_n\}$

A fuzzy Delphi method test is a fuzzy relation R: $I \times K \rightarrow [0,1]$ that gives a degree of relation of every item i, and every possible rank ***k***.

Definition 3.2

A fuzzy Delphi method iteration is a set of h fuzzy Delphi method tests filled by a set of h different experts or reviewers. Therefore it can be implemented by h fuzzy relations R^h: $I \times K \rightarrow [0,1]$.

Notation

Lets denote $\mu_{i,k}^{h} \in [0,1]$ to the degree of confidence that the referee h assigns to the i^{th} item to the rank of importance ***k.***

So $\mu_{i,k}^{h} = R^h(i_i, r_k)$

Example 3.1

The following Table 2 provides an example of a fuzzy Delphi method iteration of a set of four items done by two referees.

Referee (***h*** = 1)						
	$\mu_{i,k}^{1}$		Rank ***k***			
			k = 1	***k*** = 2	***k*** = 3	***k*** = 4
	Item ***i***	***i*** = 1	**0.8**	**0.7**	**0.3**	**0**
		i = 2	**0.5**	**0.5**	**0.5**	**0.5**
		i = 3	**1**	**0.8**	**0.7**	**0**
		i = 4	**0.2**	**0.3**	**0.6**	**0.8**
Referee (***h*** = 2)						
	$\mu_{i,k}^{2}$		Rank ***k***			
			k = 1	***k*** = 2	***k*** = 3	***k*** = 4
	Item ***i***	***i*** = 1	**0**	**0.2**	**0.5**	**0.9**
		i = 2	**0**	**0**	**0.2**	**1**
		i = 3	**0**	**0.3**	**0.6**	**0.9**
		i = 4	**0.3**	**0.4**	**0.5**	**0.7**

Table 2- Fuzzy Delphi method test of 4 items filled by 2 referees

The following fuzzy relations are the corresponding ones to the tests given in table 2:

$$R^1 = \begin{pmatrix} 0.8 & 0.7 & 0.3 & 0 \\ 0.5 & 0.5 & 0.5 & 0.5 \\ 1 & 0.8 & 0.7 & 0 \\ 0.2 & 0.3 & 0.6 & 0.2 \end{pmatrix} \qquad R^2 = \begin{pmatrix} 0 & 0.2 & 0.5 & 0.9 \\ 0 & 0 & 0.2 & 1 \\ 0 & 0.3 & 0.6 & 0.9 \\ 0.3 & 0.4 & 0.5 & 0.7 \end{pmatrix}$$

4 The fuzzy Spearman´s coefficient

The ***extended Spearman´s rank correlation coefficient*** that can be useful to measure the discrepancy of two referees in the fuzzy Delphi method. It can be applied to measure the convenience to stop the iterative process of a fuzzy Delphi method.

The classical Spearman coefficient measures the discrepancy of two referees by computing the sum of squares of the differences. The fuzzy generalization associates the maximum difference with the maximum value of the sum of the square of the centroids for every item.

Definition 4.1

The centroid or gravity center of an item i is the following expression

[3] $$c_i^2 = \frac{\sum_{k=1}^{n} \mu_{i,k} (n-k)^2}{\sum_{h=1}^{n} \mu_{i,k}}$$

Definition 4.2

The ***extended Spearman´s rank correlation coefficient***, denoted ***ESC***, is defined by:

$$ESC = 1 - \frac{\sum_{i=1}^{n}\left(\frac{\sum_{k=1}^{n}\mu_{i,k}(n-k)^2}{\sum_{k=1}^{n}\mu_{i,k}}\sum_{k=1}^{n}\left|\mu_{i,k}^{1}-\mu_{i,k}^{2}\right|\right)}{\sum_{i=1}^{n}\left(\frac{\sum_{k=1}^{n}\mu'_{i,k}(n-k)^2}{\sum_{k=1}^{n}\mu'_{i,k}}\right)} \quad [4]$$

$$= 1 - \frac{\sum_{i=1}^{n} c_i^{2}\left(\sum_{k=1}^{n}\left|\mu_{i,k}^{(1}-\mu_{i,k}^{(2}\right|\right)}{\sum_{i=1}^{n} c'^{2}_{i}}$$

$$= 1 - \frac{\sum_{i=1}^{n} c_i^{2} M_i}{\sum_{i=1}^{n} c'^{2}_{i}} \quad [5]$$

with

$$M_i = Min\left\{2, \sum_{k=1}^{n}\left|\mu_{i,k}^{1}-\mu_{i,k}^{2}\right|\right\} \quad [6]$$

Example 4.1

With the data shown in the Table 2, the values of the centroid´s of the items are:

$$c_1^2 = \frac{10.5}{2.3} \quad ; \quad c_2^2 = \frac{7}{2} \quad ; \quad c_3^2 = \frac{12}{3} \quad ; \quad c_1^2 = \frac{4.9}{2}$$

$$c'^2_1 = \frac{12.8}{2.4} \quad ; \quad c'^2_2 = \frac{12.8}{2.2} \quad ; \quad c'^2_3 = \frac{13.4}{3.3} \quad ; \quad c'^2_1 = \frac{13.5}{2.8}$$

The quantities obtained and the value of M_i are:

$$\sum_{k=1}^{4}\left|\mu_{1,k}^{1}-\mu_{1,k}^{2}\right|=\left|0.8-0\right|+\left|0.7-0.2\right|+\left|0.3-0.5\right|+\left|0-0.9\right|=2.4$$

$$M_1=Min\left\{2,2.4\right\}=2$$

$$\sum_{k=1}^{4}\left|\mu_{2,k}^{1}-\mu_{2,k}^{2}\right|=\left|0.5-0\right|+\left|0.5-0\right|+\left|0.5-0.2\right|+\left|0.5-1\right|=1.8$$

$$M_2=Min\left\{2,1.8\right\}=1.8$$

$$\sum_{k=1}^{4}\left|\mu_{3,k}^{1}-\mu_{3,k}^{2}\right|=\left|1-0\right|+\left|0.8-0.3\right|+\left|0.7-0.6\right|+\left|0-0.9\right|=2.5$$

$$M_3=Min\left\{2,2.5\right\}=2$$

$$\sum_{k=1}^{4}\left|\mu_{4,k}^{1}-\mu_{4,k}^{2}\right|=\left|0.2-0.3\right|+\left|0.3-0.4\right|+\left|0.6-0.5\right|+\left|0.8-0.7\right|=0.4$$

$$M_4=Min\left\{2,0.4\right\}=0.4$$

So, with these data, the value of ESC is, with [15]:

$$\boldsymbol{ESC=1-\frac{\frac{10.5}{2.3}\times 2+\frac{7}{2}\times 1.8+\frac{12}{3}\times 2+\frac{4.9}{2}\times 0.4}{\frac{12.8}{2.4}+\frac{12.8}{2.2}+\frac{13.4}{3.3}+\frac{13.5}{2.8}}=-0.22}$$

A value **-0.22** for the ***ESC*** supposes, certainly, a low degree of consensus between the two referees although it is difficult to evaluate it in a qualitative way. In the example, this can be explained because they show discrepancies concerning all the items.

3 Conclusions

The fuzzy Delphi method can be useful when the experts have problems filling a nitide ranking of a set of items, maybe because of uncertain information, lack of precision or lack of information. As a difference with the classical *strong* method, the referees or judges are not constrained to give different ranks to different items. It allows filling the tests giving a membership degree between every item and every rank, which can be modeled into a fuzzy relation, without the restrictions of the classical Delphi method.
This paper introduces the ***extended Spearman´s rank correlation coefficient*** that can be useful to measure the discrepancy of two referees. So it can be applied to measure the convenience to stop the iterative process of a fuzzy Delphi method.
The classical Spearman coefficient measures the discrepancy of two referees by computing the sum of squares of the differences. The fuzzy generalization associates the maximum difference with the maximum value of the sum of the square of the defined centroids for every item.
The needed extensions are defined and some examples are given.

References

1. Bellman, R.E., Zadeh, L.A., 1970. *Decision-making in a Fuzzy Environment*. Management Science, vol. 17, nº 4, pp. 141-164.A
2. Bortolan, G. and Degani, R., A review of some methods for ranking fuzzy subsets, Fuzzy Sets and Systems, 15, pp. 1-19, 1985.
3. F.J. Cabrerizo, J.M. Moreno, I.J. Pérez, E. Herrera-Viedma, Analyzing consensus approaches in fuzzy group decision making: advantages and drawbacks. Soft Computing 14:5 (2010) 451-463
4. Fodor, J., Roubens, M.,1994. *Fuzzy Preference Modelling and Multicriteria Decision Support*. Kluwer Academic Publishers, Dordrecht, The Nederlands.
5. González, A. Study of the ranking function approach through mean values, Fuzzy Sets and Systems, vol. 35, n.1, pp. 29-41, (1990).
6. Kaufman, A. and Gupta, M.M. (1988). Fuzzy Mathematical Models in Engineering and Management Sciences. North-Holland, Amsterdam.
7. Landeta, J., 1999. *El Método Delphi*. Editorial Ariel.
8. Rowe,G., Wright, G., 2001. *Expert opinions in forecasting: The role of the Delphi technique.* In: Armstrong, J.S. (Ed.), *Principles of Forecasting*. Boston, Kluwer Academic Press, pp 125-144.
9. Sackman, H (1974), Delphi Assessment: Expert Opinion, Forecasting, and Group Process.The RAND Corporation, Report #R-1283-PR, Santa Monica, California, USA.
10. Yager, R.R., A procedure for ordering fuzzy subsets of the unit interval, Information Sciences, 24 , 143-161, 1981.
11. Zadeh, L.A., 1965. *Fuzzy Sets*. Information and Control 8, pp. 338-353.
12. Zadeh, L. (1987). Fuzzy Sets and Applications: Selected Papers. John Wiley & Sons, New York.
13. Zimmermann, H.J., 1987. *Fuzzy Sets, Decision Making and Expert Systems*. Kluwer Academic Publishers. Boston.

Recursive Measures of Specificity

Luis Garmendia, Ramon Gonzalez del Campo, and Ronald R. Yager

Universidad Complutense de Madrid, Spain
Iona College, USA
rgonzale@estad.ucm.es,
lgarmend@fdi.ucm.es,
yager@panix.com

Abstract. The recursive measures of specificity are introduced to compute measures of specificity of fuzzy sets when the universe of candidates is increased with an element with any membership degree using the previously computed measure and taking a constant bounded number of operations, and so, without sorting again all the membership degrees.

Keywords: Measure of specificity, fuzzy sets

1 Introduction

Measures of specificity [4–6, 8–12] are useful measures of information for fuzzy sets to know how useful they are to choose an element from the universe. They have also been defined for interval-valued fuzzy sets, intuitionistic fuzzy sets and type-2 fuzzy sets [7, 13, 14].

In real life, the universe of discourse can grow up. Yager's Expansible measures of specificity [15] do not change when a new candidate has been added to the universe of discourse with a membership degree zero in the expanded fuzzy set.

On other side, Montero et all.[1–3] introduced the recursive aggregation operators reducing the computational cost of aggregating $n + 1$ values when the aggregation of the first n values is already known.

This paper goal is to join the ideas for recursive aggregation operators and Yager's ideas on expansible measures of specificity to solve a performance problem of computing the specificity measure of a fuzzy set when the universe is increased with a new candidate with any membership degree, from a previous computed measure of specificity and using a constant bounded number of operations without a resort of all the membership degrees of the expanded fuzzy set.

2 Preliminaries

Definition 1 *A fuzzy set μ on $X = \{e_1, \ldots, e_n\}$ is normal if there exists an element e_i in X such that $\mu(e_i) = 1$.*

Definition 2 *A fuzzy set μ on $X = \{e_1, \ldots, e_n\}$ is a singleton if there exists an element e_i in X such that $\mu(e_i) = 1$ and $\mu(e_j) = 0$ for all elements e_j in X with $i \neq j$.*

Definition 3 *[8] Let $X = \{e_1, \ldots, e_n\}$ be a finite universe and let $[0,1]^X$ be the class of fuzzy sets on X. A measure of specificity Sp is a mapping $Sp : [0,1]^X \to [0,1]$ such that:*

- *$Sp(\mu) = 1$ if and only if μ is a singleton.*
- *$Sp(\emptyset) = 0$.*
- *If μ and ν are normal fuzzy sets and $\mu \subseteq \nu$, then $Sp(\mu) \geq Sp(\nu)$.*

Definition 4 *[12] Let Sp and Sp^* be two measures of specificity on X. Sp is stricter than Sp^* if for all fuzzy set A on X $Sp(A) \leq Sp^*(A)$*

Definition 5 *[8] The linear measure of specificity of a fuzzy set A on a finite space $X = \{e_1, \ldots, e_n\}$ is:*

$$Sp_{\overrightarrow{w}}(A) = a_1 - \sum_{j=2}^{n} w_j a_j$$

where a_j is the j^{th} greatest membership degree of A and $\{w_j\}$ is a set of weights verifying:

- *$w_j \in [0,1]$*
- *$\Sigma_{j=2}^{n} w_j = 1$*

Definition 6 *[15] The fractional measure of specificity is:*

$$Sp(A) = a_1^2 / \sum_{j=1}^{n} a_j$$

where a_j is the j^{th} greatest membership degree of A.

Definition 7 *[8] The product measure of specificity is:*

$$Sp(A) = a_1 \prod_{j=2}^{n} (k a_j + (1 - a_j))$$

where k is in [0,1].

Definition 8 *[15] Let A be an fuzzy set on $X = \{e_1, \ldots, e_n\}$ and let A' be an extended fuzzy set of A on $X' = \{e_1, \ldots, e_n, e_{n+1}\}$. A measure of specificity Sp is expansible if it satisfies the following condition:*

$$\text{If } A'(e_{n+1}) = 0 \text{ then } Sp(A) = Sp(A')$$

3 Recursive Measures of Specificity

Definition 9 *A linear measure of specificity is uniformly weighted if its weights are equal.*

Lemma 1 *Let Sp be a uniformly weighted measure of specificity on a finite space* $X = \{e_1, \ldots, e_n\}$. *Then* $\{w_j\} = 1/(n-1)$ *for all* $j : 2 \ldots n$. *So the uniformly weighted linear measure of specificity is:*

$$Sp(\mu) = a_1 - \sum_{i=2}^{n} \frac{a_i}{n-1}$$

Proof. Trivial

Definition 10 *Let A be an fuzzy set on* $X = \{e_1, \ldots, e_n\}$ *and let* A' *be an extended fuzzy set of A on* $X' = \{e_1, \ldots, e_n, e_{n+1}\}$ *with* $A'(e_i) = A(e_i)$ *for all* e_i *in X and with* $A'(e_{n+1}) = b$. *A measure of specificity Sp is recursive if it satisfies the following axioms:*

1. *There exists a mapping* $\phi_A : [0,1]^2 \to [0,1]$ *such that* $Sp(A') = \phi_A(Sp(A), b)$.
2. *The computation of* ϕ_A *has a constant computational complexity* $\mathcal{O}(1)$.

The mapping ϕ_A in definition 10 is called the "expansion" mapping of Sp.

Note that a constant computational complexity forces to have a finite bounded number of operations not depending on n, where n is the cardinality of the universe X.

Note also that usual measures of specificity requires a sort of the membership degrees of the fuzzy set which takes a computational complexity $\mathcal{O}(nlog(n))$. So the recursive measures of specificity can not recompute the whole formula when a new candidate is added to the universe, but it must reuse the value of the previously computed measure of specificity.

Lemma 2 *Let Sp be a recursive measure of specificity and let* ϕ_A *be its "expansion" mapping. If* $\phi_A(x, 0) = x$ *for all X in* $[0,1]$ *then Sp is an expansible measure of specificity.*

Proof. Trivial from definition 8 and 10

Theorem 1 *If a linear measure of specificity is uniformly weighted then it is a recursive measure of specificity.*

Proof. Let A be a fuzzy set on $X = \{e_1, \ldots, e_n\}$ with sorted membership degrees $(a_1, \ldots, a_n)$ where a_j is the j^{th} greatest membership degree of A. Let A' on $X = \{e_1, \ldots, e_n, e_{n+1}\}$ be a extended fuzzy set of A with $A'(e_{n+1}) = b$.

There exists two cases:

- If $b \leq a_1$ then:
  $Sp(A') =$

$$
\begin{aligned}
&= a_1 - \tfrac{1}{n}(a_2 + \ldots + b) \\
&= a_1 - \tfrac{1}{n}(a_2 + \ldots + a_n) - \tfrac{1}{n}b \\
&= a_1 - \tfrac{n-1}{n}(a_1 - Sp(A)) - \tfrac{1}{n}b \\
&= a_1 - \tfrac{n-1}{n}a_1 + \tfrac{n-1}{n}Sp(A) - \tfrac{1}{n}b \\
&= \tfrac{1}{n}a_1 + \tfrac{n-1}{n}Sp(A) - \tfrac{1}{n}b \\
&= \tfrac{a_1+(n-1)Sp(A)-b}{n}
\end{aligned}
$$

– If $b > a_1$ then:

$$
\begin{aligned}
Sp(A') = & \\
&= b - \tfrac{1}{n}(a_1 + a_2 + \ldots a_n) \\
&= b - \tfrac{a_1}{n} - \tfrac{(a_2+\ldots a_n)}{n} \\
&= b - a_1 + \tfrac{(n-1)}{n}(a_1 - Sp(A)) \\
&= b - a_1 + \tfrac{(n-1)}{n}Sp(A)
\end{aligned}
$$

Therefore: $\phi_A(x,b) = \begin{cases} \frac{a_1+(n-1)x-b}{n}, & \text{if } b \leq a_1; \\ b - a_1 + \frac{n-1}{n}x, & \text{otherwise.} \end{cases}$

Let's consider the uniformly weighted linear measure of specificity in the following examples.

Example 1 *Let A be a fuzzy set on $X = \{e_1, e_2, e_3\}$ with the following sorted membership degrees: $A = (1, 0.2, 0)$. Then the uniformly weighted measure of specificity of A is $Sp(A) = 0.9$.*

Let $X' = X \cup \{e_4\}$ an extension of X. Let A' be a fuzzy set on X' expanding A. There are considered the following three cases: $A'(e_4) = 1$, $A'(e_4) = 0.5$ and $A'(e_4) = 0$. So:

– *If $A' = (1, 0.2, 0, 1)$ then $Sp(A') = \phi_A(0.9, 1) = \frac{1+2*0.9-1}{3} = 0.6$.*
– *If $A' = (1, 0.2, 0, 0.5)$ then $Sp(A') = \phi_A(0.9, 0.5) = \frac{1+2*0.9-0.5}{3} = 0.77$.*
– *If $A' = (1, 0.2, 0, 0)$ then $Sp(A') = \phi_A(0.9, 0) = \frac{1+2*0.9-0}{3} = 0.93$.*

The following is an example of the computation of the uniformly weighted measure of specificity of a non-normal fuzzy set, and its expanded fuzzy set, with a new candidate.

Example 2 *Let A be a fuzzy set on $X = \{e_1, e_2, e_3\}$ with the following sorted membership degrees: $A = (0.5, 0.2, 0)$. Then $Sp(A) = 0.4$. So:*

– *If $A' = (0.5, 0.2, 0, 1)$ then $Sp(A') = \phi_A(0.4, 1) = 1 - 0.5 + 2/3 * 0.4 = 0.77$.*
– *If $A' = (0.5, 0.2, 0, 0.5)$ then $Sp(A') = \phi_A(0.4, 0.5) = \frac{0.5+2*0.4-0.5}{3} = 0.27$.*
– *If $A' = (0.5, 0.2, 0, 0)$ then $Sp(A') = \phi_A(0.4, 0) = \frac{0.5+2*0.4}{3} = 0.43$.*

Note that the uniformly weighted measure of specificity is recursive but it is not expansible (see examples 1 and 2 with $b = 0$).

Theorem 2 *Let Sp the linear measure of specificity with $w_2 = 1$ and $w_j = 0$ for all $j : 3..n$. Then Sp is a recursive measure of specificity.*

Proof. Let A be a fuzzy set on $X = \{e_1, \ldots, e_n\}$ with sorted membership degrees $(a_1, \ldots, a_n)$ where a_j is the j^{th} greatest membership degree of A. Let A' on $X = \{e_1, \ldots, e_n, e_{n+1}\}$ be a extended fuzzy set of A with $A'(e_{n+1}) = b$.

There exists three cases:

- If $b \leq a_2$ then $Sp(A') = Sp(a)$.
- If $b \geq a_1$ then $Sp(A') = b - a_1 = b - Sp(A) - a_2$.
- If $a_1 > b > a_2$ then $Sp(A') = a_1 - b = Sp(A) - b + a_2$.

Therefore: $\phi_A(x, b) = \begin{cases} x, & \text{if } b \leq a_2; \\ b - x - a_2, & \text{if } b \geq a_1; \\ x - b + a_2, & \text{if } a_1 > b > a_2; \end{cases}$

Corollary 1 *[15] Let Sp the linear measure of specificity with $w_2 = 1$ and $w_j = 0$ for all $j : 3..n$. Then Sp is both an expansible and recursive measure of specificity.*

Proof. Trivial by theorem 2 and lemma 2

Example 3 *Let Sp be the linear measure of specificity with $w_2 = 1$. Let A be a fuzzy set on $X = \{e_1, e_2, e_3\}$ with the following sorted membership degrees: $A = (1, 0.8, 0)$. Then $Sp(A) = 1 - 0.8 = 0.2$:*

Let $X' = X \cup \{e_4\}$ an extension of X. Let A' be a fuzzy set on X' expanding A. There are considered the following three cases: $A'(e_4) = 1$, $A'(e_4) = 0.5$ and $A'(e_4) = 0$.

- *If $A' = (1, 0.8, 0, 1)$ then $Sp(A') = \phi_A(0.2, 1) = 1 - 0.2 - 0.8 = 0$.*
- *If $A' = (1, 0.8, 0, 0.5)$ then $Sp(A') = \phi_A(0.2, 0.5) = 0.2$.*
- *If $A' = (1, 0.8, 0, 0)$ then $Sp(A') = \phi_A(0.2, 0) = 0.2$.*

Theorem 3 *The fractional measures of specificity are recursive measures of specificity.*

Proof. Let A be a fuzzy set on $X = \{e_1, \ldots, e_n\}$ with sorted membership degrees $(a_1, \ldots, a_n)$ where a_j is the j^{th} greatest membership degree of A. Let A' on $X = \{a_1, \ldots, a_n, b\}$ be a extended fuzzy set of A with $A'(e_{n+1}) = b$.

There exists two cases:

- If $b \leq a_1$ then:

$$Sp(A') = \frac{a_1^2}{\sum_{j=1}^{n} a_j + b} = \frac{a_1^2}{\frac{a_1^2}{Sp(A)} + b} = \frac{a_1^2 Sp(A)}{a_1^2 + bSp(A)}$$

- If $b > a_1$ then:

$$Sp(A') = \frac{a_{n+1}^2}{\sum_{j=1}^{n} a_j + a_1} = \frac{a_{n+1}^2}{\frac{a_1^2}{Sp(A)} + a_1} = \frac{a_{n+1}^2 Sp(A)}{a_1^2 + a_1 Sp(A)}$$

Therefore: $\phi_A(x,b) = \begin{cases} a_1^2 x/(a_1^2+bx), & \text{if } b \le a_1; \\ b_1^2 x/(a_1^2+a_1 x), & \text{otherwise.} \end{cases}$

Corollary 2 *[15] The fractional measures of specificity are expansible and recursive measures of specificity.*

Example 4 *Let Sp be the fractional measure of specificity. Let A be a fuzzy set on* $X = \{e_1, e_2, e_3\}$ *with the following sorted membership degrees:* $A = (1, 0.8, 0)$. *Then* $Sp(A) = 0.56$.

Let $X' = X \cup \{e_4\}$ *an extension of* X. *Let* A' *be a fuzzy set on* X' *expanding* A. *There are considered the following three cases:* $A'(e_4) = 1$, $A'(e_4) = 0.5$ *and* $A'(e_4) = 0$.

- *If* $A' = (1, 0.8, 0, 1)$ *then* $Sp(A') = \phi_A(0.56, 1) = \frac{0.56}{1+0.56} = 0.36$.
- *If* $A' = (1, 0.8, 0, 0.5)$ *then* $Sp(A') = \phi_A(0.56, 0.5) = \frac{0.56}{1+0.5*0.56} = 0.44$.
- *If* $A' = (1, 0.8, 0, 0)$ *then* $Sp(A') = \phi_A(0.56, 0) = \frac{0.56}{1+0} = 0.56$.

Theorem 4 *The product measure of specificity is a recursive measure of specificity.*

Proof. Let A be a fuzzy set on $X = \{e_1, \ldots, e_n\}$ with sorted membership degrees $(a_1, \ldots, a_n)$ where a_j is the j^{th} greatest membership degree of A. Let A' on $X = \{a_1, \ldots, a_n, b\}$ be a extended fuzzy set of A with $A'(e_{n+1}) = b$.

There exists two cases:

- If $b \le a_1$ then:

$$Sp(A') = a_1 \prod_{j=2}^{n} (ka_j + (1-a_j)(kb + (1-b)) = a_1 \frac{Sp(A)}{a_1}(kb + (1-b)) = Sp(A)(kb + (1-b))$$

- If $b > a_1$ then:

$$Sp(A') = b \prod_{j=2}^{n} (ka_j + (1-a_j)(ka_1 + (1-a_1)) = b \frac{Sp(A)}{a_1}(ka_1 + (1-a_1))$$

Therefore: $\phi_A(x,b) = \begin{cases} x(kb + (1-b)), & \text{if } b \le a_1; \\ \frac{xb}{a_1}(ka_1 + (1-a_1)), & \text{otherwise.} \end{cases}$

Corollary 3 *[15] The product measure of specificity is an expansible measure of specificity.*

Proof. Trivial by lemma 2

Example 5 *Let Sp be the product measure of specificity with* $k = 0.5$. *Let A be a fuzzy set on* $X = \{e_1, e_2, e_3\}$ *with the following sorted membership degrees:* $A = (1, 0.8, 0)$. *Then* $Sp(A) = 0.6$.

Let $X' = X \cup \{e_4\}$ *an extension of* X. *Let* A' *be a fuzzy set on* X' *expanding* A. *It is considered three cases:* $A'(e_4) = 1$, $A'(e_4) = 0.5$ *and* $A'(e_4) = 0$.

- *If* $A' = (1, 0.8, 0, 1)$ *then* $Sp(A') = \phi_A(0.6, 1) = 0.6 * (0.5 * 1 + (1 - 1)) = 0.3$.
- *If* $A' = (1, 0.8, 0, 0.5)$ *then* $Sp(A') = \phi_A(0.6, 0.5) = 0.6 * (0.5 * 0.5 + (1 - 0.5)) = 0.45$.
- *If* $A' = (1, 0.8, 0, 0)$ *then* $Sp(A') = \phi_A(0.6, 0) = 0.6 * (0.5 * 0 + (1 - 0)) = 0.6$.

The following table 1 is a summary of the expansion mappings of the recursive measures of specificity.

Measure of specificity	$Sp(A)$	Expansion mapping	Expansible
Linear uniformly weighted	$a_1 - \sum_{j=2}^{n} \frac{1}{n-1} a_j$	$\phi_A(x, b) = \begin{cases} (a_1 + (n-1)x - b)/n, & \text{if } b \leq a_1; \\ b - a_1 + (n - 1/n)x, & \text{otherwise.} \end{cases}$	No
Linear with $w_2 = 1$	$a_1 - a_2$	$\phi_A(x, b) = \begin{cases} x, & \text{if } b \leq a_2; \\ b - x - a_2, & \text{if } b \geq a_1; \\ x - b + a_2, & \text{if } a_1 > b > a_2; \end{cases}$	Yes
Fractional	$a_1^2 / \sum_{j=1}^{n} a_j$	$\phi_A(x, b) = \begin{cases} a_1^2 x/(a_1^2 + bx), & \text{if } b \leq a_1; \\ b_1^2 x/(a_1^2 + a_1 x), & \text{otherwise.} \end{cases}$	Yes
Product	$a_1 \prod_{j=2}^{n} (ka_j + (1 - a_j))$	$\phi_A(x, b) = \begin{cases} x(kb + (1-b)), & \text{if } b \leq a_1; \\ \frac{xb}{a_1}(ka_1 + (1 - a_1)), & \text{otherwise.} \end{cases}$	Yes

Table 1. Some recursive measures of specificity and its expansion mapping

4 Conclusions

Recursive measures of specificity are introduced to improve the performance of the computation of such measures when the universe of candidates is increased with any membership degree. The computation is done without resorting all the membership degrees again and using a recursive formula that uses the initial fuzzy set, its measure of specificity and the membership degree of the new candidate by using an expansion mapping taking a constant computational complexity cost. Recursive measures of specificity can also be used to compute the specificity of fuzzy sets with a linear computational complexity $\mathcal{O}(n)$.

A variety of recursive measures of specificity are proposed, their expansions mappings are provided and some examples are given.

Acknowledgment

We thank the support of the research project TIN2012-32482.

References

1. A. del Amo, J. Montero, and E. Molina. Representation of consistent recursive rules. *European Journal of Operational Research*, 130(1):29–53, 2001.
2. V. Cutello and J. Montero. Recursive families of owa operators. In IEEE Press, editor, *Proceedings FUZZ-IEEE Conference*, pages 1137–1141, 1994.
3. V. Cutello and J. Montero. Recursive connective rules. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, (14):3–20, 1999.

4. L. Garmendia, R.R. Yager, E. Trillas, and A. Salvador. On t-norms based specificity measures. *Fuzzy Sets and Systems*, 133(2):237–248, 2003.
5. L. Garmendia, R.R. Yager, E. Trillas, and A. Salvador. General measures of specificity of fuzzy sets under t-indistinguishabilities. *IEEE Transactions on Fuzzy Systems*, 14(4):568–572, 2006.
6. L. Garmendia, R.R. Yager, E. Trillas, and A. Salvador. A t-norm based specificity for fuzzy sets on compact domains. *International Journal of General Systems*, 35(6):687–698, 2006.
7. R. González-del Campo, L. Garmendia, and R. Yager. Specificity for interval-valued fuzzy sets. *International Journal of Computational Intelligence Systems*, 5(3):452–459, 2012.
8. R.R. Yager. Ordinal measures of specificity. *International Journal of General Systems*, 17:57–72, 1990.
9. R.R. Yager. Similarity based measures of specificity. *International Journal of General Systems*, 19:91–106, 1991.
10. R.R. Yager. Default knowledge and measures of specificity. *International Journal of General Systems*, 61:1–44, 1992.
11. R.R. Yager. On the specificity of a possibility distribution. *Fuzzy Sets and Systems*, 50:279–292, 1992.
12. R.R. Yager. On measures of specificity. In O. Kaynak, L.A. Zadeh, B. Turksen, and I.J. Rudas, editors, *Computational Intelligence: Soft Computing and Fuzzy-Neuro Integration with Applications*. Springer, 1998.
13. R.R. Yager. Containment and specificity for type-2 fuzzy sets. *International Journal Of Fuzzy Systems*, 9:55–66, 2007.
14. R.R. Yager. On the measure of specificity of intuitionistic fuzzy sets. *NAFIPS 2008 - Annual Meeting of the North American Fuzzy Information Processing Society*, pages 677–682, 2008.
15. R.R. Yager. Expansible measures of specificity. *Int. J. General Systems*, 41(3):247–263, 2012.

Clasificación borrosa basada en disimilitud para la valoración inicial de desastres

J. Tinguaro Rodríguez, Begoña Vitoriano, Javier Montero

Facultad de Ciencias Matemáticas, Universidad Complutense de Madrid
{jtrodrig,bvitoriano,javier_montero}@mat.ucm.es

Abstract. En la gestión de desastres y emergencias, es crucial una valoración inicial correcta de las consecuencias de los fenómenos adversos para permitir una toma de decisiones adecuada. Sin embargo, no se requiere que esta valoración inicial sea necesariamente totalmente precisa, por lo que su obtención puede asimilarse con un problema de clasificación borrosa en el que las clases presentan una estructura relevante, que emana de la semántica del contexto y de los requisitos del problema de decisión. Este trabajo propone la consideración de un operador de disimilitud para la introducción de esta estructura en los procesos de aprendizaje y razonamiento de un clasificador borroso, lo que redunda en una mejora de la adaptación del clasificador a las características y los requisitos en términos de toma de decisiones del contexto de la gestión de desastres.

Keywords: toma de decisiones, clasificación borrosa, disimilitud, gestión de desastres.

1 Introducción

Tras la ocurrencia de un desastre en cualquier lugar del planeta, las agencias y ONG internacionales involucradas en la respuesta a desastres (como OCHA o IFRC) ponen en marcha procesos de análisis orientados a decidir sobre la pertinencia de una operación de ayuda y sobre la existencia o no de condiciones propicias para su realización. Por ello, las decisiones a tomar en un primer momento tienen un carácter estratégico, más centradas en valorar el grado de implicación de la organización en una posible operación de respuesta que en los contenidos específicos de tal operación.

Sin embargo, estas decisiones estratégicas tienen una influencia clave en los subsiguientes procesos de decisión logístico y sobre el terreno, que evalúan la cantidad de ayuda a enviar y cómo será transportada hasta la población afectada (ver [3]). Por esto, el proceso de decisión estratégico tiene lugar en un contexto de urgencia, pues cualquier retraso en esta etapa puede comprometer el prestigio y la reputación de la organización y retrasar la recepción de ayuda por parte de la población afectada.

Por otro lado, la toma de decisiones estratégica depende fuertemente de una valoración correcta de las consecuencias del desastre y las necesidades resultantes de la población afectada. Sin embargo, la información disponible tras el impacto de un desastre suele estar afectada por distintos tipos de *incertidumbre*. Los primeros *reports* suelen ser bastante incompletos, cuando no directamente confusos o contradictorios. Además, cuando existe, la información relevante suele estar expresada lingüís-

ticamente, por lo que puede ser vaga e imprecisa. Esto plantea dificultades para el desarrollo del mencionado proceso de decisión estratégico, en tanto que en este contexto la urgencia de las decisiones choca con la necesidad de una correcta estimación de las consecuencias del desastre, i.e. de las necesidades de la población afectada.

En trabajos anteriores de los autores se propone y describe un sistema de ayuda a la decisión (DSS) centrado en la evaluación de las consecuencias de los desastres con la primera información disponible tras su impacto. Como se discute en [10], este DSS, denominado SEDD (Sistema para la Evaluación y el Diagnóstico de Desastres), constituye uno de los primeros esfuerzos encaminados a proveer a las ONG con herramientas inteligentes para la valoración de consecuencias de los desastres específicamente diseñadas para estas organizaciones El motor de inferencia de SEDD [9] está basado en procedimientos de aprendizaje automático borroso (ver [4]), y en particular en el uso de un sistema de clasificación basado en reglas borrosas (SCBRB, ver [2]). Además, los SCBRB mejoran el rendimiento de la mayoría de técnicas estadísticas ordinarias y de métodos aprendizaje automático en la labor de proporcionar una valoración simultáneamente interpretable y precisa de las consecuencias de desastres [11].

Este trabajo estudia la estructura del problema de clasificación que subyace en la valoración provista por SEDD. En concreto, se analizan las relaciones entre la estructura impuesta en el conjunto de clases y los requisitos del contexto de decisión, mostrando que se alcanza una mejora significativa del comportamiento de SEDD en términos de su adaptación a tales requisitos al considerar e introducir ciertas estructuras en los modelos de clasificación. Como se muestra en [8], en un marco de clasificación supervisada es posible introducir relaciones entre las clases mediante la noción de antagonismo o disimilitud semántica propuesta en [7]. Por tanto, en este trabajo se discute el efecto de diferentes estructuras de disimilitud en términos de su capacidad de adaptación a algunos de los requisitos del contexto de la gestión de desastres.

2 La valoración estratégica de la severidad de los desastres como un problema de clasificación con estructura

Considérese la estimación de la variable *número de personas sin hogar*, que mide el número de personas que perdieron su casa como consecuencia de un desastre. Para las ONG, esta cantidad constituye un indicador clave de la magnitud de los esfuerzos que una potencial operación de ayuda debería acometer en materia de alojamientos temporales y gestión de emplazamientos. En la práctica, en una primera etapa, para un decisor de una ONG no es importante distinguir si 50.000 o 70.000 personas quedaron sin hogar, ya que en cualquier caso tal cantidad va a ser considerada *grande* y las decisiones estratégicas a tomar en términos de las acciones subsiguientes y el tamaño y la naturaleza de la operación necesaria van a ser similares.

Así pues, en los primeros momentos tras el impacto de un desastre, los decisores de las ONG valoran la severidad del desastre de una manera cualitativa (más que cuantitativa). Dicho de otro modo, su problema consiste en evaluar, en un contexto de información altamente incierta e imprecisa, la magnitud de las consecuencias de un desastre en relación con los escenarios y decisiones relevantes que pueden surgir de cara a la implementación de una operación de respuesta.

Por este motivo, consideramos que, en lugar de una evaluación numérica precisa, para obtener una valoración inicial que sustente una adecuada toma de decisiones estratégicas es más plausible y realista *clasificar* la severidad de las consecuencias de un desastre en términos de los escenarios relevantes para los decisores de las ONG. Así, el mencionado problema de evaluación de desastres conduce a un problema de clasificación en el que las clases se identifican con los términos lingüísticos que describen esos escenarios relevantes, como *sin víctimas mortales* o *muchos heridos*. Estas etiquetas o clases, valorando la magnitud de diferentes consecuencias relevantes de un desastre, han de ser asignadas sobre la base de una descripción o atributos del desastre dada por la primera información disponible, como el tipo de desastre, su intensidad y las características de la localización afectada (p. ej. su vulnerabilidad, ver [6]).

2.1 Estructura del conjunto de clases

Considérese ahora una de las variables a ser evaluadas para la consecución de la mencionada valoración inicial. Por ejemplo, centrémonos en la variable *número de víctimas mortales* (*CAS*). Esta variable mide el número de personas que perecieron como consecuencia del impacto del fenómeno adverso. Como se ha explicado, en una primera etapa es posible valorar esta cantidad de manera cualitativa, o lingüística. Por ejemplo, podemos medir la severidad de un escenario, en términos del número de fallecidos, por medio de las etiquetas *sin víctimas mortales*, *muy pocas*, *pocas*, *bastantes* y *muchas*. Estas etiquetas representan las clases en que ha de ser clasificado un escenario de desastre para obtener una valoración inicial de la consecuencia *CAS*.

El significado particular de estas etiquetas ha de especificarse por medio de intervalos o, más generalmente, mediante subconjuntos borrosos, del rango de la variable numérica subyacente, en este caso los enteros positivos. Además, estos significados han de estar relacionados con los diferentes escenarios relevantes en términos de las decisiones a tomar. Por ejemplo, cada etiqueta puede asociarse con un orden de magnitud creciente en el número de fallecidos, como se muestra en la Tabla 1.

Clase	*CAS*1	*CAS*2	*CAS*3	*CAS*4	*CAS*5
Etiqueta	Sin víctimas mortales	Muy pocas	Pocas	Bastantes	Muchas
Intervalo	[0,10)	[10,100)	[100,1000)	[1000,10000)	[10.000,+∞)

Tabla 1. Clases para la variable *número de víctimas mortales* (*CAS*).

Nótese que, en tanto que asociadas a diferentes órdenes de magnitud de las consecuencias, estas etiquetas pueden considerarse ordenadas de menor a mayor nivel de severidad de las consecuencias. Esto es, las clases asociadas a la variable CAS están ordenadas linealmente, i.e. $CASi < CASj$ si $i < j$. Este supuesto de linearidad de los efectos de un desastre es bastante general, ya que lo que los decisores tratan de valorar en un primer momento es precisamente el orden de severidad de esas consecuencias, o equivalentemente el orden de magnitud de los esfuerzos que una operación de ayuda debería considerar para aliviar esas consecuencias. Estos órdenes de magnitud no son necesariamente potencias de 10 de la variable numérica subyacente, sino que se asocian a los diferentes escenarios

relevantes que pueden surgir, ordenados atendiendo a la gravedad de la crisis humanitaria que implican.

Por tanto, en este marco, las clases no son ítems independientes o no relacionados, sino que conforman una estructura de valuación, en el sentido de [5]. Por ejemplo, no se comete el mismo error cuando se valora un escenario sin fallecidos (*CAS*1) como teniendo *pocas* víctimas mortales (*CAS*2), que cuando el mismo es evaluado como habiendo producido *muchos* fallecidos (*CAS*5). Así pues, el mencionado problema de valoración de la severidad de un desastre puede ser entendido como un problema de clasificación con un conjunto de clases estructurado.

2.2 Requisitos del contexto

El conjunto de clases recién introducido para la variable *CAS* en principio se ajusta a una estructura lineal, en tanto que las clases están semánticamente asociadas a diferentes órdenes de magnitud de las consecuencias. Algunas clases están más próximas entre sí que otras, por lo que se pueden distinguir diferentes niveles de error de acuerdo con tal distancia. Sin embargo, nótese que el error cometido al valorar un escenario *CAS*1 como *CAS*5 es también diferente al cometido cuando un escenario *CAS*5 se evalúa como *CAS*1. En el primer caso hay error de sobreestimación mientras que en el segundo lo hay de subestimación. Aunque los dos tipos de error son relevantes, la subestimación de los efectos de un desastre puede acarrear consecuencias mucho más peligrosas que la sobreestimación para el prestigio de la ONG y la población afectada.

Por ello, los decisores de las ONG suelen tender a evitar el riesgo de subestimación, por ejemplo llevando a cabo análisis del peor caso en el escenario bajo estudio. Así, puede requerirse que la valoración inicial de un escenario de desastre sea realizada evitando el riesgo de subestimación. Este requisito, asociado a la toma de decisiones, implica de alguna manera introducir una configuración asimétrica en la estructura lineal de las clases, ya que se cometen errores distintos dependiendo de si un escenario es subestimado o sobreestimado. La estructura del conjunto de clases ha de capturar tanto los aspectos relevantes de la semántica de las clases como los objetivos y requisitos relacionados con el contexto de decisión en que se usa el clasificador.

2.3 Estructuras de disimilitud

La asimetría de la estructura lineal del conjunto de clases fuerza a buscar estructuras más generales que las de orden. Aquí se adopta la noción de *estructura de disimilitud* propuesta en [7] para proporcionar una definición formal de la estructura de un conjunto de clases. Estas estructuras se basan en la noción de antagonismo o disimilitud semántica (también propuesta en [7]), que permite modelar relaciones de oposición entre conceptos, y en particular relaciones asimétricas. Por tanto, al adoptar esta noción, en cierto modo se está traduciendo la *distancia* semántica entre dos clases, proveniente de la ordenación lineal de las consecuencias, en un *grado de oposición* entre ellas, que puede ser asimétrico para reflejar los requisitos citados.

Como se verá en la próxima sección, las estructuras de disimilitud proporcionan un método fácil y efectivo para introducir relaciones entre clases en los modelos de clasificación, permitiendo distinguir entre *excepciones significativas* y simples *con-*

traejemplos a una regla de clasificación. Esto conduce a la aparición de un grado de confianza *negativo* en las reglas, i.e., a un marco de clasificación bipolar (ver [8]).

Así pues, denótese por $\zeta = \{C_1,...,C_N\}$ el conjunto de clases en consideración. Como se explica en [7], es posible obtener una estructura de disimilitud sobre este conjunto mediante una *matriz de disimilitud* $\Delta = (d_{ij})_{N\times N}$, tal que el valor $d_{ij} \in [0,1]$ expresa el grado en que la clase C_j es opuesta o disimilar a la clase C_i, $i,j = 1,\ldots,N_C$.

De este modo, la matriz $\Delta^I = 0$ representa una situación en que ninguna clase es opuesta a las demás. En el otro extremo, $\Delta^{III} = 1 - Id$ describe una situación en que todas las clases son totalmente opuestas entre sí. Como se verá a continuación, estas matrices permiten representar algunos de los requisitos del contexto de aplicación *dentro* de los modelos de clasificación. Por tanto, existe un amplio rango de posibilidades entre estos casos extremos para la modelización de diferentes condiciones de oposición. La elección de una matriz particular dependerá por supuesto de los requisitos semánticos específicos que hayan de cumplirse.

3 Una metodología bipolar para SEDD

Tal y como se describe en [9,11], la metodología de SCBRB de SEDD no considera explícitamente ninguna estructura en el conjunto de clases. Por ello, en esta sección se ilustran las ideas anteriores adaptando la metodología de SEDD al caso estructurado. Para este fin, se adopta la metodología de clasificación bipolar propuesta en [8].

3.1 Nociones básicas sobre SEDD

En primer lugar, SEDD puede ser entendido como un SCBRB. Como tal, las reglas que guían la clasificación han de ser aprendidas a partir de un conjunto de ejemplos de entrenamiento, y el proceso de inferencia que asigna una clase a una nueva instancia precisa que esa instancia sea descrita en los mismos términos que los ejemplos de entrenamiento. La muestra de entrenamiento de SEDD es la base de datos EM-DAT. Las variables de EM-DAT con interés para este estudio se muestran en la Tabla 2. Las variables *MAG* y *HDI* se toman como explicativas para la variable dependiente *CAS*.

Variable	Descripción
MAG (Magnitud)	*Intensidad* del fenómeno adverso: grados en la escala Richter para terremotos, etc.
HDI	Índice de Desarrollo Humano: una estimación de la vulnerabilidad del país afectado.
CAS (Fallecidos)	Número de víctimas mortales producidas por el desastre.

Table 2. Variables de EM-DAT relevantes para este estudio.

En segundo lugar, debe aplicarse un procedimiento de aprendizaje para extraer las reglas de los datos. Denotemos por $X_1,...,X_n$ los *n* atributos usados para describir un desastre, y supongamos que se dispone de un conjunto de *m* ejemplos de entrenamiento $(x_1^p,...,x_n^p;C^p)$ como muestra de aprendizaje, donde para cada $p = 1,\ldots,m$, C^p es una de las clases en $\zeta = \{C_1,...,C_N\}$. Supongamos también que se dispone de una matriz de disimilitud $\Delta = (d_{ij})_{N\times N}$ modelando las relaciones de oposición entre las

clases en ζ. Para todo $i = 1,\ldots,n$ denotemos por $A_{i1},...,A_{ic_i}$ las c_i etiquetas lingüísticas definidas para cada atributo X_i, de manera que $\mu_{A_{ij}}(x_i) \in [0,1]$ representa el grado en que el valor x_i verifica la etiqueta A_{ij}. Una premisa A de una regla viene dada por la combinación de una etiqueta A_{ij_i} de cada atributo, i.e. $A = A_{1j_1} \times ... \times A_{nj_n}$. El grado de compatibilidad entre una premisa A y un patrón $x = (x_1,...,x_n)$ se suele obtener mediante una t-norma T, i.e. $\mu_A(x) = T(\mu_{A_{1j_1}}(x_1),...,\mu_{A_{nj_n}}(x_n))$.

En estas condiciones, siguiendo [8], una regla con premisa A y consecuente una clase $C_j \in \zeta$ se evalúa mediante un par de grados de confianza

$$r_j^+(A) = \frac{\sum_{p=1,...,m/C^p=C_j} \mu_A(x^p)}{\sum_{p=1}^{m} \mu_A(x^p)}, \qquad r_j^-(A) = \frac{\sum_{p=1}^{m} \mu_A(x^p) \cdot \mu_{\Delta C_j}(C^p)}{\sum_{p=1}^{m} \mu_A(x^p)}$$

donde $\mu_{\Delta C_j}(C_i) = d_{ji}$. Nótese que $r^+, r^- \in [0,1]$ y que se cumple que $r^+ + r^- \leq 1$. Como se explica en [8], $r_j^+(A)$ y $r_j^-(A)$ estiman respectivamente la proporción de ejemplos positivos y excepciones significativas de la regla $A \Rightarrow C_j$ sobre el total de ejemplos que cumplen la premisa A. Por tanto, la estructura de disimilitud se introduce en la evaluación de las reglas por medio del grado de confianza negativo. Es posible usar diferentes procedimientos (ver por ejemplo [1]) para obtener el conjunto de premisas A para el que se han de formar reglas.

Por último, se ha de aplicar un método de razonamiento borroso (ver [2]) para evaluar un escenario de desastre $x = (x_1,...,x_n)$ bajo estudio. Siguiendo [8], aquí se aplicará el método de razonamiento VA1 (veracidad aditiva unidimensional), que calcula el grado de veracidad $t_j(A) = \max\{r_j^+(A) - r_j^-(A), 0\}$ para las R reglas $R^q : A^q \Rightarrow C_j$ disponibles, y computa un grado de evidencia $t_j(x)$ para la clasificación del escenario x en cada clase C_j como

$$t_j(x) = \frac{\sum_{q=1,...,R} \mu_{A^q}(x) \cdot t_j(A^q)}{\sum_{q=1,...,R} \mu_{A^q}(x)}.$$

Así pues, el vector $t(x) = (t_1(x),...,t_N(x))$ constituye el output final de SEDD, valorando el grado de evidencia para cada clase o nivel de consecuencias. Si es preciso facilitar una predicción nítida, se asigna el escenario x a la clase con máxima evidencia, i.e. a la clase C_h tal que $t_h(x) = \max_j t_j(x)$.

3.2 Estructuras de disimilitud para la valoración de desastres

Aquí se ilustra el efecto de diferentes estructuras de disimilitud en la valoración provista por SEDD para el caso de terremotos. Por tanto, se tiene n=2 y N=5 (se usan las clases descritas en la Tabla 1 para la variable dependiente *CAS*). La muestra de entrenamiento contiene m=386 ejemplos. Como el número máximo de premisas es pequeño (5·5=25), se adopta un método de aprendizaje basado en retículo, i.e. se

generan reglas para todas las posibles premisas, aunque se considera un umbral (δ=0.01) sobre el soporte de las reglas. Las matrices de disimilitud que se usarán son

$$\Delta_0 \equiv 0, \Delta_1 = 1 - Id, \Delta_{RL} = \begin{pmatrix} 0 & 0.2 & 0.2 & 0.5 & 1 \\ 0 & 0 & 0.2 & 0.5 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \Delta_{TA} = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Para ilustrar y comparar visualmente el efecto de cada matriz Δ, se computa el resultado del clasificador VA1 correspondiente en un mallado de puntos del espacio de entrada de los atributos *MAG* y *HDI*, obteniendo una representación gráfica de las fronteras de clase producidas por cada estructura de disimilitud. También, se computan dos medidas de error para medir el rendimiento de los diferentes clasificadores: 1) %CC representa la tasa de clasificaciones correctas obtenidas sobre la muestra de aprendizaje; 2) para medir tanto la desviación de las predicciones a la clase real como el riesgo de subestimación, se computa el coste promedio AVCOST de las predicciones sobre la muestra de entrenamiento, donde el coste de clasificar un ejemplo de la clase *i* en la clase *j* viene dado por $COST_{ij} = j - i$ si $i \leq j$ y $COST_{ij} = 2 \cdot (i - j)$ si $i > j$.

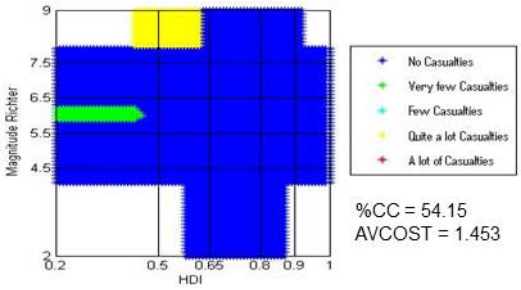


Fig. 1. Resultados del clasificador con $\Delta_0 \equiv 0$.

No oposición.

Comencemos suponiendo que no existe oposición entre las clases, i.e. $\Delta_0 \equiv 0$. En este caso, $r_j^-(A)=0$ para toda premisa *A* y consecuente C_j, por lo que $t_j(A) = r_j^+(A)$ para todas las reglas $A \Rightarrow C_j$. Por tanto, este caso corresponde a un marco de clasificación no estructurada, en el que se obtiene un clasificador borroso estándar, idéntico al usado por SEDD en [9]. Al no asumir ninguna estructura, el clasificador resultante trata las clases como ítems independientes, por lo que este estará sesgado hacia las clases con mayor proporción de ejemplos de entrenamiento. Esto se visualiza claramente en la Fig. 1, en la que se muestra el resultado del clasificador sobre el mallado. Una gran parte del espacio de entrada se asigna a la clase más baja, *CAS1= sin víctimas mortales*, i.e. aquella con mayor proporción (54.4%) de ejemplos de entrenamiento. Esto conlleva un fuerte riesgo de subestimación de las consecuencias. Además, el comportamiento del clasificador no es suave o gradual, en tanto que las predicciones presentan saltos bruscos.

Oposición total.

Analicemos ahora el otro caso extremo, dado por la matriz $\Delta_1 = 1 - Id$, en el que todas las clases son totalmente opuestas entre sí. Nótese que ahora se tiene $r_j^-(A) = 1 - r_j^+(A)$, por lo que $t_j(A) = \max\{2r_j^+(A) - 1, 0\}$. En tanto que las clases se encuentran igualmente relacionadas entre sí, se obtiene un sesgo semejante al anterior, aunque ahora una regla $A \Rightarrow C_j$ obtendrá $t_j(A) = 0$ a menos que $r_j^+(A) > 0.5$, i.e. a menos que más de la mitad de los ejemplos compatibles con la premisa *A* pertenezcan a la clase C_j. Si ninguna clase cumple esta condición, entonces $t_j(A) = 0$ para todo *j*. Por ello, una instancia *x* puede quedar sin clasificar si ninguna de las reglas activadas tiene veracidad positiva. Esto se observa claramente en la Fig. 2, pues la mitad del espacio de entrada queda sin clasificar, y la otra mitad se asigna de nuevo a la clase mayoritaria *CAS*1, la única que alcanza el umbral 0.5 en algunas premisas. En cierta medida, este clasificador puede ser asociado con un requisito de no proporcionar una valoración a menos que sea posible establecer predicciones con amplio soporte. Nótese que el 43.78% y el 38.86% de los *m* = 386 ejemplos de entrenamiento quedan sin clasificar y bien clasificados respectivamente. Esto arroja una tasa de error de 17.36% (la anterior era 45.85%), aunque al precio de no clasificar casi media muestra. También, el coste promedio es significativamente menor al quedar sin clasificar gran parte de los ejemplos de las clases altas.

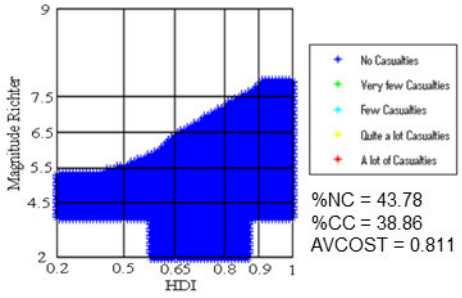


Fig. 2. Resultados del clasificador con $\Delta_1 \equiv 1 - Id$.

Orden lineal asimétrico restringido.
Para reproducir la semántica de orden lineal asociada a las clases, se toma la matriz Δ_{RL}. Esta matriz induce una estructura en la que las clases superiores son gradualmente más disimilares a las dos clases inferiores, *CAS*1 y *CAS*2, pero no simétricamente, i.e. las inferiores no son disimilares a las superiores. De este modo, las clases *CAS*1 y *CAS*2 obtienen una veracidad más baja en presencia de las superiores, precisando una mayor evidencia para ser predichas, representando la estructura lineal junto al requisito de no subestimación. En tanto que este riesgo es menor o puede ser ignorado para las clases superiores, las clases *CAS*3-*CAS*5 no son opuestas entre sí, por lo que compiten libremente como en el caso $\Delta \equiv 0$. El comportamiento de este clasificador indica ahora transiciones más suaves entre clases alejadas, y una mayor parte del espacio de entrada es asignado a clases más altas, ver Fig. 3. Además, se obtiene una tasa de clasificaciones más alta (%CC=55.18) que en el caso estándar, y se alcanza asimismo un coste promedio sensiblemente más bajo.

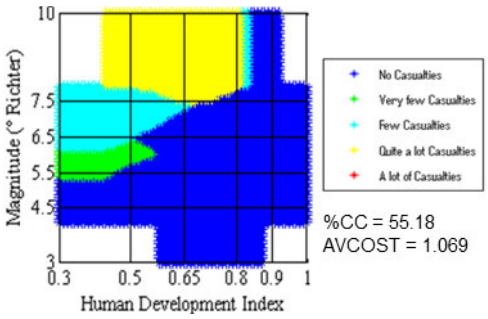


Fig. 3. Resultados del clasificador con $\Delta \equiv \Delta_{RL}$.

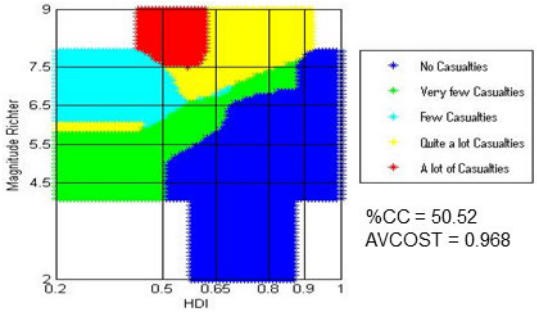


Fig. 4. Resultados del clasificador con $\Delta = \Delta_{PE}$.

Análisis del peor escenario.
Finalmente, consideremos un requisito de aversión total al riesgo de subestimación. Para este fin, se toma la matriz de disimilitud Δ_{TA}, que conduce a una estructura en que cada clase es opuesta a las clases inferiores a ella, pero no recíprocamente. Por tanto, en este marco no se considera de manera explícita una estructura lineal, sino solo un supuesto de tota asimetría entre las clases inferiores y superiores. Por tanto, cuanto más bajo es el índice de la clase, más difícil será predecirla. En este sentido, la matriz Δ_2 encaja con el requisito de un análisis del peor caso en el escenario bajo estudio. En consecuencia, como se observa en la Fig. 4, la clase *CAS*5 aparece por primera vez en los resultados del clasificador, y se obtienen transiciones incluso más suaves entre las predicciones que en el caso anterior, pudiéndose trazar una recta en el espacio de entrada que pase, ordenadamente, por todas las clases. Más significativamente, este clasificador permite distinguir una clara tendencia en las consecuencias, de manera que peores efectos se asocian con valores de *HDI* más bajos (por tanto una vulnerabilidad mayor) y mayores intensidades de los terremotos. Aunque esta tendencia es lógicamente esperable, ninguno de los clasificadores anteriores pudo reflejarla tan claramente. Además, este clasificador consigue una reducción del riesgo de subestimación, como se desprende de su coste promedio más bajo (AVCOST=0.968), que podría ser más importante atendiendo a que la tasa de clasificaciones correctas es más baja que en el caso anterior (%CC=50.52), con el consiguiente aumento de costes no nulos.

4 Conclusiones

Al considerar la producción de una valoración inicial de las consecuencias de desastres como un problema de clasificación estructurado, la estructura del conjunto de clases adquiere una fuerte relación con las características semánticas y los requisitos de los decisores del contexto de la gestión de desastres. Por ello, una adecuada caracterización de tales estructuras redunda en una mejor adaptación del clasificador a esas características y requisitos. En este trabajo se ha mostrado que la introducción de un operador de disimilitud en el conjunto de clases permite la consideración de diversas estructuras y su introducción en los procesos de aprendizaje y razonamiento del clasificador, obteniéndose con ello una mejora significativa de la precisión y la adaptación de los clasificadores a los requisitos en términos de toma de decisiones del contexto de la gestión de desastres.

Referencias

1. Casillas J, Cordón O, Herrera F (2002) COR: A Methodology to Improve Ad Hoc Data-Driven Linguistic Rule Learning Methods by Inducing Cooperation Among Rules. IEEE Trans. on Systems, Man and Cybernetics-Part B: Cybernetics 32 (4): 526- 537
2. Cordón, O., del Jesús, M. J., Herrera, F. (1999) A proposal on reasoning methods in fuzzy rule-based classification systems, International Journal of Approximate Reasoning, 20 (1) 21-45
3. Drabek, T. E., Hoetmer, G. J. (1991) Emergency management: principles and practice for local government, International City Management Association, Washington DC
4. Hullermeier, E. (2005) Fuzzy methods in machine learning and data mining: Status and prospects, Fuzzy Sets and Systems, 156 (3) 387-406
5. Montero, J., Gomez, D., Bustince, H. (2007) On the relevance of some families of fuzzy Sets, Fuzzy Sets and Systems, 158 (22) 2429-2442
6. Morrow, B. H. (1999) Identifying and mapping community vulnerability, Disasters, 23 (1) 1-18
7. Rodriguez JT, Franco CA, Vitoriano B, Montero J (2011) An axiomatic approach to the notion of semantic antagonism, Procs IFSA-AFSS'11 FT104-1/6
8. Rodríguez, J. T., Vitoriano, B., Montero, J. (2011) Rule-based classification by means of bipolar criteria. 2011 IEEE SSCI-MCDM 197-204
9. Rodríguez, J. T., Vitoriano, B., Montero, J. (2012) A general methodology for data-based rule building and its application to natural disaster management, Computers & Operations Research, 39 (4) 863-873
10. Rodríguez, J. T., Vitoriano, B., Montero, J., Omaña, A. (2008) A decision support tool for humanitarian operations in natural disaster relief, Computational Intelligence in Decision and Control, 1 805-810
11. Rodríguez, J. T., Vitoriano, B., Montero, J., Kecman V. (2011) A disaster-severity assessment DSS comparative analysis. OR Spectrum 33(3) 451-479
12. Wallace, W. A., De Balogh, F. (1985) Decision Support Systems for Disaster Management, Public Administration Review, 45 134-146

Análisis Morfológico difuso lingüístico en el ámbito de la Planificación de Escenarios

Pablo J. Villacorta, Antonio D. Masegosa y María T. Lamata

Grupo de Modelos de Decisión y Optimización (MODO),
Departamento de Ciencias de la Computación e Inteligencia Artificial,
Universidad de Granada, 18071 Granada
{pjvi, admase, mtl}@decsai.ugr.es

Resumen En el mundo moderno, las grandes empresas necesitan predecir los cambios en el mercado y anticiparse a ellos tomando las decisiones más adecuadas en el presente. Entre las herramientas especializadas para ello se encuentra la Planificación de Escenarios. Una de las técnicas usadas en Planificación de Escenarios es el Análisis Morfológico, cuya finalidad es la generación sistemática y evaluación de todas las posibles combinaciones de valores que las variables objeto de estudio pueden tomar, descartando aquellas incompatibles en la práctica. Cada combinación representa un posible escenario futuro, es decir, una situación plausible que puede llegar a darse. La variante conocida como MORPHOL evalúa un escenario mediante la probabilidad de que ocurra finalmente. Aquí se revisa una mejora publicada recientemente que considera otros criterios distintos a la probabilidad para evaluar un escenario usando valores lingüísticos. Además, se proponen mejoras futuras y nuevas líneas que están actualmente siendo investigadas, como clustering de los escenarios generados y obtención de descripciones lingüísticas de cada escenario, con ejemplos sobre un caso de estudio real.

Keywords: Planificación de Escenarios, prospectiva tecnológica, escenarios, análisis morfológico, etiquetas lingüísticas, clustering, decisión multicriterio, cálculo con palabras.

1. Introducción

En un mundo en continuo cambio como el actual, las grandes compañías son conscientes de la necesidad de buenas prácticas para adelantarse a la competencia y a los cambios, estableciendo cuáles son las medidas y decisiones adecuadas en el presente para adaptarse a estos cambios. La Planificación de Escenarios constituye un conjunto de herramientas orientadas a la estimulación del pensamiento creativo de los directivos de una empresa para considerar, de manera sistemática, la mayor cantidad posible de posibles situaciones futuras, conocidas como escenarios. Un escenario es una descripción de una situación plausible que puede darse en el futuro, reflejando diversas perspectivas de desarrollo. Los escenarios deben servir como base para actuar tomando decisiones en el presente [9].

Uno de los métodos usados en Planificación de Escenarios es el Análisis Morfológico (AM) [8], concebido como una técnica de resolución de problemas con variables difíciles de cuantificar. La finalidad de este tipo de análisis es, a grandes rasgos, la consideración de todas las posibles soluciones a un problema. Una de las variantes del AM más comúnmente usadas en Planificación de Escenarios es MORPHOL [5]. Dada una serie de variables de interés en el problema, junto al conjunto (finito) de valores que puede tomar cada una, se generan todas las posibles combinaciones en las que cada variable toma uno de sus posibles valores. Cada una de estas combinaciones es un escenario que puede llegar a darse. MORPHOL evalúa cada escenario en base a la probabilidad de que finalmente ocurra, calculada como el producto de las probabilidades de que cada variable tome el valor considerado en ese escenario. Dichas probabilidades marginales son proporcionadas por expertos. De entre todas las combinaciones generadas y evaluadas, sólo aquellas que se estiman más probables son retenidas para posteriores análisis, mientras que el resto -la gran mayoría- son descartadas.

MORPHOL presenta algunos inconvenientes, principalmente la dificultad que entraña para los expertos expresar su opinión en forma de una distribución de probabilidad, más aún cuando lo que manifiestan son probabilidades que estiman sobre hechos que no han ocurrido aún. Sería deseable facilitarles la labor permitiéndoles dar juicios lingüísticos sobre estas probabilidades, es decir, usando expresiones como *Muy probable, Casi imposible, Poco probable*, etc. Por otro lado, resultaría interesante analizar escenarios que, siendo plausibles, no están entre las más probables, puesto que podrían ser útiles para generar otras ideas, realizar cambios en las políticas para intentar que se conviertan en muy probables, analizar por qué no son tan probables, etc. Por tanto, debería poderse evaluar un escenario en términos de otros criterios, tales como deseabilidad, catastroficidad, grado de compatibilidad de sus variables... y, siguiendo con la línea anterior, los expertos también deberían poder evaluar estos criterios en lenguaje natural. Asimismo, puesto que se usarán simultáneamente varios criterios, se hace necesario el uso de algún método específico de decisión multicriterio para ordenar los escenarios en base a una relevancia global.

Para resolver estos inconvenientes, se presentó recientemente un método de Análisis Morfológico lingüístico multicriterio [10] como extensión a MORPHOL, cuyos primeros resultados son prometedores. En esta contribución, ampliaremos las posibilidades de dicha propuesta y para ello describiremos líneas que se están investigando, tales como clustering de los escenarios generados y cálculo de salidas lingüísticas como resultado de nuestro método, mostrando de esta forma las posibilidades que ofrecen dentro de este tipo de estudios prospectivos.

La estructura del trabajo es la siguiente. En la sección 2 se explica el método de AM y en particular MORPHOL, mientras que en la sección 3 se motiva la necesidad de un enfoque más robusto y se repasa nuestra propuesta. En las secciones 3.4 y 3.5 explicamos cómo aplicar clustering sobre los escenarios generados y cómo calcular salidas lingüísticas. En la sección 4 aplicamos el método a un caso de estudio y comentamos algunos resultados. Finalmente, la sección 5 se dedica a las conclusiones y trabajos futuros.

2. Análisis Morfológico y Método MORPHOL

Explicaremos los pasos del AM [8] mediante un ejemplo de diseño de un automóvil. El objetivo será la obtención de todas las posibilidades de diseño.

1. Identificar las variables de interés: *Cambios, Motor, y Fuente de energía*
2. Identificar los posibles valores (categóricos) que puede tomar cada variable: *Cambios*={*Manual (M), Automático (A)*}, *Motor*={*Combustión (C), Eléctrico (E), Híbrido (H)*} y *Fuente de energía*={*Gasolina (G), Diesel (D), Electricidad (L)*}.
3. Identificar las inconsistencias entre valores de variables distintas y formalizarlas en una matriz de consistencias cruzadas como la mostrada a la derecha. Por ejemplo, *un motor eléctrico (E) no es compatible con una fuente de energía basada en gasolina (G) ni en combustible diésel (D).*

		Cambios		Motor		
		M	A	C	E	H
Motor	C	3	3			
	E	2	3			
	H	2	3			
Fuente de energía	G	3	3	3	1	2
	D	3	3	3	1	3
	L	3	3	1	3	3

4. Generar todas las combinaciones posibles (soluciones al problema). Una solución es una situación en la cual cada variable toma uno de sus valores posibles. Ej: *(Manual, Combustión, Diesel)*, *(Automático, Híbrido, Diesel)*. En el ejemplo, se generarían $2 \times 3 \times 3$ combinaciones. A continuación, las que son factibles por contener valores incompatibles entre sí son descartadas.

Los valores de la matriz de consistencias indican el grado de consistencia o compatibilidad entre cada par de valores de dos variables distintas, donde un 1 indica incompatibilidad; un 2, compatibilidad parcial; y un 3, compatibilidad total. El grado de compatibilidad puede deberse a motivos lógicos, prácticos, legales, etc. En el ejemplo anterior, debido a las incompatibilidades de motor eléctrico con fuente de energía gasolina o diesel, se descartarán todas las soluciones de la forma *(·, E, G)* ó *(·, E, D)* donde "·" indica una indiferencia en el valor de la primera variable (*Cambios*).

El AM es un método interactivo para explorar las posibilidades, de manera que el usuario puede fijar los valores de ciertas variables y generar todas las combinaciones de las demás variables que son compatibles con los valores fijados. En cambio, el método MORPHOL sí evalúa cada escenario factible mediante la probabilidad de que finalmente ocurra. Para ello, un grupo de expertos proveen, para cada variable, una distribución de probabilidad sobre sus posibles valores (los cuales, en la terminología de Godet [5], se denominan *hipótesis* que puede tomar esa variable). Asumiendo que estas probabilidades de ocurrencia son independientes entre sí, la probabilidad de un escenario es el producto de las probabilidades de que cada variable tome el valor indicado en ese escenario. Volviendo al ejemplo de diseño del automóvil, la probabilidad de que ocurra la

combinación (M, H, E) se calculará como

$$P(Cambios = M, Motor = H, F.energia = E) =$$
$$= P(Cambios = M) \cdot P(Motor = H) \cdot P(F.energia = E)$$

Más allá de la validez de la independencia entre las probabilidades de que las variables tomen cada hipótesis, es claro que MORPHOL presenta inconvenientes. En primer lugar, es tedioso obligar a los expertos a expresar su conocimiento y opinión sobre el futuro en forma de distribución de probabilidad. El ser humano suele ser reticente a manifestar su pensamiento usando números exactos. Sería deseable que se pudiera expresar la probabilidad en términos lingüísticos [12], como *Muy probable, Casi imposible*, etc. Por otro lado, evaluar un escenario sólo en base a su probabilidad es un enfoque muy pobre. Podríamos estar obviando escenarios que, aun no estando entre los más probables, sería interesante considerar por otros motivos que también pueden ser de utilidad para el estudio. Por tanto, sería beneficioso introducir otros criterios de juicio, también lingüísticos, como se explicará en la siguiente sección. En ambos casos, se recurre a las técnicas de Cálculo con Palabras (CP) para elaborar un algoritmo.

3. Análisis Morfológico lingüístico multicriterio

Proponemos una extensión al método MORPHOL con las siguientes características principales.

3.1. Probabilidad lingüística

Modelamos la probabilidad de cada hipótesis como una variable lingüística [12]. Para capturar la incertidumbre en torno a sus valores, utilizaremos Números Difusos Trapezoidales (NDTr), representados como tuplas de números reales $\tilde{A} = (a, b, c, d), \quad a \leq b \leq c \leq d$. Cada etiqueta tendrá asociado un NDTr que se utilizará para realizar cálculos. Los valores concretos para cada etiqueta se muestran en [10] y fueron elicitados en un estudio psicológico previo [2]. A título informativo, se permiten las siguientes etiquetas para evaluar la probabilidad de ocurrencia de cada hipótesis de las variables: *Casi imposible, Muy poco probable, Poco probable, Es posible, Bastante probable, Muy probable, Casi seguro.*

A pesar de usar términos lingüísticos, estamos modelando una distribución de probabilidad, y por tanto debe cumplirse que la suma de las probabilidades lingüísticas asignadas a las hipótesis de una variable sea 1. Siguiendo el enfoque de [7], si llamamos $\tilde{\pi}_i$ al NDTr que nos da la probabilidad de la hipótesis i-ésima de una variable que puede tomar I hipótesis distintas, tiene que cumplirse[1] que $\oplus_{i=1}^{I} \tilde{\pi}_i \supseteq 1_\chi$ donde $1_\chi(x) = 1$ si $x = 1$, y 0 en otro caso ($1_\chi(x)$ es la función de pertenencia del número 1 crisp), y donde $\oplus$ indica una suma difusa. Por tanto, los expertos deben comprobar que se cumple esto al expresar las probabilidades de las hipótesis usando etiquetas lingüísticas.

[1] $\tilde{A} \supseteq \tilde{B} \leftrightarrow \mu_{\tilde{A}}(x) \geq \mu_{\tilde{B}}(x) \forall x \in \mathbb{R}$

Se utilizarán las operaciones habituales para NDTr, componente a componente (cruzadas en el caso de la división). Para calcular la probabilidad lingüística conjunta de un escenario con n variables cuyas probabilidades marginales de sus hipótesis son $(\tilde{A}_1, ..., \tilde{A}_n)$, haremos $\tilde{P} = \otimes_{i=1}^{n} \tilde{A}_i = (\prod_i a_i, \prod_i b_i, \prod_i c_i, \prod_i d_i)$.

3.2. Múltiples criterios para evaluar un escenario

Además de la probabilidad, sugerimos emplear los siguientes criterios en la evaluación de un escenario, que pueden ser útiles para descubrir escenarios interesantes aunque no estén entre los más probables:

1. Compatibilidad: a pesar de que los escenarios incompatibles se eliminan, en los restantes se puede distinguir entre escenarios parcial o totalmente compatibles. Un escenario parcialmente compatible será seguramente menos factible y por tanto menos relevante que uno con total compatibilidad.
2. Deseabilidad: un escenario deseable es aquel que deberíamos intentar alcanzar ya que representa una situación favorable para nuestros intereses.
3. Catastroficidad: indica escenarios peligrosos que deben ser evitados ya que implican serios riesgos. Este criterio no es opuesto a la deseabilidad puesto que, aunque un escenario no sea deseable, no tiene por qué implicar una amenaza seria para nuestros intereses.

Para evaluar estos criterios es posible utilizar cualquier conjunto de términos que modele adecuadamente la incertidumbre de cada juicio lingüístico en el problema que se desee resolver. Proponemos utilizar los términos {No compatible (N) = (0, 0, 0, 1), Parcialmente compatible (P) = (0, 1, 1, 2), Totalmente compatible (T) = (1, 2, 2, 2)} para evaluar la compatibilidad, y los términos {Muy baja (MB) = (1, 1, 1, 2), Baja (B) = (1, 2, 2, 3), Moderada (M) = (2, 3, 3, 4), Alta (A) = (3, 4, 4, 5), Muy alta (MA) = (4, 5, 5, 5)} para la deseabilidad y la catastroficidad.

En un problema con n variables, la deseabilidad y catastroficidad agregadas difusas de un escenario s se calculan como:

$$\tilde{D}eseab_s = \frac{1}{n}\big(\oplus_{j=1}^{n} \tilde{D}eseab_s^j\big); \quad \tilde{C}atast_s = \frac{1}{n}\big(\oplus_{j=1}^{n} \tilde{C}atast_s^j\big);$$

donde $\tilde{D}eseab_s^j$ y $\tilde{C}atast_s^j$ son los NDTr de la deseabilidad y catastroficidad de la hipótesis que toma la variable j en s. La comaptibilidad agregada difusa se calcula como

$$\tilde{C}ompat_s = \sqrt[r]{\big(\otimes_{i<j} \tilde{C}ompat_s^{ij}\big)}; \qquad r = \binom{n}{2} = \frac{n!}{2!(n-2)!}$$

siendo $\tilde{C}ompat_s^{ij}$ la compatibilidad difusa entre las hipótesis que toman las variables i y j en s, y r el número de compatibilidades mutuas entre hipótesis que deben comprobarse en cada escenario. Los escenarios cuya compatibilidad agregada es de la forma $(0, 0, 0, \cdot)$ se descartan, ya que significa que alguna de las compatibilidades mutuas que se multiplicaron era igual a *No compatible*.

3.3. Método TOPSIS difuso para la evaluación de escenarios

Para considerar simultáneamente toda la información de varios criterios en la evaluación de los escenarios, se hace necesaria la utilización de algún método de decisión multicriterio que la agregue de forma adecuada y obtenga un ranking que permita ordenarlos, de una manera más razonable que la ordenación en base solamente a su probabilidad de aparición.

Proponemos la utilización del método TOPSIS lingüístico, cuyo desarrollo detallado puede encontrarse en [3,10]. A grandes rasgos, permite a varios expertos evaluar, utilizando términos lingüísticos, una serie de alternativas en base a un conjunto de criterios, a cada uno los cuales también se les puede dar un peso lingüístico que los pondere según su importancia. El resultado es una ordenación de las alternativas según su bondad global.

En este caso, los expertos evaluarían, por separado y de forma lingüística, la probabilidad, deseabilidad, catastroficidad y compatibilidad de las hipótesis de cada variable. En caso de que exista más de un experto, se calculará el valor agregado de todos los expertos para cada hipótesis (teniendo en cuenta la importancia relativa que han dado a cada criterio). A continuación, estos valores obtenidos se utilizan para calcular, para cada escenario generado, el valor agregado de probabilidad, deseabilidad, catastroficidad y compatibilidad, tal como se indicó en la sección anterior. Los escenarios con valor de compatibilidad de la forma $(0, 0, 0, \cdot)$ se descartan, y los restantes son suministrados como entrada a TOPSIS. Por tanto, las alternativas que TOPSIS evalúa son los escenarios compatibles, y asigna a cada escenario un número real entre 0 y 1 que indica su bondad global. Ordenando decrecientemente los escenarios por este valor se obtiene un ranking.

3.4. Salidas lingüísticas

Como extensión novedosa a lo ya presentado, proponemos en primer lugar el cálculo de salidas lingüísticas para expresar el valor de cada criterio en un escenario. Hasta ahora sólo se calculaba el valor como un NDTr pero no se le asignaba una etiqueta para describir, en términos absolutos lingüísticos, el grado de probabilidad, compatibilidad, deseabilidad o catastroficidad.

El primer paso es determinar un conjunto de términos de referencia para la salida. En el caso de la probabilidad, coincide con el conjunto de términos usados para las probabilidades lingüísticas de entrada, ya que el producto de probabilidades es otra probabilidad. Para el resto de criterios, el usuario puede elegir el número de términos $\bar{N}$ que desea para la salida en función de la granularidad que necesite, siempre que $\bar{N}$ sea un número impar de etiquetas. Para calcular este conjunto de términos de salida estableceremos, para las etiquetas de entrada más pequeña, más grande y central, un escenario *ideal* que se obtiene cuando todas las hipótesis que intervienen en el escenario son evaluadas con esa etiqueta. Por ejemplo, para la deseabilidad, calculamos tres escenarios ideales:

(a) El escenario ideal de mínima deseabilidad (muy baja), dado por $s_{MB} = (MB, MB, ..., MB)$, cuya deseabilidad agregada es $\tilde{Deseab}_{s_{MB}} = \frac{1}{n} \oplus_i MB = MB$, y tomamos el extremo inferior a del soporte de ese NDTr.

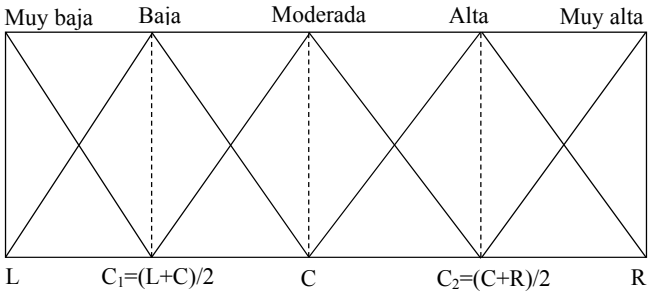


Figura 1: Cálculo del conjunto de términos de salida para deseabilidad y catastroficidad.

(b) El escenario ideal de máxima deseabilidad (muy alta), dado por $s_{MA} = (MA, MA, ..., MA)$, donde $\tilde{D}eseab_{s_{MA}} = MA$, y tomamos el extremo superior d del soporte.

(c) Procedemos de igual forma con la etiqueta central (moderada), y tomamos cualquiera de los puntos centrales b o c del núcleo NDTr resultante.

Si llamamos L, R y C a los tres puntos anteriores, respectivamente, entonces el conjunto de términos de salida se calculará a partir de ellos como muestra la Fig. 1, particularizada para el caso $\bar{N} = 5$ (si sólo se requieren 3, se ignoran las etiquetas Baja y Alta). La idea de calcular el conjunto de términos de salida a partir de escenarios ideales se explica en [11]. Una vez calculado el conjunto de términos de salida, que llamaremos $\Delta = \{\tilde{\delta}_1, ..., \tilde{\delta}_{\bar{N}}\}$, es posible asignar una de estas etiqueta lingüísticas a la deseabilidad o catastroficidad agregada difusa que se había calculado (como NDTr) para cada escenario. Dicha etiqueta será la correspondiente al NDTr más cercano del conjunto Δ, donde la distancia entre dos NDTr $\tilde{A}_1 = (a_1, b_1, c_1, d_1)$ y $\tilde{A}_2 = (a_2, b_2, c_2, d_2)$ la definimos como

$$d(\tilde{A}_1, \tilde{A}_2) = (|a_1 - a_2| + 2|b_1 - b_2| + 2|c_1 - c_2| + |d_1 - d_2|)/6$$

aunque, una vez más, es posible usar cualquier otra medida de distancia que se adapte a las necesidades del problema.

Para la compatibilidad se procede de la misma manera, puesto que para determinar cada escenario ideal para una determinada etiqueta, el producto de r compatibilidades iguales equivale a la potencia r-ésima de la etiqueta, que se cancela con la raíz r-ésima dando lugar a la propia etiqueta.

3.5. Clustering de escenarios

El número de escenarios compatibles obtenidos finalmente suele ser demasiado alto como para analizarlos todos. Puede ser interesante agruparlos en función de sus características comunes, ya sean estructurales (hipótesis que tienen en común) o su valoración por uno o más criterios (escenarios con valores similares en uno o varios criterios, pese a no ser necesariamente parecidos en sus hipótesis). De esta manera se pueden crear escenarios más generales que facilitan el

análisis al reducir de forma considerable el número de alternativas. Esta idea fue aplicada en [4], aunque de forma manual e intuitiva por los expertos.

Se propone el uso de métodos específicos de clustering donde cada escenario es un individuo descrito por un vector de características, que pueden ser sólo las hipótesis que lo componen, o bien las hipótesis junto con la valoración (defuzzificada) de los cuatro criterios, o ambas junto con el valor de bondad (en el rango [0, 1]) otorgado por TOPSIS a ese escenario (llamado *índice de proximidad*; cuanto mayor sea, mejor es una alternativa). En un futuro, también plantearemos la aplicación de técnicas de biclustering para descubrir escenarios que se parecen si sólo nos atenemos a un subconjunto de sus hipótesis y/o criterios. En todo caso, el objetivo no es calcular el mejor agrupamiento sino descubrir información útil para el resto del estudio.

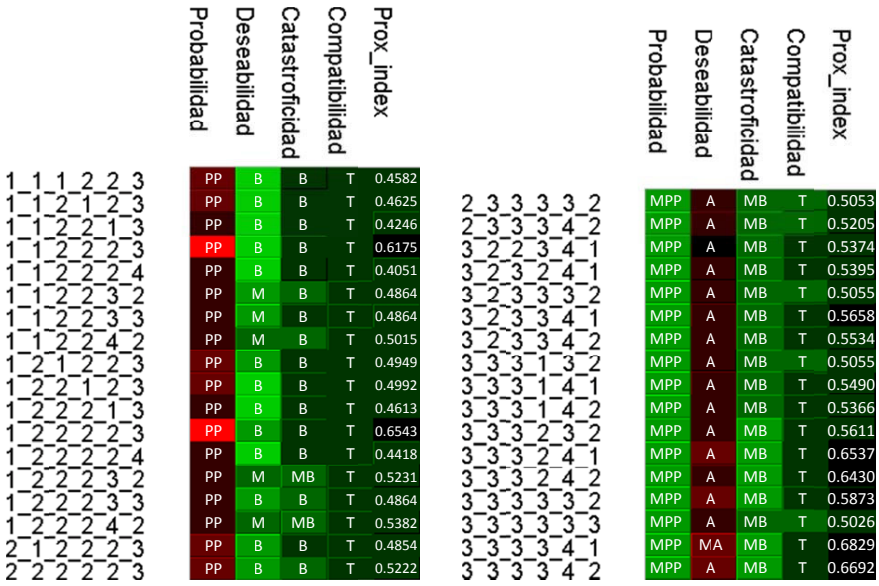
4. Aplicación a un caso de estudio

Hemos aplicado el método descrito a un problema real [6] acerca de la evolución global de factores económicos, sociales y políticos. Se consideran seis variables, de las cuales las cuatro primeras pueden tomar cada una tres posibles hipótesis distintas, y las dos últimas pueden tomar cuatro. La valoración de las hipótesis según cada criterio, y su agregación para cada escenario, está en [10].

Se generaron inicialmente $3 \times 3 \times 3 \times 3 \times 4 \times 4 = 1296$, de los cuales sólo 248 fueron retenidos. El resto se descartaron ya que alguna de las compatibilidades entre pares de hipótesis de un escenario era *No compatible*. Por tanto, el método TOPSIS difuso recibió 248 alternativas, cada una evaluada en base a 4 criterios. Aunque en [10] se muestran experimentos con tres maneras distintas de ponderar la importancia de los criterios, aquí nos centramos en la que asigna un peso (crisp) de 0.4 al criterio de Probabilidad, y 0.2 al resto.

Con el fin de ilustrar el tipo de información que puede obtenerse gracias al cálculo de salidas lingüísticas y clustering de escenarios, mostramos la salida de un típico algoritmo de clustering, K-means, aplicado a los 248 escenarios compatibles. Dado que empleamos K-means clásico, se utilizaron como atributos de cada escenario los valores defuzzificados de cada criterio, junto con el índice de proximidad que TOPSIS difuso les ha asignado. El valor de K no se conoce de antemano y debería optimizarse utilizando criterios estadísticos. En este estudio preliminar, se utilizó la implementación de K-means incluida en la herramienta BicAT [1], con $K = 10$ y distancia euclídea.

El resultado se muestra en la Fig. 2, que sólo contiene dos clusters de entre los diez obtenidos. Se han escogido por ser especialmente interesantes y poseer una interpretación muy intuitiva. Cada escenario viene descrito por las hipótesis que lo componen, numeradas del 1 al 3 en las cuatro primeras variables, y del 1 al 4 en las dos últimas. El color de cada atributo se relaciona con su valor (normalizado para toda su columna). El rojo claro indica valores (defuzzificados) más altos y el verde claro, más bajos. Mostramos también la etiqueta lingüística asignada a cada NDTr antes de la defuzzificación, como se indicó en la sección 3.4.



	Probabilidad	Deseabilidad	Catastroficidad	Compatibilidad	Prox_index
1_1_1_2_2_3	PP	B	B	T	0.4582
1_1_2_1_2_3	PP	B	B	T	0.4625
1_1_2_2_1_3	PP	B	B	T	0.4246
1_1_2_2_2_3	PP	B	B	T	0.6175
1_1_2_2_2_4	PP	B	B	T	0.4051
1_1_2_2_3_2	PP	M	B	T	0.4864
1_1_2_2_3_3	PP	M	B	T	0.4864
1_1_2_2_4_2	PP	M	B	T	0.5015
1_2_1_2_2_3	PP	B	B	T	0.4949
1_2_2_1_2_3	PP	B	B	T	0.4992
1_2_2_2_1_3	PP	B	B	T	0.4613
1_2_2_2_2_3	PP	B	B	T	0.6543
1_2_2_2_2_4	PP	B	B	T	0.4418
1_2_2_2_3_2	PP	M	MB	T	0.5231
1_2_2_2_3_3	PP	B	B	T	0.4864
1_2_2_2_4_2	PP	M	MB	T	0.5382
2_1_2_2_2_3	PP	B	B	T	0.4854
2_2_2_2_2_3	PP	B	B	T	0.5222

(a) Muy probables aunque poco deseables, con índice prox. < 0.5.

	Probabilidad	Deseabilidad	Catastroficidad	Compatibilidad	Prox_index
2_3_3_3_3_2	MPP	A	MB	T	0.5053
2_3_3_3_4_2	MPP	A	MB	T	0.5205
3_2_2_3_4_1	MPP	A	MB	T	0.5374
3_2_3_2_4_1	MPP	A	MB	T	0.5395
3_2_3_3_3_2	MPP	A	MB	T	0.5055
3_2_3_3_4_1	MPP	A	MB	T	0.5658
3_2_3_3_4_2	MPP	A	MB	T	0.5534
3_3_3_1_3_2	MPP	A	MB	T	0.5055
3_3_3_1_4_1	MPP	A	MB	T	0.5490
3_3_3_1_4_2	MPP	A	MB	T	0.5366
3_3_3_2_3_2	MPP	A	MB	T	0.5611
3_3_3_2_4_1	MPP	A	MB	T	0.6537
3_3_3_2_4_2	MPP	A	MB	T	0.6430
3_3_3_3_3_2	MPP	A	MB	T	0.5873
3_3_3_3_3_3	MPP	A	MB	T	0.5026
3_3_3_3_4_1	MPP	MA	MB	T	0.6829
3_3_3_3_4_2	MPP	A	MB	T	0.6692

(b) Muy deseables aunque muy poco probables, con índice prox. > 0.5.

Figura 2: Dos de los clusters encontrados por K-means con K = 10 y distancia euclídea.

El cluster de la Fig.2(a) muestra los escenarios más probables pero menos deseables (aunque poco catastróficos). Nótese que, al calcularse la probabilidad global como producto de seis probabilidades de las hipótesis, el resultado es siempre pequeño; de los 248 escenarios, 132 son Muy Poco Probables (MPP) y son 116 Poco Probables (PP). Aun siendo probables, globalmente son escenarios mediocres con índice de proximidad menor de 0.5.

Por otro lado el cluster de la Fig. 2(b) muestra la situación opuesta: escenarios muy deseables pero poco probables, que obtienen una buena valoración global según TOPSIS porque, pese a su baja probabilidad, tienen valores altos en otros criterios (Alta deseabilidad, Muy Baja catastroficidad). Nótese que en ambos clusters, especialmente en la Fig. 2(b), se ven similitudes entre las hipótesis componentes de los escenarios, aunque el clustering no empleó este hecho.

5. Conclusiones

Se ha descrito un método difuso y lingüístico para el Análisis Morfológico en el ámbito de la Planificación de Escenarios. El empleo de técnicas de Cálculo con Palabras ha permitido suministrar datos expresados de forma lingüística y obtener salidas también lingüísticas, más intuitivas e interpretables. Esto, junto al uso de un método multicriterio para evaluar los posibles escenarios futuros que se han generado, proporciona información más útil que el método MORPHOL

original. Además, hemos mostrado cómo pueden aplicarse técnicas de clustering a las salidas obtenidas, con el fin de descubrir otra información útil acerca de la similitud de escenarios que, sin parecerse necesariamente en las hipótesis que los componen, obtienen valoraciones parecidas; si bien ciertamente existe relación entre ambos enfoques. Los primeros resultados en este sentido son prometedores y serán estudiados en detalle.

En el futuro se experimentará con otras técnicas de clustering y de biclustering para comprobar hasta qué punto son capaces de obtener información a la que no se llegaría con un agrupamiento manual basado en la intuición.

Agradecimientos

Contribución financiada por los proyectos TIN2011-27696-C02-01 del Mº de Economía y Competitividad, P11-TIC-8001 de la Junta de Andalucía, y Fondos FEDER. El primer autor posee una beca FPU del Ministerio de Educación.

Referencias

1. Barkow, S., Bleuler, S., Prelić, A., Zimmermann, P., Zitzler, E.: BicAT: a biclustering analysis toolbox. Bioinformatics 22(10), 1282–1283 (2006)
2. Bonissone, P., Decker, K.: Selecting uncertainty calculi and granularity: An experiment in trading-off precision and complexity. In: Proc. of the Ann. Conf. on Uncertainty in AI. pp. 57 – 66 (1985)
3. Cables, E., García-Cascales, M.S., Lamata, M.T.: The LTOPSIS: An alternative to TOPSIS decision-making approach for linguistic variables. Expert Systems with Applications 39(2), 2119 – 2126 (2012)
4. Coyle, R., Yong, Y.: A scenario projection for the South China Sea: Further experience with field anomaly relaxation. Futures 28(3), 269 – 283 (1996)
5. Godet, M.: From anticipation to action: a handbook of strategic prospective. UNESCO Publishing (1994), capítulo 6
6. Godet, M., Chapuy, P., Comyn, G.: Global scenarios: Geopolitical and economic context to the year 2000. Futures 26(3), 275 – 288 (1994)
7. Halliwell, J., Shen, Q.: Linguistic probabilities: theory and application. Soft Computing 13(2), 169 – 183 (2008)
8. Ritchey, T.: Fritz Zwicky, Morphologie and Policy Analysis. In: Proc. of the EURO Conf. on Operational Analysis (1996)
9. Van Notten, P.: Writing on the wall: Scenario development in times of discontinuity. Ph.D. thesis, University of Maastrich, Masstrich, The Netherlands (2005)
10. Villacorta, P.J., Masegosa, A.D., Lamata, M.T.: Fuzzy Linguistic Multicriteria Morphological Analysis in Scenario Planning. In: Proc. IFSA-NAFIPS (2013), in press
11. Villacorta, P., Masegosa, A., Castellanos, D., Lamata, M.T.: A linguistic approach to structural analysis in prospective studies. In: Greco et al, S. (ed.) Advances in Computational Intelligence, vol. 297, pp. 150–159. Springer (2012)
12. Zadeh, L.A.: The concept of a linguistic variable and its application to approximate reasoning–I. Information Sciences 8(3), 199 – 249 (1975)

Un Algoritmo Genético para aprendizaje de Sistemas de Clasificación Basados en Reglas Difusas dentro del paradigma *one-class*

Pedro Villar[1], Ana María Sánchez[1], Rosana Montes[1], Francisco Herrera[2]

[1]Dpto. de Lenguajes y Sistemas Informáticos,
[2]Dpto. de Ciencias de la Computación e Inteligencia Artificial,
E.T.S. Ing. Informática y de Telecomunicación, Universidad de Granada.
pvillarc@ugr.es,amlopez@ugr.es,
rosana@ugr.es,herrera@decsai.ugr.es

Abstract. En este trabajo se propone un Algoritmo Genético para aprendizaje de la granularidad de un Sistema de Clasificación Basado en Reglas Difusas usado para detección de anomalías. Este problema está englobado dentro de lo que se denomina clasificación con una única clase o *one-class classification*, una situación que está presente en numerosos problemas reales en la que los ejemplos disponibles para el aprendizaje son todos de una misma clase (clase objetivo). El propósito de esta propuesta es obtener un sistema de complejidad reducida, que presente una buena tasa de clasificación sobre los ejemplos de la clase objetivo junto con una buena tasa de detección de los ejemplos anómalos, o ejemplos que no pertenecen a la clase objetivo.

Keywords: *One-Class Classification*, Sistemas de Clasificación Basados en Reglas Difusas, Algoritmos Genéticos.

1 Introducción

Dentro del aprendizaje automático, los métodos tradicionales de clasificación intentan clasificar un objeto desconocido dentro de un conjunto finito y discreto de categorías (o clases). En el paradigma de *one-class classification* [1], una de las clases está suficientemente caracterizada en los ejemplos del conjunto de entrenamiento y se suele denominar clase positiva o clase objetivo (*target*). Para el resto de clases, que caracterizan el concepto negativo (o simplemente la no pertenencia a la clase objetivo) o no hay ejemplos disponibles, o hay tan pocos ejemplos que no son representativos de ese concepto negativo. Una de las primeras aplicaciones de este paradigma es la detección de anomalías (*anomaly detection, outlayer detection*) [2] como puede ser la detección de fallos en una máquina, ya que es fácil encontrar ejemplos de funcionamiento "normal" de la misma mientras que no lo es tanto encontrar ejemplos de funcionamiento "anómalo".

En este trabajo proponemos un método de clasificación para *one-class* utilizando los Sistemas de Clasificación Basados en Reglas Difusas (SCBRDs), que

son una herramienta muy utilizada en el campo del aprendizaje automático ya que proporcionan modelos interpretables para el usuario final [3]. Desde nuestro conocimiento, no hay ninguna propuesta dentro del paradigma de *one-class* que utilice reglas difusas.

Un SCBRD tiene dos componentes principales: el Sistema de Inferencia y la Base de Conocimiento (BC). En un SCBRD lingüístico, la BC está compuesta de la Base de Reglas (BR), constituida por el conjunto de reglas difusas y la Base de Datos (BD), que contiene las funciones de pertenencia de las particiones difusas asociadas a las variables de entrada. Si no existe información experta sobre el problema a resolver, es necesario utilizar algún proceso de aprendizaje automático de la BC a partir de ejemplos [3].

Dentro de la literatura especializada, hay numerosos métodos para el aprendizaje automático de la BC de un SCBRD, tanto para el aprendizaje completo de la BC como para el aprendizaje o ajuste de alguna de sus partes, pero la mayoría están diseñados para problemas multi-clase, especialmente en el caso del aprendizaje de la BR. Por eso, dentro de nuestro método de aprendizaje de la BC proponemos un nuevo método de aprendizaje de reglas difusas específico para *one-class*, que intenta obtener un conjunto reducido de reglas que caractericen bien a la clase objetivo y utilizaremos un sistema de razonamiento difuso diseñado para *one-class*, que nos permite determinar si un nuevo ejemplo desconocido pertenece o no a la clase objetivo.

En trabajos anteriores con sistemas de reglas difusas [4], tanto de clasificación como de regresión, hemos determinado la importancia que tiene la elección de una buena BD, siendo el comportamiento del modelo especialmente sensible al número de etiquetas difusas (granularidad) que componen la partición difusa de cada variable linguística del problema. Por ese motivo, y siguiendo el paradigma de diseño de la BD de un SCBRD de trabajos anteriores [5], se propone un método de aprendizaje integrado de la BC al completo (BD y BR), que se centra en la búsqueda de una buena granularidad para la BD mediante un Algoritmo Genético (AG) que aprende el número de etiquetas de cada variable. Para la generación de la BR se utiliza el algoritmo de aprendizaje de reglas específico para *one-class* mencionado anteriormente.

Para el estudio experimental nos hemos basado en trabajos del paradigma *one-class*, centrados en detección de anomalías [6, 7] que utilizan conjuntos de datos del repositorio UCI [8]. Dichos conjuntos de datos se convierten en problemas de *one-class* al considerar todos los ejemplos de una clase como ejemplos de la clase anómala (*outlier*), eliminándolos del conjunto de entrenamiento y pasándolos al conjunto de datos de test.

El trabajo está estructurado de la siguiente manera. Primero, en la Sección 2 se realiza una breve introducción al paradigma de *one-class classification*, resaltando sus diferencias frente a la clasificación multi-clase convencional. A continuación, en la Sección 3 se detalla el algoritmo de generación de reglas difusas junto con el método usado para clasificar los ejemplos y en la Sección 4 presentamos los aspectos fundamentales del AG que se propone. La Sección 5

muestra el estudio experimental realizado. Finalmente, las conclusiones de este trabajo se exponen en la Sección 6.

2 Clasificación con una única clase

El término *one-class classification* fue propuesto por primera vez en [9], aunque varios autores han utilizado otros términos para designar paradigmas similares como son detección de anomalías (*outlayer detection* [2] o *anomaly detection* [10]), detección de novedades (*novelty detection* [11]) o aprendizaje de conceptos (*concept learning* [12]). Esa variación de términos tiene que ver más con las diferentes aplicaciones en las que se ha utilizado el paradigma de *one-class* que con diferencias en los modelos utilizados o en la forma de diseñarlos.

Dentro de la tradicional clasificación multi-clase, los métodos de aprendizaje automático se encuentran con una serie de problemas como la reducción de la tasa de error de clasificación, la capacidad de generalización del modelo, la complejidad del mismo, el tratamiento de la dimensionalidad, etc. Dentro del paradigma *one-class* esos problemas siguen apareciendo y, en algunos casos, son más difíciles de abordar. Por ejemplo, determinar una frontera entre los datos que nos permita separar las clases es más difícil cuando sólo se disponen ejemplos de una clase, ya que no tenemos información sobre la "distancia" a la que se encuentran los ejemplos anómalos en cada dirección y es difícil decidir cómo de ajustada a los ejemplos positivos debe estar esa frontera. Por tanto, es de esperar que se requieran más ejemplos para el aprendizaje que en una clasificación convencional.

Una revisión de trabajos de *one-class*, junto con una taxonomía de los mismos se puede encontrar en [13]. En dicho artículo, la taxonomía distingue tres categorías para agrupar los trabajos publicados sobre *one-class*:

- Disponibilidad de ejemplos para el conjunto de entrenamiento: dentro de este criterio se distinguen dos posibilidades, que sólo se disponga de ejemplos de la clase positiva (incluyendo o no muy pocos ejemplos de la clase anómala) o que también se disponga de ejemplos sin clasificar, entrando un poco dentro del paradigma de aprendizaje semi-supervisado.
- Metodología/algoritmos utilizados: en este caso se distinguen los algoritmos basados en máquinas vectores de soporte para *one-class* (*One-Class Support Vector Machines*, OCSVM), ya que muchas de las propuestas en la literatura especializada siguen este paradigma, y el resto de algoritmos o algoritmos no basados en OCSVM.
- Dominio de aplicación: donde se distingue el campo de clasificación de textos/documentos (ámbito muy representado dentro de los trabajos de *one-class*) y otros ámbitos de aplicación.

Debido a la gran cantidad de propuestas basadas en máquinas vectores de soporte, hemos seleccionado una implementación clásica de OCSVM [14] como algoritmo para comparar el rendimiento de nuestra propuesta.

Algunas propuestas de *one-class* presentes en la literatura especializada [6, 7] utilizan conjuntos de datos multi-clase como *benchmarks* para validar los resultados de sus propuestas. La forma usual de actuar supone convertir a dichos conjuntos de datos en problemas binarios (con dos clases), agrupando todas los ejemplos de todas las categorías excepto una dentro de una clase y los ejemplos de la categoría restante en la otra clase. A continuación, se divide el conjunto de datos total en un conjunto de datos de entrenamiento y otro de test, de forma que el conjunto de datos de entrenamiento sólo dispone de un porcentaje de ejemplos representativos de una de las dos nuevas clases (clase objetivo) y el conjunto de datos de test lo constituye el resto de ejemplos de la clase objetivo y todos los ejemplos representativos de la otra clase (clase anómala). Dentro de este esquema se distinguen dos posibilidades:

- Los ejemplos de la categoría que contiene una única clase del conjunto original constituyen el conjunto de datos de la clase positiva mientras que los ejemplos del resto de categorías caracterizarían a la clase anómala, lo que se corresponde con el paradigma de aprendizaje de conceptos.
- Los ejemplos de la categoría que contiene una única clase del conjunto original constituyen la clase anómala mientras que los ejemplos del resto de categorías serían los representantes de la clase objetivo, que se correspondería con el paradigma de detección de anomalías. Esta opción es la que hemos seguido en nuestro trabajo utilizando tres conjuntos de datos del repositorio UCI, usados en algunos de los trabajos que siguen este procedimiento [6, 7].

Obviamente, para el caso de conjunto de datos binarios (dos clases), ambos enfoques son el mismo y sólo cabría hacer consideraciones sobre la diferencia entre escoger una clase u otra como clase positiva en el caso de que el conjunto de datos fuese no balanceado.

La mayoría de propuestas para aprendizaje automático de clasificadores multi-clase utilizan alguna medida de precisión del modelo como el porcentaje de ejemplos bien clasificados. Sin embargo, cuando no se dispone de ejemplos de una clase para entrenar el modelo, ese tipo de medidas pueden llevar a conclusiones erróneas cuando se trabaja con el conjunto de datos de test, ya que es necesario encontrar un equilibrio entre el porcentaje de acierto en la clasificación de ejemplos de la clase positiva (único valor de este tipo que se podría aplicar dentro del proceso de aprendizaje) y el porcentaje de acierto sobre ejemplos de clase anómala (que sí estan disponibles en el conjunto de datos de test). En varios trabajos que siguen el mismo esquema de esta propuesta [6, 7] se utiliza para el conjunto de datos de test la medida denominada área bajo la curva o *Area Under the Curve* (AUC) [15], que se define como:

$$AUC = \frac{1 + TP_{rate} - FP_{rate}}{2} \tag{1}$$

donde TP_{rate} es el porcentaje de ejemplos de la clase objetivo bien clasificados y FP_{rate} es el porcentaje de ejemplos de la clase anómala mal clasificados (expresados en tanto por uno).

En este trabajo utilizaremos, aparte del AUC, los porcentajes de clasificación correcta para el análisis de los resultados experimentales.

3 Generación de la BR y clasificación de nuevos ejemplos

En esta sección describiremos el algoritmo que se ha usado para generar la BR, que está diseñado específicamente para el paradigma *one-class*, además de detallar el método de razonamiento difuso que, una vez determinada la BR, decide si un nuevo ejemplo pertenece a la clase positiva o no.

3.1 Algoritmo de generación de las reglas difusas

La BR está compuesta por un conjunto de reglas difusas que, en este trabajo, tienen la siguiente estructura para un problema con N variables:

$$R_k: \text{ If } X_1 \text{ is } A_1^k \text{ and } \ldots \text{ and } X_N \text{ is } A_N^k \text{ then } Y \text{ is } C$$

donde $X_1, \ldots, X_N$ son las variables consideradas en el problema y $A_1^k, \ldots, A_N^k$ son las etiquetas lingüísticas que representan los valores que toma cada variable. En este tipo de reglas, el antecedente representa un subespacio del espacio completo de búsqueda y el consecuente es la etiqueta de la clase positiva del problema (C).

El método de generación de reglas intenta generar un conjunto de reglas compacto, que caracterice de forma ajustada a los ejemplos de la clase positiva del conjunto de entrenamiento mediante la selección de reglas lo más densas posible (que agrupen a una mayor cantidad de ejemplos). El algoritmo de generación del conjunto de reglas difusas que constituyen la BR sólo decide cuáles son los antecedentes de las reglas (ya que el consecuente sería el mismo para todos, la clase positiva) y su funcionamiento se describe a continuación:

Primeramente se procede a la construcción de la BD mediante la generación de las particiones difusas de cada una de las variables del problema. Para ello, se divide el dominio de la variable entre el número de etiquetas correspondiente, generándose una partición difusa uniforme con etiquetas linguísticas definidas por conjuntos difusos triangulares.

Cada ejemplo del conjunto de entrenamiento (todos representativos de la clase positiva), tiene la siguiente estructura:

$$\{e_1, e_2, ..., e_N\}$$

Una vez establecida la BD, para cada uno de esos ejemplos se realizan las acciones siguientes:

- Seleccionar el mejor antecedente posible para ese ejemplo, buscando la etiqueta de cada variable con máximo grado de pertenencia difuso para la componente de dicha variable en el ejemplo, es decir:
  Seleccionar $A_1^k, .., A_N^k$ tal que $\mu_{A_i^k}(e_i)$ sea el máximo para $k = 1, ..v_i$, siendo v_i el número de etiquetas de la partición linguística de la variable i.

– Comprobar si esa combinación de antecedentes se ha explorado anteriormente con otro ejemplo. Si no ha sido así, se decide si se acepta o no la regla correspondiente en base al valor de dos medidas calculadas sobre dicha regla:
  - **Media_cubrimiento**, que es la media de los grados de emparejamiento de todos los ejemplos del conjunto de entrenamiento cuyo grado de emparejamiento con dicha regla sea mayor que cero. Para calcular el grado de emparejamiento de un ejemplo con el antecedente de la regla se utiliza la t-norma del mínimo.
  - **Soporte**, que es la fracción de ejemplos cubiertos por la regla (con grado de emparejamiento mayor que cero) respecto del total de ejemplos de entrenamiento.

  El propósito de utilizar estas dos medidas es generar un número compacto de reglas con elevad valor medio en el cubrimiento de los ejemplos que emparejan con ellas y que soporten a muchos ejemplos, por lo que un antecedente explorado se aceptará como regla de la BR si cumple los siguientes dos requisitos:
  - $Media_cubrimiento > umbral_cubrimiento$
  - $Soporte > umbral_soporte$

  Siendo $umbral_cubrimiento$ y $umbral_soporte$, dos parámetros del método con valores entre 0 y 1.

3.2 Mecanismo de razonamiento difuso para clasificar nuevos ejemplos

Una vez determinada la BR, el mecanismo de inferencia que decide si un ejemplo nuevo se clasifica o no como perteneciente a la clase positiva consta de dos pasos:

– Se seleccionan todas las reglas con grado de emparejamiento mayor que cero para ese ejemplo (considerando la t-norma del mínimo) y se calcula el valor medio de dichos grados de emparejamiento.
– Si dicho valor medio es mayor o igual que el parámetro $umbral_regla$ (con valores entre 0 y 1), el ejemplo se clasifica como perteneciente a la clase positiva. En caso contrario, se considera un ejemplo "anómalo".

4 Algoritmo Genético para aprendizaje de la granularidad de la BD

En este apartado se propone un método de aprendizaje automático de la BC de un SCBRD mediante un AG generacional que permite definir el número de etiquetas que tendrá la partición difusa de cada variable (granularidad). El proceso de evaluación de las posibles soluciones incluye el método de generación de la BR explicado en la sección anterior. Notaremos nuestra propuesta como AG-Gr-OC (Algoritmo Genético de aprendizaje de la Granularidad para *One-Class*). Los siguientes apartados describen los principales componentes de AG-Gr-OC.

4.1 Codificación de las soluciones

Para un problema de clasificación de N variables de entrada, el número de términos linguísticos de cada variable se codifica en un vector entero de longitud N. En este trabajo, los posibles valores considerados son el conjunto $\{2 \dots 7\}$.

4.2 Población inicial

La población inicial está dividida en dos grupos. El primero de ellos tiene g individuos, siendo g el número de posibles valores para la granularidad (en nuestro caso, $g = 6$) y todos esos individuos tienen el mismo número de etiquetas en todas sus variables (cada individuo con uno de los posibles valores). El segundo grupo contiene el resto de cromosomas y su inicialización es totalmente aleatoria.

4.3 Evaluación de los cromosomas

La evaluación de un cromosoma consta de varias etapas:

1. Generar la BD utilizando la información del cromosoma. Para cada variable se construye una partición difusa uniforme con el número de etiquetas codificado en el cromosoma utilizando funciones de pertenencia triangulares.
2. Generar la BR a partir de la BD mediante el proceso de generación de reglas descrito en la Sección 3.
3. De acuerdo al mecanismo de inferencia descrito en la sección anterior, clasificar todos los ejemplos del conjunto de aprendizaje. El valor de la función de evaluación será el número de ejemplos del conjunto de entrenamiento correctamente clasificados. El AG intentará maximizar dicho valor a lo largo de las generaciones.

4.4 Operadores genéticos

El mecanismo de selección empleado ha sido el Muestreo Universal Estocástico, propuesto por Baker en [16], incorporando una selección elitista. Los restantes operadores genéticos considerados son los siguientes:

Cruce Utilizamos el cruce clásico en un punto. Se escoge aleatoriamente el punto de cruce p (entre el rango de valores $\{2 \dots N\}$) y los dos padres se cruzan sobre la variable p.

Mutación El operador de mutación seleccionado es el propuesto por Thrift en [17]. En el gen al que corresponde mutar se cambia el valor de número de etiquetas que tiene por el inmediatamente superior o inferior (esta última decisión se toma aleatoriamente). Cuando el valor que se pretende cambiar es el mínimo (2) o el máximo (7), se realiza el único cambio posible.

5 Estudio experimental

Vamos a analizar el comportamiento de AG-Gr-OC en tres conjuntos de datos del repositorio UCI [8]: *Pima*, *Glass* y *Wisconsin breast cancer*, utilizados en otros trabajos de *one-class* para detección de anomalías [6, 7]. Para convertir esos tres problemas en problemas del paradigma *one-class*, se ha seleccionado siempre la clase menos representada dentro del conjunto de datos como clase anómala y el resto de clases se han unido en una única característica que es la clase objetivo. Se ha realizado una validación cruzada con 5 particiones distintas, por lo que se han utilizado las particiones disponibles dentro del repositorio KEEL [18], que están disponibles en la página web del proyecto KEEL (`http://www.keel.es/dataset.php`). Como dichas particiones estaban preparadas para clasificación multi-clase, el procentaje de ejemplos en cada conjunto de entrenamiento no es del 80% ya que se han eliminado todos los ejemplos de la clase anómala y se han pasado a los respectivos conjuntos de test, que contienen más del 20% de ejemplos. La Tabla 1 tiene la información básica sobre los conjuntos de datos.

Table 1. Descripción resumida de los conjuntos de datos utilizados

Conj. de datos	#Ej.	#Var.	#Clases	Clase anómala
Glass1	214	9	7	6
Wisconsin	683	9	2	malignant
Pima	768	8	2	tested-positive

Con objeto de analizar el comportamiento de nuestra propuesta, comparamos los resultados obtenidos por AG-Gr-OC con los obtenidos por un método clásico del paradigma *one-class*, como son las máquinas vectores de soporte para *one-class* (OCSVM) [14].

Los parámetros específicos del AG del método AG-Gr-OC se muestran a continuación, siendo N el número de variables del problema:

- Número de evaluaciones: $500 \cdot N$
- Tamaño de población: 100
- Probabilidad de cruce: 0.6
- Probabilidad de mutación: 0.2
- Parámetro *umbral_cubrimiento*: 0.07
- Parámetro *umbral_soporte*: 0.1
- Parámetro *umbral_regla*: 0.18

La Tabla 2 muestra los resultados obtenidos por los dos métodos (media de las cinco particiones que componen la validación cruzada). Para el caso del conjunto de entrenamiento se muestra solamente la tasa de acierto en clasificar la clase positiva mientras que para el conjunto de test se muestra la tasa de clasificación correcta total, las tasas para cada una de las dos clases (siendo CP la clase positiva y CA la clase anómala) y el AUC. Para el caso de AG-Gr-OC se muestra también la media del número de reglas de los modelos obtenidos.

Table 2. Resultados detallados para los métodos analizados

Método	Conjunto de datos	Entrenamiento	Test				Número de reglas
		$Acierto_{CP}$	$Acierto_{CP}$	$Acierto_{CA}$	$Acierto_{Total}$	AUC	
OCSVM	Glass	0.0	0.0	1.0	0.171	0.5	–
	Wisconsin	0.583	0.598	1.0	0.891	0.799	–
	Pima	0.489	0.046	0.983	0.728	0.514	–
	Media	0.357	0.215	0.994	0.596	0.604	–
AG-Gr-OC	Glass	0.771	0.585	1.0	0.659	0.792	29.4
	Wisconsin	0.967	0.969	0.625	0.950	0.956	10.0
	Pima	0.718	0.696	0.943	0.645	0.661	31.4
	Media	0.818	0.750	0.856	0.751	0.803	23.6

Como puede observarse, AG-Gr-OC obtiene mejores resultados que OCSVM en AUC sobre el conjunto de test, que es la medida más adecuada en este entorno ya que la detección correcta de la mayoría de las anomalías no debe derivar en bajos porcentajes de clasificación de nuevos ejemplos de la clase objetivo. De hecho, OCSVM sólo mejora a AG-Gr-OC en la tasa de detección de anomalías sobre el conjunto de test. Cabe destacar también que el número medio de reglas obtenido por los SCBRD generados por AG-Gr-OC es los suficientemente pequeño como para que sea efectiva una ventaja de los sistemas de reglas difusas: la interpretabilidad del modelo.

6 Conclusiones

En este trabajo se ha presentado un sistema de Clasificación Basado en Reglas Difusas dentro del paradigma *one-class* utilizando un Algoritmo Genético para determinar la granularidad más adecuada y un método específico para la generación de las reglas difusas.

Los resultados de esta primera propuesta son prometedores y esperamos mejorarlos añadiendo una selección de variables al método y reduciendo la dependencia del mismo de los parámetros umbral utilizados.

Agradecimientos

Este trabajo está soportado por el Ministerio de Ciencia e Innovación en el marco del proyecto TIN2011-28488 y por la Junta de Andalucía en el marco del proyecto P10-TIC-06858.

References

1. D. Tax and R. Duin. Uniform object generation for optimizing one-class classifiers. *Journal of Machine Learning Research* 2:155-173 2001.
2. G. Ritter and M. Gallegos. Outliers in statistical pattern recognition and an application to automatic chromosome classification. *Pattern Recognition Letters* 18:525-539 1997.

3. H. Ishibuchi, T. Nakashima, and M. Nii. *Classification and modeling with linguistic information granules: Advanced approaches to linguistic Data Mining.* Springer-Verlag, 2004.
4. O. Cordón, F. Herrera, and P. Villar. Analysis and guidelines to obtain a good uniform fuzzy partition granularity for fuzzy rule-based systems using simulated annealing. *International Journal of Approximate Reasoning*, 25(3):187–216, 2000.
5. P. Villar, A. Fernández, R. Carrasco and F. Herrera Feature Selection and Granularity Learning in Genetic Fuzzy Rule-Based Classification Systems for Highly Imbalanced Data-Sets. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 20(3):369-397, 2012.
6. G. Cabral, A. Oliveira and C. Cahú. Combining nearest neighbor data description and structural risk minimization for one-class classification. *Neural Computation & Applications*, 18:175–183 2009.
7. J. Tian, H. Gu, C. Gao and J. lian. Local density one-class support vector machines for anomaly detection. *Nonlinear Dynamics*, 64:127–139 2011.
8. A. Asuncion and D.J. Newman. UCI machine learning repository, 2007. University of California, Irvine, School of Information and Computer Sciences. `http://www.ics.uci.edu/~mlearn/MLRepository.html`
9. M. Moya, M. Koch and L. Hostetler. One-class classifier networks for target recognition applications. In *Proc. of World Congress on Neural Networks*, pages 797–801, 1993.
10. V. Chandola, A. Banerjee and V. Kumar. Anomaly detection: A Survey *ACM Computer Surveys*, 41(3): Article 15, 2009.
11. C. Bishop. Novelty detection and neural network validation. In *IEEE Proceedings on Vision, Image and Signal Processing, Special Issue on Applications of Neural Networks* 141(4):217-222, 1994.
12. N. Japkowicz. Concept-Learning in the absence of counterexamples: An autoassociation-based approach to classification. In *PhD thesis, New Brunswick Rutgers, The State University of New Jersey*, 1999.
13. S. Khan and M. Madden A Survey of Recent Trends in One Class Classification. In *Lecture notes in Artificial Intelligence*, 6206:188–197, 2010.
14. B. Schlkopf, J. Platt, J. Shawe-Taylor, A. Smola, and R. Williamson. A Survey of Recent Trends in One Class Classification. In *Neural Computation*, 13:1443-1471, 2001.
15. J. Huang and C. X. Ling. Using AUC and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 17(3):299–310, 2005.
16. J. E. Baker. Reducing bias and inefficiency in the selection algorithms. In *Proc. of the Second International Conference on Genetic Algorithms (ICGA'87)*, pages 14–21, Hillsdale, 1987.
17. P. Thrift. Fuzzy logic synthesis with genetic algorithms. In *Proc. Fourth International Conference on Genetic Algorithms (ICGA'91)*, pages 509–513, 1991.
18. J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez and F. Herrera. KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. *Journal of Multi-Valued Logic and Soft Computing* 17(2-3):255-287, 2011.

A Hierarchical Image Segmentation Algorithm

Edwin Zarrazola[1,3], Javier Yáñez[1], Daniel Gómez[2], and Javier Montero[1,4]

[1]Facultad de Matemáticas, Universidad Complutense de Madrid, P de Ciencias 3, 28040 Madrid-Spain
[2]Facultad de Estudios Estadísticos, Universidad Complutense de Madrid, Avda. Puerta de Hierro s/n, 28040 Madrid-Spain.
[3]Instituto de Matemáticas, Universidad de Antioquia, Calle 67, Número 53-108, Medellín-Colombia
[4]Instituto de Geociencas (CSIC-UCM), P de las Ciencias 3, 28040 Madrid-Spain.
{edwzar,jayage,monty}@mat.ucm.es
dagomez@estad.ucm.es

Abstract. In this paper, we develop an efficient and polynomial hierarchical clustering (*unsupervised classification*) algorithm for images. The output of this algorithm shows the cluster evolution in a divisive way: since the first iteration in which all the pixels are in the same group until the last iteration in which all the pixels belong to a singleton cluster.

Keywords: clustering, graph-based segmentation, cluster evolution, image segmentation

1 Introduction

Image segmentation is one of the most important research areas in computer vision and its applications such as pattern recognition, medicine, robotic and industry. These techniques are well known for their utility and complexity.

In general, segmentation algorithms can be classified into the following classes: clustering, edge detection, regions, histogram thresholding and graph-based techniques. The graph-based image segmentation techniques can be categorized into two classes:

1. Supervised image segmentation methods: minimum cut / maximum flow models (see [3, 4, 14, 19]), random walk models (see [1, 8–10]).
2. Unsupervised image segmentation methods: minimum spanning tree (see [5]), normalized cut (see [12, 15–17]), isoperimetric graph partitioning (see [11]).

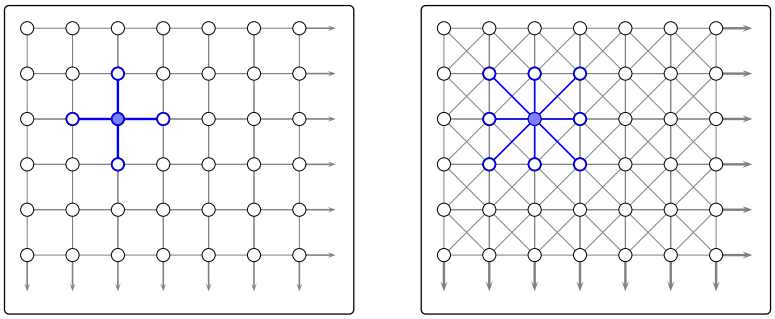
The algorithm shown in this article can be classified as an unsupervised graph-based image segmentation technique. The hierarchical approach defined here shares key elements of the segmentation algorithm proposed in [7] and extended in [6]. In this work we improve the computational complexity of these algorithms.

2 Preliminaries

Let $V = \{P_1, P_2, \ldots, P_n\}$ be the finite set of elements to be clustered. Let $E = \big\{\{P_a, P_b\} \mid P_a,\ P_b \in V\big\}$ be the set of non-ordered pairs of related (neighboring) items of V: if two elements $P_a,\ P_b \in V$ are related, then there exists an edge $e_{ab} = \{P_a, P_b\} \in E$; otherwise, $\{P_a, P_b\} \notin E$. Hence, we define a graph $G = (V, E)$ that shows the relations between the items. The graph G can be assumed to be connected; otherwise, its connected components must be analyzed separately.

Given $e_{ab} = \{P_a, P_b\} \in E$, let $d_{ab} \geq 0$ be the degree of dissimilarity between its endpoints P_a and P_b: the greater d_{ab} is, the more dissimilar P_a and P_b are. This measure is defined taking into account the specific problem and the characteristics of the elements considered. Its construction and properties are beyond the objectives of this paper.

Dealing with images, P_a can represent a single node or a set of nodes through some aggregator operator $\mathcal{A}$ (see [2]). One useful topology for images could be that one where a pixel is linked with four or eight neighbors (see Fig. 1).



(a) (b)

Fig. 1. Two topologies (networks) applied to images: (a) four neighbours, and (b) eight neighbours.

The choice of the 4-connectivity will be used in order to explain our approach. If $D = \big\{d_{ab} \mid e_{ab} \in E\big\}$, then the available information can therefore be summarized by the network

$$N = \big\{\, G\,;\, D \,\big\}. \tag{1}$$

3 Hierarchical Segmentation Algorithm

Given a binary-colored node $P_a \in V$, ($\mathrm{col}(P_a) \in \{0, 1\}$), the color that we assign to any adjacent node $P_b \in V$ depends on the measure $d_{ab} \in D$ compared with a prescribed threshold α:

$$\mathrm{col}(P_b) = \begin{cases} \mathrm{col}(P_a) & \text{if } d_{ab} < \alpha \\ 1 - \mathrm{col}(P_a) & \text{if } d_{ab} \geq \alpha \end{cases} \tag{2}$$

for all $e_{ab} \in E$. Now, for a given α_t, it says that nodes P_a and P_b lie on the same group (denoted $\mathrm{g}_t(P_b) = \mathrm{g}_t(P_a)$) if $e_{ab} \in E$ and $\mathrm{col}(P_b) = \mathrm{col}(P_a)$.

Let us observe that the binary coloring procedure given in (2) is not always consistent: a node can be colored differently depending on the choice of the chain (the sequence of adjacent edges) linking it with the first node. Inconsistent situations are therefore present when there is a cycle in the graph G and a node can be colored differently depending on the chain chosen, starting at the initial node. These cycles will be called *inconsistent* cycles.

An arbitrary spanning tree for a pixel network was defined in [6, 7]. Such a spanning tree allows a binary coloring of G, and inconsistent cycles appear when other edges are added. The number of such inconsistent cycles depends on the chosen spanning tree and the value of the threshold α. In [6, 7], when dealing with inconsistent cycles in real digital images, the large size of these pixel networks was taken into account, and a low ratio of inconsistent cycles was allowed without loss of quality of the segmentation procedure; but we now realize that the particular topology of the pixel network may allow interesting shortcuts in the arrangement of the pixels. Another approach to dealing with inconsistent cycles was introduced in [18], where the least inconsistent spanning tree was defined for any value of α; but this approach has a high computational cost and does not ensure consistency of the binary coloring procedure.

Given a network N as shown in (1) and a family of threshold values

$$\{\alpha_0 > \alpha_1 > \cdots > \alpha_{K-1} > \alpha_K\}.$$

Let $G^0 = G$; the following family of partial graphs is defined as:

$$\{\ G^t = (V, E^t)\ \} \tag{3}$$

where $t \in \{1, \ldots, K\}$, and

$$E^t = \left\{ e_{ab} \in E \ \middle|\ d_{ab} < \alpha_t \ \wedge\ \mathrm{g}_{t-1}(P_a) = \mathrm{g}_{t-1}(P_b) \right\}. \tag{4}$$

Note that $\mathrm{g}_{t-1}(P_a) = \mathrm{g}_{t-1}(P_b)$ in (4), means that P_a and P_b are in the same group to α_{t-1} (the before step). Now, for $t \in \{1, \ldots, K\}$, the family of partial graphs

$$\{\ F^t = (V, W^t)\ \} \tag{5}$$

is constructed following a two-step procedure:

1. Following a Kruskal-like algorithm, a partial graph $F^t = (V, W^t)$ of $G^{t-1} \setminus G^t = (V,\ E^{t-1} - E^t)$ is constructed: given the arrangement in decreasing order of the edges where weights are in the internal $[\alpha_t, \alpha_{t-1})$. These edges are introduced as long as their addition does not create a cycle.
2. If the partial graph F^t is a spanning forest of $G^{t-1} \setminus G^t = (V,\ E^{t-1} - E^t)$, the procedure stops; otherwise, the edges of E^t (edges with weights lower than α_t) are arranged in increasing order and are added iteratively to W^t so long as they do not form a cycle, and the process finishes when F^t becomes a spanning tree.

The connected components of F^t are neither maximum spanning trees nor minimum spanning trees because the order in which the edges are included in them is reversed between the two steps above. Indeed, in order for us to specify explicitly the algorithm outlined above, the order of the edges in the two-step procedure must take into account the case of ties in the weights of the edges; in this sense, the following remark is pertinent.

Inconsistent cycles can be distinguished into two types of cycles:

- Cycles where all edges have weights greater than or equal to the current threshold α_t. In this case, by the construction of F^t, the edge which generates the cycle has the greatest weight and, in the case of ties between some edges, is one of the last edges in the ordered list of edges. One way of breaking these ties is to consider, as a secondary criterion, the arrangement of the edges in decreasing order of the difference between the degrees of the endpoints of the edges: first, those edges with the greatest difference (in absolute value) between their endpoints.
- Cycles where an edge has a weight lower than the current threshold α. In this case, the cycle is generated by an edge which maintains its endpoints in the same cluster, and the arrangement of these edges is the opposite of that in the first case: first, those edges with the lowest difference (in absolute value) between their endpoints.

3.1 The threshold α

One of the inputs to the basic binary coloring procedure is the value of the threshold α; different values of α_t can produce different spanning forests F^t and, consequently, different colors of the nodes. The selection of these values is the keystone of the overall coloring algorithm.

It seems appropriate to follow an decreasing scheme for the values of the threshold α, with an appropriate selection of the parameter K and values α_t:

$$\alpha_0 > \overline{\alpha} = \alpha_1 > \alpha_2 > \cdots > \alpha_{K-1} \geq \underline{\alpha} > \alpha_K$$

where $\overline{\alpha} \equiv \max D$ and $\underline{\alpha} \equiv \min D$.

Similarly to the result of the binary coloring procedure, F^{t+1} and F^t will be colored differently only if there exists an edge $e_{ab} \in E$ such that $\alpha_{t+1} \leq d_{ab} < \alpha_t$. In this way, the number of different values for α is bounded by m, the number of edges in the graph; and $K \leq m$. In real situations, as the size of the graph increases, it is enough to take a value of K much lower than m.

3.2 The hierarchical segmentation of the network

In this part, we analyze the information obtained by the application of successive binary coloring procedures to the network N.

The successive binary coloring procedure previously introduced suggests a hierarchical partition process: the first clusters are defined by the connected

components of equally colored nodes, and any of them can be partitioned again by subsequent binary colorings. In this way, separated nodes cannot be joined again in this iterative process. Hence, we can understand that the earlier stages of binary coloring are more relevant than the later ones.

Note that the partitions that we produce are independent of the particular color (0 or 1) chosen, since in each iteration two nodes belonging to the same cluster will share the same cluster in the next iteration if they share the same color, independently of whether this color is 0 or 1.

If the number of successive binary colorings K is very large, the partition might, obviously, be uninformative. One way to solve this problem is to reduce the parameter K. This parameter must be chosen taking into account the requirements that, on the one hand, the partition must consider the different values of the distances and, on the other hand, the computational cost of obtaining the partition must be reduced. Moreover, after the parameter K has been fixed, the various values must satisfy the condition $\alpha_0 > \cdots > \alpha_K$. Both of these decisions depend on the size and characteristics of the problem.

Illustrative example.

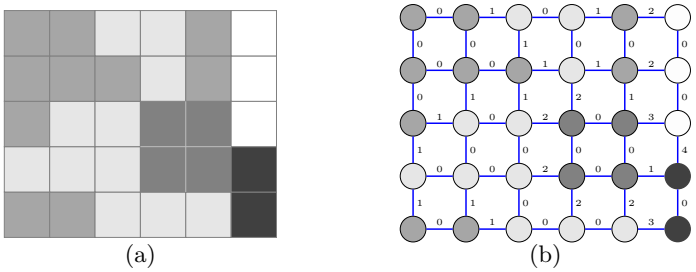
Consider a gray-scale image with the associate matrix:

$$P = \begin{bmatrix} 3\,3\,2\,2\,3\,1 \\ 3\,3\,3\,2\,3\,1 \\ 3\,2\,2\,4\,4\,1 \\ 2\,2\,2\,4\,4\,5 \\ 3\,3\,2\,2\,2\,5 \end{bmatrix} \tag{6}$$

where $P(i,j)$ in (6) represent some gray-scale measure getting homogeneous values $\{1,2,3,4,5\}$. Let

$$d\big((i,j),(i^*,j^*)\big) = \Big|P(i,j) - P(i^*,j^*)\Big|$$

be a Euclidean distance between gray-scale pixels for (i,j) and (i^*,j^*) in $[1,\ldots,5]\times[1,\ldots,6]$. A gray-scale image representation and its associate network to (6) are given in Fig. 2a and Fig. 2b respectively.



(a) (b)

Fig. 2. Gray-scale image (a), and network (b).

Consider a set of α-threshold $\{1,2,3,4\}$ (in decreasing order). The hierarchical segmentation to this set is shown in Fig. 3.

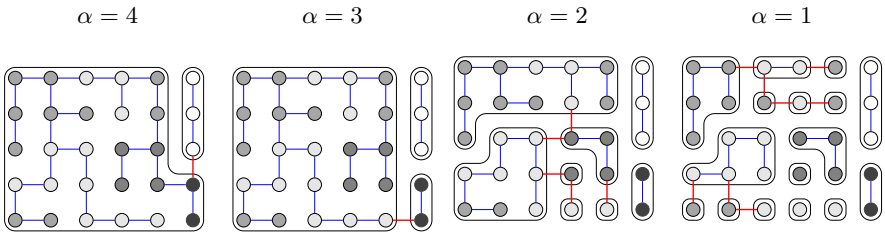


Fig. 3. Hierarchical segmentation by using four thresholds $\{1, 2, 3, 4\}$.

4 A Generalization of the Hierarchical Segmentation Algorithm

Often, we find networks that have very homogeneous regions and we want to remain compact for any threshold used in the segmentation. These groups quite similar (or similar adjacent nodes) are common in many images, so it is desirable to take into account these facts when applying a segmentation process. The hierarchical segmentation process we have developed in the Section 3 is readily adaptable to such events as shown below. Let α_τ be a given similarity threshold, and

$$\underline{E} = \{\, e_{ab} \in E \;\mid\; d_{ab} \leq \alpha_\tau \,\}. \tag{7}$$

If P_a and P_b are nodes such that $e_{ab} \in \underline{E}$; it is expected that $g_t(P_a) = g_t(P_b)$ for all $t \in \{1, \ldots, K\}$. Using (7), we define the set of edges

$$\underline{E}^t = E^t \setminus \underline{E} = \Big\{ e_{ab} \in E \;\mid\; \alpha_\tau < d_{ab} < \alpha_t \;\wedge g_{t-1}(P_a) = g_{t-1}(P_b) \Big\}. \tag{8}$$

If the given set of edges (4) is replaced by (8), a new hierarchical segmentation algorithm is defined. In the illustrative example, if $\alpha_\tau = 0$ the results are:

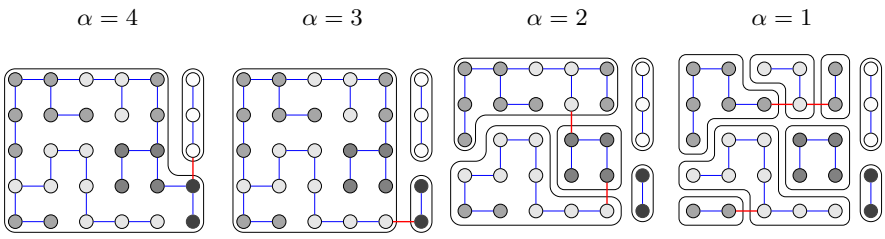


Fig. 4. Generalized Hierarchical Segmentation with $\alpha_\tau = 0$.

This algorithm generalizes the algorithm introduced in Section 3. Note that both algorithm produce the same output for $\alpha_\tau < \underline{\alpha}$.

This generalized algorithm avoids many trees which could separate similar groups of pixels and it produces a better partition, as it as follows.

We can define "adjacent groups" as those that have at least one neighbor pixel inside the other. Formally, for a given α_t, lets A_t and B_t be any two sets of pixels formed on the t-segmentation (note that, $P_a \in A_t \Leftrightarrow g_t(P_a) = A_t$, and $P_b \in B_t \Leftrightarrow g_t(P_b) = B_t$, for some $P_a, P_b \in V$). Now, it says that A_t and B_t are adjacent groups if $\big|A_t, B_t\big| > 0$, with

$$\big|A_t, B_t\big| = \text{card}\Big\{\, e_{ab} \in E \;\big|\; P_a \in A_t \;\wedge\; P_b \in B_t \Big\} \tag{9}$$

where "card" in (9) means the cardinality of the set. Note that, depending of a generated forest, it can be obtained a partitioned groups when some "adjacent" are very similar or dissimilar. One way to measure the quality of some α_t partition is through aggregation operators (see [2]), and measure how dissimilar the adjacent groups are.

Let $\mathcal{A}(C)$ be some aggregator operator defined on a set of pixels C (for instance, the average, median, max, min, among others), and let $H^{\alpha_t}_{\min}$ be the minimum distance between adjacent groups for a fixed $\mathcal{A}$, i.e;

$$H^{\alpha_t}_{\min} = \min_{|A_t,B_t|>0} \Big\{\, \|\mathcal{A}(A_t) - \mathcal{A}(B_t)\| \,\Big\}. \tag{10}$$

where $\|\cdot\|$ is a fixed distance. Note that the larger values of (10) mean more dissimilarity between groups.

In the above example, if we use the average and median as aggregator operators (i.e., $\mathcal{A}(C) = \bar{C}$ and $\mathcal{A}(C) = \tilde{C}$, respectively), the $H^{\alpha_t}_{\min}$ in (10) can be written as

$$\bar{H}^{\alpha_t}_{\min} = \min_{|A_t,B_t|>0} \Big\{ |\bar{A}_t - \bar{B}_t| \Big\} \tag{11}$$

$$\tilde{H}^{\alpha_t}_{\min} = \min_{|A_t,B_t|>0} \Big\{ |\tilde{A}_t - \tilde{B}_t| \Big\} \tag{12}$$

respectively; and then we can compare Hierarchical Segmentation Algorithm (HSA) vs Generalized Hierarchical Segmentation Algorithm (GHSA) as shown in Table 1.

	HSA			GHSA		
$\boldsymbol{\alpha}$	N	$\bar{H}^{\alpha_t}_{\min}$	$\tilde{H}^{\alpha_t}_{\min}$	N	$\bar{H}^{\alpha_t}_{\min}$	$\tilde{H}^{\alpha_t}_{\min}$
4	2	1.89	2.00	2	1.89	2.00
3	3	1.72	2.00	3	1.72	2.00
2	8	0.00	0.00	5	0.52	1.00
1	16	0.00	0.00	8	1.00	1.00

Table 1.

Note that in Hierarchical Segmentation Algorithm (HSA), $\bar{H}^{\alpha_t}_{\min}$ and $\tilde{H}^{\alpha_t}_{\min}$ defined in (11) and (12) are zero for $\alpha = 2, 1$; this means that there are adjacent groups whose average difference is zero (less quality partitions).

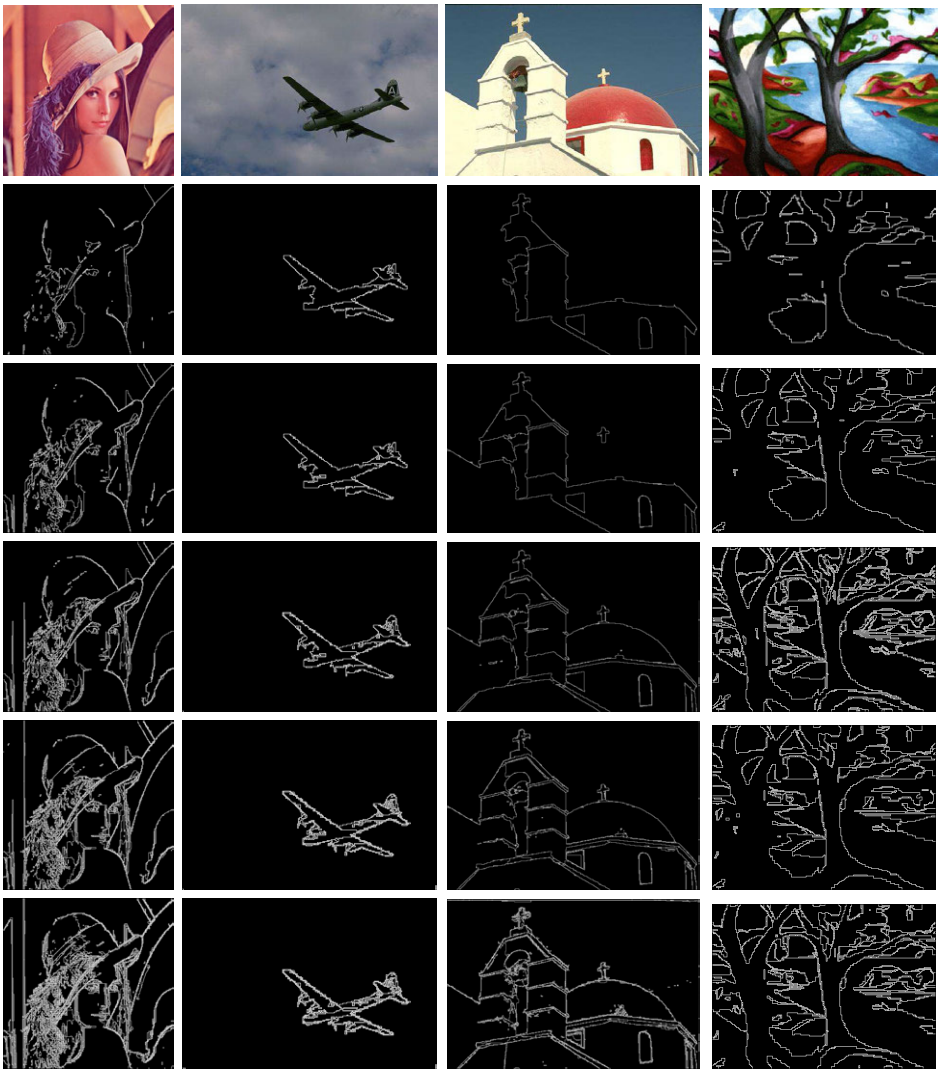


Fig. 5. Hierarchical segmentation process to 4 color images and 5 thresholds.

4.1 Some Hierarchical Segmentation in Color Images

Figure 5 shows the generalized hierarchical segmentation process to four color images to five thresholds and $\alpha_\tau = 1$.

In order to obtain a good measure of color distance, it is necessary to translate an RGB images to an another most uniform color space (i.e. CIELAB or CIELUV color spaces, see [13]). Let $P_1^{\text{rgb}} = (r_1, g_1, b_1)$ and $P_2^{\text{rgb}} = (r_2, g_2, b_2)$ be two colors on the RGB-space, its representation on the CIELAB space is given by $P_1^{\text{lab}} = (L_1, a_1, b_1)$ and $P_2^{\text{lab}} = (L_2, a_2, b_2)$ respectively. Now, one way to measure

the distance between P_1^{lab} and P_2^{lab} is

$$d_{1,2}^{\text{lab}} = \sqrt{(\Delta L)^2 + (\Delta a)^2 + (\Delta b)^2} \tag{13}$$

where $\Delta L = L_2 - L_1$, $\Delta a = a_2 - a_1$ and $\Delta b = b_2 - b_1$. The distance given in (13) is known as CIE76, and is one of the definitions of distance between colors widely used in the world, given by the CIE (International Commission on Illumination) in 1976. Although currently CIE76 is still used in various applications, there have been several versions of corrections to (13) as are CIE94 and CIE2000 among others (see [13]). However, for our purposes CIE76 will be used as a good approximation for the measure.

5 Final Comments

One of the main problems when working on theories involving segmentation thresholds is to choose a set of appropriate thresholds for the segmentation process. If we want to provide the threshold set explicitly, we need to manipulate the pixel values in any color space, and the distances between them in a measure space. However, in our proposed algorithm is more important the number of cutting edges (edges with large values) included in a given step of segmentation, that the threshold value α_i. Because of this, the only thing that matters to hierarchical segmentation proposed is the number of steps (the K value), we can even say how many cutting edges are to be incorporated in every step.

In the processing of the images of Fig. 5, the CPU time was no more than 1 minute for each hierarchical image segmentation. The algorithm was run on a 1.7GHz Intel Core i5 with 3GB of RAM. The five thresholds have been chosen automatically by the algorithm in order to include a fixed size of cut edges ($100\,t^2$ cut edges are included on each step, for $t = 1, \ldots, 5$).

Finally, our proposed algorithm can be executed using coarse nets, which reduces the computational time substantially. The theory of aggregation functions can be used for this purpose, i.e., to relate sets of pixels through some aggregation function, so that results can be obtained more quickly and preserving a good classification. Performances of different aggregator functions in networs can be a future research work.

Acknowledgement.

This research has been partially supported by the Government of Spain, grant TIN2012-32482.

References

1. S. Andrews, G. Hamarneh and A. Saad. "Fast random walker with priors using precomputation for interactive medical image segmentation". *Lecture Notes in Computer Science*, *6363*, 9–16. (2010).

2. G. Beliakov, A. Pradera and T. Calvo. "Aggregation functions: a guide for practitioners". *Springer*, (2008).
3. Y. Boykov and M. P. Jolly. "Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images". *Proceedings of International Conference on Computer Vision*, **1**, 105–112. (2001).
4. Y. Boykov and V. Kolmogorov. "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **26**(9), 1124–1137. (2004).
5. P. F. Felzenszwalb and D. P. Huttenlocher. "Efficient graph-based image segmentation". *International Journal of Computer Vision*, **59**(2), 167–181. (2004)
6. D. Gómez, J. Montero and J. Yánez. "A coloring algorithm for image classification". *Information Sciences* **176**, 3645–3657 (2006).
7. D. Gómez, J. Montero, J. Yáñez and C. Poidomani. "A graph coloring algorithm approach for image segmentation". *Omega* **35**, 173–183 (2007).
8. L. Grady. "Random walks for image segmentation". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**(11), 1768–1783. (2006).
9. L. Grady, and G. Funka-Lea. "Multi-label image segmentation for medical applications based on graph-theoretic electrical potentials". *Proceedings of the 8th ECCV Workshop on Computer Vision Approaches to Medical Image Analysis and Mathematical Methods in Biomedical Image Analysis*, 230–245. (2004).
10. L. Grady, and A. K. Sinop. "Fast approximate random walker segmentation using eigenvector precomputation". *IEEE Conference on Computer Vision and Pattern Recognition*, 1–8. (2008).
11. L. Grady and E. L. Schwartz. "Isoperimetric graph partitioning for image segmentation". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**(3), 469–475. (2006).
12. D. S. Hochbaum. "Polynomial time algorithms for ratio regions and a variant of normalized cut". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **32**(5), 889–898. (2010).
13. N. Ohta and A. Robertson "Colorimetry. Fundamentals and Applications". *John Wiley & Sons, LTd.* (2005).
14. B. Peng and O. Veksler. "Parameter selection for graph cut based image segmentation". *British Machine Vision Conference*. (2008).
15. J. Shi and J. Malik. "Normalized cuts and image segmentation". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**(8), 888–905. (2001).
16. Z. Wu and R. Leahy. "An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **15**(11), 1101–1113. (1993).
17. L. Xu, W. Li and D. Schuurmans. "Fast normalized cut with linear constraints". *IEEE Conference on Computer Vision and Pattern Recognition*, 2866–2873. (2009).
18. J. Yáñez, S. Muñoz and J. Montero. "Graph coloring inconsistencies in image segmentation". *Computer Engineering and Information Sciences* **1**, 435–440 (2008).
19. W. Yang, J. Cai, J. Zheng and J. Luo. "User-friendly interactive image segmentation through unified combinatorial user inputs". *IEEE Transactions on Image Processing*, **19**(9), 2470–2479. (2010).

Interval-Valued Overlap Functions

Benjamin Bedregal[1], Humberto Bustince[2], Javier Fernandez[2], Aranzazu Jurio[2], María Jose Asiain[2], and Laura De Miguel[2]

[1] Departamento de Informática e Matemática Aplicada
Universidade Federal do Rio Grande do Norte
Brazil
bedregal@dimap.ufrn.br
[2] Departamento Automática y Computación
Universidad Publica de Navarra
{bustince@unavarra.es}

Abstract. In this work we extend the well-known class of overlap functions to the interval-valued setting. We consider a possible definition of interval-valued overlap functions and we analyze their first properties.

Keywords. Interval-valued operators; Interval-Valued fuzzy sets, Overlap functions.

1 Introduction

Interval-valued fuzzy sets are often used if the experts have difficulties to build the membership degrees of the elements to the considered fuzzy set ([6, 8, 7]). Analogously, in problems where there are difficulties to assign given data to one class or another, overlap functions have shown themselves as very useful ([3]). In this sense, overlap functions seem to be a useful tool to deal with the type of problems for which the use of interval-valued fuzzy sets is natural.

For this reason, we consider the following problem.

To extend overlap functions to the interval-valued setting in such a way that most of the properties that make them useful in the fuzzy setting are kept.

In particular, we are going to provide an axiomatic definition and a first analysis of the properties these operators fulfill.

This work is organized as follows. We start with some basic preliminaries. Then we define the basic order relations between intervals and, in Section 4, we define the concept of interval-valued overlap function and we study some of its most important properties. We finish with conclusions and references.

2 Preliminaries

Let's start recalling the concept of aggregation function ([9]).

Definition 1. *An aggregation function is a mapping* $M : [0,1]^n \to [0,1]$ *such that*
$(M1)$ $M(0,\ldots,0) = 0$ *and* $M(1,\ldots,1) = 1$.
$(M2)$ M *is an increasing function; that is, if* $x_i \leq y_i$ *for every* $i = 1,\ldots,n$, *then* $M(x_1,\ldots,x_n) \leq M(y_1,\ldots,y_n)$.

In [3] was introduced the notion of overlap function in order to measure up to what extent a given element can be considered as included in two different classes (fuzzy sets).

Definition 2. *A mapping* $G_O : [0,1]^2 \to [0,1]$ *is an overlap function if it satisfies the following conditions:*
$(O1)$.- G_O *is symmetric.*
$(O2)$.- $G_O(x,y) = 0$ *if and only if* $xy = 0$.
$(O3)$.- $G_O(x,y) = 1$ *if and only if* $xy = 1$.
$(O4)$.- G_O *is non-decreasing.*
$(O5)$.- G_O *is continuous.*

Note that overlap functions are particular instances of aggregation functions.

Example 1. (i) The function $G_O(x,y) = (xy)^p$ is an example of overlap function for every $p > 0$.
(ii) Any continuous t-norm without zero divisors is an overlap function.

Proposition 1. *[3] Let* G_O *be an overlap function. If* G_O *is associative, then 1 is a neutral element of* G_O, *i.e.* $G_O(x,1) = 1$ *for each* $x \in [0,1]$.

3 Interval-Valued fuzzy sets

Let $L([0,1]) = \{[a,b] \in [0,1]^2 : a \leq b\}$ be the set of all closed subintervals of the unit interval $[0,1]$. An interval-valued fuzzy set (IVFS) $\mathcal{A}$ on a universe U is a function $\mu_{\mathcal{A}} : U \to L([0,1])$. In order to simplify the notation we will use the projections

$$\underline{[a,b]} = a \text{ and } \overline{[a,b]} = b \tag{1}$$

Define on $L([0,1])$ the next partial orders:

Product order: $X \leq_P Y$ iff $\underline{X} \leq \underline{Y}$ and $\overline{X} \leq \overline{Y}$; and
Inclusion order: $X \subseteq Y$ iff $\underline{Y} \leq \underline{X} \leq \overline{X} \leq \overline{Y}$

$(L([0,1]), \leq_P)$ is a continuous lattice and so it is a bounded lattice ([16, 14]). In fact the bottom is $[0,0]$, the top is $[1,1]$, the supremum and infimum of two arbitrary intervals $X, Y \in L([0,1])$ are the following intervals:

$X \vee Y = [\max(\underline{X},\underline{Y}), \max(\overline{X},\overline{Y})]$ and $X \wedge Y = [\min(\underline{X},\underline{Y}), \min(\overline{X},\overline{Y})]$

With respect to the inclusion order, notice that it does not define an infimum over $L([0,1])$. Notice that in general it is possible to define a linear order in $L([0,1])$ as follows.

Definition 3. *Fix two aggregation functions $M_1, M_2 : [0,1]^2 \to [0,1]$ such that for every $x, y, u, v \in [0,1]$ with $x \leq y$ and $u \leq v$ the identities $M_1(x,y) = M_1(u,v)$ and $M_2(x,y) = M_2(u,v)$ simultaneously hold if $x = u$ and $y = v$. Then the relation $X \preceq_{M_1,M_2} Y$ if and only if*

1. $M_1(\underline{X}, \overline{X}) < M_1(\underline{Y}, \overline{Y})$, *or*
2. $M_1(\underline{X}, \overline{X}) = M_1(\underline{Y}, \overline{Y})$ *and* $M_2(\underline{X}, \overline{X}) \leq M_2(\underline{Y}, \overline{Y})$

is a linear order in $L([0,1])$ called admissible order generated by M_1 and M_2.

A deep analysis of this kind of orders, called admissible orders, can be found in [4]. In particular, note that admissible orders generated by aggregation functions extend the usual product order between intervals.

Other operations that will be useful in this paper are:

Product: $XY = [\underline{XY}, \overline{XY}]$ for any $X, Y \in L([0,1])$;
Exponentiation 1: $X^{[k_1,k_2]} = [\underline{X}^{k_2}, \overline{X}^{k_1}]$ for any $X \in L([0,1])$ and $0 < k_1 \leq k_2$;
Complement: $X^c = [1 - \overline{X}, 1 - \underline{X}]$ for any $X \in L([0,1])$; and

When $k_1 = k_2$ usually we will write X^{k_1} instead of $X^{[k_1,k_2]}$.

Intervals X such that $\underline{X} = \overline{X}$ are called degenerate intervals or diagonals elements of $L([0,1])$ and the set of all degenerate intervals is denoted by $\mathcal{D}$.

We also extend the concept of aggregation function to the interval-valued setting.

Definition 4. *Let $\leq$ be an order which extends $\leq_P$ (but needs not be linear). An interval-valued aggregation function is a mapping $M : L([0,1])^n \to L([0,1])$ such that*

1. *M is increasing with respect to $\leq$;*
2. *$M([0,0], \ldots, [0,0]) = [0,0]$ and $M([1,1], \ldots, [1,1]) = [1,1]$.*

4 Interval-Valued Overlaps

Definition 5. *A mapping $O : L([0,1])^2 \to L([0,1])$ is an interval-valued overlap function if it satisfies the following conditions:*
($O1$).- O is symmetric.
($O2$).- $O(X,Y) = [0,0]$ if and only if $XY = [0,0]$.
($O3$).- $O(X,Y) = [1,1]$ if and only if $XY = [1,1]$.
($O4$).- O is isotone in the second component, i.e. $O(X,Y) \leq_P O(X,Z)$ when $Y \leq Z$.
($O5$).- O is Moore continuous.

Note that, by $O1$ and $O4$, interval-valued overlaps also are isotones in the first component.

Remark 1. Note that if $\leq$ is an admissible order and if O is increasing with respect to $\leq$, then $(O4)$ holds, too.

Let $\mathfrak{O}$ be the set of all interval-valued overlap functions and define the following binary relation on $\mathfrak{O}$:

$$O_1 \leq_{\mathfrak{O}} O_2 \text{ iff } O_1(X,Y) \leq_P O_2(X,Y) \text{ for all } X, Y \in L([0,1])$$

Clearly, $\leq_{\mathfrak{O}}$ is a partial order on $\mathfrak{O}$.

Proposition 2. $(\mathfrak{O}, \leq_{\mathfrak{O}})$ *is an unbounded lattice.*

Proof. Let O_1 and O_2 be interval-valued overlap functions. Then the mappings $O_1 \vee_{\mathfrak{O}} O_2, O_1 \wedge_{\mathfrak{O}} O_2 : L([0,1])^2 \to L([0,1])$ defined by

$$O_1 \vee_{\mathfrak{O}} O_2(X,Y) = O_1(X,Y) \vee O_2(X,Y)$$

and

$$O_1 \wedge_{\mathfrak{O}} O_2(X,Y) = O_1(X,Y) \wedge O_2(X,Y)$$

are clearly the supremum and infimum of O_1 and O_2. It is not hard to prove that both are also interval-valued overlap functions.

On the other hand, in order to prove that $(\mathfrak{O}, \leq_{\mathfrak{O}})$ is unbounded it is enough to note that for any interval-valued overlap function O and natural number $n \geq 2$, the functions $O^{\sqrt[n]{}}, O^n : L([0,1])^2 \to L([0,1])$ defined by

$$O^{\sqrt[n]{}}(X,Y) = O(\sqrt[n]{X}, \sqrt[n]{Y}) \text{ and } O^n(X,Y) = O(X^n, Y^n)$$

also are interval-valued overlap functions and $O^n <_{\mathfrak{O}} O <_{\mathfrak{O}} O^{\sqrt[n]{}}$. Therefore, there is neither a least nor a great interval-valued overlap function. □

In fact in above proposition if we denoted by $O^{\sqrt[\infty]{}} = \lim_{n\to\infty} O^{\sqrt[n]{}}$ and $O^\infty = \lim_{n\to\infty} O^n$, then

$$O^{\sqrt[\infty]{}}(X,Y) = \begin{cases} [0,0] & \text{if } X \vee Y = [0,0] \\ [1,1] & \text{otherwise} \end{cases}$$

and

$$O^\infty(X,Y) = \begin{cases} [1,1] & \text{if } X \wedge Y = [1,1] \\ [0,0] & \text{otherwise} \end{cases}$$

however both functions are not continuous and so are not interval-valued overlaps.

4.1 Representable interval-valued overlaps

Theorem 1. *Let O_1 and O_2 overlap functions such that $O_1 \leq_O O_2$. Then the function $\widetilde{O_1O_2} : L([0,1])^2 \to L([0,1])$ defined by*

$$\widetilde{O_1O_2}(X,Y) = [O_1(\underline{X},\underline{Y}), O_2(\overline{X},\overline{Y})] \tag{2}$$

is an interval-valued overlap function.

Proof. Trivial. □

Analogously to the notion of t-representability in [10], we said that an interval-valued overlap O is o-representable if there exist overlaps O_1 and O_2 such that $O = \widetilde{O_1O_2}$. But not every interval-valued overlap function is o-representable, for example $O(X,Y) = [\underline{X}\,\overline{X}\,\underline{Y}\,\overline{Y}, \overline{X}\,\overline{Y}]$ is clearly not o-representable.

Definition 6. *Let $F : L([0,1])^n \to L([0,1])$ be an isotone function. The left and right projections of F are the functions $\underline{F}, \overline{F} : [0,1]^n \to [0,1]$ defined by*

$$\underline{F}(x_1,\ldots,x_n) = \underline{F([x_1,x_1],\ldots,[x_n,x_n])} \text{ and} \tag{3}$$

$$\overline{F}(x_1,\ldots,x_n) = \overline{F([x_1,x_1],\ldots,[x_n,x_n])} \tag{4}$$

Proposition 3. *Let $O : \underline{L}([0,1])^2 \to L([0,1])$ be an interval-valued overlap function. Then $\underline{O}$ as well as $\overline{O}$ are overlap functions.*

Proof. For $i = 1,\ldots,4$, because O satisfy Oi, then from equation (3), $\underline{O}$ clearly satisfy the property Oi. Note that $\underline{O}$ is continuous because is a composition of two continuous functions (the interval-valued overlap O and the left projection). So, $\underline{O}$ (and analogously $\overline{O}$) is an overlap function. □

Theorem 2. *An interval-valued overlap function O is o-representable iff $O = \widetilde{\underline{O},\overline{O}}$.*

Proof. ($\Rightarrow$) If O is o-representable then by Theorem 1,(*) $O = \widetilde{O_1O_2}$ for some overlaps O_1 and O_2 such that $O_1 \leq_O O_2$. Thus,

$$\begin{aligned}
O_1(x,y) &= \underline{[O_1(x,y), O_2(x,y)]} && \text{by eq. (1)}\\
&= \underline{\widetilde{O_1O_2}([x,x],[y,y])} && \text{by eq. (2)}\\
&= \underline{O([x,x],[y,y])} && \text{by (*)}\\
&= \underline{O}(x,y) && \text{by eq. (3)}
\end{aligned}$$

Analogously, $O_2 = \overline{O}$. So, by (*), $O = \widetilde{\underline{O},\overline{O}}$.

($\Leftarrow$) Straightforward from Proposition 3. □

Lemma 1. *Let $F : L([0,1])^n \to L([0,1])$ be an isotone function. Then for any $X_i \in L([0,1])$ with $i = 1,\ldots,n$,*

$$F(X_1,\ldots,X_n) = [\underline{F}(\underline{X_1},\ldots,\underline{X_n}), \overline{F}(\overline{X_1},\ldots,\overline{X_n})]$$

iff F is inclusion monotonic.

Proof. It is an easy a trivial extension for n-ary interval-valued functions of the Theorem 8 in [12]. □

Theorem 3. *An interval-valued overlap function O is o-representable iff is inclusion monotonic.*

Proof. Straightforward from Lemma 1, Proposition 3 and theorems 1 and 2. □

In fact, in [11] the notion of t-representability was relaxed for pseudo t-representability (denoted by $\mathcal{T}_T$), generalized pseudo t-representability (denoted by $\mathcal{T}_{T,t}$) and a third type whithout particular name (denoted by $\mathcal{T}'_T$). Is clear that is we substitute the t-norm T by an overlap O, then $\mathcal{T}_O$ will be an o-representable interval-valued overlaps whereas $\mathcal{T}_{O,t}$ and $\mathcal{T}'_O$ will be not o-representable interval-valued overlaps. However, here we will provide a more generic class of "representable" interval-valued overlaps which properly contain these three classes.

Definition 7. *An interval-valued overlap O is said to be semi o-representable if there exist overlap functions $O_i : [0,1]^2 \to [0,1]$ with $i = 1, \ldots, 8$ and functions $M_1, M_2 : [0,1]^4 \to [0,1]$ such that for each $X, Y \in L([0,1])$,*

$$\begin{aligned} O(X,Y) = [&M_1(O_1(\underline{X},\overline{Y}), O_2(\overline{X},\underline{Y}), O_3(\underline{X},\underline{Y}), O_4(\overline{X},\overline{Y})),\\ &M_2(O_5(\underline{X},\overline{Y}), O_6(\overline{X},\underline{Y}), O_7(\underline{X},\underline{Y}), O_8(\overline{X},\overline{Y}))] \end{aligned} \tag{5}$$

Theorem 4. *Let O_i with $i = 1, \ldots, 8$ be overlap functions and let $M_1, M_2 : [0,1]^4 \to [0,1]$ be aggregation functions such that $O_i \leq_{\mathfrak{O}} O_{i+4}$ for $i = 1, \ldots, 4$, $M_1 \leq M_2$ and all component of M_1 as well as M_2 are necessaries . Then $O : L([0,1])^2 \to L([0,1])$ defined as in equation (5) satisfy*

1. *(O1) iff $O_1 = O_2$ and $O_4 = O_5$;*
2. *(O2) iff M_1 and M_2 has no zero divisors;*
3. *(O3) iff*
4. *(O4) iff*

Proof. (O1).-
(O2).- ($\Rightarrow$) if $O(X,Y) = [0,0] \Leftrightarrow XY = [0,0]$ then
$M_1(O_1(\underline{X},\overline{Y}), O_2(\overline{X},\underline{Y}), O_3(\underline{X},\underline{Y}), O_4(\overline{X},\overline{Y})) = 0$ and
$M_2(O_5(\underline{X},\overline{Y}), O_6(\overline{X},\underline{Y}), O_7(\underline{X},\underline{Y}), O_8(\overline{X},\overline{Y})) = 0$ iff $X = [0,0]$ or $Y = [0,0]$.

So, by (O2) property, if $X = [0,0]$ or $Y = [0,0]$ then $M_1(0,0,0,0) = 0$ and $M_2(0,0,0,0) = 0$.

($\Leftarrow$) If $XY = [0,0]$ then $X = [0,0]$ or $Y = [0,0]$. Case $X = [0,0]$,

$$\begin{aligned} O([0,0],Y) &= [M_1(O_1(0,\overline{Y}), O_2(0,\underline{Y}), O_3(0,\underline{Y}), O_4(0,\overline{Y})),\\ &\quad M_2(O_5(0,\overline{Y}), O_6(0,\underline{Y}), O_7(0,\underline{Y}), O_8(0,\overline{Y}))] && \text{by eq. (5)}\\ &= [M_1(0,0,0,0), M_2(0,0,0,0)] && \text{by (O2)}\\ &= [0,0] && \text{by (A2)} \end{aligned}$$

The case when $Y = [0,0]$ is analogous. Therefore, if $XY = [0,0]$ then $O(X,Y) = [0,0]$.

On the other hand, if $O(X,Y) = [0,0]$ then $M_1(O_1(\underline{X},\overline{Y}), O_2(\overline{X},\underline{Y}), O_3(\underline{X},\underline{Y}), O_4(\overline{X},\overline{Y})) = 0$ and $M_2(O_5(\underline{X},\overline{Y}), O_6(\overline{X},\underline{Y}), O_7(\underline{X},\underline{Y}), O_8(\overline{X},\overline{Y})) = 0$. Since, M_1 and M_2 has no zero divisors, then $O_1(\underline{X},\overline{Y}) = 0$ or $O_2(\overline{X},\underline{Y}) = 0$
$(O3)$.- Analogous;
$(O4)$ and $(O5)$.- Straightforward because composition of isotone and continuous functions are isotone and continuous. □

Corollary 1. *If there exist functions M_1, M_2 and O_i, with $i = 1,\ldots,8$ satisfying the condition of Theorem 4 then O defined as in equation (5) is a semi o-representable interval-valued overlap.*

Proposition 4. *Let O be an interval-valued overlap. If O is o-representable or there exists a t-norm T such that $O = \mathcal{T}_T$ or $O = \mathcal{T}_{T,t}$ for some $t \in [0,1]$ or $O = \mathcal{T}'_T$ then O is semi o-representable.*

Proof. For the first case, i.e. $O = \widetilde{O'O}$ for some overlaps functions O' and O, it is sufficient consider $O_3 = O'$, $O_8 = O$, $M_1(x_1,\ldots,x_4) = x_3$ and $M_2(x_1,\ldots,x_4) = x_4$. For the second case, i.e. $O = \mathcal{T}_T$, it is sufficient consider $O_i = T$ for $i = 3,5,6$, $M_1(x_1,\ldots,x_4) = x_3$ and $M_2(x_1,\ldots,x_4) = \max(x_1,x_2)$. For the third case, i.e. $O = \mathcal{T}_{T,t}$, it is sufficient consider $O_i = T$ for $i = 3,5,6,8$, $M_1(x_1,\ldots,x_4) = x_3$ and $M_2(x_1,\ldots,x_4) = \max(x_1,x_2,T(t,x_4))$. For the four case, i.e. $O = \mathcal{T}'_T$, it is sufficient consider $O_i = T$ for $i = 1,2,8$, $M_1(x_1,\ldots,x_4) = \min(x_1,x_2)$ and $M_2(x_1,\ldots,x_4) = x_4$. □

Let $F : L([0,1]) \to L([0,1])$ defined by $F(X) = m(X) + \frac{1}{2}(X - m(X))$ where $m(X)$ is the middle point of X, i.e. $m(X) = \frac{\underline{X}+\overline{X}}{2}$ ([15]). An alternative definition of F is the next:

$$F(X) = \left[\frac{\underline{X} + m(X)}{2}, \frac{\overline{X} + m(X)}{2}\right]$$

The function $O : L([0,1])^2 \to L([0,1])$ defined by

$$O(X,Y) = F(X) \wedge F(Y)$$

is an interval-valued overlap (observe that O is Moore continuous because F and the infimum are Moore continuous ([1,2])). Notice that once that F is not inclusion monotone ([15]) and therefore, by Theorem 3, O is not o-representable. Moreover, it is a simple calculation to show that

$$O(X,Y) = [\min(a\underline{X} + b\overline{X}, a\underline{Y} + b\overline{Y}), \min(b\underline{X} + a\overline{X}, b\underline{Y} + a\overline{Y})]$$

where $a = \frac{3}{4}$ and $b = 1 - a$. Thus, if there exist $M_1 : [0,1]^4 \to [0,1]$ and overlap functions O_i with $i = 1,\ldots,4$ such that

$$M_1(O_1(\underline{X},\overline{Y}), O_2(\overline{X},\underline{Y}), O_3(\underline{X},\underline{Y}), O_4(\overline{X},\overline{Y})) = \min(a\underline{X} + b\overline{X}, a\underline{Y} + b\overline{Y})$$

5 Final Remark

In this work we have introduced the concept of interval-valued overlap function. We have also analyzed its first properties. In future works we intend to carry on a deeper study of these class of functions, specially focusing on the non-representable case.

Acknowledgement

The authors of these work have been supported by projects TIN2010-15055 and TIN2011-29520 of the Spanish Ministry of Science and by projects CNPq 480832/2011-0 and CNPq 307681/2012-2 from Brazilian Government.

References

1. B.M. Acióly, B.C. Bedregal, A quasi-metric topology compatible with inclusion monotonicity on interval space, Reliable Computing **3**(3): 305–313, 1997.
2. B.C. Bedregal, A. Takahashi, Interval t-norms as interval representations of t-norms, in: Proc. of 2005 IEEE International Conference on Fuzzy Systems (Fuzz-IEEE), Reno, Nevada, May 22 - 25, 2005, pp. 909–914.
3. H. Bustince, J. Fernandez, R. Mesiar, J. Montero, R. Orduna, Overlap Functions, Nonlinear Analysis, 72, 1488–1499, 2010.
4. H. Bustince, J. Fernandez, A. Kolesárová, R. Mesiar, Generation of linear orders for intervals by means of aggregation functions, Fuzzy Sets and Systems, in press, doi:10.1016/j.fss.2012.07.015.
5. H. Bustince, J. Montero, R. Mesiar, Migrativity of Aggregation Operators, Fuzzy Sets and Systems 160(6), 766–777, 2009.
6. Bustince, H., Barrenechea, E., Pagola, M., Fernandez, J. Interval-valued fuzzy sets constructed from matrices : Application to edge detection. *Fuzzy Sets and Systems* **160** , 1819-1840, 2009.
7. Bustince, H. Interval-valued Fuzzy Sets in Soft Computing . *International Journal of Computational Intelligence Systems* **3** (2) (2010), pp. 215-222.
8. H. Bustince, M. Pagola, E. Barrenechea, J. Fernandez, P. Melo-Pinto, P. Couto, H.R. Tizhoosh, J. Montero, Ignorance functions. An application to the calculation of the threshold in prostate ultrasound images, Fuzzy Sets and Systems 161 (1) 20–36, 2010.
9. T. Calvo, A. Kolesárová, M. Komorníková and R. Mesiar: Aggregation operators: properties, classes and construction methods. In T. Calvo, G. Mayor and R. Mesiar (Eds.): *Aggregation Operators New Trends and Applications* (Physica-Verlag, Heidelberg); Pages 3-104, 2002.
10. C. Cornelis, G. Deschrijver and E.E. Kerre, Advances and challenges in interval-valued fuzzy logic, Fuzzy Sets and Systems, 157(5): 622–627, 2006.
11. G. Deschrijver, A representation of t-norms in interval-valued L-fuzzy set theory, Fuzzy Sets and Systems, 159: 1597–1618, 2008.
12. G.P. Dimuro, B.C. Bedregal, R.H.N. Santiago and R.H.S. Reiser, Interval additive generators of interval t-norms and interval t-conorms, Information Sciences 181(18):3898–3916, 2011.

13. J.Fodor, I.J. Rudas, On continuous triangular norms that are migrative, Fuzzy Sets and Systems 158, 1692-1697, 2007.
14. G. Gierz et all. *Continuous Lattices and Domains*. Cambridge University Press, Cambridge, 2003.
15. R.E. Moore. Methods and Applications for Interval Analysis. SIAM, Philadelphia, 1979.
16. D. Scott. Continous Lattices. Oxford University Computing Laboratory, Tech. Monograph PRG-7, Oxford, 1971. (http://www.sc.ox.ac.uk/files/3229/PRG07.pdf)

A measure of k-specificity of fuzzy sets and the use of maximum k-specificity

Ramon Gonzalez del Campo, Luis Garmendia, and Ronald R. Yager

Universidad Complutense de Madrid, Spain
Iona College, USA
lgarmend@fdi.ucm.es,
rgonzale@estad.ucm.es,
yager@panix.com

Abstract. The concept of specificity provides a measure of the amount of information contained in a fuzzy set. It measures the tranquility when making a decision, especially when having to choose just one element of the universe of discourse. But sometimes a decision maker has to choose more than one element out of a universe, or none of them. In this paper a generalization of specificity, called k-specificity, is proposed to measure the tranquility when choosing k elements from the information given by a fuzzy set.
This paper proposes the maximum k-specificity decision making method to decide how many elements of the universe to choose from the knowledge of a fuzzy set, by selecting an integer k that maximizes the k-specificity. The method can also define then a new defuzzyfication method and it is powerful decision making tool.
Keywords: k-specificity, cardinality, maximum specificity.

1 Preliminaries

Let $X = \{e_1, ..., e_n\}$ be a finite set.

Definition 1 *[5] Let $[0,1]^X$ be the set of fuzzy sets on a universe X. Let a_j be the j^{th} greatest membership degree of a fuzzy set μ on X. A measure of specificity of fuzzy sets is a mapping Sp:$[0,1]^X \to [0,1]$ such that:*

- *$Sp(\mu) = 1$ if and only if μ is a singleton.*
- *$Sp(\emptyset) = 0$*
- *$Sp(\mu)$ depends on a_j in this way:*
  1. *$\frac{\partial Sp(\mu)}{\partial a_1} > 0$*
  2. *$\frac{\partial Sp(\mu)}{\partial a_j} \leq 0$ for all $j \geq 2$*

It can also be defined as a weaker measure of specificity by imposing the third axiom only to normal sets:

Definition 2 *A fuzzy set μ on X is normal if there exists an element $e \in X$ such that $\mu(e) = 1$*

Definition 3 *Let $[0,1]^X$ be the set of fuzzy sets on a universe X. A weak measure of specificity of fuzzy sets is a mapping Sp:$[0,1]^X \to [0,1]$ such that:*

- $Sp(\mu) = 1$ *if* μ *is a singleton.*
- $Sp(\emptyset) = 0$
- *If* μ *and* η *are normal fuzzy sets in X and* $\mu \subset \eta$*, then* $Sp(\mu) \geq Sp(\eta)$.

Definition 4 *[6] Let Sp and Sp′ be two measures of specificity. Sp is stricter than Sp′, denoted by* $Sp \leq Sp'$*, if for all fuzzy set* μ *it verifies that* $Sp(\mu) \leq Sp'(\mu)$.

Definition 5 *[5] The linear measure of specificity of a fuzzy set* μ *on a finite space X, introduced by R.R. Yager is defined as:*

$$Sp_{\overrightarrow{w}}(\mu) = a_1 - \sum_{j=2}^{n} w_j a_j$$

where a_j *is the* j^{th} *greatest membership degree of* μ *and* $\{w_j\}$ *is a set of weights verifying:*

- $w_j \in [0,1]$
- $\Sigma_{j=2}^{n} w_j = 1$

2 Measure of *k*-specificity for Fuzzy Sets

R.R. Yager's measure of specificity is a good measure of information of a fuzzy set to know how useful a fuzzy set is to choose one element of the universe [4–6]. However, many decision makers sometimes must choose k elements from a finite universe so a generalization of measures of specificity of a fuzzy set can be usefull to measure the utility of an output fuzzy set when it must be chosen k elements.

Definition 6 *Let $[0,1]^X$ be the set of fuzzy sets on a universe X. Let a_j be the j^{th} greatest membership degree of μ and let k be an integer such that $0 \leq k \leq card(X)$. A measure of k-specificity is a mapping $Sp_k : [0,1]^X \to [0,1]$ such that:*

- $Sp_k(\mu) = 1$ *if* μ *is a crisp set of k elements.*
- $Sp_k(\emptyset) = 0$ *if* $k > 0$.
- $Sp_k(\mu)$ *depends on* a_j *as follows:*
  - *If* $k > 0$ *and* a_j *increases with* $1 \leq j \leq k$ *then* $Sp_k(\mu)$ *increases.*
  - *If* a_j *increases with* $k > j$ *then* $Sp_k(\mu)$ *decreases.*

Note that R.R. Yager's measure of specificity [5] is a particular case of k-specificity with $k = 1$.

The measure of 0-specificity is a measure of similarity between a fuzzy set and the empty set (note that $Sp_0(\emptyset) = 1$). So only when $k = 0$ the second axiom makes no sense. When $k > 0$ and it must be chosen k elements, the empty set gives no usefull information at all.

Definition 7 *Let $[0,1]^X$ be the set of fuzzy sets on X. Let a_j be the j^{th} greatest membership degree of μ and let k be an integer such that $k \geq 0$. A strong measure of k-specificity is a function $Sp_k : [0,1]^X \to [0,1]$ such that:*

- *$Sp_k(\mu) = 1$ if and only if μ is a crisp set of k elements.*
- *$Sp_k(\emptyset) = 0$ if $k > 0$.*
- *$Sp_k(\mu)$ depends on a_j as follows:*
  - *If $k > 0$ and a_j increases with $1 \leq j \leq k$ then $Sp_k(\mu)$ increases.*
  - *If a_j increases with $k < j$ then $Sp_k(\mu)$ decreases.*

Note that if Sp_k is a strong measure of k-specificity then Sp_k is a measure of k-specificity.

The following definition of linear measure of k-specificity is a generalization of R.R. Yager's linear measure of specificity.

Definition 8 *Let $[0,1]^X$ be the set of fuzzy sets on a finite universe $X = \{e_1, ..., e_n\}$. Let a_j be the j^{th} greatest membership degree of a fuzzy set μ on X. Let k be an integer with $0 \leq k \leq n$. Let $\overrightarrow{w} =$, $\overrightarrow{w}'$ be two sets of k and $n-k$ weights such that: $\sum_{i=1,..,k} w_i = 1$, $\sum_{j=k+1,..,n} w'_j = 1$ and $w_i, w'j \in [0,1]$ for all i, j.*

The linear measure of k-specificity of a fuzzy set μ is defined as follows:

$$Sp_{k,\overrightarrow{w},\overrightarrow{w}'}(\mu) = \begin{cases} \sum_{j=1}^{k} a_j w_j - \sum_{j=k+1}^{n} a_j w'_j & \text{if } k{>}0; \\ 1 - \sum_{j=1}^{n} a_j w'_j & \text{if } k{=}0. \end{cases}$$

Note that the linear measure of 1-specificity is the usual linear measure of specificity[5].

When choosing the set of k weights $\overrightarrow{w}$ and $\overrightarrow{w}'$, some consistency criteria is needed.

Definition 9 *Let $w_i = 1/k$ for $1 \leq i \leq k$ and let $w_j = 1/(n-k)$ for $k+1 \leq j \leq n$. The uniform linear measure of k-specificity of a fuzzy set μ is defined as follows:*

$$Sp_k(\mu) = \begin{cases} \sum_{j=1}^{k} a_j * 1/k - \sum_{j=k+1}^{n} a_j * 1/(n-k) & \text{if } k{>}0; \\ 1 - \sum_{j=1}^{n} a_j * 1/n & \text{if } k{=}0. \end{cases}$$

The measure of $\mathfrak{T}$-specificity is introduced in [1] to provide a logical sense to the measure of specificity and a strong generator of measures, an extension of this logical meaning is provided for a measure of $k-\mathfrak{T}$-specificity in the following definition.

Definition 10 *Let $[0,1]^X$ be the set of fuzzy sets on a universe X. Let a_j be the j^{th} greatest membership degree of a fuzzy set μ. Let $\{w_i\}, \{w'_i\}$ two sets of k and $n-k$ weights in [0,1] that verify $\sum_{i=1,..,k} w_i = 1$ and $\sum_{j=k+1,..,n} w'_j = 1$ and let $\mathfrak{T} = (T_1, S_1, T_2, N, S_2, T_3)$ such that:*

- *T_1, T_2, T_3 are t-norms.*

– S_1, S_2 *are t-conorms.*
– N *is a strong negation.*

A measure of $k - \mathfrak{T}$*-specificity is a mapping* $Sp_{\mathfrak{T},k} : [0,1]^X \to [0,1]$ *defined by:*

$$Sp_{\mathfrak{T},k}(\mu) = \begin{cases} T_1(S_{1\ i=1,..,k}\{T_2(a_i, w_i)\}, N(S_{2\ j=k+1,..,n}\{T_3(a_j, w'_j)\})) & \text{if } k{>}0; \\ N(S_{2\ j=1,..,n}\{T_3(a_j, w'_j)\}) & \text{if } k{=}0. \end{cases}$$

Lemma 1 *Let* $Sp_{\mathfrak{T},k}$ *be a measure of* $k-\mathfrak{T}$*-specificity. Let* $\mathfrak{T} = (T_1, S_1, T_2, N, S_2, T_3)$ *be a six-tuple where:*

– T_1, T_2, T_3 *are t-norms.*
– S_1, S_2 *are t-conorms.*
– N *is a strong negation.*
– S_1 *is a k-ary Lukasiewicz's t-conorm.*

If μ *is a crisp set of* k *elements then* $Sp_{\mathfrak{T},k}(\mu) = 1$.

Proof

– Case 1: $k > 0$
$Sp_{\mathfrak{T},k}(\mu) =$
$= T_1(S_{1\ i=1,..,k}\{T_2(a_i, w_i)\}, N(S_{2\ j=k+1,..,n}\{T_3(a_j, w'_j)\}))$
$= T_1(S_{1\ i=1,..,k}\{T_2(1, w_i)\}, N(S_{2\ j=k+1,..,n}\{T_3(0, w'_j)\}))$
$= T_1(S_{1\ i=1,..,k}\{w_i\}, N(S_{2\ j=k+1,..,n}\{0\}))$
$= T_1(1,1) = 1$

– Case 2: $k = 0$
$Sp_{\mathfrak{T},k}(\mu) =$
$= N(S_{2\ j=k+1,..,n}\{T_3(a_j, w'_j)\})$
$= N(S_{2\ j=k+1,..,n}\{T_3(0, w'_j)\})$
$= N(S_{2\ j=k+1,..,n}\{0\}) = 1$ □

Lemma 2 *Let* $Sp_{\mathfrak{T},k}$ *be a measure of* $k-\mathfrak{T}$*-specificity with* $k > 0$. *Then* $Sp_{\mathfrak{T},k}(\emptyset) = 0$

Proof
If μ is the empty set then $a_i = 0$ for all $i : 1..n$ and $T_2(a_i, w_i) = 0$ and $T_3(a_j, w'_j) = 0$, so $S_1(0, ..., 0) = 0$ and $N(S_2(0, ..., 0)) = 1$. Then $Sp_{\mathfrak{T},k} = T_1(0,1) = 0$ □.

Lemma 3 *Let* A, B *be two normal fuzzy sets. Let* a_j, b_j *be the* j^{th} *greatest membership degree of* μ *and* ν *respectively. If for all* $i : 1..k$ $\mu(e_i) = a_i \geq \nu(e_i) = b_i$ *and for all* $i : k+1..n$ $\mu(e_i) = a_i = \nu(e_i) = b_i$, *then:* $Sp_{\mathfrak{T},k}(\mu) \geq Sp_{\mathfrak{T},k}(\nu)$.

Proof
Trivial due to the monotonicity of T_1, S_1: $S_{1\ i=1,..,k}\{T_2(a_i, w_i)\} \geq S_{1\ i=1,..,k}\{T_2(b_i, w_i)\}$. Hence: $Sp_{\mathfrak{T},k}(\mu) \geq Sp_{\mathfrak{T},k}(\nu)$ □.

Lemma 4 *Let N be a strong negation. Let A, B be two normal fuzzy sets. Let a_j, b_j be the j^{th} greatest membership degree of μ and ν respectively. If for all $i : 1..k$ $\mu(e_i) = a_i = \nu(e_i) = b_i$ and for all $i : k+1..n$ $\mu(e_i) = a_i \leq \nu(e_i) = b_i$, then: $Sp_{\mathfrak{T},k}(\mu) \geq Sp_{\mathfrak{T},k}(\nu)$*

Proof
Trivial due to the monotonicity of T_1, S_2, T_3:

$$S_{2\ j=k+1,..,n}\{T_3(a_j, w'_j)\}) \leq S_{2\ j=k+1,..,n}\{T_3(b_j, w'_j)\}$$

so N is a strong negation then

$$N(S_{2\ j=k+1,..,n}\{T_3(a_j, w'_j)\}) \geq N(S_{2\ j=k+1,..,n}\{T_3(b_j, w'_j)\})$$

Hence: $Sp_{\mathfrak{T},k}(\mu) \geq Sp_{\mathfrak{T},k}(\nu)$ □.

Theorem 1 *Let a_j be the j^{th} greatest membership degree of a fuzzy set μ on X. Let $\{w_i\}, \{w'_i\}$ two sets of weights in [0,1] that verify $\sum_{i=1,..,k} w_i = 1$ and $\sum_{j=k+1,..,n} w'_j = 1$ and let be a set of connectives $\mathfrak{T} = (T_1, S_1, T_2, N, S_2, T_3)$ such that:*

- *T_1, T_2, T_3 are t-norms.*
- *S_1, S_2 are t-conorms.*
- *N is a strong negation.*

Let $Sp_{\mathfrak{T},k} : [0,1]^X \to [0,1]$ be defined as follows:

$$Sp_{\mathfrak{T},k}(\mu) = \begin{cases} T_1(S_{1\ i=1,..,k}\{T_2(a_i, w_i)\}, N(S_{2\ j=k+1,..,n}\{T_3(a_j, w'_j)\})), & \text{if } k{>}0; \\ N(S_{2\ j=1,..,n}\{T_3(a_j, w'_j)\}), & \text{if } k{=}0. \end{cases}$$

If S_1 is a k-ary Lukasiewicz's t-conorm, then $Sp_{\mathfrak{T},k}$ is a measure of k-specificity.

Proof

- Axiom 1:
  Trivial due to the lemma 1
- Axiom 2:
  Trivial due to the lemma 2
- Axiom 3:
  Trivial due to the lemma 3 and 4□.

Proposition 1 *The linear measure of k-specificity is a $k-\mathfrak{T}$-specificity measure.*

Proof
Let $\mathfrak{T} = (T_1, S_1, T_2, N, S_2, T_3)$ be defined as follows:

- T_1 is the Lukasiewicz's t-norm: $T_1(x, y) = max\{0, x + y - 1\}$.
- T_2 and T_3 are the product t-norm: $T_2(x, y) = T_3(x, y) = x * y$.
- S_1 and S_2 are k-ary Lukasiewicz's t-conorms: $S_1(x_1, ...x_k) = S_2(x_1, ...x_k) = min\{x_1 + ... + x_k, 1\}$.

– N is the usual negation $N(x) = 1 - x$.

Let a_j be the j^{th} greatest membership degree of μ and $\{w_i\}, \{w'_i\}$ two sets of weight as in definition 8. Then:

– Case 1: $k > 0$
$Sp_{\mathfrak{T},k}(\mu) = T_1(S_{1\ i=1,..,k}\{T_2(a_i, w_i)\}, N(S_{2\ j=k+1,..,n}\{T_3(a_j, w'_j)\})) =$
$= T_1(S_{1\ i=1,..,k}\{a_i * w_i\}, N(S_{2\ j=k+1,..,n}\{a_j * w'_j\}))$
$= max\{0, S_{1\ i=1,..,k}\{a_i * w_i\} + N(S_{2\ j=k+1,..,n}\{a_j * w'_j\}) - 1\}$
$= max\{0, S_{1\ i=1,..,k}\{a_i * w_i\} - S_{2\ j=k+1,..,n}\{a_j * w'_j\}\}$
$= max\{0, min\{1, \sum_{i=1,..,k} a_i * w_i\} - min\{1, \sum_{j=k+1,..,n} a_j * w'_j\}\}$
$= max\{0, \sum_{i=1,..,k} a_i * w_i - \sum_{j=k+1,..,n} a_j * w'_j\}$ due to $a_i, a_j \leq 1$
$= \sum_{i=1,..,k} a_i * w_i - \sum_{j=k+1,..,n} a_j * w'_j$
– Case 2: $k = 0$
$Sp_{\mathfrak{T},k}(\mu) = T_1(S_{1\ i=1,..,k}\{T_2(a_i, w_i)\}, N(S_{2\ j=k+1,..,n}\{T_3(a_j, w'_j)\})) =$
$= N(S_{2\ j=k+1,..,n}\{a_j * w'_j\})$
$= 1 - S_{2\ j=k+1,..,n}\{a_j * w'_j\}$
$= 1 - min\{1, \sum_{j=k+1,..,n} a_j * w'_j\}$
$= 1 - \sum_{j=k+1,..,n} a_j * w'_j$

so $Sp_{\mathfrak{T},k}(\mu)$ is the linear measure of k-specificity □.

Lemma 5 *Let $\overrightarrow{w} = \{w_q\}$, $\overrightarrow{w} = \{w'_j\}$ be two sets of weights with $w_q > 0$ for $q : 1..k$ and $w'_j > 0$ for $j : k + 1..n$. Let $Sp_{k,\overrightarrow{w},\overrightarrow{w}'}$ be the linear measure of k-specificity. Then, if $Sp_{k,\overrightarrow{w},\overrightarrow{w}'}(\mu) = 1$ then μ is a crisp set with cardinality k.*

Proof

– Let's suppose that exists any $a_j > 0$ for $j : k + 1..n$, then $\sum_{j=k+1}^{n} a_j w'_j > 0$ and $\sum_{q=1}^{k} a_q w_q \leq \sum_{q=1}^{k} w_q = 1$ so $Sp_{k,\overrightarrow{w},\overrightarrow{w}'}(\mu) = \sum_{j=1}^{k} a_j w_j - \sum_{j=k+1}^{n} a_j w'_j < 1$ which is contradiction with $Sp_{k,\overrightarrow{w},\overrightarrow{w}'}(\mu) = 1$.
– Let's suppose that exists any $a_q < 1$ for $q : 1..k$, so $Sp_{k,\overrightarrow{w},\overrightarrow{w}'}(\mu) = \sum_{q=1}^{k} a_q w_q - \sum_{j=k+1}^{n} a_j w'_j \leq 1$ due to $\sum_{q=1}^{k} a_q w_q < \sum_{q=1,..,k} w_q = 1$.

Accordingly $a_q = 1$ with $q : 1..k$ and $a_j = 0$ with $j : k + 1..$ by reductio ad absurdum so μ is a crisp set□.

Lemma 6 *The uniform linear measure of k-specificity is a strong measure of k-specificity.*

Proof: Trivial due to the lemma 5□.

Lemma 7 *Let Sp_k the uniform linear measure of k-specificity. Let N be the negation operator $N(x) = 1 - x$. Then $Sp_k(\mu) = Sp_{n-k}(N(\mu))$ for all $k : 0..n$.*

Proof
Let μ be a fuzzy set on X. Let a_j and b_j be the j^{th} greatest membership degree of a fuzzy set μ and $N(\mu)$ respectively. It is verified that $b_j = 1 - a_{n-j+1}$. Then:
For $k > 0$:

$$\begin{aligned}
&Sp_{n-k}(N(\mu)) = \\
&\quad = \textstyle\sum_{q=1}^{n-k}(1/(n-k))b_q - \sum_{q=n-k+1}^{n}(1/(n-(n-k)))b_q \\
&\quad = \textstyle\sum_{q=1}^{n-k}(1/(n-k))b_q - \sum_{q=n-k+1}^{n}(1/k)b_q \\
&\quad = \textstyle\sum_{q=1}^{n-k}(1/(n-k))(1-a_{n-q+1}) - \sum_{q=n-k+1}^{n}(1/k)(1-a_{n-q+1}) \\
&\quad = \textstyle\sum_{q=n-k+1}^{n}(1/k)a_{n-q+1} - \sum_{q=1}^{n-k}(1/(n-k))a_{n-q+1} \\
&\quad = \textstyle\sum_{q=k}^{1}(1/k)a_{n-q+1} - \sum_{q=n}^{k+1}(1/(n-k))a_{n-q+1} \\
&\quad = Sp_k(\mu)
\end{aligned}$$

For $k = 0$:

$$\begin{aligned}
&Sp_0(N(\mu)) = \\
&\quad = 1 - \textstyle\sum_{q=1}^{n}(1/n)b_q \\
&\quad = 1 - \textstyle\sum_{q=1}^{n}(1/n)(1-a_{n-q+1}) \\
&\quad = \textstyle\sum_{q=1}^{n}(1/n)(1-a_{n-q+1}) \\
&\quad = Sp_n(\mu)\ \square
\end{aligned}$$

3 Measure of Maximum *k*-specificity for Fuzzy Sets

Definition 11 *Let $[0,1]^X$ be the set of fuzzy sets on X. Let a_j be the j^{th} greatest membership degree of μ. Let Sp_k be a measure of k-specificity. A measure of maximum k-specificity is a function $Sp_{max} : [0,1]^X \to [0,1]$ such that:*

$$Sp_{max}(\mu) = max_{k:0..card(\{a_j\})}\{Sp_k(\mu)\}$$

A requirement of consistency in the determination of the set of weights $\{w\}$ and $\{w'\}$ to compute every Sp_k must be established. For the uniform measure of k-specificity we use $w_i = 1/k$ and $w'_i = 1/(n-k)$ for every k.

Lemma 8 *Let Sp_k a measure of k-specificity with $\{w_j > 0\}$ and $\{w'_j > 0\}$. $Sp_{max}(\mu) = 1$ if and only if μ is a crisp set.*

Proof

- *If μ is a crisp set then $Sp_{max}(\mu) = 1$:*
  Let k_0 be the cardinality of μ. Then $Sp_{k_0}(\mu) = 1$ and $Sp_{max}(\mu) = 1$. Note that also $Sp_{max}(\emptyset) = 1$ because $Sp_0(\emptyset) = 1$.
- *If $Sp_{max}(\mu) = 1$ then μ is a crisp set*:
  If $Sp_{max}(\mu) = 1$ then there exists a k such that $Sp_k(\mu) = 1$ so by lemma 5 μ is a crisp set of k elements $\square$.

Definition 12 *Let Sp_k the uniform linear measure of k-specificity. The uniform measure of maximum k-specificity is*

$$Sp_{max}(\mu) = max\{1 - \sum_{j=0}^{n} a_j/n,\ max_{k:1..card(X)}\{\sum_{j=1}^{k} a_j w_j - \sum_{j=k+1}^{n} a_j w'_j\}\}$$

Example 1 *Let $X = \{e_1, e_2, e_3, e_4, e_5\}$ be a finite set and let μ be the following fuzzy sets on X: $\mu_6 = \{1/e_1, 0.9/e_2, 0.2/e_3, 0.1/e_4, 0.1/e_5\}$*

k	$Sp_k(\mu)$
0	*0.54*
1	*0.68*
2	*0.82*
3	*0.60*
4	*0.54*
5	*0.46*

The uniform measure of maximum k-specificity for μ is $Sp_{max}(\mu) = 0.82$, for $k = 2$, so we can decide to defuzzificate X choosing $\{e_2, e_3\}$.

Lemma 9 *Let Sp_k the uniform linear measure of k-specificity. Then $Sp_{max}(\mu) \geq 1/2$.*

Proof
Let a_j be the j^{th} greatest membership degree of a fuzzy set μ on X. Let m be $\sum_{i=1}^{n}(1/n)a_i$.

- If $m = 1/2$ then $Sp_0(\mu) = Sp_n(\mu) = 1/2$ so $Sp_{max}(\mu) \geq 1/2$.
- If $m > 1/2$ then $Sp_n(\mu) > 1/2$ so $Sp_{max}(\mu) \geq 1/2$.
- If $m < 1/2$ then $Sp_0(\mu) > 1/2$ so $Sp_{max}(\mu) \geq 1/2$ □.

Lemma 10 *Let Sp_{max} be a uniform measure of maximum k-specificity. Then $Sp_{max}(\mu)$ takes the minimum value for the constant fuzzy set $1/2$ ($\mu(e_i) = 1/2$ for all e_i in X).*

Proof Trivial by lemma 9 □.

4 A new generalization of the concept of cardinality for a fuzzy set

It is possible to define a generalization of the concept of cardinality of fuzzy sets by using the maximum k-specificity principle.

Definition 13 *Given a measure of k-specificity Sp_k, the cardinality for a fuzzy set is an integer $k \in \{0, ..., n\}$ that maximizes $Sp_k(\mu)$:*

$$Card(\mu) = \{k \mid Sp_k(\mu) \geq Sp_q(\mu)\ for\ all\ q\ in\ \{0..n\}\}$$

Lemma 11 *When μ is a crisp set, the maximum k-specificity cardinality of μ is the classical cardinality.*

Proof
Let μ be a crisp set of q elements on a universe X. By the first axiom of measures of k-specificity $Sp_k(\mu) = 1$ so it is maximum for $k = q\square$.

Lemma 12 *The cardinality of a fuzzy set always exists but it is not always unique.*

Proof
It exists by definition. It is not always unique. A counterexample is provided.

Let $Sp_k(\mu)$ be the uniform linear measure of k-specificity. Let X be $\{e_1\}$ and let $\mu = \{0.5/e_1\}$ be a fuzzy set. Then $Sp_0(\mu) = 0.5$, $Sp_1(\mu) = 0.5$ so $Card(\mu) = \{0, 1\}$ $\square$.

Example 2 *Using the fuzzy set and the uniform measures of k-specificity of the example 1 the cardinality of μ is 2.*

5 A new maximum *k*-specificity defuzzification method: how many elements to choose

Measures of k-specificity can also be useful to decide how many elements are optimal to choose from the information of an output fuzzy set. The max k-specificity cardinality of a fuzzy set can also be used to define a defuzzification method by using the following algorithm: *Maximum k-specificity defuzzyfication method*, defined as follows:

1. Compute k the maximum k-specificity cardinality of μ
2. A crisp defuzzified subset X_μ can be chosen as $X_\mu = \{e_1, ..., e_k\}$

This defuzzification method can be useful to decide how many elements of X should be chosen maximizing the utility of the given information of the fuzzy set.

Lemma 13 *The computational complexity of the Maximum k-specificity using the linear measure of k-specificity defuzzyfication method is $O(n^2)$ where n is the cardinality of the universe X.*

Proof
The computational complexity of computing $Sp_k(\mu)$ is $O(n)$ but it must be computed $n + 1$ times $\square$.

Example 3 *Using the fuzzy set and the measures of k-specificity of the example* **??** *the optimal decision for μ is $\{e_1, e_2\}$*

6 Conclusions

A generalization of measures of specificity is proposed to measure the utility of the information of a fuzzy set in order to choose k elements: the measures of k-specificity. Usually R.R. Yager's measures of specificity are 1-measures of specificity. The linear measure of specificity has been extended as well.

It is proposed a t-norm based definition of a measure of k-specificity, called $Sp_{\mathfrak{T},k}$, giving it a logical sense of "having k elements and no others".

The measures of k-specificity can be used to define a new concept of maximum k-specificity measure and an extension of the concept of cardinality so fuzzy sets which can be useful to decide how many elements are optimal to choose, and then providing a new defuzzification method, by maximizing the utility of information to pick up k elements from a universe of discourse.

Finally, the inverse relation between a uniform uniform linear measure of k-specificity and weak entropy measures of fuzzyness is shown.

References

1. L. Garmendia, R.R. Yager, E. Trillas, and A. Salvador. On t-norms based specificity measures. *Fuzzy Sets and Systems*, 133(2):237–248, 2003.
2. A. De Luca and S. Termini. A definition of a non-probabilistic entropy in the setting of fuzzy sets theory. *Information and Control*, 20:301–312, 1972.
3. B. Schweizer and Sklar A. *Probabilistic Metric Spaces*. North-Holland, 1984.
4. R.R. Yager. Measuring tranquility and anxiety in decision making: an application off uzzy sets. *International Journal of General Systems*, 8:139–146, 1982.
5. R.R. Yager. Ordinal measures of specificity. *International Journal of General Systems*, 17:57–72, 1990.
6. R.R. Yager. On measures of specificity. In O. Kaynak, L.A. Zadeh, B. Turksen, and I.J. Rudas, editors, *Computational Intelligence: Soft Computing and Fuzzy-Neuro Integration with Applications*. Springer, 1998.

Constructing (multi) distances from t-norms

Javier Martín and Gaspar Mayor

University of the Balearic Islands,
Cra. Valldemossa km 7.5, 07122 Palma, Spain
{javier.martin,gmayor}@uib.es

Abstract. In this article the issue of constructing distances (metrics) from t-norms is taken up. We extend known results to pairs (T, S) of a t-norm and a t-conorm. In the case $S = T^*$, the dual of T, we show a very simple continuous non-strict Archimedean t-norm that is not a copula but generates a distance. We apply the same construction to obtain, from pairs (T, S), multi-argument distances (multidistances).

Keywords: distance, multidistance, t-norm, t-conorm, 1-Lipschitz condition, copula

1 Introduction

In [1] (see also [2]), C. Alsina introduced the idea of constructing distances from a t-norm T and its dual T^*, $T^*(x, y) = 1 - T(1 - x, 1 - y)$. Thus, given a t-norm T, define $d_T \colon [0, 1]^2 \to [0, 1]$ by:

$$d_T(x, y) = \begin{cases} 0 & \text{if } x = y, \\ T^*(x, y) - T(x, y) & \text{if } x \neq y. \end{cases} \tag{1}$$

Alsina established in that paper the following result:

– If the t-norm T is a copula then d_T is a distance.

Recently, Y. Ouyang provided an example of a continuous strict Archimedean t-norm that is not a copula and that generates a distance, thus proving that for continuous t-norms the reciprocal of the Alsina's theorem is not true (see [2, 4]). We present here a very simple example of a continuous non-strict Archimedean t-norm that also makes false the reciprocal of the Alsina's result. The characterization of those t-norms that induce distances is still an open problem.

In this paper, we extend the definition (1) to pairs (T, S) where T is a t-norm and S a t-conorm not precisely the dual of T. We obtain a sufficient condition on a pair (T, S) in order to induce a distance $d_{(T,S)}$ and also a necessary condition in terms the distance $d_{(W,W^*)}$, where W denotes the Łukasiewicz t-norm.

In [3] we introduced the concept of multi-argument distance (multidistance, for short) where the conventional definition of a distance is extended to apply to collections of more than two elements. The generation of ordinary distances

from a pair (T, S) can be also applied to generate multidistances with similar conditions at those used in the ordinary case.

In Section 2 we describe the basic definitions and results we are going to use in the development of the paper. Section 3 is devoted to the construction of distances from a t-norm and a t-conorm. The generation of multidistances is studied in Section 4.

2 Preliminaries

We begin with the definitions of t-norm, t-conorm and copula, and some properties and basic examples [2].

Definition 1. *Let us consider functions $T, S: [0,1]^2 \to [0,1]$. We say that T is a t-norm if it verifies, for all $x, y, z \in [0,1]$,*

- $T(x,1) = T(1,x) = x$ *(neutral element),*
- $T(x,y) \leqslant T(x',y')$ *when $x \leqslant x'$, $y \leqslant y'$ (monotonicity),*
- $T(x,y) = T(y,x)$ *(commutativity),*
- $T(T(x,y),z) = T(x,T(y,z))$ *(associativity).*

and S is a t-conorm if the following holds:

- $S(x,0) = S(0,x) = x$ *(neutral element),*
- $S(x,y) \leqslant S(x',y')$ *when $x \leqslant x'$, $y \leqslant y'$ (monotonicity),*
- $S(x,y) = S(y,x)$ *(commutativity),*
- $S(S(x,y),z) = S(x,S(y,z))$ *(associativity).*

Proposition 1. *A continuous t-norm T is Archimedean if and only if it has an additive generator, that is, a strictly decreasing and continuous function $f: [0,1] \to [0,\infty]$ with $f(1) = 0$ such that*

$$T(x,z) = f^{(-1)}(f(x) + f(y)),$$

where $f^{(-1)}: [0,\infty] \to [0,1]$ is the pseudo-inverse of f, defined by

$$f^{(-1)}(x) = \begin{cases} f^{-1}(x) & \text{if } x \leqslant f(0), \\ 0 & \text{otherwise.} \end{cases}$$

For a t-norm T, its dual t-conorm is the function defined by

$$T^*(x,y) = 1 - T(1-x, 1-y).$$

Example 1. Basic t-norms are the following:

$$\begin{aligned} M(x,y) &= \min(x,y) && \text{(minimum)}, \\ Z(x,y) &= \begin{cases} x & \text{if } y = 1, \\ y & \text{if } x = 1, \\ 0 & \text{otherwise.} \end{cases} && \text{(drastic t-norm)}, \\ \Pi(x,y) &= xy, && \text{(product)}, \\ W(x,y) &= \max(x+y-1, 0) && \text{(Łukasiewicz t-norm).} \end{aligned}$$

Their dual t-conorms are, respectively,

$$\begin{aligned} M^*(x,y) &= \max(x,y) && \text{(maximum)},\\ Z^*(x,y) &= \begin{cases} x & \text{if } y=0,\\ y & \text{if } x=0,\\ 1 & \text{otherwise.}\end{cases} && \text{(drastic t-conorm)},\\ \Pi^*(x,y) &= x+y-xy, && \text{(probabilistic sum)},\\ W^*(x,y) &= \min(x+y,1) && \text{(Łukasiewicz t-conorm or bounded sum)}. \end{aligned}$$

Another binary operations we will deal with, related in some sense to t-norms, are the copulas.

Definition 2. *A copula is a function* $C\colon [0,1]^2 \to [0,1]$ *verifying these properties, for* $x, y, x', y' \in [0,1]$*:*

- $C(x,y) + C(x',y') \geqslant C(x,y') + C(x',y)$ *when* $x \leqslant x', y \leqslant y'$,
- $C(x,0) = C(0,x) = 0$
- $C(x,1) = C(1,x) = x$

The relation between t-norms and copulas is established in the following characterizations.

Proposition 2.

- *A copula is a t-norm if and only if it is associative.*
- *A t-norm is a copula if and only if it satisfies the Lipschitz condition with constant 1.*
- *A continuous Archimedean t-norm is a copula if and only if its additive generator is convex.*

Remember that a function $F\colon [0,1]^2 \to [0,1]$ satisfies the Lipschitz condition with constant 1 if

$$|F(x,y) - F(x',y')| \leqslant |x-x'| + |y-y'|,$$

for all pairs $(x,x'), (y,y') \in [0,1]^2$.

Example 2. Among the four basic t-norms, M, Π and W are associative and commutative copulas, while Z is not a copula.

We recall here also the definition of distance.

Definition 3. *A function* $d\colon X \times X \to [0,\infty)$ *is a* distance *on a set* X *if the following properties are satisfied, for all* $x, y, z \in X$*:*

1) $d(x,y) = 0$ *if and only if* $x = y$,
2) $d(x,y) = d(y,x)$,
3) $d(x,y) \leqslant d(x,z) + d(z,y)$.

The generalization to measure the distance of more than two points is the so called multidistance. For more details see [3].

Definition 4. *A function* $D\colon \bigcup_{n\geqslant 1} X^n \to [0,\infty)$ *is a* multidistance *on* X *when the following properties hold, for all* n*, for any* $x_1,\dots,x_n,z\in X$*:*

1) $D(x_1,\dots,x_n)=0$ *if and only if* $x_1=\cdots=x_n$*,*
2) $D(x_1,\dots,x_n)=D(x_{\pi(1)},\dots,x_{\pi(n)})$ *for any permutation* π *of* $1,\dots,n$*,*
3) $D(x_1,\dots,x_n)\leqslant D(x_1,z)+\cdots+D(x_n,z)$*.*

3 Distances constructed from t-norms and t-conorms

Definition 5. *If* T *is a t-norm and* S *a t-conorm, we define on* $[0,1]^2$*:*

$$d(x,y)=\begin{cases} S(x,y)-T(x,y) & \text{if } x\neq y,\\ 0 & \text{if } x=y.\end{cases}$$

We will denote this function by $d_{(T,S)}$ *when we want to emphasize that it comes from the pair* (T,S)*.*

The function d so defined takes values on $[0,1]$ and fulfills some of the properties required to be a distance.

Proposition 3. *For any pair* (T,S)*,*

i) $0\leqslant d(x,y)\leqslant 1$*,*
ii) $d(x,y)=d(y,x)$*,*
iii) $d(x,y)=0$ *if and only if* $x=y$*.*
iv) The triangular inequality $d(x,y)\leqslant d(x,z)+d(z,y)$ *holds whenever* $\min(x,y)\leqslant z\leqslant \max(x,y)$*.*

Proof. The triangular inequality can be written as

$$S(x,y)+T(x,z)+T(z,y)\leqslant T(x,y)+S(x,z)+S(z,y). \tag{2}$$

Let us suppose $x<y$. If $x\leqslant z\leqslant y$ then $S(x,y)\leqslant S(z,y)$, $T(x,z)\leqslant T(x,y)$ by the monotonicity of T and S. Also, $T(z,y)\leqslant z\leqslant S(x,z)$ and therefore the inequality is fulfilled.

The following results give sufficient and necessary conditions for the pair (T,S) to ensure that d is a distance.

Proposition 4. *If* T *and* S *satisfy the Lipschitz condition with constant 1, then* (T,S) *is a solution of* (2) *and so* $d_{(T,S)}$ *is a distance.*

Proof. Consider these inequalities:

$$\begin{aligned} T(x,z)&\leqslant \min(x,z),\\ -S(x,z)&\leqslant -\max(x,z),\\ T(z,y)-T(x,y)&\leqslant \max(z-x,0),\\ S(x,y)-S(z,y)&\leqslant \max(x-z,0).\end{aligned}$$

Adding them we obtain:

$$T(x,z)-S(x,z)+T(z,y)-T(x,y)+S(x,y)-S(z,y)\leqslant 0,$$

which in turn implies (2).

Remark 1.

i) If T satisfies the 1-Lipschitz condition, then its dual T^* also satisfies it. Thus, if T is 1-Lipschitz (T is a copula) then the pair (T, T^*) verifies (2) and $d_{(T,T^*)}$ is a distance. This result is established in [2], where the reverse is enunciated as an open problem:

   - If T is a continuous t-norm and $d_{(T,T^*)}$ is a distance, then is T necessarily 1-Lipschitz (a copula)?

   The answer is negative: in a short note [4], a continuous triangular norm that is not a copula but generates a metric is provided.
   Another simpler counterexample is $T = \frac{W}{\Pi^*}$. Note that this t-norm, which is

$$T(x,y) = \begin{cases} 0 & \text{if } x+y \leqslant 1, \\ \frac{x+y-1}{x+y-xy} & \text{if } x+y \geqslant 1, \end{cases}$$

   is non strict Archimedean, and it is rational on the complement on its 0-region.

ii) Note that different pairs can produce the same distance. For example, (M, S) and (T, M^*) have the same associated distance if and only if (T, S) is a Frank's pair, that is, a solution of the functional equation (see [2])

$$T(x,y) + S(x,y) = x + y.$$

   Note that in this case T and S are dual.

Example 3.

i) (M, M^*) is the only pair from which we deduce the Euclidean distance:

$$d(x,y) = |x - y|.$$

ii) The pair (W, W^*) defines the distance (with $x \neq y$):

$$d(x,y) = \begin{cases} x+y & \text{if } x+y \leqslant 1, \\ 2-(x+y) & \text{if } x+y \geqslant 1. \end{cases} \tag{3}$$

iii) The pair (Π, Π^*) determines the distance (with $x \neq y$)

$$d(x,y) = x + y - 2xy.$$

iv) The pair (Π, M^*) determines the distance (with $x \neq y$)

$$d(x,y) = \max(x,y)(1 - \min(x,y)).$$

   Note that Π is a copula and M^* is not the dual (for any strong negation) of Π.

Proposition 5. *If (T, S) is a solution of (2), then*

$$d_{(M,M^*)} \leqslant d_{(T,S)} \leqslant d_{(W,W^*)}. \tag{4}$$

Proof. In (2), if $z = 0$ then $S(x, y) - T(x, y) \leqslant x + y$, and for $z = 1$ we have $S(x, y) - T(x, y) \leqslant 2 - (x + y)$. Putting together the inequalities and taking into account (3), we have $d_{(T,S)} \leqslant d_{(W,W^*)}$.

The lower bounding follows from the fact that $S \geqslant M^*$ and $T \leqslant M$ for all T, S.

Remark 2.

i) From (4) we can deduce the following boundings, for T and S:

$$T(x, y) \geqslant \max(\min(x, y) + 2\max(x, y) - 2, 0),$$
$$S(x, y) \leqslant \min(2\min(x, y) + \max(x, y), 1).$$

This means, in particular that the zero and one regions of T and S can not exceed the shaded areas in Fig. 1.

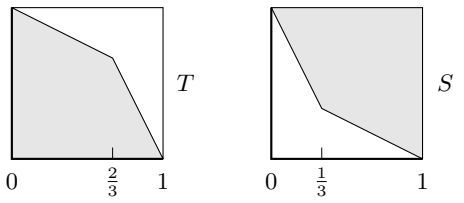


Fig. 1. Maximum zero and one regions of T and S.

For instance, with the drastic t-norm and t-conorm we have that $d_{(Z,S)}$ and $d_{(T,Z^*)}$ are not distances for any T, S.

Note that the lower bounding of T is neither a t-norm, because it is not associative, nor a copula. The upper bounding of S is not also associative, and so it is not a t-conorm.

ii) Condition (4) is not sufficient to ensure (2). For example, let us define the functions:

$$T(x, y) = \begin{cases} \min(x, y) & \text{if } \min(x, y) \leqslant a, \\ \max(a, x + y - 1) & \text{if } \min(x, y) > a, \end{cases}$$

$$S(x, y) = \begin{cases} \max(x, y) & \text{if } \min(x, y) \leqslant a, \\ W^*(x, y) & \text{if } \min(x, y) > a, \end{cases}$$

for $a < \frac{1}{2}$. It can be checked that:

- T is a t-norm and S is a t-conorm,
- $d_{(T,S)} \leqslant d_{(W,W^*)}$,

- but $d_{(T,S)}$ is not a distance: taking $a = z = 0.2$, $x = 0.4$ and $y = 0.5$, we have $d(x, y) = 0.7$ but $d(x, z) + d(z, y) = 0.2 + 0.3 = 0.5$.

iii) If $d_{(T,S)}$ is a distance, then $d_{(S^*,T^*)}$ is also a distance, defined by:

$$d_{(S^*,T^*)}(x, y) = d_{(T,S)}(1 - x, 1 - y).$$

We can say that $d_{(S^*,T^*)}$ is the dual of $d_{(T,S)}$.
The pair (M, Π^*) produces the function

$$d_{(M,\Pi^*)}(x, y) = d_{(\Pi,M^*)}(1 - x, 1 - y).$$

Note that in this case, $d_{(\Pi,M^*)}(1-x, 1-y) = d_{(\Pi,M^*)}(x, y)$, that is, $d_{(M,\Pi^*)} = d_{(\Pi,M^*)}$.

iv) $d_{(T,S)} = d_{(S^*,T^*)}$ if and only if $f(x, y) = f(1 - x, 1 - y)$ for all x, y in $[0, 1]$, where $f(x, y) = T(x, y) + S(1 - x, 1 - y)$.

4 Multidistances constructed from t-norms and t-conorms

It will be shown in this section that the definition of a distance d based on a pair (T, S) and the results relating to these distances, that is, sufficient and necessary conditions in order for $d_{(T,S)}$ to be a distance (Props. 4 and 5), can be translated in a natural way to multidistances.

Definition 6. *If T is a t-norm and S a t-conorm, we define on $\bigcup_{n\geqslant 1}[0, 1]^n$:*

$$D(x_1, \ldots, x_n) = \begin{cases} 0 & \text{if } x_1 = \cdots = x_n, \\ S(x_1, \ldots, x_n) - T(x_1, \ldots, x_n) & \text{otherwise.} \end{cases}$$

We will denote this multi-argument function by $D_{(T,S)}$ when we want to emphasize that it comes from the pair (T, S).

Note that the associativity of t-norms and t-conorms plays an essential role in this definition.

The function D so defined has range $[0, 1]$ and verifies some properties of the definition of multidistance.

Proposition 6. *For any pair (T, S):*

i) $0 \leqslant D(x_1, \ldots, x_n) \leqslant 1$,
ii) D *is symmetric,*
iii) $D(x_1, \ldots, x_n) = 0$ *if and only if* $x_1 = \cdots = x_n$.

Therefore, D is a multidistance if and only if for all n, for any non constant list $(x_1, \ldots, x_n)$ and for all z in $[0, 1]$, different from $x_1, \ldots, x_n$,

$$\sum_{i=1}^{n} T(x_i, z) - T(x_1, \ldots, x_n) \leqslant \sum_{i=1}^{n} S(x_i, z) - S(x_1, \ldots, x_n). \tag{5}$$

Proposition 7. *If T and S satisfy the Lipschitz condition with constant 1, then (T, S) is a solution of* (5) *and so $D_{(T,S)}$ is a multidistance.*

Proof. We proceed by induction on n.

The base case $n = 2$ is Prop. 4.

Let us suppose that (5) holds for n. Adding this inequality to these other ones:

$$T(z, x_{n+1}) - T(x_1, \ldots, x_n, x_{n+1}) \leqslant (z - T(x_1, \ldots, x_n)) \vee 0,$$
$$S(x_1, \ldots, x_n, x_{n+1}) - S(z, x_{n+1}) \leqslant (S(x_1, \ldots, x_n) - z) \vee 0,$$

we obtain

$$\sum_{i=1}^{n+1} T(x_i, z) - T(x_1, \ldots, x_n, x_{n+1}) \leqslant \sum_{i=1}^{n+1} S(x_i, z) - S(x_1, \ldots, x_n, x_{n+1}),$$

and so (5) also holds for $n + 1$.

Example 4.

i) The pair (M, M^*) defines the multidistance

$$D(x_1, \ldots, x_n) = \bigvee_{i,j} |x_i - x_j|.$$

ii) The pair (W, W^*) defines the multidistance (for any non-constant list):

$$D(x_1, \ldots, x_n) = \begin{cases} \sum_{i=1}^n x_i & \text{if } \sum_{i=1}^n x_i \leqslant 1, \\ 1 & \text{if } 1 \leqslant \sum_{i=1}^n x_i \leqslant n-1, \\ n - \sum_{i=1}^n x_i & \text{otherwise.} \end{cases} \tag{6}$$

iii) The pair (Π, Π^*) determines the multidistance (for any non-constant list):

$$D(x_1, \ldots, x_n) = \sum_{j=1}^{n} (-1)^{j+1} \sum_{i_1 < \cdots < i_j} x_{i_1} \cdots x_{i_j} - x_1 \cdots x_n.$$

Proposition 8. *If (T, S) is a solution of* (5)*, then*

$$D_{(M,M^*)} \leqslant D_{(T,S)} \leqslant D_{(W,W^*)}. \tag{7}$$

Proof. Condition (5) for $z = 0$ and $z = 1$ reduces to

$$S(x_1, \ldots, x_n) - T(x_1, \ldots, x_n) \leqslant x_1 + \cdots + x_n,$$
$$S(x_1, \ldots, x_n) - T(x_1, \ldots, x_n) \leqslant 2 - (x_1 + \cdots + x_n).$$

Considering (6), we have $D_{(T,S)} \leqslant D_{(W,W^*)}$.

The other inequality is again a consequence of $T \leqslant M$ and $S \geqslant M^*$, for all T, S.

5 Conclusions

In this paper the generation of distances from a t-norm and its dual t-conorm, proposed in [1], has been extended to included the more general case of any pair of a t-norm and a t-conorm. A necessary and a sufficient condition for a pair to generate a distance have been provided.

This way of generating distances from these pairs has been translated to multidistances, where the transitivity of the t-norm and a t-conorm has all its meaning.

As a future work, and once established that it is not necessary for a t-norm to be a copula, in order to generate with its dual a distance, it will be very interesting to find out the characterization of t-norms not being copulas that generate distances.

Acknowledgments. The authors acknowledge the support of the Spanish DGI grant MTM2009-10962.

References

1. Alsina, C.: On some metrics induced by copulas. In Walter, W. (Ed.), General Inequalities, vol. 4, pp. 397. Birkhäuser Verlag, Basel (1984)
2. Alsina, C., Frank, M.J., Schweizer, B.: Associative Functions: Triangular Norms and Copulas. World Scientific Publishing Company, Singapore (2006)
3. Martín, J., Mayor, G.: Multi–argument Distances. Fuzzy Sets and Systems 167, 92–100 (2011)
4. Ouyang, Y.: A note on metrics induced by copulas. Fuzzy Sets and Systems 191, 122–125 (2012)

Sobre uninormas y nulnormas (α, T)-migrativas

M. Mas, M. Monserrat, D. Ruiz-Aguilera, and J. Torrens

Departamento de Ciencias Matemáticas e Informática
Universitat de les Illes Balears
Carretera de Valldemossa, km 7.5, 07122 Palma
{mmg448,mma112,daniel.ruiz,jts224}@uib.es

Resumen En este trabajo se introducen las nociones de uninormas y nulnormas α-migrativas sobre una t-norma T fija. Se analizan los casos en los que la uninorma U pertenece a alguna de las clases más usuales y se caracterizan las soluciones de la migratividad para todas las posibles combinaciones de uninormas y nulnormas sobre t-normas, con algunas hipótesis de continuidad.

Palabras clave: migratividad, uninormas, nulnormas, t-normas

1. Introducción

En las últimas décadas, el estudio de las funciones de agregación ha sido desarrollado con profusión, principalmente por la gran cantidad de sus aplicaciones. Esta es una de las principales razones por el creciente interés en el campo de las funciones de agregación, y este interés está recogido por las publicaciones de algunos monográficos dedicados por completo a este tipo de funciones [2,6].

Uno de los temas principales en el estudio de estos conectivos desde el punto de vista teórico va dirigido hacia la caracterización de aquellos que verifican ciertas propiedades que pueden ser útiles en cada contexto. El estudio de estas propiedades para ciertas funciones de agregación habitualmente involucra la resolución de ecuaciones funcionales. Una de estas propiedades es la α-*migratividad*, introducida en [9]. Para algún $\alpha \in [0,1]$ y una función $F : [0,1]^2 \to [0,1]$, esta propiedad se describe como

$$F(\alpha x, y) = F(x, \alpha y) \quad \text{para todos } x, y \in [0,1]. \tag{1}$$

El interés de esta propiedad viene a partir de sus aplicaciones, por ejemplo, en procesos de toma de decisiones [4], cuando una información parcial repetida necesita ser fusionada en un resultado global, o en procesamiento de imágenes, ya que en este contexto la migratividad expresa la invariabilidad de una propiedad dada bajo un reescalado proporcional de alguna parte de la imagen [20].

La propiedad de migratividad (y generalizaciones sucesivas) se ha estudiado para t-normas en [11,12,13], para semicópulas, cuasi-cópulas y cópulas en [1,8,20] y para funciones de agregación en general en [3,4,21]. En la ecuación (1) el producto αx puede ser reemplazado por alguna t-norma T_0 obteniendo la propiedad

para t-normas llamada (α, T_0)-migratividad, que puede ser escrita como

$$T(T_0(\alpha, x), y) = T(x, T_0(\alpha, y)) \quad \text{para todos } x, y \in [0,1] \tag{2}$$

siendo T_0 una t-norma y $\alpha \in [0,1]$. Esta generalización de la migratividad para t-normas ha sido estudiada recientemente en [12,13]. Por dualización, una definición similar se puede dar para t-conormas como se expuso en [18]. Además, este estudio ha sido extendido a uninormas con el mismo elemento neutro en [19] (el caso de uninormas representables también fue resuelto en [3]).

De todas maneras, todos los estudios previos tienen un punto en común: se trabaja con funciones de agregación (t-normas, t-conormas o uninormas) que tienen el mismo elemento neutro, con la única excepción de las uninormas representables, que fueron investigadas en [3]. Esta condición no es necesaria para encontrar soluciones de la propiedad de migratividad como veremos en el presente trabajo. Como primer paso en esta dirección, presentaremos en este artículo un estudio completo de aquellas uninormas con elemento neutro $e \in]0,1[$ que son α-migrativas sobre t-normas. Lo haremos considerando uninormas en las clases más conocidas. Haremos también un estudio similar para nulnormas α-migrativas sobre t-normas.

El artículo está organizado en diferentes secciones. Después de esta introducción, se incluye una sección de preliminares para establecer la notación necesaria y recordar las definiciones básicas, especialmente sobre uninormas y nulnormas. En la sección 3 se introduce la definición de uninorma (α, T)-migrativa para una t-norma dada T, analizando algunas propiedades iniciales. A continuación se caracterizan las uninormas (α, T)-migrativas que pertenecen a las clases más conocidas de uninormas, es decir, uninormas en $\mathcal{U}_{\text{mín}}$ y $\mathcal{U}_{\text{máx}}$, uninormas idempotentes y uninormas representables. En la sección 4 se caracterizan las nulnormas (α, T)-migrativas y en la última sección se introducen las conclusiones y el trabajo futuro.

2. Preliminares

Supondremos conocida la teoría básica de t-normas y t-conormas. Las definiciones, notaciones y resultados sobre este tipo de operadores se pueden hallar en [15]. Únicamente daremos en esta sección algunos resultados básicos sobre uninormas y nulnormas. Más detalles sobre uninormas se pueden hallar en [14], para el caso de idempotentes en [23] y para nulnormas en [5,17].

Definición 1. *Una aplicación binaria* $U : [0,1]^2 \to [0,1]$ *se dice que es una* uninorma *si es conmutativa, asociativa, creciente en cada variable y existe un elemento neutro* $e \in [0,1]$ *tal que* $U(e, x) = x$ *para todo* $x \in [0,1]$.

Evidentemente, una uninorma con elemento neutro $e = 1$ es una t-norma y una uninorma con elemento neutro $e = 0$ es una t-conorma. Para cualquier otro valor del elemento neutro $e \in]0,1[$, la uninorma se comporta como una t-norma

en $[0,e]^2$, como una t-conorma en $[e,1]^2$ y sus valores se hallan entre el mínimo y el máximo en el conjunto de puntos $A(e)$ dado por

$$A(e) = [0,e[\times]e,1] \cup]e,1] \times [0,e[.$$

De forma general, denotaremos una uninorma con elemento neutro e, t-norma subyacente T y t-conorma subyacente S, por $U \equiv \langle T,e,S\rangle$. Toda uninorma verifica que $U(0,1) \in \{0,1\}$ y si una uninorma U satisface $U(1,0) = 0$ se dice *conjuntiva* y si $U(1,0) = 1$ se dice *disyuntiva*. Por otra parte, las clases más usuales de uninormas son las siguientes:

- Uninormas en $\mathcal{U}_{\text{mín}}$ (respectivamente $\mathcal{U}_{\text{máx}}$), son aquellas que valen el mínimo (respectivamente máximo) en la región $A(e)$.
- Uninormas *idempotentes*, son aquellas que verifican $U(x,x) = x$ para todo $x \in [0,1]$.
- Uninormas *representables*, son aquellas que tienen un generador aditivo.

Recordemos a continuación la estructura de cada una de las clases de uninormas mencionadas.

Teorema 1 ([14]). *Sea $U : [0,1]^2 \to [0,1]$ una uninorma con elemento neutro $e \in]0,1[$. Entonces, las secciones $x \mapsto U(x,1)$, $x \mapsto U(x,0)$ son continuas en cada punto excepto quizás en e si y solo si U viene dada por una de las siguientes formas.*

(a) Si $U(0,1) = 0$, entonces

$$U(x,y) = \begin{cases} e \cdot T\left(\frac{x}{e}, \frac{y}{e}\right) & si\ (x,y) \in [0,e]^2, \\ e + (1-e) \cdot S\left(\frac{x-e}{1-e}, \frac{y-e}{1-e}\right) & si\ (x,y) \in [e,1]^2, \\ \text{mín}(x,y) & si\ (x,y) \in A(e), \end{cases} \tag{3}$$

donde T es una t-norma y S una t-conorma.

(b) Si $U(0,1) = 1$, entonces U tiene la misma estructura cambiando simplemente el mínimo por el máximo en la región $A(e)$.

La clase de uninormas de la forma dada en el caso (a) lo denotaremos por $\mathcal{U}_{\text{mín}}$ y la clase de uninormas de la forma dada en el caso (b) por $\mathcal{U}_{\text{máx}}$. Una uninorma de $\mathcal{U}_{\text{mín}}$ con t-norma subyacente T, t-conorma subyacente S y elemento neutro e la denotaremos por $U \equiv \langle T,e,S\rangle_{\text{mín}}$ y, de forma similar, una uninorma de $\mathcal{U}_{\text{máx}}$ la denotaremos por $U \equiv \langle T,e,S\rangle_{\text{máx}}$.

Una caracterización de las uninormas idempotentes fue dada en [16], posteriormente corregida en [23] como sigue.

Teorema 2 ([23]). *U es una uninorma idempotente con elemento neutro $e \in [0,1]$ si y solo si existe una función decreciente $g : [0,1] \to [0,1]$, simétrica respecto de la diagonal principal, con $g(e) = e$ tal que*

$$U(x,y) = \begin{cases} \text{mín}(x,y) & si\ y < g(x)\ \ o\ (y = g(x), x < g(g(x))), \\ \text{máx}(x,y) & si\ y > g(x)\ \ o\ (y = g(x), x > g(g(x))), \\ \text{mín}(x,y)\ \ o\ \ \text{máx}(x,y) & si\ y = g(x), x = g(g(x)), \end{cases}$$

siendo conmutativa en los puntos (x, y) *tales que* $y = g(x)$ *con* $x = g(g(x))$.

Una uninorma idempotente U con elemento neutro e y función asociada g la denotaremos por $U \equiv \langle g, e\rangle_{\text{ide}}$ y la clase de todas las uninormas idempotentes por $\mathcal{U}_{\text{ide}}$. Obviamente, para todas estas uninormas la t-norma subyacente T es el mínimo y la t-conorma subyacente S es el máximo.

Definición 2 ([14]). *Sea* $e \in]0,1[$. *La aplicación* $U : [0,1]^2 \to [0,1]$ *es una* uninorma representable *si y solo si existe una función estrictamente creciente y continua* $h : [0,1] \to [-\infty, +\infty]$ *tal que* $h(0) = -\infty$, $h(e) = 0$, $h(1) = +\infty$,

$$U(x, y) = h^{-1}(h(x) + h(y))$$

para todo $(x, y) \in [0,1]^2 \setminus \{(0,1), (1,0)\}$ *y* $U(0,1) = U(1,0) \in \{0,1\}$. *La función* h *se denomina habitualmente un* generador aditivo *de* U.

Observación 1. Recordemos que no existen uninormas continuas con elemento neutro $e \in]0,1[$, es decir, que no sean t-normas o t-conormas. De hecho, las uninormas representables han sido caracterizadas como aquellas que son continuas en $[0,1]^2 \setminus \{(1,0),(0,1)\}$ (ver [22]) y también, como aquellas que son estrictamente crecientes en el cuadrado unidad abierto (ver [10]). Denotaremos por $\mathcal{U}_{\text{rep}}$ la clase de las uninormas representables y, una uninorma U con elemento neutro e y generador aditivo h, la denotaremos por $U \equiv \langle h, e\rangle_{\text{rep}}$. Claramente, la t-norma y la t-conorma subyacentes en este tipo de uninormas son siempre t-normas y t-conormas estrictas.

Recordemos la definición de nulnorma, llamada también t-operador en [17].

Definición 3 ([5,17]). *Una aplicación* $F : [0,1]^2 \to [0,1]$ *se dice que es una* nulnorma *si es conmutativa, asociativa, creciente en cada variable y existe un elemento* $k \in [0,1]$ *llamado elemento absorbente que verifica* $F(k, x) = k$ *para todo* $x \in [0,1]$ *y*

$$F(0, x) = x \quad \textit{para todo } x \leq k \quad y \quad F(1, x) = x \quad \textit{para todo } x \geq k.$$

En este caso, cuando $k = 0$ obtenemos una t-norma y cuando $k = 1$ obtenemos una t-conorma. En general, el elemento absorbente viene dado siempre por $k = F(1,0)$. La estructura de las nulnormas es como sigue.

Teorema 3 ([17]). *Sea* $F : [0,1] \times [0,1] \longrightarrow [0,1]$ *una nulnorma con elemento absorbente* $F(1,0) = k \neq 0, 1$. *Entonces*

$$F(x, y) = \begin{cases} kS\left(\frac{x}{k}, \frac{y}{k}\right) & \textit{si } (x, y) \in [0, k]^2 \\ (1-k)T\left(\frac{x-k}{1-k}, \frac{y-k}{1-k}\right) + k & \textit{si } (x, y) \in [k, 1]^2 \\ k & \textit{en cualquier otro caso,} \end{cases}$$

donde S *es una t-conorma y* T *es una t-norma.*

Una nulnorma F con elemento absorbente k, t-conorma subyacente S y t-norma subyacente T la denotaremos por $F = \langle S, k, T\rangle$.

Finalmente, recordemos que un estudio detallado de las t-normas que son α-migrativas sobre el mínimo, el producto y la t-norma de Łukasiewicz se puede hallar en [12] y una extensión de dicho trabajo a todas las t-normas continuas incluyendo sumas ordinales puede hallarse en [13]. De este modo, los resultados referentes a migratividad de t-normas que usaremos en este trabajo para enunciar los correspondientes a uninormas y nulnormas, pueden hallarse en los citados trabajos.

Notemos que la ecuación (2), escrita para funciones de agregación en general, es un caso particular de la ecuación general de asociatividad y fue considerada en [7] en el estudio de la consistencia de representaciones recursivas de funciones de agregación n-arias a partir de funciones de agregación binarias.

3. Migratividad de uninormas sobre t-normas

Empecemos introduciendo la definición de migratividad de una uninorma U sobre una t-norma T.

Definición 4. *Dada una t-norma T y $\alpha \in [0,1]$, una uninorma U se dice que es* α-migrativa *sobre T o* $(\alpha, T)-$migrativa *si*

$$U(T(\alpha, x), y) = U(x, T(\alpha, y)) \quad \textit{para todos } x, y \in [0,1]. \tag{4}$$

Dado que los casos de t-normas y t-conormas son ya conocidos, en este estudio consideraremos únicamente uninormas con elemento neutro $e \in]0,1[$. Notemos primeramente que, para los valores extremos de α, la migratividad de uninormas sobre t-normas es trivial y se comporta como sigue.

Lema 1. *Sea U una uninorma con elemento neutro $e \in]0,1[$ y T una t-norma, entonces*

i) U es 1-migrativa sobre T.
ii) U es 0-migrativa sobre T si y solo si U es conjuntiva.

Como consecuencia del lema anterior, a partir de ahora consideraremos $\alpha \in]0,1[$. En este caso se puede deducir fácilmente el siguiente resultado.

Proposición 1. *Sea U una uninorma con elemento neutro $e \in]0,1[$, T una t-norma y $\alpha \in]0,1[$. Si U es α-migrativa sobre T, se verifica:*

i) U tiene que ser conjuntiva.
ii) $U(\alpha, y) = U(1, T(\alpha, y))$ para todo $y \in [0,1]$.

En vista del resultado anterior consideraremos solo uninormas conjuntivas en nuestro estudio. Asimismo, del apartado *ii*) anterior se deduce que los valores que tome una uninorma sobre su frontera, es decir, los valores $U(1, y)$ con $y \in [0,1]$ serán especialmente importantes. Por este motivo dedicamos la siguiente subsección al estudio de dichos valores para una uninorma en general.

3.1. Las uninormas sobre la frontera

En esta sección estudiaremos los valores que toma una uninorma U sobre la frontera, $B = [0,1]^2 \backslash]0,1[^2$. Está claro que cualquier uninorma conjuntiva satisface $U(x,0) = U(0,y) = 0$ para todos $x, y \in [0,1]$ y $U(x,1) = U(1,y) = 1$ para todos $x, y \in [e,1]$, mientras que cualquier uninorma disyuntiva satisface $U(x,1) = U(1,y) = 1$ para todos $x, y \in [0,1]$ y $U(x,0) = U(0,y) = 0$ para todos $x, y \in [0,e]$. Investigamos ahora los valores que toma la uninorma en los puntos restantes.

Observemos que, para todas las clases de uninormas mencionadas en los preliminares, se satisface también que

$$U(1,y) \in \{1,y\} \qquad \text{para todo} \quad y \in [0,1], \tag{5}$$

para las uninormas conjuntivas y

$$U(0,y) \in \{0,y\} \qquad \text{para todo} \quad y \in [0,1], \tag{6}$$

para las uninormas disyuntivas. De hecho, hasta donde sabemos, no hay uninormas conjuntivas incumpliendo la propiedad (5) y tampoco hay uninormas disyuntivas incumpliendo la propiedad (6), por eso conjeturamos que esta propiedad sobre la frontera se cumple en general para todas las uninormas conjuntivas y disyuntivas respectivamente. Para tratar esta cuestión, empecemos pues con la siguiente definición.

Definición 5. *Sea U una uninorma conjuntiva. Decimos que U es* localmente interna sobre la frontera *si satisface la propiedad (5). Análogamente, si U es una uninorma disyuntiva, decimos que U es* localmente interna sobre la frontera *si satisface la propiedad (6).*

Gracias a esta definición, la conjetura anterior se puede escribir como sigue:

Conjetura: Cualquier uninorma es localmente interna sobre la frontera.

No tenemos ninguna demostración de esta conjetura, pero podemos dar los siguientes resultados parciales sobre uninormas conjuntivas, los cuales se pueden dualizar fácilmente a las disyuntivas.

Lema 2. *Sea U una uninorma conjuntiva. Entonces, para cualquier $x \in]0,e[$, $U(1,x) < e$ o $U(1,x) = 1$.*

Proposición 2. *Cualquier uninorma conjuntiva $U \equiv \langle T, e, S\rangle$ con t-norma subyacente T continua, es localmente interna sobre la frontera.*

Claramente, obtenemos resultados análogos para las uninormas disyuntivas simplemente dualizando los razonamientos. Concretamente, tenemos el siguiente resultado:

Proposición 3. *Sea $U \equiv \langle T, e, S\rangle$ una uninorma disyuntiva. Tenemos que:*

i) Para cualquier $x \in]e,1[$, $U(0,x) > e$ o $U(0,x) = 0$.
ii) Si la t-conorma subyacente S es continua, entonces U es localmente interna sobre la frontera.

3.2. Migratividad de las uninormas que son localmente internas sobre la frontera

Volvamos a la propiedad de migratividad, en este caso para las uninormas que son localmente internas sobre la frontera.

Proposición 4. *Sea U una uninorma conjuntiva con elemento neutro $e \in]0,1[$ que sea localmente interna sobre la frontera. Sean T una t-norma y $\alpha \in]0,1[$. Si U es α-migrativa sobre T, entonces $T(\alpha, e) = \alpha$ y $\alpha < e$.*

Entonces, a partir de este momento sólo consideraremos $\alpha \in]0, e[$. Ahora demostraremos que para las uninormas localmente internas sobre la frontera, la migratividad de U sobre T equivale a que U y T tengan la misma sección α.

Proposición 5. *Sean U una uninorma conjuntiva con elemento neutro $e \in]0,1[$ localmente interna sobre la frontera, T una t-norma y $\alpha \in]0, e[$. Los enunciados siguientes son equivalentes:*

i) U es (α, T)-migrativa.
ii) $U(\alpha, y) = T(\alpha, y)$ para todo $y \in [0,1]$.

Ya que las uninormas son compensatorias en la región $A(e)$ y las t-normas están por debajo del mínimo, el resultado anterior demuestra, en particular, que no hay uninormas (localmente internas sobre la frontera) α-migrativas sobre una t-norma arquimediana cualquiera. Entonces, si nos restringimos a partir de ahora a las t-normas continuas, para encontrar uninormas (α, T)-migrativas, T tiene que ser el mínimo o una suma ordinal no trivial.

También gracias a la Proposición 5, para caracterizar las uninormas U que sean α-migrativas sobre una t-norma T, solamente necesitamos encontrar uninormas con la misma sección α que la de T. Separaremos el estudio por casos, según la clase de uninormas consideradas. Empezamos con las uninormas de $\mathcal{U}_{\text{mín}}$.

Teorema 4. *Sea $U \equiv \langle T_U, e, S_U \rangle$ una uninorma de $\mathcal{U}_{\text{mín}}$ con elemento neutro $e \in]0,1[$ y T_U continua. Sean T una t-norma continua y $\alpha \in]0, e[$.*

a) Si $T(\alpha, \alpha) = \alpha$, entonces U es α-migrativa sobre T si y solo si $U(\alpha, \alpha) = \alpha$. En este caso, ambas T_U y T tienen que ser sumas ordinales

$$T_U = \left(\left\langle 0, \frac{\alpha}{e}, T_{1U} \right\rangle, \left\langle \frac{\alpha}{e}, 1, T_{2U} \right\rangle \right) \text{ y } T = \left(\langle 0, \alpha, T_1 \rangle, \langle \alpha, 1, T_2 \rangle \right)$$

para algunas t-normas continuas T_{1U}, T_{2U}, T_1 y T_2.

b) Si $T(\alpha, \alpha) < \alpha$, entonces U es α-migrativa sobre T si y sólo si ambas T y T_U son sumas ordinales de la forma $T = (\ldots, \langle a, b, T_A \rangle, \ldots)$ y $T_U = \left(\ldots, \left\langle \frac{a}{e}, \frac{b}{e}, T_{UA} \right\rangle, \ldots\right)$, donde $a < \alpha < b \leq e$, T_A y T_{UA} son arquimedianas, y con T_{UA} que sea $\frac{\alpha - a}{b - a}$-migrativa sobre T_A.

Para las uninormas representables el resultado es negativo, es decir, no hay uninormas representables α-migrativas sobre las t-normas, tal y como indica el teorema siguiente.

Teorema 5. *Sean U una uninorma representable y $\alpha \in]0,1[$. Entonces U no es (α, T)-migrativa, cualquiera que sea la t-norma T.*

El caso de las uninormas idempotentes se resuelve en el teorema siguiente.

Teorema 6. *Sean $U \equiv \langle g, e\rangle_{\text{ide}}$ una uninorma idempotente conjuntiva con $e \in]0,1[$, T una t-norma y $\alpha \in]0, e[$. Los enunciados siguientes son equivalentes:*

i) U es (α, T)-migrativa.
ii) $U(\alpha, y) = T(\alpha, y) = \min(\alpha, y)$ para todo $y \in [0,1]$.

Observemos que cuando T es continua, la sección α de T viene dada por el mínimo si y solo si α es un elemento idempotente de T. En este caso, teniendo en cuenta la estructura de las uninormas idempotentes, el teorema anterior se puede escribir como sigue.

Teorema 7. *Sean $U \equiv \langle g, e\rangle_{\text{ide}}$ una uninorma idempotente conjuntiva con $e \in]0,1[$, T una t-norma continua y $\alpha \in]0, e[$. Los enunciados siguientes son equivalentes:*

i) U es (α, T)-migrativa.
ii) α es un elemento idempotente de T, $g(1) \geq \alpha$ y $U(\alpha, 1) = \alpha$.

4. Nulnormas migrativas sobre t-normas

De forma similar al caso de uninormas podemos definir la migratividad de nulnormas sobre t-normas.

Definición 6. *Dada una t-norma T y $\alpha \in [0,1]$, una nulnorma F se dice que es* α-migrativa *sobre T o* $(\alpha, T)-$migrativa *si*

$$F(T(\alpha, x), y) = F(x, T(\alpha, y)) \quad \text{para todos } x, y \in [0,1]. \tag{7}$$

Dado que los casos de t-normas y t-conormas son ya conocidos, en este estudio consideraremos únicamente nulnormas con elemento absorbente $k \in]0,1[$. Notemos primeramente que, para los valores extremos de α, la migratividad de nulnormas sobtre t-normas es igualmente trivial.

Lema 3. *Sea F una nulnorma con elemento absorbente $k \in]0,1[$ y T una t-norma. Entonces,*

i) F es 1-migrativa sobre T.
ii) F nunca es 0-migrativa sobre T.

Como consecuencia del lema anterior, a partir de ahora consideraremos $\alpha \in]0,1[$. En este caso, se puede deducir fácilmente el siguiente resultado.

Proposición 6. *Sea F una nulnorma con elemento absorbente $k \in]0,1[$, T una t-norma y $\alpha \in]0,1[$. Si F es α-migrativa sobre T, se verifica:*

i) $\alpha \geq k$ *y* $T(\alpha, k) = k$.
ii) $F(\alpha, y) = T(\alpha, y)$ *para todo* $y \geq k$.

En el caso de que la t-norma considerada sea continua se puede ver que las condiciones de la proposición anterior son también suficientes, con lo que se pueden caracterizar completamente las nulnormas que son α-migrativas sobre una t-norma T continua.

Teorema 8. *Sea* $F = \langle S_F, k, T_F \rangle$ *una nulnorma con elemento absorbente* $k \in]0,1[$, T *una t-norma continua y* $\alpha \in]0,1[$. *Son equivalentes:*

i) F *es* α*-migrativa sobre* T.
ii) $\alpha \geq k$ *y* $F(\alpha, y) = T(\alpha, y))$ *para todo* $y \geq k$.

Además, en el caso en que T_F *sea también continua, entonces* F *es* α*-migrativa sobre* T *si y solo si* $\alpha \geq k$ *y existe un* $a \in [k, \alpha]$ *de forma que* T_F *y* T *son sumas ordinales* $T_U = \left(\left\langle 0, \frac{a-k}{1-k}, T_{1F} \right\rangle, \left\langle \frac{a-k}{1-k}, 1, T_{2F} \right\rangle\right)$ *y* $T = (\langle 0, a, T_1 \rangle, \langle a, 1, T_2 \rangle)$, *siendo* T_{2F} $\frac{\alpha-k}{1-k}$ *migrativa sobre* T_2.

5. Conclusiones y trabajo futuro

Las uninormas que son migrativas sobre uninormas con el mismo elemento neutro fueron estudiadas en [19]. De todas formas, la hipótesis de que tengan el mismo elemento neutro no es necesaria para encontrar soluciones de la propiedad de migratividad. En esta línea, en este trabajo se presenta un primer paso en el estudio de uninormas migrativas con elementos neutros diferentes. Específicamente en este artículo se ha estudiado la migratividad de uninormas con elemento neutro $e \in]0,1[$ sobre t-normas. Como resultado, se ha demostrado que no hay uninormas migrativas sobre el producto (que es la ecuación de migratividad original) ni sobre ninguna t-norma arquimediana. Por el contrario, aparecen muchas soluciones cuando nos centramos en la t-norma mínimo, o con cualquier t-norma que sea suma ordinal, y se han conseguido y caracterizado todas las soluciones variando la uninorma en alguna de las clases más conocidas. Un trabajo similar se ha realizado para nulnormas.

Nuestro trabajo futuro en este tema estará dirigido al estudio de uninormas que son migrativas sobre uninormas con elementos neutros diferentes y a un estudio análogo para nulnormas. Creemos que los resultados obtenidos en este estudio pueden ser útiles, porque probablemente se podrán utilizar razonamientos similares a los presentados en este trabajo.

Referencias

1. Beliakov, G., Calvo, T.: On migrative means and copulas, in: Proceedings of Fifth International Summer School on Aggregation Operators, AGOP 2009, Palma de Mallorca, 2009, pp. 107-110

2. Beliakov, G., Pradera, A., Calvo, T.: Aggregation Functions: A Guide for Practicioners. Springer, Berlin Heidelberg, (2007)
3. Bustince, H., De Baets, B., Fernandez, J., Mesiar, R., Montero, J.: A generalization of the migrativity property of aggregation functions, Information Sciences **191** 76–85 (2012)
4. Bustince, H., Montero, J., Mesiar, R.: Migrativity of aggregation functions. Fuzzy Sets and Systems **160** 766–777 (2009)
5. Calvo, T., De Baets, B., Fodor, J.: The functional equations of Frank and Alsina for uninorms and nullnorms. Fuzzy Sets and Systems, **120**, 385-394 (2001)
6. Calvo, T., Mayor, G., Mesiar, R. (eds.): Aggregation operators. New trends and applications. Studies in Fuzziness and Soft Computing, 97. Physica-Verlag, Heidelberg (2002)
7. Cutello, V., Montero, J.: Recursive connective rules. International Journal of Intelligent Systems, **14**, 3-20 (1999)
8. Durante, F., Ricci, R. G.: Supermigrative semi-copulas and triangular norms. Information Sciences **179**, 2689–2694 (2009)
9. Durante, F., Sarkoci, P.: A note on the convex combination of triangular norms. Fuzzy Sets and Systems **159**, 77–80 (2008)
10. Fodor, J., De Baets, B.: A single-point characterization of representable uninorms. Fuzzy Sets and Systems **202**, 89–99 (2012)
11. Fodor, J., Rudas, I.J.: On continuous triangular norms that are migrative. Fuzzy Sets and Systems **158**, 1692–1697 (2007)
12. Fodor, J., Rudas, I.J.: An extension of the migrative property for triangular norms. Fuzzy Sets and Systems **168**, 70–80 (2011)
13. Fodor, J., Rudas, I.J.: Migrative t-norms with respect to continuous ordinal sums. Information Sciences **181**, 4860–4866 (2011)
14. Fodor, J., Yager, R.R., Rybalov, A.: Structure of uninorms. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **5**, 411–427 (1997)
15. Klement, E.P., Mesiar, R., Pap, E.: Triangular norms. Kluwer Academic Publishers, Dordrecht (2000)
16. Martín, J., Mayor, G., Torrens, J.: On locally internal monotonic operations. Fuzzy Sets and Systems **137**, 27–42 (2003)
17. Mas, M., Mayor, G., Torrens, J.: t-Operators. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, **7**, 31-50 (1999)
18. Mas, M., Monserrat, M., Ruiz-Aguilera, D., Torrens, J.: On migrative t-conorms and uninorms. In: Communications in Computer and Information Science, vol. 299, pp. 286–295. Springer, Heidelberg (2012)
19. Mas, M., Monserrat, M., Ruiz-Aguilera, D., Torrens, J.: An extension of the migrative property for uninorms. Information Sciences, (2013).
20. Mesiar, R., Bustince, H., Fernandez, J.: On the α-migrativity of semicopulas, quasicopulas and copulas. Information Sciences **180**, 1967–1976 (2010)
21. Ricci, R.G.: Supermigrative aggregation functions, in: Proceedings of the 6th International Summer School on Aggregation Operators, AGOP-2011, Benevento, Italy, 2011, pp. 145–150.
22. Ruiz, D., Torrens, J.: Distributivity and conditional distributivity of a uninorm and a continuous t-conorm. IEEE Transactions on Fuzzy Systems **14**, 180–190 (2006)
23. Ruiz-Aguilera, D., Torrens, J., De Baets, B., Fodor, J.: Some remarks on the characterization of idempotent uninorms. In: Hüllermeier, E., Kruse, R., Hoffmann, F. (eds.) Lecture Notes in Artificial Intelligence 6178, pp. 425–434. Springer, Heidelberg (2010)

Caracterización de las (h, e)-implicaciones

S. Massanet y J. Torrens

Departament de Ciències Matemàtiques i Informàtica,
Universitat de les Illes Balears,
Campus UIB, Crta. Valldemossa Km. 7.5, E-07122 Palma de Mallorca,
{s.massanet, jts224}@uib.es

Resumen En esta comunicación, se caracterizan las (h, e)-implicaciones, una familia de implicaciones borrosas introducida recientemente. Su caracterización se obtiene gracias a la introducción de una extensión de las implicaciones de Yager. Estas nuevas clases están basadas en la generalización del factor interno x, en el caso de las implicaciones f-generadas, o del $\frac{1}{x}$, en el caso de las implicaciones g-generadas, a una función más general. Finalmente, se estudian algunas propiedades y características de las nuevas familias introducidas.

Palabras clave: Implicaciones borrosas, implicaciones de Yager f-generadas, implicaciones de Yager g-generadas, (h, e)-implicaciones.

1. Introducción

Las implicaciones borrosas se han convertido en una de las operaciones más importantes en la lógica borrosa. Su importancia radica en el hecho de que desarrollan una función similar a la de la implicación clásica en la lógica binaria. Las implicaciones borrosas generalizan las clásicas en el sentido que restringidas a $\{0, 1\}^2$ coinciden con la implicación clásica. Hoy en día, estos operadores se modelan mediante funciones $I : [0, 1]^2 \to [0, 1]$ monótonas que satisfacen las condiciones de frontera exigidas. En los últimos años, un gran número de investigadores han dedicado sus esfuerzos a estos conectivos lógicos como prueba la publicación del artículo recopilatorio [8] y de los libros [2] y [4], dedicados en su totalidad a estos operadores.

En este campo, la caracterización y representación de las implicaciones borrosas es uno de los puntos de interés de la comunidad investigadora. En [13], se indica que las caracterizaciones a través de propiedades algebraicas son esenciales para comprender el comportamiento de los diferentes modelos de implicaciones borrosas. Además, debido al gran número de aplicaciones que tienen, existen muchos modelos diferentes de implicaciones borrosas para responder al problema concreto que deben de modelar. Así, el estudio de las caracterizaciones ha sido una constante a lo largo del tiempo como se puede apreciar por la gran cantidad de trabajos publicados sobre el tema. Por ejemplo, las (S, N) (excepto cuando la negación N no es continua) y las R-implicaciones están caracterizadas completamente (ver [3] y [5] respectivamente), mientras que las QL y las

D-operaciones sólo lo están en algunos casos particulares ([7]). Además recientemente, se han caracterizado, entre otras clases, las (U, N)-implicaciones en [4], las RU-implicaciones en [1] y las implicaciones de Yager f y g-generadas en [11].

Otra de las familias de implicaciones que recientemente ha sido caracterizada es la de las h-implicaciones (ver [12]). Estas implicaciones fueron introducidas en [9] mediante el uso de generadores aditivos de uninormas representables de forma análoga a la estrategia usada por Yager en las definiciones de las implicaciones f y g-generadas [15]. En [9], además de las h-implicaciones, también se introdujeron las (h, e)-implicaciones, una familia de implicaciones que satisface la propiedad $I(e, y) = y$ para todo $y \in [0, 1]$. Sin embargo, aunque las h-implicaciones han podido ser caracterizadas a través de las implicaciones de Yager, las (h, e)-implicaciones aún no han podido ser caracterizadas. En este trabajo, se caracterizarán estas implicaciones mediante el uso de unas nuevas familias de implicaciones que extienden las implicaciones de Yager. Recientemente, se han propuesto varias generalizaciones de estas implicaciones [6,10,14]. La novedad de la extensión que se propone es el uso de una función continua con rango en $[0, +\infty]$ en la definición de las implicaciones f y g-generadas en lugar de los factores x y $\frac{1}{x}$, respectivamente.

La comunicación se organiza como sigue. En la siguiente sección, se recuerdan las definiciones y propiedades básicas sobre implicaciones borrosas que se usarán en este trabajo. En la sección 3, se estudia la caracterización de las (h, e)-implicaciones y se introducen las nuevas extensiones de las implicaciones de Yager. A continuación, en la sección 4, se determinan las propiedades adicionales que verifican estas nuevas familias. El trabajo concluye con las conclusiones y el trabajo futuro que se pretende llevar a cabo.

2. Preliminares

Recordemos algunos conceptos y resultados que se usarán en el resto del trabajo. En primer lugar, sigue la definición de negación borrosa.

Definición 1 ([5, Definición 1.1]). *Una función decreciente $N : [0, 1] \to [0, 1]$ se dice que es una negación borrosa, si $N(0) = 1$ y $N(1) = 0$. Una negación borrosa N es*

(i) estricta, si es continua y estrictamente decreciente.
(ii) fuerte, si es una involución, i.e., $N(N(x)) = x$ para todo $x \in [0, 1]$.

A continuación, se recuerda la definición de implicación borrosa.

Definición 2 ([5, Definición 1.15]). *Un operador binario $I : [0, 1]^2 \to [0, 1]$ se dice que es una implicación borrosa, si satisface:*

(I1) $I(x, z) \geq I(y, z)$ *cuando* $x \leq y$, *para todo* $z \in [0, 1]$.
(I2) $I(x, y) \leq I(x, z)$ *cuando* $y \leq z$, *para todo* $x \in [0, 1]$.
(I3) $I(0, 0) = I(1, 1) = 1$ *e* $I(1, 0) = 0$.

Observar que de la definición se deriva que $I(0,x) = 1$ e $I(x,1) = 1$ para todo $x \in [0,1]$ mientras que los valores simétricos $I(x,0)$ e $I(1,x)$ no quedan determinados. Algunas propiedades de las implicaciones borrosas que se usarán posteriormente son las siguientes:

- El *principio de neutralidad por la izquierda*,

$$I(1,y) = y, \quad y \in [0,1]. \tag{\textbf{NP}}$$

- El *principio de intercambio*,

$$I(x,I(y,z)) = I(y,I(x,z)), \quad x,y,z \in [0,1]. \tag{\textbf{EP}}$$

- La *ley de importación* con una t-norma T,

$$I(T(x,y),z) = I(x,I(y,z)), \quad x,y,z \in [0,1]. \tag{\textbf{LI}}$$

- La *propiedad de orden*,

$$x \leq y \Longleftrightarrow I(x,y) = 1, \quad x,y \in [0,1]. \tag{\textbf{OP}}$$

- El *principio de identidad*,

$$I(x,x) = 1, \quad x \in [0,1]. \tag{\textbf{IP}}$$

Definición 3 ([4, Definición 1.4.15]). *Sea I una implicación borrosa. La función N_I definida como $N_I(x) = I(x,0)$ para todo $x \in [0,1]$, se dice que es la* negación natural *asociada a I.*

Finalmente, se recuerdan las definiciones de las implicaciones f y g-generadas de Yager.

Definición 4 ([4, Definición 3.1.1]). *Sea $f : [0,1] \to [0,\infty]$ una función continua y estrictamente decreciente con $f(1) = 0$. La función $I_f : [0,1]^2 \to [0,1]$ definida por*

$$I_f(x,y) = f^{-1}(x \cdot f(y)), \quad x,y \in [0,1],$$

entendiendo que $0 \cdot \infty = 0$, se dice que es una implicación f-generada. *La función f es un f-generador de la implicación I_f.*

Definición 5 ([4, Definición 3.2.1]). *Sea $g : [0,1] \to [0,\infty]$ una función continua y estrictamente creciente con $g(0) = 0$. La función $I_g : [0,1]^2 \to [0,1]$ definida por*

$$I_g(x,y) = g^{(-1)}\left(\frac{1}{x} \cdot g(y)\right), \quad x,y \in [0,1],$$

entendiendo que $\frac{1}{0} = \infty$ y $\infty \cdot 0 = \infty$, se dice que es una implicación g-generada, *donde la función $g^{(-1)}$ es la pseudo-inversa de g definida como*

$$g^{(-1)}(x) = \begin{cases} g^{-1}(x) & si\ x \in [0,g(1)], \\ 1 & si\ x \in [g(1),\infty]. \end{cases}$$

La función g es un g-generador de la implicación I_g.

3. Caracterización de las (h, e)-implicaciones

En [9], se introdujeron entre otras clases de implicaciones, las h-implicaciones y las (h, e)-implicaciones. Estas dos familias se generan a partir de un generador aditivo de una uninorma representable de forma similar a como las implicaciones f y g-generadas de Yager son generadas a partir de generadores aditivos de t-normas y t-conormas continuas y Arquimedianas, respectivamente. A continuación, se recuerda la definición de las (h, e)-implicaciones.

Definición 6 ([9, Definición 8]). *Sea $e \in (0,1)$ fijo y $h : [0,1] \to [-\infty, \infty]$ una función continua y estrictamente creciente tal que $h(0) = -\infty$, $h(e) = 0$ y $h(1) = +\infty$. La función $I^{h,e} : [0,1]^2 \to [0,1]$ definida por*

$$I^{h,e}(x,y) = \begin{cases} 1 & si\ x = 0, \\ h^{-1}\left(\frac{x}{e} \cdot h(y)\right) & si\ x > 0, y \leq e, \\ h^{-1}\left(\frac{e}{x} \cdot h(y)\right) & si\ x > 0, y > e, \end{cases}$$

se llamará (h, e)-implicación. La función h se llamará un h-generador de la implicación $I^{h,e}$.

Las h-implicaciones I^h se definen de forma similar cambiando los factores $\frac{x}{e}$ y $\frac{e}{x}$ por x y $\frac{1}{x}$, respectivamente. Las (h, e)-implicaciones se propusieron con el objetivo de modificar las h-implicaciones para que verificaran (NP_e), es decir, $I(e, y) = y$ para todo $y \in [0,1]$, el equivalente a (NP) para implicaciones derivadas de uninormas. Una vez introducidas ambas familias, se caracterizaron las h-implicaciones en [12] mediante el método de generación de implicaciones borrosas a partir de dos dadas basado en un umbral horizontal. Este método se basa en un escalado adecuado de la segunda variable de las implicaciones iniciales.

Teorema 1 ([12, Teorema 3]). *Sean I_1, I_2 dos implicaciones y $e \in (0,1)$. La función $I_{I_1 - I_2} : [0,1]^2 \to [0,1]$, llamada la implicación generada con umbral horizontal e a partir de I_1 y I_2, definida por*

$$I_{I_1-I_2}(x,y) = \begin{cases} 1 & si\ x = 0, \\ e \cdot I_1\left(x, \dfrac{y}{e}\right) & si\ x > 0, y \leq e, \\ e + (1-e) \cdot I_2\left(x, \dfrac{y-e}{1-e}\right) & si\ x > 0, y > e, \end{cases}$$

es una implicación borrosa.

Concretamente, las h-implicaciones fueron caracterizadas como un caso particular del método del umbral horizontal usando las implicaciones f y g-generadas de Yager.

Teorema 2 ([12, Teorema 2, Observación 3]). *Sean $I : [0,1]^2 \to [0,1]$ una función y $e \in (0,1)$. Entonces I es una h-implicación respecto a e si, y sólo si, existen implicaciones f y g-generadas con $f(0) = g(1) = +\infty$, I_f e I_g*

respectivamente, tales que $I = I_{I_f - I_g}$. *Además, en este caso, los generadores* h, f *y* g *se encuentran relacionados de la siguiente forma:*

$$\begin{aligned} f(x) &= -h(e \cdot x) \quad \textit{para todo } x \in [0,1], \\ g(x) &= h(e + (1-e) \cdot x) \quad \textit{para todo } x \in [0,1], \\ h(x) &= \begin{cases} -f\left(\dfrac{x}{e}\right) & \textit{si } x \leq e, \\ g\left(\dfrac{x-e}{1-e}\right) & \textit{si } x > e. \end{cases} \end{aligned}$$

Sin embargo, aunque las (h, e)-implicaciones son también un caso particular de las implicaciones con umbral horizontal[1] ya que satisfacen $I^{h,e}(x, e) = e$ para todo $x > 0$, hasta ahora no habían podido ser aún caracterizadas. El siguiente resultado proporciona una primera aproximación a la caracterización.

Teorema 3. *Sean* $I : [0,1]^2 \to [0,1]$ *una función y* $e \in (0,1)$. *Entonces* I *es una* (h, e)*-implicación respecto a* e *si, y sólo si, existen un* f*-generador con* $f(0) = +\infty$ *y un* g*-generador con* $g(1) = +\infty$ *tales que* I *viene dada por*

$$I(x,y) = I_{I_1 - I_2}(x,y) = \begin{cases} 1 & \textit{si } x = 0, \\ e \cdot f^{-1}\left(\frac{x}{e} \cdot f\left(\frac{y}{e}\right)\right) & \textit{si } x > 0, y \leq e, \\ e + (1-e) \cdot g^{-1}\left(\frac{e}{x} \cdot g\left(\frac{y-e}{1-e}\right)\right) & \textit{si } x > 0, y > e, \end{cases}$$

donde I_1 *e* I_2 *son las implicaciones borrosas dadas por*

$$I_1(x,y) = f^{-1}\left(\frac{x}{e} \cdot f(y)\right), \quad I_2(x,y) = g^{-1}(\frac{e}{x} \cdot g(y)).$$

Además, en este caso, los generadores h, f *y* g *están relacionados del siguiente modo:*

$$\begin{aligned} f(x) &= -h(e \cdot x) \quad \textit{para todo } x \in [0,1], \\ g(x) &= h(e + (1-e) \cdot x) \quad \textit{para todo } x \in [0,1], \\ h(x) &= \begin{cases} -f\left(\frac{x}{e}\right) & \textit{si } x \leq e, \\ g\left(\frac{x-e}{1-e}\right) & \textit{si } x > e. \end{cases} \end{aligned}$$

Como se puede observar, las (h, e)-implicaciones se han caracterizado de forma similar a las h-implicaciones. Sin embargo, la presencia del factor e en su definición provoca que no sean generadas a partir de dos implicaciones de Yager con el método de generación con umbral horizontal. El siguiente ejemplo ilustra este hecho.

[1] Las implicaciones generadas con umbral horizontal se caracterizaron en [12] como aquellas que satisfacen $I(x, e) = e$ para todo $x > 0$.

Ejemplo 1. Considerando el h-generador respecto a $e = \frac{1}{2}$ dado por $h_1(x) = ln\left(\frac{x}{1-x}\right)$, se obtiene la siguiente (h, e)-implicación

$$I^{h_1}(x,y) = \begin{cases} 1 & \text{si } x = 0, \\ \frac{y^{2x}}{(1-y)^{2x}+y^{2x}} & \text{si } x > 0, y \leq \frac{1}{2}, \\ \frac{y^{\frac{1}{2x}}}{(1-y)^{\frac{1}{2x}}+y^{\frac{1}{2x}}} & \text{si } x > 0, y > \frac{1}{2}. \end{cases}$$

Usando el teorema anterior, esta implicación se ha obtenido a partir de las implicaciones

$$I_1(x,y) = f^{-1}\left(\tfrac{x}{e} \cdot f(y)\right) = \frac{2y^{2x}}{(2-y)^{2x}+y^{2x}},$$
$$I_2(x,y) = g^{-1}(\tfrac{e}{x} \cdot g(y)) = \frac{(1+y)^{\frac{1}{2x}} - (1-y)^{\frac{1}{2x}}}{(1+y)^{\frac{1}{2x}} + (1-y)^{\frac{1}{2x}}},$$

donde $f(x) = -\ln\left(\frac{x}{2-x}\right)$ y $g(x) = \ln\left(\frac{1+x}{1-x}\right)$ mediante el método de generación con umbral horizontal.

Notar que en el ejemplo anterior, I_1 e I_2 no son implicaciones de Yager ya que $\frac{x}{e} > 1$ para todo $e < x \leq 1$ y $\frac{e}{x} > 1$ para todo $0 \leq x < e$. En particular, ninguna satisface (NP) y por lo tanto, no pertenecen a ninguna de las familias más conocidas de implicaciones borrosas.

4. Implicaciones de Yager extendidas

En esta sección, se pretenden introducir dos nuevas clases de implicaciones borrosas que incluyan las implicaciones I_1 e I_2 obtenidas en el Teorema 3. Estas nuevas clases de implicaciones serán parecidas a las implicaciones de Yager pero admitiendo factores internos con rango $[0, +\infty]$. Si las definiciones de las implicaciones de Yager se observan con atención, se pueden considerar x y $\frac{1}{x}$ como casos particulares de funciones más generales con un rango potencialmente mayor.

4.1. Extensión de las implicaciones f-generadas

La extensión se basará en una generalización de la función $g(x) = x$ a una función $g : [0,1] \to [0, +\infty]$ continua y estrictamente creciente con $g(0) = 0$.

Definición 7. *Sean $f : [0,1] \to [0,\infty]$ una función continua y estrictamente decreciente con $f(1) = 0$ y $g : [0,1] \to [0, +\infty]$ una función continua y estrictamente creciente con $g(0) = 0$. La función $I_{f,g} : [0,1]^2 \to [0,1]$ definida por*

$$I_{f,g}(x,y) = f^{(-1)}(g(x) \cdot f(y)), \quad x, y \in [0,1]$$

con el convenio $0 \cdot \infty = 0$, se llamará un operador (f, g)-generado.

Una primera diferencia con las implicaciones f-generadas de Yager es que en este caso, es necesario el uso de la pseudo-inversa ya que cuando $f(0) < +\infty$, $g(x) \cdot f(y)$ puede ser mayor que $f(0)$. Sin embargo, se puede omitir la pseudo-inversa en la expresión de la siguiente forma

$$I_{f,g}(x, y) = f^{-1}(\text{mín}\{g(x) \cdot f(y), f(0)\}),\ x, y \in [0, 1].$$

Observación 1. Esta extensión es diferente a la propuesta en [14]. Aunque se trata de una idea similar, las (f, g)-implicaciones definidas en el artículo anterior se generan a partir de funciones $g : [0, 1] \to [0, 1]$ estrictamente crecientes con $g(0) = 0$ y $g(1) = 1$, no necesariamente continuas. En la extensión que se propone aquí, se extiende el rango de la función g a $[0, +\infty]$ y en consecuencia, $g(1) \neq 1$ en general. Obviamente, si $g(1) = 1$ entonces ambas propuestas coinciden y de hecho, las implicaciones (f, g)-generadas son en realidad φ-conjugadas de implicaciones f-generadas con f-generador $f \circ g^{-1}$ y $\varphi = g$ (ver Observación 2.1 en [14]).

El siguiente teorema determina cuando los operadores (f, g)-generados son implicaciones borrosas en el sentido de la definición 2.

Teorema 4. *Un operador (f, g)-generado $I_{f,g}$ es una implicación borrosa si, y sólo si, se cumple una de las siguientes condiciones:*

1. $f(0) = +\infty$.
2. $f(0) < +\infty$ *y* $g(1) \geq 1$.

Cuando un operador (f, g)-generado sea de hecho una implicación borrosa, usaremos el término (f, g)-implicación y llamaremos a (f, g) una pareja admisible de generadores. Como se puede comprobar fácilmente, la implicación I_1 del Teorema 3 es una (f, g)-implicación ya que $f(0) = +\infty$ y $g(x) = \frac{x}{e}$ con $e \in (0, 1)$.

A continuación, se estudiarán algunas propiedades iniciales de las (f, g)-implicaciones. En primer lugar, el siguiente resultado establece cuando esta familia satisface (NP).

Proposición 1. *Sea (f, g) una pareja admisible de generadores. Entonces $I_{f,g}$ satisface (NP) si, y sólo si, $g(1) = 1$.*

Los siguientes resultados tratan la verificación de (EP) y (LI) por parte de esta nueva familia de implicaciones.

Proposición 2. *Sea (f, g) una pareja admisible de generadores. Entonces las siguientes afirmaciones son equivalentes:*

1. *$I_{f,g}$ satisface (EP).*
2. *$f(0) = +\infty$ o ($f(0) < +\infty$ y $g(1) = 1$).*

Proposición 3. *Sea (f, g) una pareja admisible de generadores y T una t-norma. Entonces las siguientes afirmaciones son equivalentes:*

1. *La pareja de funciones $(I_{f,g}, T)$ satisface (LI).*

2. $g(1) = 1$ *y* $T = (T_P)_g$, *i.e.*, $T(x, y) = g^{-1}(g(x) \cdot g(y))$ *para todo* $x, y \in [0, 1]$.

En cambio, las (f, g)-implicaciones no verifican nunca ni (IP) ni (OP).

Proposición 4. *Sea* (f, g) *una pareja admisible de generadores. Entonces la* (f, g)*-implicación* $I_{f,g}$ *no satisface ni (IP) ni (OP).*

Finalmente, se estudia la negación natural de estas implicaciones que puede llegar a ser una negación borrosa fuerte.

Proposición 5. *Sea* (f, g) *una pareja admisible de generadores. Entonces se cumplen las siguientes propiedades:*

1. *Si* $f(0) = +\infty$, *la negación natural* $N_{I_{f,g}}$ *es la negación mínima o de Gödel* N_{D_1}.
2. *Si* $f(0) < +\infty$, *la negación natural* $N_{I_{f,g}}$ *viene dada por*

$$N_{I_{f,g}}(x) = \begin{cases} f^{-1}(g(x) \cdot f(0)) & \text{si } g(x) \leq 1, \\ 0 & \text{si } g(x) > 1. \end{cases}$$

3. *Si* $f(0) < +\infty$ *y* $g(1) = 1$, *la negación natural* $N_{I_{f,g}}$ *es una negación estricta.*
4. *Si* $f(0) < +\infty$, $g(1) = 1$ *y* $f^{-1}(g(x) \cdot f(0)) = g^{-1}\left(\frac{f(x)}{f(0)}\right)$, *la negación natural* $N_{I_{f,g}}$ *es una negación fuerte.*

4.2. Extensión de las implicaciones *g*-generadas

A continuación, se propone una extensión similar de las implicaciones g-generadas generalizando $f(x) = \frac{1}{x}$ a una función $f : [0, 1] \to [0, +\infty]$ continua y estrictamente decreciente con $f(0) = +\infty$.

Definición 8. *Sean* $g : [0, 1] \to [0, \infty]$ *una función continua y estrictamente creciente con* $g(0) = 0$ *y* $f : [0, 1] \to [0, +\infty]$ *una función continua y estrictamente decreciente con* $f(0) = +\infty$. *La función* $I_{g,f} : [0, 1]^2 \to [0, 1]$ *definida por*

$$I_{g,f}(x, y) = g^{(-1)}(f(x) \cdot g(y)), \quad x, y \in [0, 1]$$

con el convenio $0 \cdot (+\infty) = +\infty$ *y* $\frac{1}{0} = +\infty$, *se llamará un* operador (g, f)-generado.

En este caso, también se puede omitir la pseudo-inversa en la expresión de la siguiente forma

$$I_{g,f}(x, y) = g^{-1}(\text{mín}\{f(x) \cdot g(y), g(1)\}), \ x, y \in [0, 1].$$

De igual modo que para las (f, g)-generadas, se determinan qué operadores (g, f)-generados son de hecho implicaciones en el sentido de la definición 2.

Teorema 5. *Un* (g, f)*-operador* $I_{g,f}$ *es una implicación borrosa si, y sólo si, se cumple una de las siguientes condiciones:*

1. $g(1) = +\infty$.
2. $g(1) < +\infty$ *y* $f(1) \geq 1$.

Cuando un (g, f)-operador sea una implicación borrosa, se usará el término (g, f)-implicación y en este caso, (g, f) será una pareja admisible de generadores. Una sencilla comprobación, muestra que la implicación I_2 del Teorema 3 es una (g, f)-implicación ya que $g(1) = +\infty$ y $f(x) = \frac{e}{x}$ con $e \in (0, 1)$ y por lo tanto, el siguiente resultado es inmediato.

Corolario 1. *Sean* $I : [0, 1]^2 \to [0, 1]$ *una función y* $e \in (0, 1)$. *Entonces* I *es una* (h, e)*-implicación respecto a* e *si, y sólo si,* $I = I_{I_1 - I_2}$ *donde* I_1 *es la implicación* (f, g)*-generada con* $f(x) = -h(e \cdot x)$ *y* $g(x) = \frac{x}{e}$ *e* I_2 *es la implicación* (g, f)*-generada con* $g(x) = h(e + (1 - e) \cdot x)$ *y* $f(x) = \frac{e}{x}$.

Para finalizar, se estudiarán algunas propiedades adicionales de las (g, f)-implicaciones. En primer lugar, se determina cuando satisfacen (NP).

Proposición 6. *Sea* (g, f) *una pareja admisible de generadores. Entonces* $I_{g,f}$ *satisface (NP) si, y sólo si,* $f(1) = 1$.

Después, se analizan las propiedades (EP) y (LI) para esta familia de implicaciones.

Proposición 7. *Sea* (g, f) *una pareja admisible de generadores. Entonces* $I_{g,f}$ *siempre satisface (EP).*

Proposición 8. *Sea* (g, f) *una pareja admisible de generadores y* T *una t-norma. Entonces las siguientes afirmaciones son equivalentes:*

1. *La pareja de funciones* $(I_{g,f}, T)$ *satisface (LI).*
2. $f(1) = 1$ *y* $T(x, y) = f^{-1}(f(x) \cdot f(y))$ *para todo* $x, y \in [0, 1]$.

A diferencia de las (f, g)-implicaciones, las (g, f)-implicaciones satisfacen en algunos casos (IP) y (OP).

Proposición 9. *Sea* (g, f) *una pareja admisible de generadores. Entonces* $I_{g,f}$ *satisface (IP) si, y sólo si,* $g(1) < +\infty$ *y* $f(x) \geq \frac{g(1)}{g(x)}$ *para todo* $x \in [0, 1]$.

Proposición 10. *Sea* (g, f) *una pareja admisible de generadores. Entonces las siguientes afirmaciones son equivalentes:*

1. $I_{g,f}$ *satisface (OP).*
2. $g(1) < +\infty$ *y existe una constante* $c \in (0, +\infty)$ *tal que* $f(x) = \frac{c}{g(x)}$.
3. $I_{g,f}(x, y) = f^{-1}\left(\text{máx}\left\{f(1), \frac{f(y)}{f(x)}\right\}\right)$

Finalmente, la negación natural es siempre la negación borrosa mínima.

Corolario 2. *Sea* (g, f) *una pareja admisible de generadores. Entonces la negación natural* $N_{I_{g,f}}$ *es la negación borrosa mínima o de Gödel* N_{D_1}.

5. Conclusiones y trabajo futuro

En este trabajo, se han caracterizado las (h, e)-implicaciones a partir de dos nuevas familias de implicaciones, introducidas como extensiones de las implicaciones f y g-generadas de Yager. Estas extensiones se basan en considerar un factor interno más general que los habituales $g(x) = x$ y $f(x) = \frac{1}{x}$ en las implicaciones f y g-generadas, respectivamente. Además, se han determinado las condiciones bajo las cuáles se verifican algunas propiedades adicionales de las implicaciones borrosas. Como trabajo futuro, sería interesante considerar una extensión de las (h, e)-implicaciones mediante funciones que generalicen $\frac{x}{e}$ y $\frac{e}{x}$.

Agradecimientos. Este trabajo ha sido financiado por el proyecto MTM2009-10320 con fondos FEDER.

Referencias

1. I. Aguiló, J. Suñer, and J. Torrens. A characterization of residual implications derived from left-continuous uninorms. *Inform Sciences*, 180(20):3992–4005, 2010.
2. M. Baczyński, G. Beliakov, H. Bustince, and A. Pradera. *Advances in Fuzzy Implication Functions*, volume 300 of *Studies in Fuzziness and Soft Computing Series*. Springer-Verlag GmbH, 2013.
3. M. Baczyński and B. Jayaram. On the characterization of (S,N)-implications. *Fuzzy Sets and Systems*, 158:1713–1727, 2007.
4. M. Baczyński and B. Jayaram. *Fuzzy Implications*, volume 231 of *Studies in Fuzziness and Soft Computing*. Springer, Berlin Heidelberg, 2008.
5. J. C. Fodor and M. Roubens. *Fuzzy Preference Modelling and Multicriteria Decision Support*. Kluwer Academic Publishers, Dordrecht, 1994.
6. H.-W. Liu. On a new class of implications: (g, min)-implications and several classical tautologies. *Int J Uncertain Fuzz*, 20(1):1–20, 2012.
7. M. Mas, M. Monserrat, and J. Torrens. QL versus D-implications. *Kybernetika*, 42:351–366, 2006.
8. M. Mas, M. Monserrat, J. Torrens, and E. Trillas. A survey on fuzzy implication functions. *IEEE Transactions on Fuzzy Systems*, 15(6):1107–1121, 2007.
9. S. Massanet and J. Torrens. On a new class of fuzzy implications: h-implications and generalizations. *Information Sciences*, 181(11):2111 – 2127, 2011.
10. S. Massanet and J. Torrens. On a generalization of Yager's implications. In S. Greco et al., editor, *Advances on Computational Intelligence, Part II – Proc. IPMU 2012*, volume 298 of *Communications in Computer and Information Science*, pages 315–324, Catania, Italy, 2012. Springer.
11. S. Massanet and J. Torrens. On the characterization of Yager's implications. *Information Sciences*, 201:1–18, 2012.
12. S. Massanet and J. Torrens. Threshold generation method of construction of a new implication from two given ones. *Fuzzy Sets and Systems*, 205:50–75, 2012.
13. E. Trillas, M. Mas, M. Monserrat, and J. Torrens. On the representation of fuzzy rules. *Int. J. Approx. Reasoning*, 48(2):583–597, 2008.
14. A. Xie and H. Liu. A generalization of Yager's f-generated implications. *International Journal of Approximate Reasoning*, 54(1):35 – 46, 2013.
15. R. Yager. On some new classes of implication operators and their role in approximate reasoning. *Information Sciences*, 167:193–216, 2004.

Una Aplicación de la Similitud Semántica en la mejora de la selección de Información Clínica

Israel Alonso[1], Francisco P. Romero[2], David Contreras[1], Jose A. Olivas[2]

[1] Depto. de Sistemas Informáticos, Univ. Pontificia de Comillas - ICAI
c/ Alberto Aguilera, 23, 28015 Madrid, España
{ialonso,davidcb}@dsi.icai.upcomillas.es
[2] Dept. Tecnologías y Sistemas de la Información, Univ. de Castilla La Mancha,
Paseo de la Universidad, 4, 13071 Ciudad Real, España
{FranciscoP.Romero,JoseAngel.Olivas}@uclm.es

Resumen Este trabajo se centra en la utilización de técnicas semánticas para la recuperación de información médica a partir de unos criterios de inclusión especificados por el usuario. En la propuesta se incluye un método de representación conceptual tanto de los criterios como de los documentos sobre los que se busca. Este método está basado principalmente en el uso de diferentes métricas de similitud semántica entre conceptos extraídas del MetaTesauro UMLS (Unified Medical Language System). Además se incluye un algoritmo de ordenación de los documentos utilizando relaciones borrosas.

Keywords: Recuperación de Información, Similitud Semántica, Documentación Clínica, UMLS.

1. Introducción

Actualmente, con el masivo uso de Internet, la cantidad de información biomédica a la que un profesional de la salud tiene acceso es realmente difícil de manejar. Es por ello que cada vez se hace más necesario contar con sistemas de recuperación de información (S.R.I) que permitan, además de localizar la información necesaria, permitir ordenarla y entregarla de forma eficiente a los usuarios con el fin de satisfacer sus necesidades de información sobre enfermedades, síntomas, etc.

Dentro de este entorno se están desarrollando muchos sistemas que hacen uso de recursos externos como ontologías que demuestran que la integración la integración de conocimiento puede ayudar a mejorar los sistemas [1]. Dentro de estos recursos semánticos cabe destacar los recogidos en el Metatesauro UMLS[3] (Unified Medical Language System) como son entre otros: Medical Subject Headings [4] (MeSH), Systematized Nomenclature of Medicine – Clinical Terms [5] (SNOMED CT) e International Classification of Diseases, Ninth Revision (ICD9)[6].

[3] http://www.nlm.nih.gov/research/umls/
[4] http://www.ncbi.nlm.nih.gov/mesh
[5] http://www.ihtsdo.org/snomed-ct/
[6] http://www.cdc.gov/nchs/icd9.htm

La utilización de estos recursos semánticos permite el desarrollo de modelos de representación, acceso y tratamiento de la semántica de los documentos dentro de este dominio de conocimiento.

Este trabajo está centrado en la recuperación de información clínica relevante para estudios epidemiológicos, clínicos o de cohortes. Dentro de estos estudios es especialmente relevante la selección de los pacientes sobre los que se va a trabajar a partir de unos ciertos criterios de inclusión y exclusión. Estos criterios normalmente son expresados mediante consultas en lenguaje natural como "Pacientes con Demencia Senil" (*Patients with Dementia*) o "Pacientes que no han presentado pneumothorax" (*Patients with No evidence of pneumothorax*). Dado que la información clínica existente está normalmente desestructurada y organizada con otros propósitos. Es necesario identificar y seleccionar los conceptos que definen y representan tanto los criterios de inclusión de la consulta, como el contenido de la propia información clínica.

En este documento, se define un modelo de representación conceptual tanto de la información clínica (documentos clínicos, etc.) como de los criterios de búsqueda de las consultas, mediante la identificación y extracción de los conceptos médicos contenidos en ellos. Además se propone un sistema de evaluación que permite calcular la relevancia o proximidad semántica de una consulta, frente a un conjunto de documentos clínicos. De esta forma, se pretende emular el comportamiento de un experto a la hora de seleccionar y determinar aquella información clínica que es relevantes o de importancia con respecto a los criterios de inclusión en un estudio de cohortes o epidemiológico. Como material para los ejemplos y experimentos de este trabajo se ha contado con los informes facilitados en el TREC 20111 Medical Track[7], cuyo objetivo fue el fomento de la investigación sobre tecnologías que permitan la recuperación de documentos clínicos electrónicos, en función del contenido semántico de los campos de texto libre incluidos.

El resto del trabajo está organizado como sigue: En la sección 2 se presenta el modelo de representación conceptual tanto de documentos como de consultas basado en conceptos presentes en el metatesauro UMLS. En la sección 3 se presenta el método de calificación por relevancia de la información recuperada. En la sección 4 se presentan los resultados del estudio preliminar llevado a cabo con el fin de comprobar la validez de la propuesta, y finalmente, en la sección 5, se presentan las conclusiones a las que se ha llegado y las líneas de trabajo futuro que se presentan.

2. Representación Conceptual de Información Médica

Uno de los puntos fundamentales de este trabajo es el análisis, extracción y representación tanto de la información contenida en cada uno de los documentos clínicos como de los criterios de búsqueda de cada consulta. Con este fin, es necesario realizar el tratamiento del texto libre almacenado, extrayendo las frases

[7] http://trec.nist.gov/pubs/call2011.html/

que lo conforman para posteriormente asociar e identificar los conceptos contenidos en el MetaTesauro UMLS que reflejan dicha información. Como resultado de este análisis se obtiene una representación del conocimiento contenido en cada documento o consulta, en forma de vector o conjunto de conceptos UMLS que nos permitirá posteriormente definir la similitud semántica entre ambos.

2.1. Representación de los documentos clínicos

Este proceso consiste en el tratamiento y transformación del conocimiento contenido en el texto libre y los códigos ICD9 de cada documento, para su representación basada en conceptos UMLS.

Para el procesamiento, análisis y división del texto en sentencias y frases, se realizó un estudio de herramientas como: OpenNLP[8] , ScopeFinder[9] y MetaMap[10]. Siendo seleccionado entre ellos, el programa MetaMap [2], por su alta integración con el MetaTesauro UMLS en la obtención de los resultados, así como por su eficiente tratamiento de la negación. MetaMap permite extraer e identificar del texto de los documentos, todos aquellos conceptos biomédicos existentes en el MetaTesauro UMLS y marcar aquellos que se encuentran en el ámbito lingüístico de la negación, gracias a su implementación del algoritmo NegEx [3]. Es importante resaltar, que MetaMap es un software altamente configurable que incluye en su última pre-release del 2013, nuevas extensiones que permiten la evaluación de la negación para todos los tipos semánticos y no solamente algunos, como ocurría en versiones anteriores o en las otras herramientas.

Con el resultado que MetaMap nos ofrece al procesar un documento clínico se plantea representar conceptualmente cada uno de los documentos clínicos mediante el conjunto de conceptos que aparecen en él que se va a denotar por C_v (*Concept of Visit*).

$$C_v = \left[C_v^1, \ldots, C_v^m, \right] \tag{1}$$

siendo m el número de conceptos distintos detectado en el mismo. En este contexto, hay que hacer notar que se está trabajando con documentos clínicos almacenados en XML (siguiendo un estándar o no). Es por ello, que los propios documentos son procesados previamente para extraer el texto libre que contienen. Por ejemplo, dentro del esquema de los documentos facilitados por TREC serán las etiquetas $< report_text >$, $< subtype >$, $< type >$ y, $< chief_complaint >$ las que contengan texto libre, y etiquetas como $< admit_diagnosis >$ y las que contienen códigos ICD9. Por ejemplo, si aplicamos MetaMap a la frase "*Chronic dementia.*" el resultado es el siguiente:

```
Processing 00000000.tx.4: Chronic dementia.

Phrase: "Chronic dementia."
```

[8] http://incubator.apache.org/opennlp/
[9] http://scopefinder.sourceforge.net/
[10] http://metamap.nlm.nih.gov/

```
1. Meta Mapping (888):
   694   C0205191:chronic (chronic) [Temporal Concept]
   861   C0497327:Dementia (Dementia) [Mental or  Behavioral Dysfunction]
2. Meta Mapping (888):
   694   C0205191:chronic (chronic) [Temporal Concept]
   861   C0011265:dementia (Presenile dementia)  [Mental or Behavioral Dysfunction]
```

De este resultado se puede extraer un conjunto de conceptos procedente de la unión del concepto con identificador (CUI) $C0205191$ para el texto "*Chronic*", junto con los conceptos $C0497327$ y $C0011265$ para el texto "*dementia*". Por lo tanto, los elementos del conjunto C_v serían los siguientes:

$$C_v = [C0205191, C0497327, C0011265] \tag{2}$$

De igual manera, y como ejemplo de representación de la negación, el resultado ofrecido por MetaMap al tratar la la frase "*No evidence of pneumothorax*" es el siguiente:

```
Processing 00000000.tx.3: No evidence of pneumothorax.

Phrase: "No evidence of pneumothorax."
1. Meta Mapping (810):
   833 N C0332120:Evidence of (Evidence of)  [Functional Concept]
   604 N C0032326:Pneumothorax (Pneumothorax)  [Disease  or Syndrome]
2. Meta Mapping (810):
   833 N C0332120:Evidence of (Evidence of)  [Functional Concept]
   604 N C1963215:Pneumothorax (Pneumothorax Adverse  Event) [Finding]
```

En este resultado se puede observar como aquellos conceptos identificados en un contexto de negación, son marcados con el valor "N". Estos conceptos, no se tendrán en cuenta a la hora de evaluar su similitud con respecto a los conceptos de la consulta, ya que representan la ausencia de ellos.

2.2. Representación de las sentencias de búsqueda

Cuando el profesional expresa unos criterios de inclusión y/o exclusión para un determinado estudio, lo hace mediante la utilización de una sentencia de búsqueda o consulta como lo haría en cualquier buscador o sistema de recuperación de información.

El proceso de representación de la sentencia de búsqueda es muy similar al proceso de representación de la información clínica, por lo que de la misma manera se realiza la extracción y representación de los conceptos contenidos en cada una de las consultas. En este caso no es necesaria la extracción del texto contenido en ningún documento XML, ya que la consulta es una frase en texto libre, definida por el usuario y que refleja sus necesidades de información o criterio de búsqueda. De esta manera, la frase o sentencia de búsqueda será procesada mediante MetaMap, obteniendo como resultado un vector denominado C_t (*Concepts of the topic*) que representa la semántica de dicha consulta, siendo n el número de conceptos distintos detectado en la misma.

$$C_t = \left[C_t^1, \ldots, C_t^n, \right] \tag{3}$$

Así por ejemplo, para la búsqueda correspondiente al topic 105 de TREC, "Patients with dementia" el resultado es el siguiente:

```
Processing 00000000.tx.1: Patients with dementia

Phrase: "Patients with dementia"
1. Meta Mapping (746):
   790   C0030705:Patients (Patients) [Patient or  Disabled Group]
   623   C0497327:Dementia (Dementia) [Mental or  Behavioral Dysfunction]
2. Meta Mapping (746):
   790   C0030705:Patients (Patients) [Patient or  Disabled Group]
   623   C0011265:dementia (Presenile dementia)  [Mental or Behavioral Dysfunction]
```

De esta manera, se extraen los 2 conceptos biomédicos incluidos en UMLS ($C0497327$, $C0011265$) que representan el texto "Dementia". Junto con el concepto ($C0030705$), para el texto "patients".

En interesante observar, que en el proceso de extracción de conceptos se puede realizar una expansión implícita de estos. Ya que, por ejemplo, en el caso del concepto ($C0011265$), se puede hacer referencia a las cadenas "dementia" y "Presenile dementia" indistintamente.

3. Calificación de Relevancia basada en Similitud Semántica

En esta sección, se presenta el proceso de construcción de la matriz de similitud semántica que permitirá establecer la relevancia de la información contenida en un documento clínico, con respecto a una consulta o tema de búsqueda.

3.1. Construcción de la relación borrosa de similitud semántica

Una vez extraídos los conceptos biomédicos y representados tanto los documentos clínicos y los temas de búsqueda en vectores de conceptos, el paso siguiente consiste en obtener un grado pertenencia del documento clínico con respecto al criterio de inclusión de la búsqueda lo cual va a ser representado mediante una relación borrosa. El concepto de relación borrosa es una generalización del concepto de relación de la teoría clásica de conjuntos. Mientras que una relación entre dos conjuntos clásicos describe la existencia o no de asociación entre los elementos de ambos conjuntos, una relación borrosa describe el grado de asociación o interacción entre los elementos de dos o más conjuntos borrosos.

De esta manera, se define una relación borrosa S que representa a la similitud semántica entre los dos conjuntos de C_v y C_t.

Esta similitud semántica es un valor entre 0 y 1 el cual va a ser calculado a partir de la aplicación de la métrica path[4], sobre la totalidad de los recursos contenidos en UMLs y sus relaciones jerárquicas directas (PAR/CHD) lo que

viene dado por la función $F_{path-UMLS}$, calculada utilizando la librería UMLS-Similarity [5].

Dado que estos valores son discretos se puede expresar una relación borrosa en forma de matriz, denominada matriz relacional borrosa, cuyos elementos toman valores en el intervalo $[0, 1]$.

$$S = \begin{pmatrix} F_{path-UMLS}(C_v^1, C_t^1) & \dots & F_{path-UMLS}(C_v^1, C_t^n) \\ \vdots & \ddots & \vdots \\ F_{path-UMLS}(C_v^m, C_t^1) & \dots & F_{path-UMLS}(C_v^m, C_t^n) \end{pmatrix}$$

Con el fin de contar con la mayor cantidad de conocimiento posible en este dominio todos los recursos contenidos en el MetaTesauro UMLS son utilizados. Esto es así dado que si únicamente se utilizara un único recurso se podrían perder relaciones y conceptos que podrían ser relevantes dentro del documento. Por otro lado, la elección de la métrica *path* para el cálculo de la similitud semántica entre conceptos ofrece los resultados más aproximados a los definidos por los expertos [6]. Esta métrica ha sido aplicada únicamente sobre las relaciones jerárquicas directas (PAR/CHD) representadas sobre UMLS. El uso de las relaciones jerárquicas indirectas (RB/RN), aunque supone una muy ligera mejora en los resultados, también conlleva un mayor e inaceptable coste computacional.

3.2. Construcción de la relación borrosa de similitud semántica

La relevancia de un documento clínico frente a un tema de búsqueda, se obtiene en función de la localización de los valores de proximidad en la matriz similitud semántica. De esta manera se definen tres niveles de relevancia: "relevante", parcialmente relevante, y no relevante. Sin embargo, el concepto de relevancia se ha definido en sentido impreciso y por lo tanto no es posible establecer niveles de relevancia mediante intervalos de valores *precisos ya que existe la suficiente nitidez para distinguir entre los diferentes niveles.*

Una relación binaria borrosa como S, se puede definir como unión de α-cortes de un determinado nivel α de la siguiente manera:

$$S = U\alpha \; S\alpha\epsilon\nabla$$

siendo ∇ el conjunto nivel de S y donde αS_α es una relación borrosa definida por:

$$m\alpha S\alpha(c_v^i, c_t^j) = \alpha * mS\alpha(x, y) \; \forall(c_v^i, c_t^j)\epsilon C_v, C_t$$

De esta manera podemos construir un conjunto nivel para cada uno de los niveles de relevancia especificados, en concreto, los conjuntos nivel $S_{0,5}$, $S_{0,2}$ y S_0 van a ser los conjuntos representativos de los niveles "relevante", "parcialmente relevante" y "no relevante" mencionados anteriormente. A partir de estos conjuntos nivel se obtendrá de forma directa o de mediante la adicción de diferentes cálculos el nivel de relevancia del documento.

De esta manera, un documento será "relevante" para la consulta en el caso de que el conjunto nivel $S_{0,5}$ no sea vacío, es decir, la matriz resultante no sea la matriz nula ya que esto supone que existe información en el documento clínico basada en mismos conceptos o sinónimos de los contenidos en el criterio de inclusión. Por otro lado, un documento será "no relevante" en el caso de que ambos conjuntos $S_{0,5}$ y $S_{0,2}$ sean vacíos.

En el caso de que el conjunto nivel $S_{0,5}$ sea vacío, pero no el $S_{0,2}$ no se puede asegurar que el documento sea "parcialmente relevante". Esta situación va a hacer necesario realizar nuevos cálculos para distinguir entre aquellos pares (C_v^i, c_t^j) con valor en el conjunto nivel cuyo valor $F_{path-UMLS}$ cuenta con una tendencia positiva, caso en el cual el documento será calificado como "parcialmente relevante", o con una tendencia negativa, caso en el cual el documento será calificado como "no relevante".

Como ejemplo ilustrativo de este caso, suponemos que se cuenta se cuenta con dos textos, un texto A que contiene el concepto "Psychotic disorder" y otro texto B que contiene el concepto "Heart Disease". Por otro lado, se cuenta con un tema de búsqueda "Patients with dementia" (topic 105 del TREC Medical Track 2011) que es representado también mediante dos conceptos. Por lo tanto:

$C_d^A = [C0033975]$
$C_d^B = [CC0018799]$
$C_t = [C0497327, C0011265]$

Los valores de $F_{path-UMLS}$ obtenidos son 0,33 para ambos documentos, los cuales se localizan dentro del conjunto de nivel $S_{0,2}$ que teóricamente corresponde al nivel "parcialmente relevante" pero no es una conclusión definitiva. Para determinar la tendencia del valor de $F_{path-UMLS}$ se aplica la métrica *path* únicamente sobre recursos especializados de información clínica y diagnostica (SNOMEDCT, ICD9CM y sus combinaciones), es decir, se calcula $F_{path-SNOMED}$, $F_{path-ICD9}$ y $F_{path-SNOICD9}$. Los incrementos en ciertas distancias (disminución de similitud) usando estos recursos, determinan que los resultados previos aplicando UMLS como recurso, acortaron el camino entre dos conceptos jerárquicamente distantes dentro de la taxonomía.

En el ejemplo propuesto anteriormente, en ambos casos se observa como aquellos conceptos con tendencias positivas ("Psychotic disorder"), mantienen valores de similitud que permitirían mantener el valor dentro del conjunto nivel $S_{0,2}$. Sin embargo, aquellos conceptos con tendencia negativa obtienen una similitud semántica fuera de este conjunto nivel. (Tabla 1).

Tabla 1. Resultados

	Psychotic disorder	**Heart Disease**
$F_{path-SNOMED}$	0.33	0.16
$F_{path-ICD9}$	0.25	0.16
$F_{path-SNOICD9}$	0.33	0.11

4. Experimento Preliminar

Un experimento preliminar de alcance reducido se ha llevado a cabo con el fin de verificar la viabilidad de la propuesta. En este caso se han seleccionado de forma aleatoria 4 consultas (*topics*) y 180 documentos de entre los datos facilitados en el TREC 2011 Medical Track. Las consultas seleccionadas han sido las siguientes:

- "women with osteopenia"
- "patients with dementia"
- "patients with hearing loss"
- "patients admitted with chest pain and assessed with CT angiography"

Para cada uno de los pares (consulta, documento) se ha obtenido un nivel de relevancia que se ha comparado con el proporcionado por los expertos y por el proporcionado por la línea base , que en este caso ha venido dada por la utilización de Lucene [7] como herramienta de recuperación de información. Como medidas de rendimiento, se han calculado, siguiendo su definición estándar, los valores de precisión (P), exhaustividad (R) y métrica F (la media armónica entre precisión y exhaustividad) [8]. Para la evaluación del rendimiento medio para todas las categorías se utilizará el método basado en *micro-medias* (*micro-averaging*).

Los resultados pueden observarse en la Tabla 2. En ellos se puede observar que se obtienen mejores resultados tanto en precisión como exhaustividad que en la línea base propuesta. Así mismo, también se detecta que la expansión implícita de los conceptos que se realiza en la propuesta provoca que la precisión sea más baja que la exhaustividad por lo que se deben añadir nuevos mecanismos al propuesto con el fin de mejorar los resultados en cuanto a precisión.

Tabla 2. Resultados

	P	**R**	**F**
Propuesta	**75 %**	**81 %**	**78 %**
Lucene	74 %	68 %	72 %

5. Conclusiones

En este trabajo, se ha propuesto un método de representación tanto de documentos como de consultas, así como un procedimiento de ordenación en base al análisis de las matrices de similitud semántica entre los conceptos contenidos. Si bien, los resultados preliminares han sido prometedores, se requiere una mejora en el proceso de automatización de la generación de matrices semánticas con el fin de evaluar su rendimiento sobre la totalidad del repositorio de documentos clínicos contenidos en el TREC Medical Track 2011.

Como trabajos futuros se plantea el uso de los métodos propuestos para la representación de los documentos clínicos con el fin de mejorar tareas como el clustering y la clasificación de información clínica.

Referencias

1. Patwardhan S, Pedersen T.: Using WordNet-based context vectors to estimate the semantic relatedness of concepts. Making Sense of Sense: Bringing Psycholinguistics and Computational Linguistics Together. 2006:1
2. Aronson A., Lang F.: An overview of MetaMap: historical perspective and recent advances. J Am Med Inform Assoc. 2010 May-Jun;17(3) : 229–3.
3. Chapman W., Bridewell W., Hanbury P., Cooper G.F., Buchanan B.G.: A simple algorithm for identifying negated findings and diseases in discharge summaries. J Biomed Inform 2001:301-10.
4. Pedersen T., Pakhomov, S. Patwardhan S., Chute C.: Measures of semantic similarity and relatedness in the biomedical domain. J.B.I. 2007;40(3):288-99.
5. McInnes B., Pedersen T., Pakhomov S.: UMLS-interface and UMLS-similarity: Open source software for measuring paths and semantic similarity. AMIA annual symposium proceedings; American Medical Informatics Association; pp. 431-35. 2009.
6. Alonso, I., Contreras, D., Romero, F.P.: Impacto de las relaciones jerárquicas inter/intra-recursos en la similitud semántica entre conceptos del dominio biomédico. Actas XVI Congreso Español sobre Tecnologías y Lógica Fuzzy (ESTYLF 2012), Valladolid, España, pp. 438-442, 2012.
7. Hatcher E., Gospodnetic O.: Lucene in Action. Manning Publications. 2005.
8. Van Rijsbergen C. : Information Retrieval. Butterworth, London, 1979.

Evaluating the Library 2.0 Services via Computing with Words

Francisco J. Cabrerizo[1], Rubén Heradio[1], David Fernández-Amorós[2], Manuel Herrera[3], and Enrique Herrera-Viedma[4]

[1] Department of Software Engineering and Computer Systems,
Distance Learning University of Spain (UNED), 28040 Madrid, Spain
cabrerizo@issi.uned.es,rheradio@issi.uned.es
[2] Department of Languages and Computer Systems,
Distance Learning University of Spain (UNED), 28040 Madrid, Spain
david@lsi.uned.es
[3] Department of Sociology III,
Distance Learning University of Spain (UNED), 28040 Madrid, Spain
mherrera@poli.uned.es
[4] Department of Computer Science and Artificial Intelligence,
University of Granada, 18071 Granada, Spain
viedma@decsai.ugr.es

Abstract. Web 2.0 tools have changed the relationship between users and libraries by improving the involvement of users in the libraries activities. To use such tools efficiently, it is essential to know if they are meeting user requirements and expectations, and how they could be improved to increase user satisfaction. The aim of this paper is to present a model to evaluate the quality of the Library 2.0 services. To do so, the quality assessments are defined using user's perceptions on the quality of the Library 2.0 services provided though the websites of the libraries. We assume a fuzzy linguistic modeling to represent the user's perceptions and apply tools of fuzzy computing with words based on the LOWA and LWA operators to compute the quality assessments of the Library 2.0 services. In addition, the model uses the LibQUAL+ methodology, allowing for the identification of specific 2.0 services in need of improvement and of those outstandingly satisfied by the library.

Keywords: Library 2.0, quality, computing with words, LibQUAL+.

1 Introduction

Libraries play a notable role in the educational progress, enabling and facilitating the exchange and growth of information, knowledge and culture among teachers, students and the general public. This purpose is nowadays enhanced and facilitated by the use of technology and, in recent times, by the so-called Web 2.0 [1, 2].

The term Web 2.0 was coined by O'Reilly [3] to describe the trends and business models that survived the technology sector market crash of the 1990s.

He noted that the companies which had survived the collapse were collaborative in nature, interactive, dynamic, and users created the content in these sites as much as they consumed it [2]. A Web 2.0 site may allow users to interact and collaborate with each other in a social media dialogue as creators of user-generated content in a virtual community, in contrast to websites where people are limited to the passive viewing of content. Developing the idea of Web 2.0 in the library context, the concept of Library 2.0 emerges.

Library 2.0 is a model for a modernized form of library service that reflects a transition within the library world in the way that services are delivered to users. The focus is on user-centered change and participation in the creation of content and community. Library 2.0 has multiple facets reflecting the typical means of user participation that Web 2.0 enables. These facets include blogging, tagging, social bookmarking, social networking, podcasting and so on.

As libraries are service institutions, better service will be provided if the nature and needs of users are known. As the Library 2.0 model enables and encourages participation of the user/client not only in the use of the service but also in its management, the evaluation of the Library 2.0 services needs to be judged by its users. According to user comments, observed weaknesses and strengths can be understood and, in order to eliminate defects and develop strengths, proposals can be provided to this end. Furthermore, focusing on users in the libraries and the efforts to resolve their expectations, it makes the academic libraries more dynamic [4].

Different quality evaluation models of digital libraries based on user perception have been proposed [5]. However, these models have not taken into account the new dimension of the libraries, i.e, the impact of Web 2.0 on library websites.

The aim of this paper is to propose a quality evaluation model of the Library 2.0 services. We present a model measuring the quality level of the Library 2.0 services offered by libraries through their websites according to the users' perceptions. Conventional measurement tools used by users to express their opinions are devised on numerical values, but, as the natural language is the standard representation of those concepts that humans use for communication, it seems natural that they use words (linguistic terms) instead of numerical values to provide their opinions. The use of words or sentences rather than numbers is, in general, less specific, more flexible, direct, realistic, and adequate form to express perceptions. These characteristics indicate the applicability of fuzzy set theory [6] in capturing the user's perceptions, which aids in measuring the ambiguity of concepts that are associated with human beings subjective judgment. Since the evaluation is resulted from the different evaluator's view of linguistic variables, its evaluation must therefore be conducted in an uncertain, fuzzy environment. For this reason, we use an ordinal fuzzy linguistic modeling [7] to represent user's perceptions and tools of fuzzy computing with words to compute the quality assessments. Furthermore, it is important to note that using the LibQUAL+ methodology [8], the model is able to identify both the Library 2.0 services in which the service levels should be improved and the Library 2.0 services satisfied outstandingly by the library.

The rest of this paper is set out as follows. Section 2 presents the fuzzy linguistic approach for computing with words and the LibQUAL+ methodology. Section 3 describes in detail the model we propose. Finally, Section 4 presents some conclusions.

2 Preliminaries

In this section, the LibQUAL+ methodology for quality evaluation of libraries is described and the fuzzy linguistic approach for computing with words, which is used to design our quality evaluation model of Library 2.0 services, is presented.

2.1 LibQUAL+ Methodology

In 1999, a major project to develop a standardized measure of library service quality was undertaken by the Association of Research Libraries (ARL) in collaboration with Texas A&M University. The result of this project was LibQUAL+ [8], which is an extension of the SERVQUAL (for SERVice QUALity) tool [9]. SERVQUAL has been carefully tested and widely accepted after a dozen years of application in the private sector and elsewhere. Grounded in the gap theory of service quality, the singular percept of SERVQUAL is that "only customers judge quality; all other judgments are essentially irrelevant" [10]. According to the gap model, service quality is the gap between customer's expectations and perceptions. When experiences exceed expectations, the quality of the service is high, and vice versa. Service quality is conceptualized as a gap between customers' minimum/desired expectations of service quality and their perceptions of the service quality actually received. A positive gap indicates that the service performance has exceeded customers' expectations, whereas a negative gap indicates that the service performance has fallen short of the expected service.

Following that idea, LibQUAL+ is a survey administered by the ARL to measure library user's perception of library service quality and to help libraries identify service areas needing improvement [8]. To do so, the LibQUAL+ survey is composed of 22 core questions that measure perceptions concerning three dimensions of library service quality. For each question, respondents are asked to indicate their minimum acceptable service level, their desired service level, and the perception of the actual service provided by the library by giving a score from one to nine. The minimum service level and the desired service level reflect the importance of that service to the user: a low level means that it is not considered very important, and vice versa – when the minimum or desired service level receive high scores, the issue is important. An adequacy gap (the perceived quality in relation to the accepted minimum level) and a superiority gap (the perceived quality in relation to the desired service) are determined based on the answers.

2.2 A Fuzzy Linguistic Approach for Computing with Words

Many problems present vague and imprecise aspects [11]. In such problems, the information cannot be assessed precisely in a quantitative form, but it may be done in a qualitative one, and thus, the use of a linguistic approach is necessary. For example, when attempting to qualify phenomena related to human perception, we are often led to use words in natural language instead of numerical values. The fuzzy linguistic approach is an approximate technique appropriate to deal with vague and imprecise aspects of problems. It models linguistic information by means of linguistic terms supported by linguistic variables [12], whose values are not numbers but words or sentences in a natural or artificial language. A linguistic variable is defined by means of a syntactic rule and a semantic rule.

The ordinal fuzzy linguistic approach is a very useful kind of fuzzy linguistic approach used for modeling the linguistic aspects in problems [7]. It facilitates the fuzzy linguistic modeling very much because it simplifies the definition of the semantic and syntactic rules. It is defined by considering a finite and totally ordered label set $S = \{s_i\}$, $i \in \{0, \ldots, \mathcal{T}\}$, in the usual sense, i.e., $s_i \geq s_j$ if $i \geq j$, and with odd cardinality. Typical values of cardinality used in the linguistic models are odd values, such as 7 or 9, with an upper limit of granularity of 11 or no more than 13, where the mid term represents an assessment of "approximately 0.5", and the rest of the terms being placed symmetrically around it. The semantics of the linguistic term set is established from the ordered structure of the label set by considering that each linguistic term for the pair $(s_i, s_{\mathcal{T}-i})$ is equally informative. For example, we can use the following set of nine labels to provide the user evaluations: $\{N = None, EL = Extremely\ Low, VL = Very\ Low, L = Low, M = Medium, H = High, VH = Very\ High, EH = Extremely\ High, T = Total\}$.

An advantage of the ordinal fuzzy linguistic approach is the simplicity and quickness of its computational model. It is based on the symbolic computation [7, 13] and acts by direct computation on labels by taking into account the order of such linguistic assessments in the ordered structure of linguistic terms. This symbolic tool seems natural when using the fuzzy linguistic approach, because the linguistic assessments are simply approximations which are given and handled when it is impossible or unnecessary to obtain more accurate values. Usually, the ordinal fuzzy linguistic model for computing with words is defined by establishing (i) a negation operator, $Neg(s_i) = s_j \mid j = \mathcal{T} - i$, (ii) comparison operators based on the ordered structure of linguistic terms: Maximization operator: $MAX(s_i, s_j) = s_i$ if $s_i \geq s_j$; and Minimization operator: $MIN(s_i, s_j) = s_i$ if $s_i \leq s_j$, and (iii) adequate aggregation operators. In the following, we present two aggregation operators based on symbolic computation to complete the ordinal fuzzy linguistic computational model.

The LOWA operator. An important aggregation operator of ordinal linguistic values based on symbolic computation is the LOWA operator [7]. The *Linguistic Ordered Weighted Averaging* (LOWA) is an operator used to aggregate non-weighted ordinal linguistic information, i.e., linguistic information values with equal importance [7].

Definition 1. *Let $A = \{a_1, \ldots, a_m\}$ be a set of labels to be aggregated, then the LOWA operator, ϕ, is defined as:*

$$\begin{aligned}\phi(a_1, \ldots, a_m) &= W \cdot B^T = \mathcal{C}^m\{w_k,\ b_k,\ k = 1, \ldots, m\} \\ &= w_1 \odot b_1 \oplus (1 - w_1) \odot \mathcal{C}^{m-1}\{\beta_h,\ b_h,\ h = 2, \ldots, m\}\ ,\end{aligned} \tag{1}$$

where $W = [w_1, \ldots, w_m]$ is a weighting vector, such that, $w_i \in [0, 1]$ and $\Sigma_i w_i = 1$. $\beta_h = w_h / \Sigma_2^m w_k$, $h = 2, \ldots, m$, and $B = \{b_1, \ldots, b_m\}$ is a vector associated to A, such that, $B = \sigma(A) = \{a_{\sigma(1)}, \ldots, a_{\sigma(m)}\}$, where, $a_{\sigma(j)} \leq a_{\sigma(i)}\ \forall\ i \leq j$, with σ being a permutation over the set of labels A. $\mathcal{C}^m$ is the convex combination operator of m labels and if $m = 2$, then it is defined as:

$$\mathcal{C}^2\{w_i,\ b_i,\ i = 1, 2\} = w_1 \odot s_j \oplus (1 - w_1) \odot s_i = s_k\ , \tag{2}$$

such that, $k = min\{\mathcal{T},\ i + round(w_1 \cdot (j - i))\}$, $s_j, s_i \in S$, $(j \geq i)$, where "round" *is the usual round operation, and $b_1 = s_j$, $b_2 = s_i$. If $w_j = 1$ and $w_i = 0$, with $i \neq j$, $\forall i$, then the convex combination is defined as: $\mathcal{C}^m\{w_i,\ b_i,\ i = 1, \ldots, m\} = b_j$.*

The LOWA operator is an "or-and" operator [7] and its behavior can be controlled by means of W. In order to classify OWA operators with regards to their localization between "or" and "and", Yager [14] introduced a measure of *orness*, associated with any vector W : $orness(W) = \frac{1}{m-1} \sum_{i=1}^{m} (m - i) w_i$. This measure characterizes the degree to which the aggregation is like an "or" (MAX) operation. Note that an OWA operator with $orness(W) \geq 0.5$ will be an *orlike*, and with $orness(W) < 0.5$ will be an *andlike* operator.

An important question of the LOWA operator is the determination of the weighting vector W. In [14], it was defined an expression to obtain W that allows to represent the concept of fuzzy majority [15] by means of a fuzzy linguistic nondecreasing quantifier Q [16]:

$$w_i = Q(i/n) - Q((i - 1)/n),\ i = 1, \ldots, n\ . \tag{3}$$

When a fuzzy linguistic quantifier Q is used to compute the weights of LOWA operator ϕ, it is symbolized by ϕ_Q.

The LWA operator. Another important aggregation operator of ordinal linguistic values is the *Linguistic Weighted Averaging* (LWA) operator [13]. It is based on the LOWA operator and is defined to aggregate weighted ordinal fuzzy linguistic information, i.e., linguistic information values with not equal importance.

Definition 2. *The aggregation of a set of weighted linguistic opinions, $\{(c_1, a_1), \ldots, (c_m, a_m,)\}$, $c_i, a_i \in S$, according to the LWA operator, Φ, is defined as:*

$$\Phi[(c_1, a_1), \ldots, (c_m, a_m)] = \phi(h(c_1, a_1), \ldots, h(c_m, a_m))\ , \tag{4}$$

where a_i represents the weighted opinion, c_i the importance degree of a_i, and h is the transformation function defined depending on the weighting vector W used for the LOWA operator ϕ, such that, $h = MIN(c_i, a_i)$ if $orness(W) \geq 0.5$, and $h = MAX(Neg(c_i), a_i)$ if $orness(W) < 0.5$.

We have chosen the LOWA and the LWA operators as the basis of our evalution model due to the following reasons: (i) both operators are complementary (the LWA operator is defined from the LOWA operator) and this simplifies the design of the evaluation model, (ii) both operators act by symbolic computation and, therefore, linguistic approximation processes are unnecessary and this simplifies the processes of computing with words, and (iii) the concept of fuzzy majority represented by linguistic quantifiers plays a role in the computation process and so, the assessments on Library 2.0 services are obtained according to the majority of evaluations provided by the users.

3 A Model to Evaluate the Quality of Library 2.0 Services using Tools of Computing With Words

In this section, we present a model using tools of fuzzy computing with words to evaluate the quality of the Library 2.0 services. The main characteristics of the model are: (i) it presents a set of subjective criteria related to the Library 2.0 services, (ii) the quality level of the Library 2.0 services offered by the library through its website is measured using user's perceptions on those services, (iii) it uses the ordinal fuzzy linguistic modeling [7] to represent the users' perceptions and performs computing with words based on the linguistic aggregation operators LOWA [7] and LWA [13] to compute the quality assessments, and (iv) it uses the LibQUAL+ methodology in order to identify both the Library 2.0 services in which the service levels should be improved and the Library 2.0 services already satisfied by the library.

The quality evaluation model presens two elements: (i) an evaluation scheme that contains the subjective criteria, and (ii) a computation method to generate quality assessments of Library 2.0 services.

3.1 Evaluation Scheme

To elicit user opinion regarding the quality of Library 2.0 services, we adapt the Linh's checklist [17], which is a questionnaire with 95 *yes–or–no* questions. In [17], Linh analyzes what types of Web 2.0 technologies have been applied in Australasian university libraries, as well as their purposes and features. Linh uses a checklist that includes features and criteria that emerged from the content analysis of literature on Web 2.0 in libraries. The checklist is based primarily on the usability evaluation of library websites and the list of checkpoints for web content accessibility guidelines 1.0. However, as this number of questions is excessive, we are going to define a low number of subjective criteria being easily understandable by the users in order to avoid user rejection.

We propose the evaluation scheme shown in Figure 1, which is composed of twenty four subjective criteria about the Web 2.0 tools commonly used in the libraries. Furthermore, as we are interested in obtaining the quality level of the 2.0 services offered by the library, for all of its twenty four items (subjective criteria), users are asked to indicate the minimum level of service that they would

***Category 1 – Use of RSS*:**

1. RSS usefulness to inform you about library news and events
2. RSS usefulness to inform you about new books, journals and e-resources databases
3. Adequacy of instructions on how to use RSS
4. Adequacy of links on library's web site/pages to download RSS
5. Usability of RSS items (e.g., items in RSS are searchable and classified into topics)

***Category 2 – Use of Blogs*:**

6. Blog usefulness to inform you about library news and events
7. Blog usefulness to inform you about new books, journals and e-resources databases
8. Blog usefulness to publish book and journal reviews/discussions
9. Blog usefulness for information literacy
10. Adequacy of instructions on how to use Blogs
11. Blog recentness (i.e., how recent are the latest postings)
12. Adequacy of blog links (i.e., they point to relevant Internet resources, to similar blogs, etc)
13. Blog usability (e.g., entries are searchable by keywords, entries are browsable by topics or by date, etc)

***Category 3 – Use of Podcasts*:**

14. Podcast usefulness to inform you about library news and events
15. Podcast usefulness to inform you about new books, journals and e-resources databases
16. Podcast usefulness to provide guidance to use resources and other library facilities
17. Podcast usefulness for information literacy
18. Adequacy of instructions on how to use Podcasts
19. Podcast usability (e.g., a transcript accompanies each podcast, podcasts are searchable by keywords, podcasts are browsable by topics, etc)

***Category 4 – Use of Wikis*:**

20. Wiki usefulness as subject guides
21. Wiki usefulness to provide resource listings
22. Adequacy of instructions on how to use Wikis
23. Wiki editability (e.g., users can create new pages, edit an existing page, upload files, etc)
24. Wiki usability (e.g., it provides a link to the library home page, a keyword search engine, etc)

Fig. 1. Evaluation scheme to assess the quality of Library 2.0 services.

find acceptable, the desired service level they expect, and their perceived service level (i.e., its formulation is similar to the LibQUAL+ survey). In this way, we are able to identify both the 2.0 services in which the service levels should be improved and the 2.0 services satisfied outstandingly by the library.

3.2 Computation Method

We have designed a computation method to generate quality assessment in academic libraries that has two main characteristics:

- *It is a user-centered computation method.* The quality assessment is obtained from individual linguistic judgments provided by their users rather than from assessments obtained objectively by means of the direct observation of the Library 2.0 services.
- *It is a majority guided computation method.* The quality assessments are values representative of the majority of individual judgments provided by the users of the Library 2.0 functionalities. The aggregation to compute the quality assessments is developed by means of the LOWA and LWA operators.

The computation method is composed of the following steps:

1. Users express their opinion by filling the questionnaire described in Subsection 3.1. To do so, the linguistic term set presented in Section 2.2 can be used. As a result, for each one of the users $u_j \in \{u_1, u_2, \ldots, u_n\}$ and each questionnaire item $i_k \in \{i_1, i_2, \ldots, i_{24}\}$, there is a tuple ($\mathrm{MSL}_{jk}$, DSL_{jk}, PPL_{jk}), that encodes the minimum service level, the desired service level, and the perceived performance level, provided by the user u_j on the item i_k, respectively.
2. To get the global user opinion regarding each item i_k, (MSL_k, DSL_k, PPL_k), the LWA operator is used. The users of the library do not play equal roles in measuring Library 2.0 service quality: i.e., some users should be more influential than others in some questionnaire items as it is not always valid that all group of users have equal importance with respect to the decision being made. This is because the degree of relevancy, knowledge, and experience may not be equal among them. Therefore, it is assigned a relative linguistic importance degree, $\mathrm{UI}(u_j, i_k) \in S$, for each user, u_j, on each questionnaire item, i_k. This importance degree could be obtained from a set of experts or the staff members of the library and it may be different for each library. Then, for each item i_k, its correspondent tuple (MSL_k, DSL_k, PPL_k) is computed using the LWA operator as:

$$\begin{aligned} \mathrm{MSL}_k &= \Phi_Q((\mathrm{UI}(u_1, i_k), \mathrm{MSL}_{1k}), \ldots, (\mathrm{UI}(u_n, i_k), \mathrm{MSL}_{nk})), \\ \mathrm{DSL}_k &= \Phi_Q((\mathrm{UI}(u_1, i_k), \mathrm{DSL}_{1k}), \ldots, (\mathrm{UI}(u_n, i_k), \mathrm{DSL}_{nk})), \\ \mathrm{PPL}_k &= \Phi_Q((\mathrm{UI}(u_1, i_k), \mathrm{PPL}_{1k}), \ldots, (\mathrm{UI}(u_n, i_k), \mathrm{PPL}_{nk})), \end{aligned} \tag{5}$$

   where MSL_k, DSL_k and PPL_k are the linguistic measures that represents the minimum service level, the desired service level and the perceived performance level, respectively, of the library with respect to item i_k, according to the majority (represented by the fuzzy linguistic quantifier Q) of linguistic evaluation judgments provided by the group of users.
3. To get the global user opinion regarding all items, (MSL, DSL, PPL), the LOWA operator is use. Tuple (MSL, DSL, PPL) is computed using the

LOWA operator as:

$$\begin{aligned} \text{MSL} &= \phi_Q(\text{MSL}_1, \ldots, \text{MSL}_{24}), \\ \text{DSL} &= \phi_Q(\text{DSL}_1, \ldots, \text{DSL}_{24}), \\ \text{PPL} &= \phi_Q(\text{PPL}_1, \ldots, \text{PPL}_{24}), \end{aligned} \tag{6}$$

where MSL, DSL and PPL are the linguistic measures that represents the global minimum service level, the global desired service level and the global perceived performance level, respectively, achieved for the library, according to the majority (represented by the fuzzy linguistic quantifier Q) of linguistic evaluation judgments provided by the group of users about all items.

4. Gap analysis is done for each item. According to LibQUAL+, the minimum and the desired scores establish the boundaries of a *zone of tolerance* within which the perceived scores should desirably float. The difference between the perceived and minimum scores is called the *Service Adequacy* (SA) gap, and the difference between the desired and perceived scores is called the *Service Superiority* (SS) gap. The computation of SA and SS relies on the *linguistic distance* defined as:

$$D(s_i, s_j) = s_k \ , \text{ where } k = \begin{cases} i - j \text{ if } \ i > j \\ j - i \text{ otherwise} \end{cases} \tag{7}$$

So, for each item i_k, SA_k and SS_k are computed as:

$$\begin{aligned} \text{SA}_k &= D(\text{PSL}_k, \text{MSL}_k) \\ \text{SS}_k &= D(\text{DSL}_k, \text{PSL}_k) \end{aligned} \tag{8}$$

The cases when the perceived level of service falls out of the zone of tolerance are denoted as SA^- and SS^+. SA^- means that the library is not meeting its users' minimum expectations, i.e., the perceived score is lower than the minimum one. Likewise, SS^+ means that the library is exceeding its users' desired expectations, i.e., the perceived score is higher than the desired one. Therefore, SA^- can be used to identify Library 2.0 services needing improvement, whereas SS^+ is an indicator of the extent to which Library 2.0 services are exceeding the desired expectations of the users.

4 Conclusions

In this paper we have proposed an evaluation model of Library 2.0 services that provides quality assessment according to users opinions. Considerable use has been made of fuzzy set technology to provide the ability to describe the information by using linguistic label in a way that is particularly user friendly. Furthermore, we have applied automatic tools of fuzzy computing with words based on the LOWA and LWA operators to compute quality assessments of academic libraries. In addition, using the LibQUAL+ methodology, the proposed model is able to identify Library 2.0 services which should be improved and Library 2.0 services satisfied outstandingly by the library.

Acknowledgments

This work has been developed with the financing of FEDER funds in FUZZYLING-II Project TIN2010-17876, the Andalusian Excellence Projects TIC-05299 and TIC-5991, and "Proyecto de Investigación del Plan de Promoción de la Investigación UNED 2011 (2011/PUNED/0003)".

References

1. Anfinnsena, S., Ghineaba, G., Cesareb, C.: Web 2.0 and folksonomies in a library context. International Journal of Information Management 31, 63–70 (2011)
2. Maness, J.M.: Library 2.0 theory: Web 2.0 and its implications for libraries. Webology 3, http://www.webology.ir/2006/v3n2/a25.html, (2006)
3. O'Reilly, T.: What is Web 2.0? Design patterns and business models for the next generation of software. O'Reilly Media, Inc., http://www.oreilly.de/artikel/web20.html, (2005)
4. Asemi, A., Kazempour, Z., Rizi, H.A.: Using LibQUAL+TM to improve services to libraries: A report on academic libraries of Iran experience. The Electronic Library 28, 568–579 (2010)
5. Heradio, R., Fernádez-Amorós, D., Cabrerizo, F.J., Herrera-Viedma, E.: A review on quality evaluation of digital libraries based on user's perceptions. Journal of Information Sciences 38, 269–283 (2012)
6. Zadeh, L.A.: Fuzzy sets. Information and Control 8, 338–353 (1965)
7. Herrera, F., Herrera-Viedma, E., Verdegay, J.L.: Direct approach processes in group decision making using linguistic OWA operators. Fuzzy Sets and Systems 79, 175–190 (1996)
8. Association of Research Libraries: LibQUAL+: charting library service quality. ARL, http://www.libqual.org
9. Parasuraman, A., Zeithaml, V.A., Berry, L.L.: A multiple-item scale for measuring consumer perceptions of service quality. Journal of Retailing 64, 12–40 (1988)
10. Zeithaml, V.A., Berry, L.L., Parasuraman, A.: Delivering Quality Services - Balancing Customer Perceptions and Expectations. The Free Press, New York (1990)
11. Herrera, F., Alonso, S., Chiclana, F., Herrera-Viedma, E.: Computing with words in decision making: foundations, trends and prospects. Fuzzy Optimization and Decision Making 8, 337–364 (2009)
12. Zadeh, L.A.: The concept of a linguistic variable and its applications to approximate reasoning. Part I. Part II. Part III. Information Sciences 8, 199–249. Information Sciences 8, 301–357. Information Sciences 9, 43–80, (1975)
13. Herrera, F., Herrera-Viedma, E.: Aggregation operators for linguistic weighted information. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 27, 646–656 (1997)
14. Yager, R.R.: On ordered weighted averaging aggregation operators in multicriteria decision making. IEEE Transactions on Systems, Man and Cybernetics 18, 183–190 (1988)
15. Kacprzyk, J.: Group decision making with a fuzzy linguistic majority. Fuzzy Sets Systems 18, 105–118 (1986)
16. Zadeh, L.A.: A computational approach to fuzzy quantifiers in natural languages. Computer & Mathematics with Applications 9, 149–184 (1983)
17. Linh, N.C.: A survey of the application of Web 2.0 in Australasian university libraries. Library Hi Tech 26, 630–653 (2008)

Sistema Borroso de Clasificación de Notas Clínicas de Texto Libre en base a Calidad de Datos

Francisco P. Romero[1], Mauro J. Ourezabal[2], Jose A. Olivas[1], Jesús Serrano-Guerrero[1], Ismael Caballero[1]

[1] Dept. Tecnologías y Sistemas de la Información, Univ. de Castilla La Mancha, Paseo de la Universidad, 4, 13071 Ciudad Real, España
{FranciscoP.Romero,JoseAngel.Olivas,Jesus.Serrano}@uclm.es
[2] Servicio de Oncología, Hospital Universitario Rey Juan Carlos, C/ Gladiolo s/n, 28933 Móstoles, Madrid, España
mauro.oruezabal@hospitalreyjuancarlos.es

Resumen En este trabajo se presenta un sistema de clasificación de notas clínicas basado en criterios de calidad de datos. El método propuesto parte del procesamiento del texto de la nota con el fin de extraer aquellos conceptos que más contribuyen a su significado. Con estos conceptos y su distribución en el texto se obtiene una calificación de la nota clínica según unos criterios de calidad predeterminados como la completitud, la corrección, la utilidad y la brevedad.

Keywords: Relaciones Borrosas, Calidad de Datos, Información Clínica

1. Introducción

Actualmente, en el mundo occidental, el grado de informatización de las instituciones sanitarias es muy alto. Esta situación ha supuesto la generación de grandes volúmenes de notas e informes médicos que se encuentran almacenados en formato electrónico. La digitalización de los contenidos hace que la cantidad de información a la que un profesional de la medicina puede acceder sea cada vez mayor. En muchas ocasiones, la información que estos profesionales buscan, no es el historial de un paciente o los resultados de unos análisis de laboratorio, sino que necesitan obtener información que les aporte conocimiento. Este conocimiento se obtiene a partir del contenido de las descripciones de los casos médicos que los propios profesionales elaboran a diario, y que suelen incluir observaciones sobre enfermedades, síntomas, reacciones, etc.

Las descripciones de los casos médicos pueden ser útiles para profesionales y estudiantes de Medicina, tanto en el fundamento de sus diagnósticos como en su labor de aprendizaje. Dado el gran volumen y el constante crecimiento de esta información, el grado de satisfacción de los usuarios respecto a la información a la que puede acceder y tiene que asimilar es muy pequeña. Una estrategia que se puede utilizar para paliar este problema es la aplicación de técnicas que permitan

clasificar la información en base a diferentes criterios (relevancia, temática, etc.) es por ellos que los métodos de clasificación de documentos de forma supervisada o no supervisada cobran especial importancia.

Las herramientas necesarias para construir un sistema clasificador de historias clínicas van desde la gestión de la calidad de datos de las fuentes de información hasta la clasificación de estas informaciones según diferentes criterios, pasando por algunos procesos intermedios como el tratamiento de abreviaturas o la representación conceptual de los documentos.

El resto del trabajo se organiza como sigue: en la sección 2 se describirán brevemente aquellos conceptos claves para el desarrollo de esta propuesta; en la Sección 3 se detallará el sistema propuesto, y en la Sección 4 se presentará las pruebas llevadas a cabo para verificar la viabilidad de la propuesta; para concluir en la sección 5 con las líneas de trabajo futuro.

2. Fundamentos

En esta sección se describen brevemente los dos conceptos clave en los que se basa esta propuesta: la calidad de datos y las relaciones borrosas.

2.1. Calidad de Datos

Los marcos de trabajo para la gestión de la calidad de los datos (en adelante DQ), proporcionan a los usuarios un conjunto de técnicas y herramientas fundamentales para evaluar la calidad de los contenidos de un texto. De esta manera, conociendo el nivel global de calidad de datos de un texto, un usuario o un sistema pueden decidir si utilizarlo o descartarlo. Para determinar si el texto es útil es necesario analizar si los datos contenidos para comprobar presentan un nivel adecuado de calidad estudiado con respecto a ciertas dimensiones de calidad de datos. Teniendo en cuenta que la "adecuación para el uso" depende de la tarea y del rol del agente o *stakeholder* que maneje las fuentes y los textos, es necesario identificar a partir de los requisitos de usuario, aquellas dimensiones que mejor representan las necesidades de calidad de datos para cada uno de los tipos de usuarios.

Un intento de desarrollar un conjunto de medidas para evaluar la calidad de la documentación clínica se puede ver en [1]. Como resultado de este trabajo se desarrolla un modelo de 22 dimensiones denominado "*22-item PDQI*" (Physician Documentation Quality Instrument) en el cual se incluyen dimensiones como la claridad, la completitud, la consistencia y la coherencia entre otras. A su vez se destaca que las principales dimensiones que influyen en la calidad de datos de un nota clínica son la comprensibilidad, la corrección, la compacidad y la organización. A este respecto, y con el fin de manejar un menor número de dimensiones en [2] se desarrolla el 9-item PDQI en el que se seleccionan los 9 items más relevantes del modelo anterior. Por otro lado, en [3] se identifican cinco dimensiones importantes como son la completitud, la corrección, la concordancia, la plausibilidad y la oportunidad, identificándose a su vez siete grandes categorías

de métodos de evaluación de calidad de datos sobre información clínica entre los que cabe destacar la comparación con normas de oro, la detección de elementos clave y la comparación entre notas y entre fuentes de datos.

2.2. Relaciones Borrosas

El concepto de relación borrosa es una generalización del concepto de relación de la teoría clásica de conjuntos. Mientras que una relación entre dos conjuntos clásicos describe la existencia o no de asociación entre los elementos de ambos conjuntos, una relación borrosa describe el grado de asociación o interacción entre los elementos de dos o más conjuntos borrosos.

Formalmente, una relación borrosa R entre dos variables x e y, cuyos dominios están en X e Y respectivamente, se define por una función que proyecta pares ordenados $X \star Y$ a su grado en la relación, la cual es un número entre 0 y 1, por ejemplo:

$R : X \star Y \leftarrow [0, 1]$

Si los valores posibles de x e y son discretos o pueden ser discretizados, se puede expresar una relación borrosa en forma de matriz, denominada matriz relacional borrosa, cuyos elementos toman valores en el intervalo $[0, 1]$.

Las relaciones borrosas se usan en sistemas de inferencia borrosa (*if* X *then* Y) de la siguiente forma: dada una relación borrosa bidimensional y los valores posibles de una variable, inferir los valores posibles de la otra variable. Para combinar las relaciones difusas se usan operaciones de composición como la composición max-min propuesta por Zadeh, la max-producto, o la suma-producto. Supongamos que es una relación binaria borrosa R en el universo $X \star Y$. A' y B' son conjuntos borrosos en XeY respectivamente. Si conocemos R y A' podríamos conocer B' mediante la denominada Regla composicional de inferencia:

$B' = A'(x) \circ R(x, y)$

$B'(y) = supx\epsilon X$

$min[A'(x), R(x, y)]$

donde $R(x, y) = I(A(x), B(y))$ siendo I una función de implicación.

3. Sistema Clasificador de Notas Clínicas

Este sistema clasificador de notas clínicas está basado en diferentes dimensiones de calidad de datos que deben ser calculadas tras la representación de cada una de las notas. Esta representación se va a basar en los conceptos que han sido tratados en los textos así como en la estructura que presentan.

3.1. Representación Conceptual de las Notas Clínicas

Para poder calcular la diferentes dimensiones de calidad de datos propuestas en este trabajo es necesario detectar las entidades conceptuales y la estructura

de la nota. Estas entidades conceptuales van a ser conceptos normalizados existentes en el MetaTesauro UMLS (Unified Medical Language System)[3]. Con el fin de detectar estos conceptos dentro del texto de la nota se utiliza la herramienta MetaMap [4] proporcionada por la National Library of Medicine (NLM). Esta herramienta detecta todas las variaciones terminológicas que pueden existir dentro de un texto a partir de las características morfo-sintácticas de los vocablos. MetaMap ofrece unos buenos resultados de exhaustividad pero es necesario incorporar otros procesos al tratamiento de los textos clínicos si se requiere obtener una precisión aceptable.

MetaMap nos va a permitir obtener de cada texto el identificador del concepto dentro de UMLS y su tipo semántico. El tipo semántico viene dado por la función semántica dentro del dominio biomédico que tiene el concepto en el contexto en el que aparece. Existe una amplia lista de tipos semánticos, concretamente 113, que incluyen todas las posibles funciones del concepto como diagnóstico, enfermedad, prueba diagnóstica, etc. En este trabajo, el tipo semántico cobra especial relevancia ya que permite la utilización de reglas que capturan el estilo y las propiedades lingüísticas que pueden dar lugar a la estructura de la nota clínica.

3.2. Estructura de las Notas Clínicas

Para valorar un texto médico hay muchos parámetros que hay que tener en cuenta. Uno de los más importantes que se va a considerar como criterio de calidad es que la nota siga una estructuración de una historia clínica orientada a problemas que denominada *SOAP* (Subjetivo, Objetivo, Evaluación y Plan)[5] que presenta los siguientes elementos relevantes en el proceso asistencial (explícitos o no):

- Subjetiva (S): Se consignan impresiones subjetivas tanto del médico como del enfermo. En esta zona de la nota debemos encontrar conceptos que son síntomas o signos (tipo Semántico *Sign or Symptom* en UMLS) y enfermedades previas del paciente (*Diseases or Syndrome*).
- Objetiva (O): Deben aparecer los datos del examen físico y constantes vitales. Se pueden encontrar las diferentes actividades diagnósticas que se hayan realizado (*Health Care Activity, Diagnostic Procedure*, etc.).
- Evaluación (A): Se deben encontrar afirmaciones sobre el problema encontrado (*Disease or Syndrome*, etc.).
- Plan (P): Se debe contar con la planificación de las conductas que se tomarán, como por ejemplo diagnósticos, terapias, seguimientos y actuaciones (*Therapeutic or Preventive Procedure*).

Esta estructuración va a permitir que se puedan calcular algunas de las cinco dimensiones de calidad de las que van utilizarse para calificar las notas en este sistema y que se explican a continuación.

[3] http://www.nlm.nih.gov/research/umls/

3.3. Dimensiones de Calidad de Datos

En este trabajo se han determinado algunas de las dimensiones indicadas en en [3] como son la completitud y la corrección (recomendadas a su vez en [6]), así como otras dimensiones seleccionadas en otros estudios como [1],[2] como la utilidad, la legibilidad y la brevedad. Todas ellas se detallan a continuación.

- **Completitud**: Un nota es completa cuando representa todas las características relevantes del proceso asistencial. Las métricas relacionadas con esta dimensión vienen dadas de controlar que todos los conceptos relevantes en el proceso asistencial se encuentran en alguna parte del texto. La proporción de presencia de estos elementos nos dará un grado de completitud posible.
- **Corrección**: Un texto es correcto desde el punto de vista semántico si cada elemento del proceso asistencial se representa utilizando estructuras adecuadas, es decir, si cada una de las partes contiene elementos que corresponden a esa parte. Por ejemplo, no sería lógico que en las impresiones subjetivas del profesional se mezclaran diagnósticos o acciones a realizar.
- **Utilidad**: La nota es útil siempre y cuando proporcione información relevante dentro de un contexto identificado con una consulta. En este caso se entiende como relevancia de una nota con respecto a una consulta en la capacidad del texto para responder a la necesidad de información expresada por la consulta. Siguiendo esta definición, se puede medir esta dimensión mediante la similitud conceptual entre la consulta y la nota.
- **Legibilidad**: La nota debe ser clara, sin ambigüedad y sin secciones que son díficiles de entender. Existen diferentes métodos estadísticos que permiten evaluar el grado de legibilidad de los textos que suelen basarse en la corrección de ortográfica y gramatical de éstos y el uso de vocabulario frecuente. Esto se realiza mediante la utilización de los resultados ofrecidos por el análisis del texto realizado mediante MetaMap.
- **Brevedad**: La nota es breve, concreta y sin redundancias. La brevedad se calcula a partir del número de conceptos tratados en el texto, teniendo en cuenta si existe repetición de éstos dentro de la propia nota clínica.

Los resultados de estas mediciones se ofrecerán en forma de intervalos de confianza, por ejemplo, si la nota presenta todas las partes de una nota SOAP, pero una de las partes, por ejemplo el plan, no se ve del todo completa, entonces el valor de la completitud vendrá dado por el intervalo [0,8 0,9].

3.4. Clasificador de Notas

El profesional comienza su indagación sobre el paciente siempre haciendo una pregunta sobre lo que quiere buscar, generalmente suele ser una pregunta sobre una determinada enfermedad (*"Infarto Agudo de Miocardio"*), o relacionada con un síntoma (*"Dolor torácico"*). Esta consulta va a servir para recuperar una serie de notas clínicas que van a ser evaluadas según las diferentes dimensiones de calidad de datos mencionadas anteriormente. Con el fin de realizar una manipulación personalizada de estas dimensiones se cuenta con una lista o vector sobre

el que se asignaran diferentes prioridades a las distintas medidas de calidad de datos obtenidas. Esto es así, dado que según el rol del profesional sus necesidades pueden ser muy distintas, por ejemplo, un médico consultará por enfermedades, síntomas, pruebas diagnósticas o tratamientos; mientras que a un investigador le puede interesar grupos relacionales de diagnóstico, la rareza de la enfermedad atendida,etc. y por último a un gestor le pueden interesar temas relacionados con la calidad y seguridad del paciente.

La metodología de clasificación de notas clínicas se basa en la definición de una relación borrosa ϕ entre notas y dimensiones de calidad de datos sobre la cual se va a aplicar la composición suma producto.

De esta manera, se tienen en cuenta dos conjuntos, por un lado, D que tiene por elementos los criterios de calidad de datos que nos permitirán calificar/evaluar a cada nota según cada una de las dimensiones explicada en el apartado anterior ($D = \{d_1, \ldots, d_5\}$) y por otro lado N que tiene como elementos cada una de las notas resultantes de la consulta del profesional ($N = \{n_1, n_2, \ldots, n_m\}$). Por ejemplo, para el tratamiento del problema representamos la relación borrosa ϕ que se muestra en la siguiente matriz, suponiendo que la consulta ha devuelto tres notas.

$$\begin{pmatrix} [0{,}8\ 0{,}9] & [0{,}6\ 0{,}7] & [0{,}7\ 0{,}8] & [0{,}6\ 0{,}7] & [0{,}9\ 1{,}0] \\ [0{,}9\ 1{,}0] & [0{,}3\ 0{,}4] & [0{,}7\ 0{,}8] & [0{,}5\ 0{,}6] & [0{,}8\ 0{,}9] \\ [0{,}9\ 1{,}0] & [0{,}8\ 0{,}9] & [0{,}6\ 0{,}8] & [0{,}9\ 1{,}0] & [0{,}9\ 0{,}9] \end{pmatrix}$$

En la matriz anterior se han evaluado mediante intervalos de confianza en $[0, 1]$ para cada nota n_j con $j = 1, 2, 3, 4, 5$ (filas), los criterios de calidad de datos d_i con $i = 1, 2, 3$ (columnas). Las valoraciones se han realizado de forma que el valor asignado es mayor cuanta mayor calidad de datos presenta la nota con respecto a ese criterio. De esta manera, se dispone de una evaluación nota-criterio cuyos valores han sido calculados de forma automática.

A partir de la relación borrosa planteada se puede construir una lista ordenada por preferencia de las notas. Para lo cual se realizan el siguiente proceso:

1. Se utiliza la lista de dimensiones de calidad de datos priorizada que exista para el rol del usuario. Esta lista va a ser manejada en forma de vector borroso V cuyos valores van a ser los pesos con los que se haya valorado a cada una de las dimensiones de calidad de datos pre-establecida:

   $$V = [w_1, w_2, w_3, w_4, w_5]$$

   por ejemplo:

   $$V = [0,7 \;\; 0,8 \;\; 0,6 \;\; 0,9 \;\; 0,7].$$

2. Este vector debe ser normalizado mediante la división de cada valor por la suma de todos los valores del vector), por lo que se obtiene el vector V^N cuya suma de todos sus valores es 1. Siguiendo el ejemplo propuesto:

   $$V^N = [0,19 \;\; 0,22 \;\; 0,16 \;\; 0,24 \;\; 0,19]$$

3. Finalmente, se calcula $\phi \star V^N$, siendo la composición suma-producto (operación $\star$) . En nuestro ejemplo:

$$\begin{pmatrix} [0{,}71\ 0{,}81] \\ [0{,}62\ 0{,}72] \\ [0{,}83\ 0{,}0,92] \end{pmatrix}$$

Con este resultado se puede llegar a calificar cada nota según su calidad de datos. En este caso la n_3 y n_1 tendrían una calidad alta y n_1 una calidad media. Además nos permite ofrecer al usuario la notas ordenadas de la siguiente forma: $n_3 > n_1 > n_2$.

4. Experimento

Con el fin de evaluar esta propuesta de calificación de notas clínicas de texto libre se ha utilizado como base el conjunto de notas clínicas procedentes del BLULab NLP Repository[4] de la Universidad de Pittsburgh que fueron facilitadas para la 2011 Text REtrieval Conference (TREC) Medical Records Track [5]. Esta base documental consta de más de cien mil documentos clínicos de diversos tipos y naturaleza.

Este experimento preliminar se ha llevado a cabo sobre un conjunto concreto de notas clínicas extraídas mediante una consulta de propósito general sobre esta colección de notas. En concreto el propósito de la consulta es extraer aquellas notas que correspondan a los cursos clínicos de urgencias por lo que la cadena de la consulta ha sido "*Emergency Clinical Course*". Este tipo de documentos no es muy común pero puede ser muy útil para los médicos residentes que tienen que realizar diferentes estancias en Urgencias y pueden así aprender como estructurar y redactar un documento de este tipo y con ello su proceso asistencial.

El resultado de esta consulta fueron 34 documentos clínicos los cuales fueron calificados por un experto y ordenados según su calidad de datos. Esta clasificación teórica es comparada con la ofrecida por el sistema lo cual nos va a medir la capacidad de la propuesta para reproducir el razonamiento experto. A su vez, con el fin de disponer de una línea base se ha utilizado Lucene [7] como herramienta de recuperación de información. En este caso se ha utilizado el *score* de Lucene y su división en tres niveles por densidad como calificador de las notas clinicas recuperadas.

Como medidas de rendimiento, se han calculado, siguiendo su definición estándar, los valores de precisión (P), exhaustividad (R) y métrica F (la media armónica entre precisión y exhaustividad) [8]. Para la evaluación del rendimiento medio para todas las categorías se utilizará el método basado en *micro-medias* (*micro-averaging*).

La propuesta logra un resultado aceptable en cuanto a la precisión (91 %), exhaustividad (84 %) y métrica F (87 %) de las clasificaciones efectuadas (Tabla 1). Comparativamente además se consiguen unos mejores resultados que las

[4] http://www.dbmi.pitt.edu/nlpfront
[5] http://trec.nist.gov/pubs/call2011.html/

herramientas clásicas de recuperación de información, lo cual implica que la utilización de criterios de calidad de datos es un procedimiento de ordenación más próximo al que podría hacer un experto, ya que los procedimientos clásicos únicamente se basan en un único aspecto la proximidad más o menos semántica entre la consulta realizada y el documento regresado.

Tabla 1. Resultados

	P	**R**	**F**
Propuesta	**91 %**	**84 %**	**87 %**
Lucene	74 %	62 %	67 %

5. Conclusiones

En este trabajo se ha propuesto un sistema que permite clasificar las notas clínicas de texto libre según criterios de calidad de datos. De esta manera se permite evaluar el grado de calidad de datos de las notas procesadas lo cual puede ser un buen indicador para medir la calidad asistencial.

Además, esta clasificación permite que las notas que tengan un alto grado de calidad puedan ser utilizadas en un futuro para la extracción de conocimiento mediante la aplicación de herramientas que intenten extraer conceptos, por ejemplo, agregaciones de conceptos, modo de pensamiento del médico, etc. y con ello ayudar a los expertos en la extracción de hipótesis que generarán conocimiento.

Referencias

1. Stetson, P.D., Morrison, F.P., Bakken, S., Johnson, S.B.: Research Paper: Preliminary Development of the Physician Documentation Quality Instrument. JAMIA **15**(4) (2008) 534–541
2. Stetson, P.D., Bakken, S., Wrenn, J.O., Siegler, E.L.: Assessing Electronic Note Quality Using the Physician Documentation Quality Instrument (PDQI-9). Applied Clinical Informatics **3**(2) (April 2012) 164–174
3. Weiskopf, N.G., Weng, C.: Methods and dimensions of electronic health record data quality assessment: enabling reuse for clinical research. JAMIA **20**(1) (2011) 144–151
4. Aronson A., Lang F.: An overview of MetaMap: historical perspective and recent advances. J Am Med Inform Assoc. 2010 May-Jun;17(3) : 229–3.
5. Cantale Carlos R.: Historia Clínica Orientada a Problemas El Generalista N° 1, 29–32, 2002.
6. Sharique H., Padman R.: Analyzing the Effect of Data Quality on the Accuracy of Clinical Decision Support Systems: A Computer Simulation Approach AMIA 2006 Symposium, 324–328, 2006.
7. Hatcher E., Gospodnetic O.: Lucene in Action. Manning Publications. 2005.
8. Van Rijsbergen C. : Information Retrieval. Butterworth, London, 1979.

Uso de OpenCyc y un Corpus de Ontologías para el Aprendizaje de Ontologías a partir de Mapas Conceptuales

Amhed Suárez-Rodríguez [1], Alfredo Simón-Cuevas [1], José A. Olivas[2]

[1] Facultad de Ingeniería Informática, Instituto Superior Politécnico José Antonio Echeverría, Cujae, Calle 114, No. 11901 e/ 119 y 127, Marianao, La Habana, Cuba
{asuarez, asimon}@ceis.cujae.edu.cu

[2] Departamento de Tecnologías y Sistemas de Información, Universidad de Castilla-La Mancha, Paseo de la Universidad 4, 13071-Ciudad Real, España.
JoseAngel.Olivas@uclm.es

Abstract. El trabajo presenta la extensión de un método de aprendizaje de ontologías a partir de una conceptualización representada en forma de mapa conceptual. El método es concebido en tres fases: pre-procesamiento, mapping semántico y codificación, y ha sido extendido con el uso de OpenCyc y un corpus de ontologías, como fuentes de conocimiento para el proceso de inferencia semántica en el mapa conceptual. La propuesta es evaluada experimentalmente con 54 conceptualizaciones bien formadas y de dominio medioambiental, con resultados prometedores para continuar profundizando en esta línea de trabajo.

Keywords: aprendizaje de ontologías, inferencia semántica, y mapas conceptuales

1 Introducción

En sistemas donde el conocimiento requiere ser semánticamente formalizado, para su procesamiento computacional, el uso de ontologías es una de las alternativas más prometedoras. Esto ha impulsado el desarrollo de metodologías y herramientas para la construcción de ontologías, así como lenguajes formales para su representación, dando lugar a la *ingeniería de ontología* [3]. Sin embargo, el diseño y construcción de ontologías, los editores disponibles, así como los lenguajes formales actuales, resultan aún complejos para personas no expertas en este tema. Esta complejidad se puede reducir usando un tipo de representación de conocimiento, más natural e intuitiva para las personas, e integrable con las ontología, en el sentido de que estas sean obtenidas automáticamente o semi-automáticamente. Los mapas conceptuales (MC) constituyen uno de esos tipos de representación de conocimiento.

Los MCs [11] son un recurso gráfico para organizar y representar el conocimiento más significativo de un tema, en lenguaje natural y de forma flexible e intuitiva para las personas. Están compuestos de conceptos y relaciones etiquetadas que forman

proposiciones, y son esencialmente útiles para construir una conceptualización. Estas bondades han motivado a su uso en el proceso de construcción de ontologías, por ejemplo, en la representación informal de la conceptualización de la ontología a construir [5],[9],[13] y en métodos para su construcción automática en OWL [1],[8],[4],[6],[14].

2 Mapas Conceptuales en la Ingeniería de Ontologías

Los MCs son una herramienta gráfica para organizar y representar conocimiento en lenguaje natural, y muy intuitivos para las personas. Se componen de conceptos y relaciones etiquetadas con frases-enlace que forman proposiciones, las cuales son consideradas unidades semánticas o de significado [11]. Generalmente, tienen una estructura jerárquica y expresan el entendimiento de lo más significativo de un dominio de conocimiento determinado, aunque pueden integrar varios dominios. Su utilidad en la ingeniería de ontologías ha sido reconocida por varios autores, fundamentalmente, para la representación informal de la conceptualización de la ontología [5],[9],[13], y en métodos para su construcción automática [1],[8],[4],[6],[14].

La integración entre los MCs y las ontologías se sustenta en correspondencias identificadas entre sus componentes [7],[13], a pesar de la falta de formalización semántica que caracteriza a los primeros. Muchos de los métodos definidos para este propósito muestran limitaciones, en cuanto a la capacidad para la inferencia semántica sobre el MC, la reducción de flexibilidad en la construcción de los MC a partir de la incorporación de formalismos, y el no tratamiento de la ambigüedad presente en los conceptos del MC. Se reportan mejoras a estos aspectos en [14], aunque aún se mantiene como una necesidad incrementar la capacidad del mecanismo de inferencia semántica en los MCs para obtener ontologías más completas, y al mismo tiempo, conservando las bondades que aporte su flexibilidad. En correspondencia, en este trabajo se propone incrementar estas capacidades dotando al método de construcción de ontologías de otros recursos de conocimiento ya consolidados, demostrándose experimentalmente esta contribución. Concretamente, se ha extendido el método reportado en [14], con la ontología superior OpenCyc y un mecanismo para el tratamiento de un corpus de ontologías. La propuesta está inspirada en la utilidad, y creciente tendencia, de reutilizar conocimiento existente, consolidado y formalizado, en función de la creación de nuevos conocimientos.

3 Recursos de Conocimiento para la Inferencia Semántica

El conocimiento en un MC no está formalizado, por tanto, su semántica está implícita, ya que estos han sido definidos para ser usados e interpretados por personas y no por sistemas computacionales. El uso de recursos de conocimiento, en particular WordNet [10], ha sido considerado con anterioridad para inferir la semántica asociada a conceptos y relaciones en propuestas de aprendizaje de ontologías con MCs [1][13].

WordNet se compone de una gran red semántica de *synset*, el cual representa sentidos de palabras, estando presente una palabra en más de un *synset* si es polisémica.

Los *synset* se conectan mediante varios tipos de relaciones semánticas [10], tales como: *hiponimia*/*hiperonimia* y *meronimia*/*holonimia*. En [1][13] este tipo de relaciones se emplean para inferir si un concepto constituye una clase, subclase o una propiedad. WordNet ha demostrado ser útil en este proceso de inferencia, pero resulta ser insuficiente ya que solo aporta resultados satisfactorios para conceptualizaciones de dominio general y cuyos conceptos se formen por términos simples, constituyendo una limitante. Es por ello que en este trabajo se propone OpenCyc, como complemento de WordNet para entornos de dominio general, y un corpus de ontologías.

OpenCyc es la versión libre de la ontología superior Cyc, que comprende una taxonomía de conceptos, relaciones, propiedades, restricciones e instancias codificadas en lenguaje CycL. En OpenCyc se describen objetos y eventos de la vida diaria, y útiles en diversas aplicaciones, tales como la desambiguación semántica, en el mapeo de ontologías, en la integración semántica de metadatos, pero no en el aprendizaje de ontologías con mapas conceptuales. En este sentido, OpenCyc es útil en la inferencia de clases, subclases e instancias a partir de la identificación de conceptos relacionados mediante predicados del tipo:

isa: Predicado para indicar que una instancia es miembro del conjunto C (#$isa I C), equivalente a I INSTANCE_OF C.

genls: Predicado para la representación de relaciones de clase/subclase. (#$genls C' C), significa que C' es una subclase de C.

A través de estos tipos de predicado se encuentra representada la mayor parte de la información de esta ontología general.

El empleo de un corpus de ontologías tiene el propósito de inferir información semántica para identificar relaciones del tipo clase/subclase, clase/instancia, clase/propiedad, entre otras, fundamentalmente en entornos de dominio específico. De esta forma, se trata de suplir la necesidad de información semántica de referencia que puede ocurrir si solo se usa WordNet y OpenCyc. El corpus podría estar formado por una o varias ontologías OWL, preferiblemente de dominio específico. Aunque existan varias ontologías, no todas aportan la misma información, y hasta se podría llegar a conflictos semánticos. Una forma de reducir esto es incluir en el procesamiento del corpus un mecanismo de selección de aquellas ontologías más relevantes para el MC que se está procesando. De esta forma, también se gana en eficiencia, en cuanto al tratamiento del conocimiento almacenado en el corpus.

La selección de ontologías relevantes para un MC determinado requiere de la comparación entre el conocimiento formalizado en primero y el representado en el segundo, sin embargo, esto no es posible directamente. Una solución es llevar el conocimiento de ambas representaciones a un formato común. En este sentido, se aplica el método reportado en [10] para extraer la conceptualización de cada ontología del corpus en forma de MC. Luego se procede al análisis de relevancia, el cual está basado en la comparación entre cada una de las conceptualizaciones obtenidas de las ontologías y el MC que se procesa. La comparación se realiza de forma automática usando el algoritmo de comparación de MCs [10]. Este algoritmo posibilita obtener un grado de similitud entre dos MCs, usando pocos recursos computacionales para su ejecución, y es uno de los más usados para estos fines. El proceso finaliza con la selección

de las ontologías de referencia del corpus como bases de conocimiento para el proceso de inferencia semántica. Son seleccionadas ontologías de referencia aquellas cuyas conceptualizaciones estén dentro del 25 % de las que más similares al MC. En trabajos anteriores [1],[8],[4],[6],[14], no se reporta el uso de un corpus de ontologías en el proceso de inferencia semántica en MC para la construcción de ontologías.

4 Método de Aprendizaje de Ontología Extendido

El método ha sido concebido como un proceso estructurado en tres fases: *pre-procesamiento*, *mapping semántico* y [14], según se muestra en la Figura 1, y cuatro componentes se desarrollaron para ejecutar los diferentes procesamientos: *parser*, *desambiguador*, *intérprete semántico* y un *codificador-OWL* [14]. OpenCyc y el corpus de ontologías son incluidos para extender los recursos de conocimiento usados por el método, y son tratados en la fase de *pre-procesamiento*.

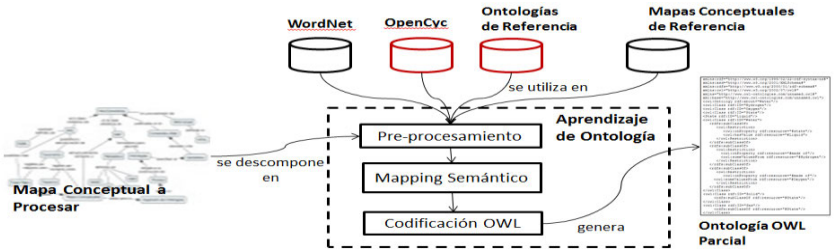


Fig. 1. Fases y recursos usados por el método de aprendizaje de ontologías.

En el pre-procesamiento el parser extrae todas las proposiciones del MC y el desambiguador determina el sentido de cada concepto usando WordNet y el algoritmo reportado en [12]. Luego, se recupera información presente en los recursos de conocimiento externo, relacionada con los conceptos y proposiciones del MC, y que es relevante para el proceso de inferencia que lleva a cabo el intérprete semántico. La información recuperada es almacenada en varios conjuntos, a saber:

- Un conjunto $C_{clase/subclase}$ creado con:
  - pares de conceptos *(C, C')*, si en WordNet, entre el *synset* de *C'* y *C* existe algún camino a través de relaciones de *hiperonimia*, y los pares *(C', C)*, si el camino es de relaciones de *hiponimia* entre el *synset* de *C'* y el de *C*;
  - pares de conceptos *(C, C')*, si en OpenCyc, está presente el predicado *(#$ genls C' C)*, o se puede transitar por un camino, a través de predicados *genls*, entre *(#$ genls C' Z)* y *(#$ genls Z C)*, siendo *Z* el conjunto de conceptos que relaciona a *C* y *C'*.
  - los pares de conceptos *(C, C')*, si en una de las ontologías de referencia, *C* constituye una *owl: class* y en *C'* se especifica *< rdf:subClassOf ... "C"/>*.
- Un conjunto $C_{clase/instancia}$ creado con:

  - pares de conceptos *(C, C')*, si OpenCyc incluye el predicado *(#$ isa C' C)*, o si en una de las ontologías de referencia se codifica *C'* como *<C rdf:ID="C"/>*.
- Un conjunto $C_{mero/holo}$ creado con:
  - pares de conceptos *(C, C')*, si en WordNet, el *synset C* está conectado directamente mediante una relación de *meronimia* con el de *C'*, o si la conexión es inversa a través de una relación de *holonimia*.
- Un conjunto $C_{mero/holo\text{-}tipo}$ creado con:
  - proposiciones *(C, C', tipo-relación)*, si en WordNet, el *synset C* está conectado directamente mediante una relación de *meronimia no básica* con el de *C'*; y *(C', C, tipo-relación)*, la relación es inversa y se establece a través de tipos de relación de *holonimia*. En *tipo-relación* se almacena el tipo de relación identificada.
- Un conjunto $C_{prop/objeto}$ creado con:
  - proposiciones *(C, Pr, C')*, si en una de las ontologías de referencia, *Pr* se especifica como *owl:ObjetctProperty*, donde *C* constituye su dominio y *C'* su rango.
- Un conjunto $C_{rest/objeto}$ creado con:
  - proposiciones *(C, Pr, C')*, si en una de las ontologías de referencia, *C* se codifica como *owl:class*, y *Pr* como una propiedad de este, con la restricción de tener a C' como valor particular.

En el *mapping semántico*, el *intérprete semántico* se encarga de inferir la semántica implícita en la estructura proposicional del MC, según las especificaciones OWL [15]. El intérprete se ha diseñado como una máquina de inferencia basada en reglas, con razonamiento hacia adelante, donde se combina: la semántica de frases-enlaces predefinidas, la topología del MC, así como la información recuperada de WordNet, OpenCyc y las Ontologías de Referencia. Las reglas tienen asociado un *valor de relevancia (VR)* para el control de su ejecución, y se formalizan como sigue [14]:

- Reglas para inferir clases simples y relaciones entre clases (almacenan el resultado en el conjunto de pares de conceptos *C_CS*):
  R1 (VR = 8):
  a. Si $P = (C_o, f\text{-}e, C_d) \in PP\text{-}MC \wedge f\text{-}e \in CC$ *Entonces* $PP\text{-}MC = PP\text{-}MC - \{P\}$, $C_CS = C_CS \cup \{(C_o, C_d)\}$.
  b. Si $P = (C_o, f\text{-}e, C_d) \in PP\text{-}MC \wedge f\text{-}e \in CC^{-1}$ *Entonces* $PP\text{-}MC = PP\text{-}MC - \{P\}$, $C_CS = C_CS \cup \{(C_d, C_o)\}$.
  R2 (VR = 8): Si $P = (C_o, f\text{-}e, Cd) \in PP\text{-}MC \wedge f\text{-}e \notin CC \wedge f\text{-}e \notin CI \wedge [((C_o, C_d) \in C_{clase/subclase})]$ *Entonces* $PP\text{-}MC = PP\text{-}MC - \{P\}$, $C_CS = C_CS \cup \{(C_o, C_d)\}$, $CC = CC \cup \{f\text{-}e\}$.
- Reglas para inferir clases complejas (almacenan el resultado en el conjunto *Unión*):
  R3 (VR = 6): Si $U = (CU, \{C_i \mid (CU, C_i) \in C_CS\}) \wedge U \notin$ *Unión Entonces Unión = Unión* $\cup \{U\}$.
  R4 (VR = 6): Si $I_C = (CI, \{C_i \mid (C_i, CI) \in C_CS\}) \wedge I_C \notin Intersección_{clases}$ *Entonces* $Intersección_{clases} = Intersección_{clases} \cup \{I_C\}$.
  R5 (VR = 2): Si $I_{C\text{-}Pr} = (CI, C \mid ((C, CI) \in C_CS, (C', CI) \notin C_CS), Pr \mid (CI, Pr, V) \in C_CPV_{tieneValor}, V) \wedge I_{C\text{-}Pr} \notin Intersección_{clase\text{-}propiedad}$ *Entonces* $Intersección_{clase\text{-}propiedad} = Intersección_{clase\text{-}propiedad} \cup \{I_{C\text{-}Pr}\}$.
- Regla para inferir instancias (almacenan el resultado en el conjunto *C_CI*):

R6 (VR = 8):
a. Si P = (C_o, f-e, C_d)∈ PP-MC ∧ f-e ∈ CI ∧ (C_o, C_d) ∉ $C_{clase/subclase}$ Entonces PPMC = PP-MC – {P}, C_CI = C_CI ∪ {(C_o, C_d)}.
b. Si P = (C_o, f-e, C_d) ∈ PP-MC ∧ f-e ∉ CI ∧ [(C_o, C_d) ∈ $C_{clase/Instancia}$] Entonces: PP-MC = PP-MC – {P}, C_CI = C_CI ∪ {(C_o, C_d)}.

- Reglas para inferir propiedades simples (almacenan sus resultados en los conjuntos *C_CP*, *C_CPV*):
  R7 (VR = 8): Si P = (C_o, f-e, C_d) ∈ PP-MC ∧ (f-e ∈ CP ∨ (C_o, C_d) ∈ $C_{mero/holo}$) v (P ∈ $C_{prop/objeto}$) Entonces PP-MC = PP-MC – {P}, C_CP = C_CP ∪ {(C_o, f-e, C_d)}, CP = CP ∪ {f-e}.
  R8 (VR = 8):
  a. Si P = (C_o, f-e, C_d)∈ PP-MC ∧ f-e ∈ CPV Entonces PP-MC = PP-MC – {P}, C_CPV = C_CPV ∪ {(C_o, f-e, C_d)};
  b. Si P = (C_o, f-e, C_d)∈ PP-MC ∧ (C_o, C_d, tipo-relación)∈ $C_{mero/holo\text{-}tipo}$ Entonces PPMC = PP-MC – {P}, C_CPV = C_CPV ∪ {(C_o, f-e, C_d)}, CPV = CPV ∪ {f-e};
  c. Si P = (C_o, f-e, C_d)∈ PP-MC ∧ (P ∈ $C_{restr/prop}$) Entonces PP-MC = PP-MC – {P}, C_CPV = C_CPV ∪ {(Co, f-e, Cd)}.
- Regla para inferir restricciones de propiedades (almacenan el resultado en el conjunto $C_CPV_{tieneValor}$):
  R9 (VR = 6): Si (C, Pr, V) ∈ C_CPV ∧ ∃(C', I)∈ C_CI ∧ V = I Entonces $C_CPV_{tieneValor}$ = $C_CPV_{tieneValor}$ ∪ {(C, Pr, V)}.
  Reglas para inferir características de propiedades(almacenan el resultado en el conjunto $Pr_{funcional}$ y $Pr_{simétrica}$):
  R10 (VR = 2): Si (C, Pr, V)∈ $C_CPV_{tieneValor}$ ∧ (∀C ∀ $_{(C, f\text{-}e, C')∈PP\text{-}BC}$[(f-e = Pr ∨ Sinónimo(f-e,Pr))→(C'=V ∨ Sinónimo(C',V))]) Entonces $Pr_{funcional}$=$Pr_{funcional}$ ∪ {Pr}.
  R11 (VR = 2): Si (C, Pr, V) ∈ C_CPV ∧ (∀V ∀ $_{(V, f\text{-}e, C')∈PP\text{-}BC}$[(f-e = Pr ∨ Sinónimo (f-e, Pr)) → (C' = C ∨ Sinónimo (C', C)]) Entonces $Pr_{simétrica}$ = $Pr_{simétrica}$ ∪ {Pr}.

siendo *P:* una proposición; *PP-MC:* conjunto de proposiciones del MC; C_o*:* concepto origen de la proposición; C_d*:* concepto destino de la proposición, *f-e:* frase-enlace; *CC:* f-e predefinidas que expresan relaciones taxonómicas, y CC^{-1} sería su inversa; *CI:* f-e que expresan relación entre una clase y su instancia; *CP:* f-e que expresan relación entre un concepto clase y otro que sería su propiedad; y *CPV:* f-e que expresa el identificador de una propiedad; y *Sinónimo(,)*: función que identifica si dos términos son sinónimos (según WordNet);

En la fase de codificación el codificador-OWL establece la correspondencia entre la semántica inferida y almacenada en diferentes conjuntos, con las especificaciones del lenguaje OWL DL [19], a partir de una convención de codificación definida. Al concluir la codificación, se obtiene una ontología OWL parcial, que formaliza semánticamente todo o parte del conocimiento representado en el MC. La ontología parcial resultante puede ser refinada y completada a través del editor de ontologías Protégé.

5 Resultados Experimentales y Conclusiones

La nueva propuesta de método fue aplicado sobre 54 MCs, como muestra de experimentación (superior a la considerada en [14]) para demostrar los aportes de OpenCyc y el corpus de ontología para el aprendizaje de ontologías desde MCs. Los MCs se tomaron, fundamentalmente, de artículos científicos ya publicados, en sustitución de una colección de referencia que no está disponible. Estaban presente 23 conceptos y 26 proposiciones en cada MC, como promedio, y todas las conceptualizaciones eran de dominio medioambiental, para facilitar los análisis comparativos con propuestas anteriores. Cada MC constituyó una conceptualización independiente, y a partir de cada uno se construyó una ontología parcial en OWL.

Se evaluó la calidad y cantidad de información semántica inferida en los MCs, a través del conocimiento representado en ellos y que fue semánticamente descrito en las ontologías resultantes. El segundo elemento constituyó el objetivo fundamental de análisis para validar la propuesta, ya que la misma va dirigida a incrementar dicho aspecto, al dotar al método de conocimiento adicional. Se utiliza la *cobertura* [14] como métrica para el cómputo de los resultados. La cobertura mide cuánto conocimiento, en cantidad de proposiciones, se logra formalizar en la ontología OWL, y se formula mediante la razón entre la cantidad de proposiciones codificadas y las que se debían codificar. Una proposición se considera codificada si sus conceptos, y la relación entre ellos son descritos semánticamente en la ontología.

Se realizaron dos experimentos, cada uno de los cuales incluyó análisis comparativo con el antecedente de la nueva propuesta reportado en [14]:

1. se evalúo la cobertura del método usando OpenCyc y el corpus de ontologías por separado, para demostrar sus aportes de forma independiente.
2. se evalúo la cobertura del método combinando todos los recursos de conocimiento, y al extenderlo en cuanto a la inferencia de clases.

En el primer experimento se utilizó el corpus del proyecto SWEET[1] (siglas del inglés, Semantic Web for Earth and Environmental Terminology), con alrededor de 100 ontologías OWL sobre dominio medioambiental y uno de los más reconocidos en este ámbito [9]. La Fig. 2 muestra el comportamiento de la cobertura del método, usando solo OpenCyc, respecto a su versión anterior [14] con la misma muestra, de la misma manera se muestra en la Fig. 3, pero usando en este caso el corpus SWEET.

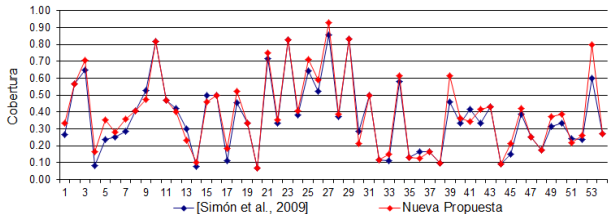


Fig. 2. Resultados comparativos de *cobertura* usando solo OpenCyc.

[1] http://sweet.jpl.nasa.gov/index.html

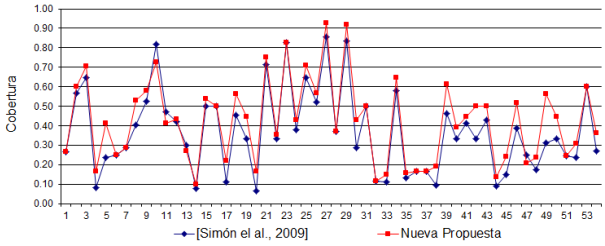


Fig. 3. Resultados comparativos de *cobertura* usando solo el corpus de ontologías SWEET.

Con el uso solo de OpenCyc se alcanza una cobertura del 40 % (tres puntos porcentuales superior a [14]), donde el 12 % de la inferencia se produjo a partir de este recurso, demostrando así sus beneficios. Con el corpus SWEET se alcanza un 45 % de cobertura (ocho puntos porcentuales superior a [14]), y alrededor de un 19 % de la semántica total inferida es resultante del conocimiento formalizado en sus ontologías, evidenciando también la validez de la propuesta. En el uso del corpus de ontología, también es estudiado el comportamiento de la cobertura respeto al nivel de vínculo temático entre cada MC y las ontologías de referencia, dado que estas son seleccionadas automáticamente para cada MC y por tanto pueden ser diferentes. Para medir la relevancia del conjunto ontologías recuperadas del corpus, respecto a un MC, se definió la métrica *grado de relevancia* (GR), cuya formulación se muestra a continuación:

$$GR = \frac{\sum_{i=0}^{n} S(MC_B, O_i)}{n} \quad \textbf{(1)}$$

donde: $S(MC_B, O_i)$ es el valor de similitud entre el MC a procesar (MC_B) y el obtenido de cada ontología i recuperada (O_i) según el algoritmo de comparación de MC reportado en [2], y n la cantidad de ontologías recuperadas. En la Fig. 5 se muestra como tendencia que, mientras más correspondencia existe entre el dominio de la conceptualización del MC y el de las ontologías de referencia, mayor es la cobertura del método y por tanto mayor capacidad de inferencia semántica en un proceso de aprendizaje de ontología a partir de MC.

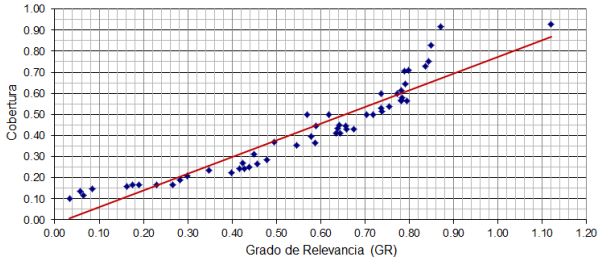


Fig. 4. Resultado de la cobertura respecto al grado de relevancia de las ontologías de referencia.

Se ha combinado WordNet con OpenCyc y con las ontologías de referencia, no obstante, no se han apreciado contradicciones en la semántica inferida. La propuesta

de selección de ontologías de referencia dentro del corpus, a partir de una medida de cercanía temática y estructural con el MC, contribuye a reducir las posibles contradicciones o ambigüedades sobre conceptos que puedan surgir en el proceso de inferencia.

Finalmente, se evaluó la *cobertura* alcanzable por el método extendido, tanto con OpenCyc, como con el corpus SWEET, y los resultados se muestran en la Fig. 5. La *cobertura* alcanza el 47 %, siendo 14 % superior a la obtenida por [14] sobre la misma muestra.

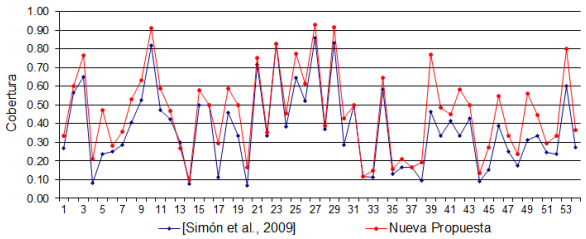


Fig. 5. Resultado de la cobertura entre la nueva propuesta y su versión anterior [14].

De esta forma se demuestran los beneficios que aportan OpenCyc y un corpus de ontologías, soportado por un mecanismo de selección de ontologías relevantes, en la inferencia de la semántica implícita en un MC para la construcción automática de ontologías. Los resultados resultan prometedores, aunque no fueron superiores, debido, fundamentalmente, a que 55% de las proposiciones analizadas tenían al menos un concepto no incluido en ninguno de los recursos de conocimiento, y tampoco resultaron útiles sus frases-enlace, y no se pudo inferir información semántica de sus conceptos. En este sentido, se consideró incluir una regla en el intérprete semántico que identifique como *clase*, todos los conceptos formalizados como *clases* en alguna de las ontologías de referencia, y están incluidos en proposiciones no codificadas. La Fig. 6 muestra los resultados de esta nueva propuesta de extensión, con la que se alcanza un 52 % de *cobertura* y incrementa en un 21 % respecto a [14].

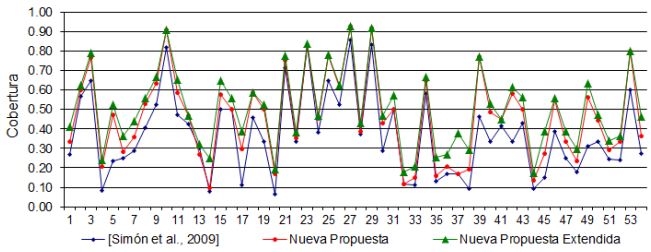


Fig. 6. Resultado de la cobertura de la nueva propuesta extendida respecto las anteriores [14].

Los resultados experimentales demuestran que la incorporación de otros recursos de conocimiento, adicionales a WordNet, contribuye significativamente al incremento de las capacidades de los métodos para el aprendizaje de ontologías desde MCs.

Referencias

1. Brilhante V., Macedo G., y Macedo S.: Heuristic Transformation of Well-Constructed Conceptual Maps into OWL Preliminary Domain Ontologies, Proc. of 2nd Workshop on Ontologies and their Applications, Brazil, 2006.
2. Cañas A., J., Leake D., y Maguitman A.: Combining Concept Mapping with CBR: Towards Experience-B ased Support for Knowledge Modeling. Proc. of FLAIRS Conference 2001, AAAI, 2001, 286-290.
3. Corcho O., Fernández M., y Gómez A., Ontological Engineering: What are Ontologies and How Can We Build Them?, Semantic Web Services: Theory, Tools and Applications, J. Cardoso Ed., IGI Global, 2007, 44-70.
4. García A., Norena A., Betancourt A., García L., y Sequeda, J. F.: CMAPS supporting the development of OWL ontologies. Proc. of Int. Semantic Web Conf., 2008.
5. Giouvanakis T., Fragidis G., Kehris E., y Samaras H.: Exploiting Concept Mapping in a Semantic Web Environment, Proc. of the Fifth IEEE Int. Conf. on Advanced Learning Technologies, IEEE Computer Society Press, 2005, 432-434.
6. Gómez H., Díaz B., y González A.: Two Layered Approach to Knowledge Representation Using Conceptual Maps Description Logic, Proc. of 1st Int. Conf. on Concept Mapping, Pamplona, Spain, 2004.
7. Graudina V., y Grundspenkis J.,: Concept Map Generation from OWL Ontologies, Proc of 3rd Int. Conf. on Concept Mapping, Estonia and Finland, 2008, 173-180.
8. Hayes P. J., Eskridge T. C., Saavedra R., Reichherzer T., Mehrotra M., y Bobrovnikoff D.: Collaborative Knowledge Capture in Ontologies, Proc. of 3rd Int. Conf. on Knowledge Capture, Canadá, 2005, 99-106.
9. McGuinness D., Fox P., Cinquini L., West P., Benedict J., y García J.: Current and future uses of OWL for Earth and Space science data frameworks: successes and limitations, AGU Fall Meeting 2007, 2007.
10. Miller G., Beckwidth R., Fellbaum C., Gross D., y Miller K. J.: Introduction to WordNet: An On-line Lexical Database, Int. Journal of Lexicography, 3(4), 1993, 235-244.
11. Novak, J. D., y Gowin, D. B.: Learning how to learn, Cambridge University Press, 1984.
12. Simón-Cuevas A., Ceccaroni L., Rosete-Suárez A., y de la Iglesia M.: A Concept Sense Disambiguation Algorithm for Concept Maps, Proc. of 3rd Int. Conf. on Concept Mapping, Estonia y Finland, 2008, 14-21.
13. Simón-Cuevas A., Ceccaroni L., Rosete-Suárez A., Suárez-Rodríguez A., y Victoria-Oquendo R.: A Support to Formalize a Conceptualization from a Concept Map Repository, Proc. of 3rd Int. Conf. on Concept Mapping, Estonia y Finland, 2008, 68-75.
14. Simón-Cuevas A., Ceccaroni L., Rosete-Suárez A., y Suárez-Rodríguez A.: Formal Modeling Method applied to Environmental-Knowledge Engineering, CISIS 2009, IEEE CS Press, 2009, 1132-1137.
15. Smith M. K., Welty Ch., y McGuinness D.: OWL Web Ontology Language Guide, W3C, 2004.

Una Aplicación de los Prototipos Deformables Borrosos para predecir el rendimiento académico de nuevos estudiantes a partir de la información generada por Sistemas Tutores

Mª Rosario Vázquez[1], Francisco P. Romero[2], Jose A. Olivas[2]

[1] Dependencia Ciencias de la Información, Univ. Autónoma del Carmen,
Av. 56 No. 4, 24180 Ciudad del Carmen, Campeche, México
mvazquez@pampano.unacar.mx
[2] Dept. Tecnologías y Sistemas de la Información, Univ. de Castilla La Mancha,
Paseo de la Universidad, 4, 13071 Ciudad Real, España
{FranciscoP.Romero,JoseAngel.Olivas}@uclm.es

Resumen En este trabajo se presenta una aplicación de los *Prototipos Deformables Borrosos*, con el fin de obtener un modelo cognitivo que permita predecir el futuro rendimiento académico de estudiantes que hacen uso de sistemas tutores para resolver problemas algebraicos. Se utilizan los Prototipos Deformables Borrosos para representar y manipular los diferentes tipos de estudiantes, con el fin de obtener patrones de comportamiento académico que permitan determinar las futuras actuaciones de nuevos estudiantes en base a su similitud con los patrones representados. Sin embargo, debido a la imprecisión inherente a la información manipulada, es posible que exista solapamiento entre las representaciones borrosas de los prototipos, por lo que se hace uso de conjuntos borrosos de tipo 2 (Interval Type 2 Fuzzy Sets).

Keywords: Prototipos Deformables Borrosos, Sistemas Tutores, Conjuntos intervalares borrosos de tipo 2

1. Introducción

La Inteligencia Artificial se ha aplicado con éxito en la educación tal es el caso de los Sistemas tutores, que son sistemas de software que apoyan al aprendizaje. En la actualidad, estos sistemas se adaptan al progreso de cada estudiante descomponiendo una tarea dada, en una serie de actividades con el fin de ayudar al estudiante a llegar a la solución de la misma. Actualmente existe una gran variedad de sistemas tutores enfocados a diversos cursos académicos tales como: Álgebra y Mysql, etc.

Por lo general, los sistemas tutores generan un archivo de registro (bitácora) donde se almacena cada una de las acciones realizadas por sus usuarios (estudiantes). Habitualmente este archivo contiene un gran volúmen de datos por lo que resulta un desafío su análisis. Sin embargo, se ha despertado un gran interés

en analizar estas grandes colecciones de datos para predecir el futuro rendimiento académico del estudiante con el fin de planear estrategias que contribuyan a incrementar su nivel académico [1]. Una muestra de este interés se ve reflejado en el desafío correspondiente al año 2010 del concurso anual KDD Cup [3], que consistió en predecir si un estudiante respondería correctamente al primer intento algún punto específico de un problema determinado. Para ello, el concurso proporcionó archivos de registro generados por 2 sistemas tutores.

En este trabajo se pretende obtener a partir de estos archivos de registro, un modelo cognitivo que permita predecir el futuro rendimiento académico de los estudiantes. El resto del trabajo se organiza como sigue: en la sección 2 se obtiene el modelo de predicción a través del descubrimiento de comportamientos prototípicos de estudiantes que se representan a través de conjuntos borrosos y es donde interviene la propuesta de este trabajo. En la sección 3 se explica la evolución del estudiante conforme avanza en el curso académico. Finalmente en la sección 4 se presentan las conclusiones.

2. Análisis de datos generados por sistemas tutores

En este trabajo se toma como referencia la teoría de *Prototipos Deformables Borrosos* [2], que incluye el uso de un proceso denominado *Descubrimiento de Conocimiento Prototípico Borroso (DCPB)*. Con el DCPB se obtienen prototipos que describen comportamientos prototípicos. Posteriormente estos prototipos se representan formalmente a través de números borrosos triangulares y/o trapezoidales, que permitirán obtener los grados de afinidad de una nueva situación con los mismos. Finalmente, los prototipos que se obtienen en el proceso DCPB se deforman de acuerdo al grado de afinidad de los prototipos borrosos con la nueva situación. Esta deformación se realiza con el fin de describir la situación real.

2.1. Descubrimiento de Conocimiento

El proceso *Descubrimiento de Conocimiento Prototípico Borroso* (DCPB) [2] se apoya en el KDD (Knowledge Discovery in Databases), con el fin de obtener patrones de comportamiento que permitan estimar el futuro comportamiento de nuevas situaciones con base a su similitud con los patrones obtenidos. A continuación se describen brevemente cada una de las fases del DCPB aplicadas al proceso de predicción del rendimiento académico de estudiantes, a partir de datos generados por sistemas tutores para el aprendizaje de Álgebra.

Colección de datos Se utilizó la colección de datos *algebra_2005_2006_train* proporcionada por el concurso anual KDD Cup 2010. Esta colección fue generada por un sistema tutor para la enseñanza de Álgebra llamado *Carnegie Learning*

[3] https://pslcdatashop.web.cmu.edu/KDDCup/

Algebra System [4], que contiene 809694 registros que pertenecen a 574 estudiantes de diferentes instituciones educativas que utilizaron el sistema tutor durante el período escolar 2005-2006. El sistema tutor proporciona una serie de tareas (ejercicios, problemas) que el estudiante debe resolver. Estas tareas se descomponen en un conjunto de actividades llamadas pasos, y están estructuradas en unidades. De esta manera, cada registro de la colección de datos pertenece a una actividad (paso) de un problema.

Selección y Preproceso La colección contiene 19 características tales como: el identificador del estudiante, el nombre del problema resuelto, la unidad a la que pertenece el mismo, los nombres de los pasos en que se descompuso el problema, el tiempo empleado en resolver un paso, las ayudas adicionales (pistas) que el estudiante recibió del sistema entre otros.

Debido a la multidimensionalidad del conjunto de datos se seleccionó un subconjunto de características que influyen en el rendimiento académico del estudiante tales como: el tiempo empleado para resolver un paso, la respuesta del paso (1 correcto, 0 incorrecto), el número de intentos realizados para resolver el paso, así como las pistas solicitadas al sistema tutor. Los atributos identificador del estudiante, la unidad, el problema y el nombre del paso permiten identificar los registros de cada estudiante por lo que se incluyeron en el subconjunto de características seleccionadas.

En la etapa de preproceso se trataron los datos con valores vacíos ó valores desconocidos. También se incrementó la dimensionalidad de los datos, dividiendo algunos atributos en otros; tal es el caso del atributo *hierarchy unit* que contenía el nombre de la unidad y la sección a la que pertenece cada problema, por lo que el atributo se dividió para facilitar la manipulación de las secciones que conforman a cada unidad.

Transformación Para detectar los tipos de rendimiento académico que pudieran tener los estudiantes al resolver problemas de Álgebra se llevaron a cabo un conjunto de tareas:

- Se calculó el rendimiento del estudiante por problema. El eje central de todo sistema tutor son las tareas (ejercicios, problemas) que el estudiante debe resolver. Estas tareas se descomponen en un conjunto de actividades llamadas pasos. Para obtener la evolución académica de cada estudiante es necesario analizar la manera en que resuelve cada una de las actividades involucradas en el problema, y comparar su rendimiento con los estudiantes que resolvieron los mismos problemas con el fin de determinar la dificultad a la que se enfrentó al resolver el problema, la rapidez con que lo resolvió, así como la respuesta del problema.
  Dificultad: Partiendo del hecho de que los sistemas tutores descomponen un problema en una serie de actividades (pasos), de acuerdo al progreso del

[4] http://www.carnegielearning.com/secondary-curricula/algebra1/

estudiante; el incremento de pasos en un problema supone la dificultad para resolverlo. De esta manera, la dificultad esta definida en la Ecuación 1.

$$dificultad = \frac{PR - PP}{PP} \tag{1}$$

Donde: PR se refiere a la cantidad de actividades en que el sistema tutor descompuso el problema. PP se refiere al promedio de pasos realizados por los estudiantes que resolvieron el mismo problema. Un resultado negativo significa que el problema representa una menor dificultad para el estudiante que para el promedio de estudiantes que resolvieron el mismo problema. Un resultado igual a cero significa que la dificultad fue la misma que la que experimentó el promedio de estudiantes respecto al mismo problema. Finalmente, un resultado mayor que cero indica que el problema representó mayor dificultad que para el promedio de estudiantes.
Tasa de corrección: También se calculó la tasa de corrección del problema a través de la Ecuación 2.

$$tasaCorrectitud = \frac{RI - RC}{PR} \tag{2}$$

Donde: RC se refiere a los pasos contestados correctamente en un problema. RI se refiere a los pasos contestados incorrectamente en un problema. PR se refiere al número de pasos en que el sistema tutor descompuso el problema para cada estudiante.Un resultado negativo significa que el estudiante resolvió correctamente una mayor cantidad de problemas. Un resultado igual a cero significa que el estudiante resolvió la misma cantidad de problemas correctos que incorrectos. Finalmente, un resultado mayor que cero indica que el estudiante resolvió correctamente una cantidad menor que el promedio de estudiantes.
Factor de rapidez: Para obtener el factor de rapidez con que un estudiante resolvió un problema, se utilizó la Ecuación 3.

$$factorRapidez = \frac{TE - TP}{TP} \tag{3}$$

Donde: TE se refiere al tiempo que le llevo al estudiante resolver el problema. TP se refiere al tiempo promedio empleado por los estudiantes que resolvieron el mismo problema. Un resultado negativo significa que el problema fué resuelto en un tiempo menor que el promedio de estudiantes que resolvieron el mismo problema. Un resultado igual a cero significa que la dificultad fue la misma que la que experimentó el promedio de estudiantes respecto al mismo problema. Finalmente, un resultado mayor que cero indica que al estudiante le tomó mas tiempo que el promedio de estudiantes resolver el problema. En este punto se eliminó el atributo que contenía el nombre del paso (actividad).

- Se calculó el rendimiento académico de cada estudiante por unidad a través de una operación de agregación sobre el rendimiento que obtuvo en los problemas resueltos en cada unidad. En este punto se observa que la evolución del estudiante se puede representar a manera de ilustración en una función

sigmoidal (ver Figura 1), que de acuerdo al conocimiento experto es posible dividirla en 3 sectores que permitirán rastrear la evolución acad'emica del estudiante a lo largo del curso. Estos sectores se etiquetaron como: Bajo (S1), ya que la cantidad de problemas resueltos correctamente es mínima; Promedio (S2) existe un incremento el número de problemas resueltos correctamente; y Alto (S3) ya que existe una estabilización en el número de problemas resueltos correctamente. En este punto se eliminaron los atributos que contenía el nombre del problema y la sección a la que pertenecá el problema.

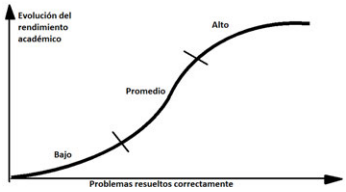


Figura 1. Evolución del rendimiento estudiantil

- Se calculó el rendimiento global por estudiante, a través de una operación de agregación sobre el rendimiento que obtuvo en las unidades cursadas. De esta manera cada estudiante estará representado por un solo registro. En este punto se eliminó el atributo que contenía el nombre de la unidad cursada.

Por otra parte, con el fin de detectar la similitud entre los estudiantes en la forma de resolver problemas, se realizó un proceso de clustering jerárquico aglomerativo del que se obtuvieron 5 clusters. Realizado el proceso de clustering se ordenaron los registros de los estudiantes en función de su rendimiento académico, que se obtuvo a través de la medida heurística definida en la Ec. 4.

$$rendimiento = \frac{dificultad * tasaCorrectitud}{factorRapidez} \tag{4}$$

Donde: *dificultad* se refiere a la dificultad a la que se enfrentó el estudiante para resolver un problema (ver Ecuación 1); *tasaCorrectitud* se refiere al porcentaje de correctitud de un problema (ver Ecuación 2), y *factorRapidez* se refiere al tiempo promedio empleado para resolver un problema (ver Ecuación 3).

Esta medida heurística permitió clasificar a los estudiantes con base a su rendimiento. De esta manera se obtuvieron 5 tipos de rendimiento etiquetados de la siguiente forma: *Bajo, Promedio, Casi alto, Alto* y *Muy alto*.

Minería de datos En la etapa de transformación se obtuvieron los tipos de rendimiento académico comparando el rendimiento de los estudiantes al resolver problemas con respecto a sus compañeros. Sin embargo para obtener prototipos que describan los diferentes tipos de rendimiento académico que se obtuvieron en

la etapa de transformación se realizó un mapeo con los datos originales. Es decir, ya se tienen identificados los estudiantes que caen bajo cada prototipo por lo que se aplicaron funciones de resumen sobre los datos originales. De este modo, cada prototipo está definido por el porcentaje de correctitud de los problemas resueltos por el estudiante, el número de intentos para resolver problemas, el tiempo (segundos) empleado así como la cantidad de ayuda solicitada (ver Tabla 1). Una vez que se obtuvieron los prototipos de datos fue necesaria la intervención del conocimiento experto para determinar los 3 sectores (S1, S2, S3) que componen a cada prototipo. Estos sectores permitirán rastrear la evolución del estudiante a lo largo del curso, sin embargo para los prototipos de la Tabla 1 esta información fue omitida por cuestiones de espacio. En este punto se eliminó el atributo que contenía el identificador del estudiante, ya que no era de utilidad.

	Prototipos de datos				
Dominio	**Bajo**	**Promedio**	**Casi alto**	**Alto**	**Muy alto**
Porcentaje de correctitud promedio/problema	35	62	75	84	96
Mínimo porcentaje de correctitud/problema	19	56	72	82	92
Máximo porcentaje de correctitud/problema	50	67	78	87	100
Num. de intentos/problema	2	1	1	1	1
Tiempo (segundos)/problema	50	41	33	30	25
Cantidad de ayuda solicitada/problema	1	0	0	0	0

Tabla 1. Prototipos que se obtuvieron del proceso de descubrimiento de conocimiento

2.2. Representación formal de los prototipos

Los prototipos se representaron formalmente a través de números borrosos, que permitirán obtener el grado de afinidad de una nueva situación con cada uno de los mismos. Por simplicidad los prototipos se representaron a través de funciones de pertenencia triangulares y trapezoidales. Para representarlos gráficamente fue necesario calcular sus centros y la base de los triángulos.

- *Centro del prototipo*: se utilizó la metodología propuesta por [3] para calcular el centro del prototipo.
  - Se ordenaron los casos que caen bajo cada prototipo con base al rendimiento que se obtuvo en la etapa de transformación (ver Ecuación 4). De esta manera se tiene que: $y'_i <= y'_{i+1}$
  - Se calcula la diferencia entre los datos adyacentes. De esta forma se tiene que $dif = y'_{i+1} - y'_i$
  - Calcular el grado de similitud entre los datos adyacentes a través de la Ecuación 5.

$$s_i = \begin{cases} 1 - \frac{dif_i}{C * \sigma_s} & \text{si } dif <= C * \sigma_s \\ 0 & \text{en otro caso} \end{cases} \quad (5)$$

Donde: σ es la desviación standard de los datos que caen bajo cada prototipo; *dif* es la diferencia entre los casos adyacentes pertenecientes a cada prototipo;

C es un valor definido por el usuario que indica el grado de similitud entre los casos que caen bajo cada prototipo. De esta manera, a través de la Ecuación definida en [3] se obtuvieron los centros de los prototipos.

- *Base del triangulo*: con el fin de obtener prototipos borrosos simétricos se utilizó la distancia desde el cero al centro del primer prototipo [2] como distancia standard para construir la base de los 3 triángulos.

Prototipos representados a través de conjuntos borrosos de tipo 2 Es necesario considerar que entre los distintos tipos de rendimiento académico existen determinadas similitudes que vienen dadas por ciertos factores tales como: la rapidez con que resuelve un problema, las pistas solicitadas entre otros. Por lo que es posible que exista solapamiento entre los prototipos, este solapamiento refleja la similitud entre los mismos. Para representar y manipular esta imprecisión se hace uso de conjuntos borrosos de tipo 2.

De esta manera una vez que los prototipos han sido descubiertos mediante el proceso DCPB y representados formalmente a través de números borrosos. Para representar y manipular el solapamiento de los prototipos se utiliza el *Teorema de Representación* [4] con el fin de obtener un conjunto borroso de Tipo 2 de Intervalo (IT2FS, Interval Type-2 Fuzzy Set). Un IT2FS está delimitado por dos conjuntos borrosos ordinarios (de tipo 1): el UMF (Upper Membership Function) y el LMF (Lower Membership Function), que representan la alta y baja pertenencia con el IT2FS. El área entre UMF y LMF es el FOU (Footprint Of Uncertainty), que permite modelar la imprecisión existente.

De este modo, un prototipo representado a través de un IT2FS estará compuesto por la unión de todos sus prototipos embebidos (ver Figura 2). Para obtener el UMF y el LMF de los IT2FSs que representarán a los prototipos borrosos se siguió la metodología definida por Liu y Mendel [5], que se describe brevemente a continuación:

- Se numeraron los parámetros de los T1FSs triangulares embebidos en el FOU_{Alto}, de tal manera que a_i y b_i representan los extremos izquierdo y derecho del soporte del *i-ésimo* T1FS; c_i es el centro del *i-ésimo* T1FS embebido.
- Se obtiene el soporte del UMF_{Alto} (α_{UMF}) a través del mínimo de los a_i ($\underline{a}$), y el máximo de los b_i ($\bar{b}$).
  $\alpha_{UMF_{Alto}} = (min(a_i), max(b_i))$
- Se calcularon los centroides inferior ($\underline{c}$) y superior ($\bar{c}$) del UMF_{Alto}. Para calcular el centroide inferior se obtiene el mínimo de los c_i, y para obtener el centroide superior se obtiene el máximo de los c_i: $\underline{c} = min(c_i)$, $\bar{c} = max(c_i)$
  De esta manera, se obtuvo un UMF_{Alto} con forma trapezoidal y definido por los puntos $(\underline{a}, 0), (\underline{c}, 1), (\bar{c}, 1), (\bar{b}, 0)$.
- Se obtuvo el soporte del LMF_{Alto} (α_{LMF}) a través del máximo de los a_i ($\bar{a}$) y el mínimo de los b_i ($\underline{b}$). El LMF_{Alto} tiene forma triangular por lo que para calcular su centro, y su altura máxima se emplean las siguientes ecuaciones.

$$p = \frac{\underline{b}(\bar{c} - \bar{a}) + \bar{a}(\underline{b} - \underline{c})}{(\bar{c} - \bar{a}) + (\underline{b} - \underline{c})} \quad (6) \qquad \mu_p = \frac{\underline{b} - p}{\underline{b} - \underline{c}} \quad (7)$$

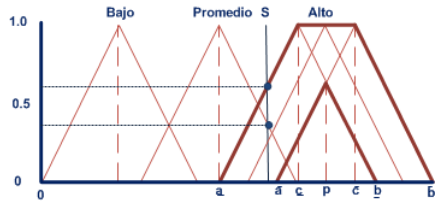


Figura 2. Prototipos borrosos representados a través de IT2FS

En la teoría de *Prototipos Deformables Borrosos* cada prototipo es representado por un número borroso y está asociado con su correspondiente prototipo de datos. Los prototipos de datos asociados con los prototipos borrosos embebidos en el IT2FS se agregaron con el fin de obtener uno solo que los represente. En la Tabla 2 se muestran los 3 prototipos de datos que se utilizarán en el proceso de predicción. Para cada prototipo se determinan 3 sectores (S1, S2 y S3) definidos por el conocimiento experto. Cada prototipo está definido por el porcentaje de correctitud de los problemas resueltos por el estudiante, el número de intentos para resolver problemas, el tiempo (segundos) empleado así como la cantidad de ayuda solicitada

	Prototipos de datos								
	Bajo			**Promedio**			N		
Dominio	**S1**	**S2**	**S3**	**S1**	**S2**	**S3**	**S1**	**S2**	**S3**
Porcentaje de correctitud promedio/problema	8	25	42	55	63	71	79	87	96
Mínimo porcentaje de correctitud/problema	0	17	33	51	60	68	75	84	93
Máximo porcentaje de correctitud/problema	16	32	50	59	67	74	83	92	100
Num. de intentos/problema	3	2	1	2	1	1	2	1	1
Tiempo (segundos)/problema	106	50	22	100	41	20	100	30	18
Cantidad de ayuda solicitada/problema	2	1	0	1	0	0	1	0	0

Tabla 2. Prototipos de datos

3. Aproximación de los prototipos a una nueva situación

Se tomó como medida objetiva la opinión de los expertos para determinar el conjunto de características que definirán a cada nuevo estudiante, y que se describen a continuación:

- *Capacidad cognitiva:* Se refiere al conocimiento que posee el estudiante del tema y vendrá dada por: Alta/0.8, Media/0.5, Baja/0.2.

- *Capacidad para comprender el problema:* Se refiere a la capacidad que posee el estudiante para interpretar lo que se plantea y vendrá dada por: Alta/0.75, Media/0.45, Baja/0.15.
- *Complejidad del problema:* vendrá dada por: Alta/0.85, Media/0.45, Baja/0.15.

De esta manera, una nueva estimación se inicia cuando el usuario (profesor) introduce los valores de las características que definen a cada estudiante. Posteriormente, se agregan los valores de las características con el fin de obtener el valor que definirá a cada nueva situación. Se obtienen los grados de compatibilidad de la situación con cada prototipo, y finalmente se modifican los prototipos afines.

Por ejemplo, supóngase que el primer día en que un estudiante utiliza el sistema tutor, el usuario (profesor) introduce los siguientes valores: Capacidad cognitiva: Media/0.5, Interpretación: Alta/0.75, Complejidad del problema: Media/0.45. De esta manera, aplicando operaciones de agregación sobre estas características se obtiene el valor que definirá a ese nuevo estudiante S=0.56. De este modo, el estudiante (S) tiene afinidad con el prototipo *Promedio* y el prototipo *Alto* (FOU_{Alto}) (ver Figura 2).

La afinidad del nuevo caso con el prototipo *Promedio* se obtiene a través de la función de pertenencia triangular definida en la literatura, para este ejemplo es de 0.45 ($\mu_{Promedio} = 0{,}45$). Por otra parte, para obtener la afinidad del nuevo caso con el prototipo *Alto* (FOU_{Alto}) es necesario calcular las afinidades del nuevo caso con el UMF (Upper Membership Function) y el LMF (Lower Membership Function). El UMF y el LMF son funciones de pertenencia de tipo 1. Para obtener el LMF es necesario calcular su altura y su centro de acuerdo a las ecuaciones definidas por Liu-Mendel en [5]. Posteriormente, para obtener la pertenencia real de la situación con el prototipo *Alto* en este trabajo se tomaron en cuenta las consideraciones definidas en la Ecuación 8.

$$\mu_{FOUi} = \begin{cases} UMF_i & \text{si } LMF_i = 0 \\ UMF_i - \frac{UMF_i - LMF_i}{2} & \text{si } LMF_i > 0 \end{cases} \tag{8}$$

Donde: UMF_i y LMF_i corresponden a la pertenencia superior e inferior con el i-ésimo IT2FS; y μ_{FOU_i} es la pertenencia real con el i-ésimo prototipo.

Para el ejemplo anterior, el $UMF_{Alto} = 0{,}55$ y el $LMF_{Alto} = 0$ por lo que de acuerdo con la Ecuación 8, la afinidad real es de 0.55 ($\mu_{Alto} = 0{,}55$).

Para describir el comportamiento del nuevo estudiante se deforman los prototipos de datos (ver Tabla 2), de acuerdo al grado de afinidad de la nueva situación con los prototipos borrosos previamente definidos. La técnica de deformación utilizada fue la definida en la Ecuación 9.

$$Situacion_j(p_1, p_2, ..p_n) = \frac{\sum_{i=1}^{n} \mu_i v_i}{\sum_{i=1}^{n} \mu_i} \tag{9}$$

Donde: p_n son las características de la j-ésima situación; μ_i es el grado de pertenencia del i-ésimo prototipo con la j-ésima situación; v_i es el i-ésimo prototipo de datos.

Dominio	S1	S2	S3
Porcentaje de correctitud promedio/problema	68	76	85
Mínimo porcentaje de correctitud/problema	64	73	82
Máximo porcentaje de correctitud/problema	72	81	88
Num. de intentos/problema	2	1	1
Tiempo (segundos)/problema	100	35	19
Cantidad de ayuda solicitada/problema	1	0	0

Tabla 3. Estimación del rendimiento académico real del nuevo estudiante

La estimación real del rendimiento del estudiante (ver Tabla 3), se dividió en 3 sectores que permitirán rastrear su evolución académica a lo largo del curso. De esta manera, al inicio de la interacción con el sistema tutor, se sitúa al estudiante en el primer sector del prototipo real estimado. Al finalizar cada unidad se verifica si ha habido un cambio de sector por lo que se compara el rendimiento del estudiante *(r)* en la unidad recién cursada (anterior), con los vectores correspondientes a cada sector, a través de cualquier medida de distancia, por ejemplo, la distancia Euclídea. El sector donde se posicionará la evolución será aquel con menor distancia.

4. Conclusiones

En este trabajo se hizo uso de los *Prototipos Deformables Borrosos* con el fin de predecir el rendimiento de un estudiante dentro de un sistema tutor. Debido a la imprecisión inherente a la información utilizada, es posible que exista solapamiento entre las representaciones borrosas de los prototipos, por lo que p se hizo uso de conjuntos borrosos de tipo 2 de intervalo (IT2FS).

Referencias

1. Romero, C., Ventura, S.: Educational data mining: A survey from 1995 to 2005. Expert System with Applications **33**(1) (2007) 135–146
2. Olivas, J.A.: Contribución a un estudio experimental de predicción basada en Categorías Deformables Borrosas. Tesis de doctorado, Universidad de Castilla-La Mancha (2000)
3. Hong, T.P., Lee, C.Y.: Induction of fuzzy rules and membership functions from training examples. Fuzzy sets and systems **84** (1996) 33–47
4. Mendel, J.M., Bob, R.I.: Type-2 fuzzy sets made simple. IEEE Transactions on fuzzy systems **10**(2) (2002) 117–127
5. Liu, F., Mendel, J.M.: An interval approach to fuzzistics for interval type-2 fuzzy sets. In: Fuzzy Systems Conference, 2007. FUZZ-IEEE 2007. (2007) 1–6

Un sistema difuso evolutivo para la detección de excepciones en descubrimiento de subgrupos

C.J. Carmona, P. González, and M.J. del Jesus

Grupo de investigación SIMIDAT, Departamento de Informática, Universidad de Jaén
{ccarmona,pglez,mjjesusolzwarth}@ujaen.es

Resumen El descubrimiento de subgrupos (SD) es una técnica de minería de datos cuyo principal objetivo es buscar descripciones de subgrupos de datos que sean estadísticamente inusuales con respecto a una propiedad de interés. En esta técnica se prefiere obtener reglas generales que describan tantos ejemplos como sea posible, lo que puede provocar que algunos ejemplos se describan incorrectamente. Estos ejemplos negativos se pueden agrupar en excepciones. El trabajo presenta un sistema difuso evolutivo para la detección de excepciones asociadas a reglas previamente obtenidas mediante algoritmos de SD. El objetivo es obtener una nueva descripción con mayor precisión que el sugbrupo original. El algoritmo se puede aplicar a los resultados de cualquier algoritmo de SD, y su utilidad se verifica mediante un estudio experimental.

Keywords: Descubrimiento de subgrupos, reglas de excepción, sistemas difusos evolutivos, algoritmos evolutivos multiobjetivo.

1. Introducción

El descubrimiento de subgrupos (*Subgroup Discovery*, SD) [11,16,8] es una técnica de inducción supervisada que obtiene reglas descriptivas mediante aprendizaje supervisado. El objetivo es encontrar reglas interesantes con respecto a una variable de interés, que aporten información desconocida, que confirmen información ya conocida por intuición o que ofrezcan incluso información sorprendente que contradiga el conocimiento de los expertos.

Aunque el conocimiento extraído debe ser sencillo para que sea útil a los expertos, la búsqueda de modelos sencillos en los algoritmos de SD puede provocar una reducción de la precisión si se obtienen reglas generales con ejemplos negativos. La detección de estos ejemplos negativos y su descripción mediante reglas con excepciones [9] puede mejorar el conocimiento extraído sobre la propiedad de interés, no sólo a través de una mejora de la precisión de las reglas de SD sino ofreciendo además conocimiento novedoso y sorprendente a los expertos.

La tarea de SD suele implicar la optimización de diferentes medidas de calidad relacionadas con aspectos de precisión, simplicidad e interés de las descripciones de subgrupos, que suelen representarse mediante reglas. Los algoritmos evolutivos han mostrado un buen comportamiento en procesos de aprendizaje de reglas y problemas multiobjetivo.

Este trabajo presenta un algoritmo de post-procesamiento basado en un sistema difuso evolutivo multiobjetivo para la detección de excepciones en subgrupos. La propuesta busca, a través de un enfoque multiobjetivo, excepciones en subgrupos previamente obtenidos mediante algún algoritmo de SD. El comportamiento del algoritmo se verifica mediante un estudio experimental. El trabajo se organiza de la siguiente forma: la sección 2 describe los conceptos fundamentales utilizados. El algoritmo para la detección de excepciones asociadas a reglas de SD se presenta en la sección 3. La sección 4 detalla el estudio experimental y la sección 5 las conclusiones.

2. Preliminares

2.1. Descubrimiento de subgrupos

El concepto de SD fue introducido por Kloesgen [11] y Wrobel [16]. Se define como, dada una población de individuos y una propiedad de estos individuos en la que estamos interesados, descubrir subgrupos de la población que sean estadísticamente "más interesantes", es decir, que sean tan grandes como sea posible y que tengan una distribución estadística lo más inusual posible respecto a la propiedad de interés (o variable objetivo).

Por tanto, el objetivo del SD es extraer conocimiento descriptivo acerca de una propiedad de interés de los datos [13]. El conocimiento se representa mediante patrones que deben caracterizar los datos representados de forma que los expertos puedan entenderlos. Así, en SD no es necesario obtener relaciones completas, sino que se buscan relaciones parciales, representadas normalmente mediante reglas de tipo $Cond \rightarrow VarObjetivo$.

Uno de los aspectos más importantes en un algoritmo de SD es el relativo a las medidas de calidad utilizadas para analizar el interés de los subgrupos obtenidos. En la literatura de SD se ha utilizado una gran variedad de medidas de calidad, que pueden ser divididas en distintos grupos dependiendo de su objetivo principal[8]: complejidad, generalidad, precisión e interés. Se puede encontrar una revisión de los algoritmos de SD existentes y sus aplicaciones en [8].

2.2. Minería de reglas de excepción

La mayoría de los métodos de minería de datos se centran en la extracción de reglas generales con alto soporte y confianza. Sin embargo, reglas con bajo soporte podrían aportar conocimiento interesante y sorprendente a los expertos. La minería de reglas de excepción fue introducida por Hussain y otros en [9] como la extracción de reglas con bajo soporte y alta confianza. En este sentido, una excepción es interesante cuando se diferencia del resto [14].

Se pueden distinguir dos enfoques al buscar reglas de excepción [9]: dirigido (o subjetivo) que obtiene un conjunto de reglas de excepción, cada una de las cuales contradice una creencia especificada por el experto; y no dirigido (u objetivo) que obtiene un conjunto de pares de regla general / excepción.

Aplicado a los resultados de cualquier algoritmo de SD, la detección de excepciones podría llevar a una mejora en la precisión y en la descripción, puesto que se descubrirían dentro del subgrupo pequeñas áreas con ejemplos incorrectamente descritos o negativos.

3. Un sistema difuso evolutivo multiobjetivo para la detección de excepciones en subgrupos

Se presenta aquí un algoritmo de post-procesamiento para la detección de excepciones en SD. El objetivo es buscar, dentro de un subgrupo dado, pequeñas áreas que describan excepciones al conocimiento representado por este subgrupo, formadas por ejemplos cuyo valor de la variable objetivo sea distinto al del subgrupo inicial. El concepto se explica en la figura 1, que representa mediante un círculo sombreado en gris un subgrupo correspondiente al valor o de la variable objetivo. Este subgrupo cubre todos los ejemplos del valor o, además de algunos ejemplos del otro valor de la variable objetivo (valor x). A pesar de cubrir algunos ejemplos negativos, la representación del conocimiento utilizada es simple y puede ser muy útil para los expertos. Estos ejemplos negativos (marcados en un rombo sombreado en gris oscuro) constituyen excepciones del subgrupo. La nueva regla, formada por el subgrupo inicial junto con sus excepciones, aporta una mejora en la descripción y un incremento de la precisión.

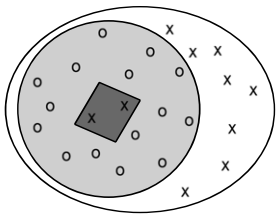


Figura 1. Detección de excepciones dentro de un subgrupo

Para lograr este objetivo, se propone un sistema difuso evolutivo (*Evolutionary Fuzzy System*, EFS) multiobjetivo [7]. Los algoritmos evolutivos son métodos de búsqueda de propósito general que se han demostrado muy eficaces en procesos de aprendizaje de reglas, y la utilización de un enfoque multiobjetivo resulta muy adecuada cuando se deben optimizar diferentes medidas de calidad, como ocurre en SD.

El sistema parte de un conjunto de i subgrupos obtenidos por cualquier algoritmo de SD ($S_R = \{R_1,\ R_2, \ldots,\ R_i\}$) representados por reglas del tipo:

$$R_i : IF\ Cond_i\ THEN\ VarObjetivo \tag{1}$$

en el que $Cond_i$ representa el antecedente de la regla i.

A cada una de las reglas o subgrupos se les aplica el proceso evolutivo, obteniendo una nueva regla que incorpora las excepciones encontradas. El resultado final, una vez aplicado el proceso a todos los subgrupos iniciales, es un conjunto de subgrupos con excepciones, uno por cada subgrupo del conjunto inicial.

El algoritmo puede utilizarse sobre conjuntos de datos con variables continuas y/o categóricas, obteniendo subgrupos modificados que serán del mismo tipo (difusos y/o nítidos) que los originales. Las características principales de la propuesta se presentan a continuación.

3.1. Representación de los individuos

Los individuos de la población son excepciones al subgrupo inicial. Se utiliza un modelo de representación entero con tantos genes como variables tiene el conjunto de datos original, sin considerar la variable objetivo. La codificación se realiza según el enfoque "*Cromosoma = Regla*" [7], en que que el cromosoma sólo representa el antecedente.

El algoritmo puede trabajar con dominios difusos y/o nítidos (y por tanto con reglas difusas y/o nítidas) obteniendo subgrupos modificados del mismo tipo. La representación indicada se utiliza tanto para variables categóricas como continuas. Para variables continuas, el algoritmo utiliza la lógica difusa para manejar las variables continuas sin necesidad de aplicar una discretización previa, ya que esto que podría ocasionar una pérdida de calidad. Así, las variables continuas se consideran como variables lingüísticas, y los conjuntos difusos correspondientes a las variables lingüísticas son los definidos por los algoritmos de SD para obtener los subgrupos iniciales. La figura 2 muestra un ejemplo de variable con particiones difusas uniformes y funciones de pertenencia triangulares correspondientes a cinco etiquetas lingüísticas (*Muy Bajo*, *Bajo*, *Medio*, *Alto*, *Muy Alto*). En el caso de variables categóricas, cada gen estará compuesto por un valor entero que codifique el valor de la variable correspondiente.

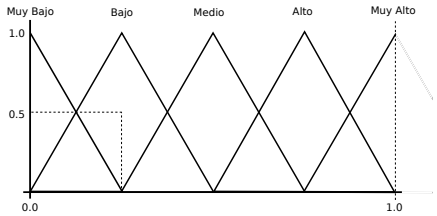


Figura 2. Uso de la lógica difusa para una variable con cinco etiquetas lingüísticas

En el individuo no se representa el consecuente de la regla puesto que está prefijado. Como los individuos de la población representan excepciones al subgrupo inicial, el valor de la variable objetivo de los individuos debe ser distinto al del subgrupo inicial (si la variable objetivo tiene dos clases, las excepciones tendrán

el valor contrario al del subgrupo inicial; si tiene más de dos clases, el valor de la variable objetivo de las excepciones será cualquiera distinto al del subgrupo inicial). Por lo tanto, si el valor de la variable objetivo del subgrupo original es $VarObjetivo$, el valor de la variable objetivo de las excepciones es $\overline{VarObjetivo}$.

Cada regla de excepción de un subgrupo mantiene los valores de los atributos del subgrupo original, además de incorporar nuevos atributos. Consideremos el siguiente ejemplo de regla difusa de descripción de subgrupos: $IF\ x_1 = Medio\ AND\ x_3 = Bajo\ THEN\ VarObjetivo$. Una excepción a este subgrupo, como $IF\ x_1 = Medio\ AND\ x_3 = Bajo\ AND\ x_5 = Medio\ THEN\ \overline{VarObjetivo}$, mantiene los valores del subgrupo original e incorpora nuevos atributos. Por lo tanto, las excepciones son reglas más específicas que los subgrupos originales de los que provienen.

3.2. Algoritmo evolutivo multiobjetivo

El algoritmo busca, dentro del espacio delimitado por el antecedente de la regla, pequeños grupos de ejemplos con un valor de la variable objetivo distinto al establecido en el consecuente del subgrupo; es decir, busca conjuntos de ejemplos incorrectamente descritos por el subgrupo. Esta búsqueda se realiza mediante un algoritmo evolutivo multiobjetivo que utiliza el enfoque NSGA-II [4], de acuerdo con un proceso en dos pasos:

1. *Detección de conjuntos de excepciones asociadas al subgrupo.* Se aplica el algoritmo para extraer excepciones formadas por los mismos valores de las variables presentes en el subgrupo además de otras variables adicionales, y correspondientes a un valor distinto de la variable objetivo del subgrupo.
2. *Generación de los subgrupos modificados.* Se obtiene el subgrupo modificado mediante la exclusión de las excepciones encontradas en el paso anterior:

$$R'_i : IF\ Cond_i\ AND\ \overline{Exc_i}\ THEN\ VarObjetivo \tag{2}$$

donde Exc_i representa las condiciones de las excepciones asociadas a la regla R_i (sin las variables comunes).

El proceso evolutivo termina cuando el algoritmo alcanza un número establecido de evaluaciones, y devuelve como resultado el conjunto de individuos no repetidos del frente de Pareto que alcancen un umbral establecido de confianza. El algoritmo se ejecuta para cada subgrupo inicial (obteniendo subgrupos con excepciones) por lo que el número de subgrupos modificados obtenido es el mismo que el de subgrupos iniciales.

El algoritmo utiliza expresiones adaptadas de las medidas de sensibilidad [11] y confianza [5] tanto para evaluar las excepciones como para guiar el proceso de aprendizaje:

- **Sensibilidad**. Mide la proporción de verdaderos positivos que son correctamente identificados:

$$Sens(Exc) = \frac{TP}{TP + FN} = \frac{n(\overline{VarObjetivo} \cdot Cond)}{n(\overline{VarObjetivo})} \tag{3}$$

donde $n(\overline{VarObjetivo})$ son los ejemplos que tienen el valor contrario de la variable objetivo del subrupo inicial, y $n(\overline{VarObjetivo} \cdot Cond)$ son los ejemplos cubiertos por la excepción con el valor contrario de la variable objetivo al del subgrupo.

- **Confianza difusa**. Mide la proporción de ejemplos correctamente identificados respecto al valor contrario de la variable objetivo:

$$FCnf(Exc) = \frac{\sum_{E^k \in E / E^k \in \overline{VarObjetivo}} APC(E^k, Exc)}{\sum_{E^k \in E} APC(E^k, Exc)} \tag{4}$$

donde $E^k = (e_1^k, e_2^k, \ldots, e_v^k), VarObjetivo^k)\ /k = 1, \ldots,\ N,\quad VarObjetivo^k \in\ T$ es un ejemplo del conjunto E, v es el número de variables del ejemplo, e_v^k es el valor de la variable v en el ejemplo E^k, $VarObjetivo^k$ es el valor de la variable objetivo para el ejemplo E^k y APC es el grado de compatibilidad entre un ejemplo y el antecedente de una regla de excepción (cuando se aplica en dominios nítidos, el grado de compatibilidad entre un ejemplo y el antecedente de una regla es 0 ó 1).

3.3. Operadores genéticos

El algoritmo evolutivo de post-procesamiento incluye operadores de selección por torneo y cruce multi-punto, además de otros operadores específicos:

- *Inicialización orientada*. Se genera una población de individuos que contienen los mismos valores del subgrupo inicial, además de nuevos valores para el resto de variables. Para conseguir esto, parte de los individuos de la población se generan de forma sesgada, y el resto de forma aleatoria. Los valores de las variables que forman parte del subgrupo inicial se copian directamente en los nuevos individuos de la población, mientras que el resto de valores se obtienen de la siguiente forma: los valores del 75 % de los individuos se generan considerando que sólo un máximo del 90 % de las variables pueden formar parte de la regla; para el resto (25 %) los valores se generan de forma aleatoria. Este operador permite comenzar el proceso evolutivo con un conjunto de reglas muy específicas puesto que los individuos generados son reglas en las que intervienen un alto número de variables.
- *Mutación orientada*. Se deriva de la mutación estándar [6], pero impidiendo la modificación de los valores que formaban parte del subgrupo inicial. Además, la mutación del valor de una variable que no formaba parte del subgrupo inicial no siempre implica su eliminación, puesto que se asigna a esta variable uno de los valores posibles que sea diferente del actual.
- *Reinicialización orientada basada en cobertura*. El algoritmo utiliza una modificación del operador definido para el algoritmo NMEEF-SD [3]. En el operador original, se realiza una comprobación antes de generar la población de la siguiente generación, para ver si el Pareto evoluciona o no. Si el Pareto no evoluciona, todos los individuos no repetidos del Pareto se introducen en

la población de la siguiente generación y el resto de individuos hasta completar la población se generan de forma que cubran ejemplos del conjunto de datos no cubiertos por los individuos que forman el Pareto. La modificación utilizada consiste en que los individuos generados deben mantener los mismos valores en las variables que intervenían en el subgrupo inicial, y por tanto serán más específicos. Los individuos generados se asocian a ejemplos no cubiertos por el Pareto, al seleccionar aleatoriamente un ejemplo no cubierto y codificar los valores del individuo con respecto a este ejemplo.

4. Estudio experimental

La experimentacón se ha realizado con conjuntos de datos del repositorio KEEL[1] [2], cuyas propiedades se muestran en la tabla 1 (donde n_v es el número de variables, n_{vD} el número de variables discretas y n_{vC} el de variables continuas y N el número de ejemplos). Se ha utilizado validación cruzada con 10 particiones, y tres ejecuciones para los algoritmos no determinísticos.

Tabla 1. Conjuntos de datos utilizados del repositorio KEEL

♯	*Nombre*	n_v	n_{vD}	n_{vC}	N
1	Australian	14	8	6	690
2	Crx	15	12	3	690
3	Heart	13	6	7	270
4	Hepatitis	19	13	6	155
5	Monk-2	6	6	0	432
6	Housevotes	16	16	0	435
7	Saheart	9	4	5	462
8	Tic-tac-toe	9	9	0	958

Para mostrar las ventajas de la propuesta se utilizan dos conocidos algoritmos de SD para obtener los subgrupos iniciales: NMEEF-SD [3] y Apriori-SD [10]. A los resultados de estos algoritmos se les aplica la propuesta para obtener subgrupos con excepciones. La evaluación de los subgrupos con excepciones se realiza mediante expresiones adaptadas para las medidas de calidad de relevancia (Rel) [11], atipicidad (Ati) [12], sensibilidad (Sen) [11] y confianza (Cnf) [1].

- Relevancia de un subgrupo con excepciones (R'):

$$Rel'(R'_i) = 2 \cdot \sum_{k=1}^{n_c} (TP_{R'_i})_k \cdot log \frac{(TP_{R'_i})_k}{((TP+FN)_{R_i})_k \cdot \frac{((TP+FP)_{R'_i})_k}{N}} \quad (5)$$

donde $TP_{R'_i} = TP_{R_i} - FP_{Exc_i}$, TP_{R_i} es el número de ejemplos correctamente descritos de la regla, FP_{Exc_i} es el número de ejemplos incorrectamente descritos por el conjunto de excepciones asociadas a la regla, $(TP+FN)_{R_i}$ es el número de ejemplos para los valores de la variable objetivo, $(TP+FP)_{R'_i} =$

[1] `http://www.keel.es`

$(TP + FP)_{R_i} - (TP + FP)_{Exc_i}$, $(TP + FP)_{R_i}$ es el número de ejemplos cubiertos por la regla y $(TP + FP)_{Exc_i}$ es el número de ejemplos cubiertos por el conjunto de excepciones asociadas a la regla inicial.

- Atipicidad de un subgrupo con excepciones:

$$Atip'(R'_i) = \left(\frac{TP_{R'_i}}{(TP+FP)_{R'_i}} - \frac{(TP+FN)_{R_i}}{N} \right) \cdot \frac{(TP+FP)_{R'_i}}{N} \tag{6}$$

- Sensibilidad de un subgrupo con excepciones:

$$Sens'(R'_i) = \frac{TP_{R'_i}}{(TP+FN)_{R_i}} \tag{7}$$

- Confianza difusa de un subgrupo con excepciones:

$$FCnf'(R'_i) = \frac{\sum_{E^k \in E / E^k \in VarObjetivo} APC(E^k, R'_i)}{\sum_{E^k \in E} APC(E^k, R'_i)} \tag{8}$$

donde $APC(E^k, R'_i) = APC(E^k, R_i) - APC(E^k, Exc_i)$.

La tabla 2 muestra la media de los resultados de los algoritmos de SD y la aplicación posterior del post-procesamiento propuesto, donde n_r representa la media del número de subgrupos obtenidos, n_v es la media del número de variables para cada subgrupo y los valores de las medidas de calidad indicadas (REL, ATI, SEN y CNF) son las medias de los resultados de las distintas reglas de los conjuntos de reglas. Por cuestiones de espacio, sólo se muestran aquí los resultados medios y los de los tests estadísticos realizados.

Tabla 2. Resultados obtenidos por los algoritmos

Algoritmo	n_r	n_v	REL	ATI	SEN	CNF
Apriori-SD	5.70	2.30	3.219	0.092	**0.587**	0.708
Apriori-SD+Exceptions	9.36	7.64	**3.559**	**0.102**	0.567	**0.740**
NMEEF-SD	4.45	2.59	3.748	0.113	**0.820**	0.824
NMEEF-SD+Exceptions	11.35	7.58	**4.522**	**0.126**	0.788	**0.864**

Para el análisis y comparación de los resultados de los distintos algoritmos, se ha utilizado el test de Wilcoxon [15]. Se ha utilizado un nivel de confianza de $\alpha = 0{,}10$ en todos los experimentos. La tabla 3 muestra los resultados del test de Wilcoxon para cada medida de calidad con los correspondientes $p - val$, y el resultados de la *Hipótesis*. En general, los resultados obtenidos muestran diferencias significativas en la mayoría de las medidas de calidad a favor de los resultados una vez aplicado el enfoque de post-procesamiento propuesto.

Como muestran las tablas 2 y 3, los resultados después de aplicar el post-procesamiento mejoran los obtenidos por los algoritmos de SD. Se obtienen pequeñas reducciones de los valores de sensibilidad de los algoritmos originales

Tabla 3. Test de Wilcoxon para la comparación de Apriori-SD/NMEEF-SD+Excepciones Vs. Apriori-SD/NMEEF-SD

Algoritmo		$p-val$	*Hipótesis*
Apriori-SD	REL	0.092	Rechazado por Apriori-SD+Exceptions
	ATI	0.030	Rechazado por Apriori-SD+Exceptions
	SEN	0.017	Rechazado por Apriori-SD
	CNF	0.025	Rechazado por Apriori-SD+Exceptions
NMEEF-SD	REL	0.021	Rechazado por NMEEF-SD+Exceptions
	ATI	0.028	Rechazado por NMEEF-SD+Exceptions
	SEN	0.018	Rechazado por NMEEF-SD
	CNF	0.012	Rechazado por NMEEF-SD+Exceptions

en comparación con los resultados de esta propuesta en todos los experimentos. Puesto que la sensibilidad cuantifica la relación de ejemplos cubiertos por la variable objetivo, el subgrupo con excepciones podrá obtener como máximo el mismo resultado que el subgrupo inicial; por tanto, no se pueden mejorar los resultados en sensibilidad ya que los subgrupos modificados sólo miden los ejemplos del valor de la variable objetivo del subgrupo inicial. Para el resto de medidas de calidad, el uso de excepciones mejora la precisión y el interés de los subgrupos iniciales.

5. Conclusiones

En este trabajo se ha presentado un algoritmo de post-procesamiento basado en un EFS multiobjetivo que permite mejorar los subgrupos obtenidos por cualquier algoritmo de SD. El objetivo es detectar excepciones dentro de los subgrupos, con dos ideas: por un lado, describir pequeños espacios en los datos con un comportamiento distinto dentro del subgrupo; y por otro, incrementar la precisión de los subgrupos al detectar ejemplos dentro de los subgrupos que pueden ser interesantes para los expertos.

El estudio experimental desarrollado, soportado por tests estadísticos, muestra que la aplicación del algoritmo de post-procesamiento a los resultados obtenidos por un algoritmo de SD permite que los valores de las medidas de calidad (confianza y sensibilidad) de los subgrupos modificados con las excepciones encontradas mejoren respecto a los obtenidos por los subgrupos iniciales. Además, no sólo se mejoran los valores de las medidas de calidad utilizadas en el proceso evolutivo, sino también otras medidas de calidad consideradas en SD. El algoritmo es plenamente aplicable a problemas reales en los que los expertos necesiten obtener información para mejorar la descripción de los datos disponibles.

Reconocimientos

Este trabajo ha sido parcialmente financiado por el Ministerio de Economía y Competitividad bajo el proyecto TIN2012-33856 (Fondos FEDER), y por el Plan Andaluz de Investigación bajo el proyecto TIC-3928 (Fondos FEDER).

Referencias

1. R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A.I. Verkamo, *Fast discovery of association rules*, Advances in Knowledge Discovery and data mining (U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, eds.), AAAI Press, 1996, pp. 307–328.
2. J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, *KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework*, Journal of Multiple-Valued Logic and Soft Computing **17** (2011), no. 2-3, 255–287.
3. C. J. Carmona, P. González, M. J. del Jesus, and F. Herrera, *NMEEF-SD: Non-dominated Multi-objective Evolutionary algorithm for Extracting Fuzzy rules in Subgroup Discovery*, IEEE Transactions on Fuzzy Systems **18** (2010), no. 5, 958–970.
4. K. Deb, A. Pratap, S. Agrawal, and T. Meyarivan, *A fast and elitist multiobjective genetic algorithm: NSGA-II*, IEEE Transactions Evolutionary Computation **6** (2002), no. 2, 182–197.
5. M. J. del Jesus, P. González, F. Herrera, and M. Mesonero, *Evolutionary Fuzzy Rule Induction Process for Subgroup Discovery: A case study in marketing*, IEEE Transactions on Fuzzy Systems **15** (2007), no. 4, 578–592.
6. D. E. Goldberg, *Genetic Algorithms in search, optimization and machine learning*, Addison-Wesley Longman Publishing Co., Inc., 1989.
7. F. Herrera, *Genetic fuzzy systems: taxomony, current research trends and prospects*, Evolutionary Intelligence **1** (2008), 27–46.
8. F. Herrera, C. J. Carmona, P. González, and M. J. del Jesus, *An overview on Subgroup Discovery: Foundations and Applications*, Knowledge and Information Systems **29** (2011), no. 3, 495–525.
9. F. Hussain, H. Liu, E. Suzuki, and H. Lu, *Exception rule mining with a relative interesting measure*, Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining, LNAI, vol. 1805, 2000, pp. 86–97.
10. B. Kavsek and N. Lavrac, *APRIORI-SD: Adapting association rule learning to subgroup discovery*, Applied Artificial Intelligence **20** (2006), 543–583.
11. W. Kloesgen, *Explora: A Multipattern and Multistrategy Discovery Assistant*, Advances in Knowledge Discovery and Data Mining, American Association for Artificial Intelligence, 1996, pp. 249–271.
12. N. Lavrac, P. A. Flach, and B. Zupan, *Rule Evaluation Measures: A Unifying View*, Proceedings of the 9th International Workshop on Inductive Logic Programming, LNCS, vol. 1634, Springer, 1999, pp. 174–185.
13. N. Lavrac, B. Kavsek, P. A. Flach, and L. Todorovski, *Subgroup Discovery with CN2-SD*, Journal of Machine Learning Research **5** (2004), 153–188.
14. E. Suzuki, *Data mining methods for discovering interesting exceptions from an unsupervised table*, Journal of Universal Computer Science **12** (2006), no. 6, 627–653.
15. F. Wilcoxon, *Individual comparisons by ranking methods*, Biometrics **1** (1945), 80–83.
16. S. Wrobel, *An Algorithm for Multi-relational Discovery of Subgroups*, Proceedings of the 1st European Symposium on Principles of Data Mining and Knowledge Discovery, LNAI, vol. 1263, Springer, 1997, pp. 78–87.

Modelo Lingüístico de Evaluación de QoS de Red basado en Distancias

M. Espinilla[1], S. Gramajo[2]

[1]Departamento de Informática, Universidad de Jaén
Campus Las Lagunillas, Andalucía, Jaén – España (23071)

mestevez@ujaen.es

[2] Grupo de Investigación sobre Inteligencia Artificial. Universidad Tecnológica Nacional – Facultad Regional Resistencia. French 414 Resistencia, Chaco - Argentina (3500)

sergio@frre.utn.edu.ar

Resumen. Uno de los propósitos buscados para los procesos de Calidad de Servicio (QoS) en redes de ordenadores es establecer la prioridad de los servicios de red usados dentro de una organización en una situación determinada. Ante la previsión de un nuevo escenario, es necesario realizar una evaluación del mismo por un grupo evaluador para establecer el conjunto de prioridades de servicios de red de acuerdo a su nivel de importancia para el nuevo escenario. El proceso de asignación de prioridades a los servicios de red generalmente involucra subjetividad e incertidumbre por lo que el uso de etiquetas lingüísticas es adecuado para su valoración. En base a las valoraciones individuales, es calculada la Configuración de Servicios de Red Colectiva (CSR-C) que debe ser implementada y probada en la organización para el nuevo escenario, lo cual conlleva una serie inconvenientes. En determinadas ocasiones puede que dichos inconvenientes no rindan el suficiente beneficio, ya que puede existir en la organización una configuración similar o suficientemente cercana a la CSR-C obtenida que ofrezca unos niveles de garantía de calidad suficientes para la nueva situación. En esta contribución, proponemos un nuevo modelo de evaluación de QoS para servicios de red basado en distancias que solicita al grupo evaluador un umbral de cercanía factible. Dicho umbral representa el margen de tolerancia de las prioridades proporcionadas por cada evaluador para garantizar la QoS de red para el nuevo escenario. A partir de las evaluaciones individuales se obtiene la configuración adecuada, teniendo en consideración las configuraciones ya implementas en la organización, la CSR-C y el umbral de cercanía factible colectivo.

1 Introducción

Uno de los procesos clave en el ámbito de Calidad de Servicio (QoS) en la red de una organización es el proceso de establecer la prioridad de los servicios de red usados

por la misma [1][2]. El objetivo de dicho proceso es intentar mejorar la transmisión de datos en el tiempo, garantizando cierto nivel de calidad y prioridad a cada servicio de red [3]. Dichos servicios pueden denominarse Tipos de Tráfico (ToTs) y su importancia radica en el hecho de que en determinadas situaciones, existen algunos de ellos que son críticos para el funcionamiento de una organización mientras que otros pueden no ser tan importantes o tener una importancia relativa. Así, en una organización existen un conjunto de Configuraciones de Servicios de Red establecidas (CSRs), cada una de ellas asociada a una situación concreta o escenario. Los administradores de red son los encargados de activar la CSR apropiada en función del escenario por el que esté pasando la organización y así adecuar la prioridad de cada ToT a las necesidades específicas del escenario.

El modelo presentado en [4] ha dado muy buenos resultados a la hora de establecer la CSR de un escenario en una organización. En dicho modelo, la CSR es fijada calculando la prioridad para cada ToT de la organización, teniendo en cuenta tanto las valoraciones de los administradores de la red como de los usuarios que la utilizan. Dado que la información involucrada en el proceso de valoración de prioridades presenta vaguedad e imprecisión, dicho modelo propone expresar las preferencias a través de términos lingüísticos [5], ofreciendo un marco de evaluación adecuado que facilita y mejora la resolución del problema de asignación de prioridades.

A partir de las valoraciones del grupo evaluador es calculada la CSR colectiva (CSR-C) para la situación bajo estudio, la cual se considera la apropiada. A partir de ahí, los administradores de red realizan el proceso de implementación y pruebas de tal configuración. Básicamente, la implementación consiste en asignar a cada ToT la prioridad fijada y, posteriormente, realizar un conjunto de pruebas con el objetivo de testear la configuración. Dicho proceso implica una serie de inconvenientes asociados a la organización, por lo que la decisión de implementar una nueva configuración no es trivial [6]. Por un lado, los administradores deben asumir la tarea de implementar y probar una nueva configuración y, por otro lado, los usuarios de la red padecen los inevitables perjuicios que origina el probar una nueva configuración en la red: imposibilidad de usar algunos servicios, lentitud en la red o en determinados servicios críticos, etc.

En determinadas situaciones puede ocurrir que los perjuicios de implementar y probar la CSR-C obtenida no rinda el suficiente beneficio para la organización. Este hecho se debe a que puede existir en el conjunto de CSRs de la organización una configuración similar o suficientemente cercana a la CSR-C obtenida, la cual ofrezca unos niveles de garantía la servicios de red suficiente para el nuevo escenario.

Como solución a este problema, en esta contribución se propone que ante un nuevo escenario los evaluadores proporcionen junto a las prioridades de los ToT un umbral de cercanía factible (*ucf*). Dicho umbral representa el margen de tolerancia de las prioridades proporcionadas por cada evaluador para garantizar una QoS de red suficiente en el nuevo escenario. El modelo de evaluación computará un umbral de cercanía factible colectivo (*ucf -c*), el cual será utilizado para escoger entre la CSR-C o la CSR más próxima a la CSR-C. Si esta última cubre una QoS de red suficente, será la configuración escogida para el nuevo escenario y se evitarán los perjuicios de desarrollar una nueva configuración. Dado que el *ucf* es un concepto que también

presenta vaguedad e imprecisión, se propone que sea valorado a través de etiquetas lingüísticas, al igual que las prioridades de los ToTs.

Para llevar a cabo nuestra propuesta, la contribución se estructura del siguiente modo: la Sección 2 revisa conceptos necesarios para la correcta comprensión de nuestra propuesta. La Sección 3 presenta el nuevo modelo lingüístico de evaluación basado en distancias para QoS de red. La Sección 4 presenta un ejemplo ilustrativo del modelo propuesto. Por último, el trabajo finaliza en la Sección 5 con las conclusiones.

2 Preliminares

En esta sección, revisamos diferentes conceptos y herramientas que son utilizadas en nuestra propuesta. Inicialmente revisamos el esquema de análisis de decisión en el cual se basará el modelo lingüístico de evaluación de QoS de red basado en distancias. Posteriormente, revisamos el enfoque lingüístico difuso y el modelo de representación de información lingüística difusa 2-tupla, los cuales serán utilizados para manejar la información en el modelo propuesto en esta contribución.

2.1 Análisis de Decisión y Evaluación

El análisis de decisión es una disciplina perteneciente a la teoría de decisión que ha proporcionado resultados exitosos en diferentes problemas de evaluación como: evaluación sensorial o evaluación de desempeño [7][8][9]. El objetivo principal de la evaluación es sintetizar la información involucrada en el proceso en una valoración que ofrezca información útil sobre el elemento evaluado. Para conseguir tal fin, es necesario establecer un marco de evaluación donde se fijan los diversos elementos que intervienen en el proceso, se recoge la información y, finalmente, se calcula la valoración para cada elemento evaluado. El análisis de decisión es una excelente herramienta para los procesos de evaluación, ya que proporciona una gran variedad de métodos para evaluar alternativas, considerando los criterios relevantes en el problema y que puede tener en consideración la opinión de múltiples expertos [10]. El esquema general que sigue el modelo que presentamos en esta contribución es mostrado en la Figura 1, el cual corresponde al proceso de análisis de decisión.

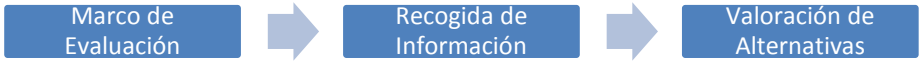


Fig. 1. Fases del Proceso de Análisis a la Decisión

2.2 Enfoque Lingüístico Difuso

La información involucrada en el proceso de evaluación de QoS de red presenta vaguedad e incertidumbre, ya que la información está basada en el conocimiento y en la percepción que los administradores y usuarios de la red tienen sobre ella.

El enfoque lingüístico difuso [5] ha dado muy buenos resultados a la hora de modelar dicha información. Para ello, representa la información vaga e imprecisa a través de variables lingüísticas. Cuando se trabaja con información lingüística es necesario seleccionar el conjunto de etiquetas lingüística y su semántica y debe ser tenido en cuenta la granularidad de la incertidumbre, es decir, la capacidad de discriminación entre los distintos valores de información. Valores típicos de granularidad son valores impares, tales como 7 ó 9, donde la etiqueta central representa una valoración de aproximadamente 0.5 y el resto de etiquetas se encuentran distribuidas simétricamente a su alrededor. Una posibilidad es generar todas las etiquetas lingüísticas sobre una escala en la cual se define un orden total y la semántica viene determinada por números difusos definidos en el intervalo [0,1] [11]. Un ejemplo de conjunto de 7 términos lingüísticos es ilustrado en la Figura 2.

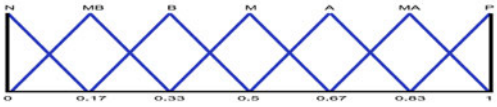


Fig. 2. Conjunto de 7 Términos Lingüísticos

2.3 Modelo Lingüístico 2-Tupla

El uso de variables lingüísticas implica realizar procesos de computación con palabras. El modelo de representación lingüístico basado en 2-tuplas [12] mejora la precisión en los procesos de computación con palabras respecto a otros modelos [13][14]. Este modelo se basa en el concepto de traslación simbólica y representa la información a partir de una 2-tupla (s_i, α), donde $s_i \in S = \{s_0, \ldots, s_g\}$ es un término lingüístico y $\alpha \in [-0.5, 0.5)$ es un valor numérico que representa la translación simbólica.

Definición 1.Sea $\beta \in [0, g]$ el valor obtenido en una operación simbólica, $S = \{s_0, \ldots, s_g\}$ un conjunto de términos lingüísticos y $\bar{S} = S \times [-0.5, 0.5)$ el conjunto de 2-tuplas asociadas a S, la información equivalente a β se obtiene a través de la función $\Delta(\beta): [0, g] \rightarrow \bar{S}$ definida por:

$$\Delta(\beta) = (s_i, \alpha), \text{con} \begin{cases} s_i, i = \text{round}(\beta) \\ \alpha = \beta - i, \end{cases} \quad (1)$$

Donde *round* es el operador usual de redondeo que asigna el número entero más cercano a β. Conviene señalar que Δ es biyectiva, siendo la función $\Delta^{-1}(\bar{p}) = \Delta^{-1}(s_i, \alpha) : \bar{S} \rightarrow [0, g]$ definida como $\Delta^{-1}(s_i, \alpha) = i + \alpha$.

Comentario 1. A partir de las funciones anteriores, la conversión de un término lingüístico en una 2-tupla consiste en añadir el valor cero como traslación simbólica: $H: s_i \in S \rightarrow (s_i, 0) \in \bar{S}$.

El modelo de representación basado en 2-tupla tiene asociado un modelo computacional que permite realizar procesos de computación con palabras sin pérdida de información. En la literatura podemos encontrar que han sido extendidos diversos operadores de agregación y distancias para 2-tuplas [15]. A continuación, se define el operador de agregación de media ponderada para 2-tuplas lingüísticas, el cual será utilizado en esta contribución.

Definición 2. Sea $A = \{(r_1, \alpha_1), \ldots, (r_n, \alpha_n)\}$ un conjunto de 2-tuplas lingüísticas y $W = \{w_1, \ldots, w_n\}$ un vector numérico con sus pesos asociados, tal que $\sum_{j=1}^{n} w_j = 1$ y $w_j \in [0,1]$, la 2-tupla que simboliza la media ponderada $\bar{x}^w$ se define como:

$$\bar{x}^w\big((r_1, \alpha_1), \ldots, (r_n, \alpha_n)\big) = \Delta\Big(\sum_{j=1}^{n} w_j \Delta^{-1}(s_j, \alpha_j)\Big)$$

3 Modelo lingüístico de evaluación de QoS de red basado en distancias

En esta sección, presentamos el modelo lingüístico de evaluación de QoS de red basado en distancias para evaluar un nuevo escenario en una compañía y obtener la configuración de servicios de red adecuada, considerando el umbral de cercanía factible. El esquema de dicho modelo es el siguiente:

Fase 1 – Establecimiento del marco de evaluación:
- Identificar evaluadores, configuraciones de servicios de red establecidas (CSRs) y tipos de tráfico (ToTs). Establecer el nuevo escenario objeto de evaluación. Establecer conjunto de términos lingüísticos.

Fase 2- Recogida de las valoraciones individuales
- Prioridades de los ToT para el nuevo escenario y umbral de cercanía factible para el nuevo escenario

Fase 3 – Elección de la configuración adecuada
- Agregación de las valoraciones individuales para el nuevo escenario
  - Calcular la CSR colectiva (CSR-C)
  - Calcular umbral de cercanía factible colectivo (ucf-c)
- Calculo de la distancia entre las CSRs y la CSR-C
- Selección de la CSR adecuada para el nuevo escenario

En las siguientes secciones se describen cada una de las fases mencionadas en detalle.

3.1 Establecimiento del marco de evaluación

En esta fase se fijan todos los elementos que intervienen en el proceso de evaluación de QoS de red para un nuevo escenario.

La compañía cuenta con un conjunto finito de configuraciones de servicio de red $X = \{CSR_i, i = 1, \ldots, m\}$ que han sido implementadas y probadas. Cada CSR_i se caracteriza por un vector que contiene la prioridad de cada ToT $CSR_i = \{\bar{p}_{i1}, \ldots, \bar{p}_{in}\}$, siendo $\bar{p}_{ij}$ la prioridad del ToT t_j, expresada en 2-tuplas, $\bar{p}_{ij} = (s_l, \alpha)_{ij}$ donde $s_l \in S = \{s_0, \ldots, s_g\}$ y $\alpha \in [-0.5, 0.5)$.

Además, se establece el nuevo escenario objeto de evaluación A, y el conjunto de ToT de la organización $T = \{t_j, j = 1, \ldots, n\}$, cuya prioridad será valorada por el conjunto de evaluadores $E = \{e^k, k = 1, \ldots, z\}$ en el conjunto de términos lingüísticos $S = \{s_0, \ldots, s_g\}$. El grupo de evaluadores estará compuesto por administradores y usuarios de la red en la organización

3.2 Recogida de las valoraciones individuales

Una vez que se ha fijado el marco de evaluación, se obtienen las opiniones del conjunto de evaluadores. Para ello, cada evaluador e^k , proporciona la prioridad del tipo de tráfico t_j, del nuevo escenario A, en una etiqueta lingüística del conjunto de términos lingüísticos $S = \{s_0, \ldots, s_g\}$. Así para cada evaluador e^k, se obtendrá el siguiente vector de utilidad $A^k = \{p_{a1}^k, \ldots, p_{an}^k\}$, $p_{aj}^k \in S$.

Además, es necesario recoger la opinión de cada evaluador sobre el umbral de cercanía factible de las prioridades proporcionadas, es decir, el margen o tolerancia para obtener una configuración similar o suficientemente cercana que garantice la QoS de la red para el nuevo escenario. Así, cada evaluador e^k, proporciona su valoración sobre el umbral de cercanía factible en el conjunto de términos lingüísticos fijado en el marco de evaluación, $ucf^{\,k} \in S$.

3.3 Elección de la configuración adecuada

En esta fase es seleccionada la CSR adecuada para el nuevo escenario, teniendo en consideración los umbrales de cercanía factibles proporcionados por los evaluadores. Para ello, primero, en base a las valoraciones individuales, se calcula una CSR colectiva (CSR-C) para el nuevo escenario y el umbral de cercanía factible colectivo (*ucf-c*). A continuación, se calcula la distancia entre el conjunto de CSRs establecidas en la compañía y la CSR-C con el objetivo de buscar la configuración más cercana a la colectiva. Finalmente, se compara la distancia de dicha configuración con el *ucf-c* para escoger entre la CSR-C o la CSRs más cercana a la colectiva (CSR_{min}). A continuación, se describen en detalle cada uno de estos pasos.

3.3.1 Agregación de las valoraciones individuales

En esta etapa se realizan los procesos computacionales para obtener una prioridad colectiva para cada ToT t_j en el nuevo escenario A, éstas definirán la CSR-C para el nuevo escenario. Además, en esta etapa es calculado el *ucf-c* que representará la tolerancia o margen máximo para que una CSR establecida en la compañía sea escogida en lugar de la CSR-C. Para llevar a cabo los procesos computacionales, primero es necesario transformar las valoraciones individuales en términos lingüísticos en 2-tuplas lingüísticas, siguiendo el Comentario 1, $\bar{A}^k = \{\bar{p}_{a1}^k, \ldots, \bar{p}_{an}^k\}$, $\overline{ucf}^{\,k}$.

A continuación son seleccionados operadores de agregación para 2-tuplas para obtener una prioridad colectiva $\bar{\imath}_{aj}$ correspondiente al ToT t_j y al *ucf-c*.

$$CSR - C = (\bar{\imath}_{a1}, \ldots, \bar{\imath}_{an}) = \{AG(\bar{p}_{a1}^1, \ldots, \bar{p}_{a1}^z), \ldots, AG(\bar{p}_{an}^1, \ldots, \bar{p}_{an}^z)\}, \quad (2)$$

$$\overline{ucf - c} = AG\{\overline{ucf}^{\,1}, \ldots, \overline{ucf}^{\,z}\} \quad (3)$$

3.3.2 Calculo de la distancia entre CSR-C y CSRs

El objetivo de esta etapa es calcular la distancia entre la $CSR - C$ para el nuevo escenario y las CSRs en la compañía. Dicha distancia permitirá calcular la cercanía en 2-tupla entre cada CSR desarrollada por la compañía previamente y la configuración

colectiva obtenida para el nuevo escenario en la fase anterior. La distancia entre dos 2-tuplas es definida como [16]:

$$\bar{d}_{CSR_i=} d(CSR - C, CSR_i) = \Delta\left(\sum_{j=1}^{n} \left|\Delta^{-1}(\bar{r}_{aj}) - \Delta^{-1}(\bar{p}_{aj})\right|\right) \in \bar{S} \quad (4)$$

3.3.3 Selección de la configuración adecuada.

En esta fase es seleccionada la configuración adecuada al escenario objeto de evaluación en la compañía con el fin de escoger entre la $CSR - C$ o la CSR más cercana (CSR_{min}), cuando esta última cubra la garantías de QoS de la red para el nuevo escenario. Para ello, primero se selecciona la CSR que tenga la menor distancia con la $CSR - C$.

$$CSR_{min} = CSR_i \mid \bar{d}_{min} = \bar{d}_{CSR_i} = Min(\bar{d}_{CSR_1}, \ldots, \bar{d}_{CSR_m}) \quad (5)$$

A continuación, es comparada la distancia mínima $\bar{d}_{min}$, con el umbral de cercanía factible colectivo $\overline{ucf - c}$, calculado previamente.

Si la distancia mínima de la CSR_{min} rebasa el *ucf-c* , $\bar{d}_{min} > \overline{ucf - c}$, la configuración adecuada para el nuevo escenario corresponde a $CSR - C$. En este caso, se implementará y se realizarán las pruebas necesarias para desarrollar tal configuración en la compañía, ya que no existe en la organización ninguna configuración que satisfaga la QoS de la red para este nuevo escenario.

En caso contrario, es decir, que la distancia mínima de la CSR_{min} se encuentre por debajo del *ucf-c* , $\bar{d}_{min} \leq ucf - c$, la configuración adecuada para el nuevo escenario corresponde a la CSR_{min}. En este caso, dentro de la organización se ha seleccionado la configuración más cercana a la $CSR - C$ que cubre los niveles de QoS de la red adecuados para la nueva situación. Por tanto, no será necesario desarrollar una nueva configuración, evitando los problemas asociados.

4 Ejemplo Ilustrativo

En esta sección se presenta un ejemplo ilustrativo del modelo lingüístico de evaluación de QoS de red basado en distancias. El ejemplo ilustrativo está centrado en la evaluación del escenario *Jornadas* en un centro educativo. A continuación, se describen cada uno de las fases realizadas para llevar a cabo tal evaluación.

4.1 Establecimiento del Marco de Evaluación

Suponemos un centro educativo donde coexisten diferentes tipos de usuarios y administradores de red. Los usuarios usan una red LAN que tiene un enlace a Internet por el cual los diversos usuarios utilizan los servicios de red que necesitan. Dicho centro posee tres configuraciones de servicios de red establecidas en su organización $X = \{CSR_1, CSR_2, CSR_3\}$, que corresponde respectivamente a: *horario escolar*, *hora-*

rio extraescolar, *horario fin de semana*. Las configuraciones están caracterizadas por la prioridad asignada a los cuatro ToTs $T = \{t_1, t_2, t_3, t_4\}$ que corresponden a *transferencia de datos*, *mensajería*, *multimedia*, *videoconferencia*. Las prioridades de los ToTs de cada configuración están establecidas en el conjunto de términos lingüísticos mostrados en la Figura 2, siendo sus prioridades mostradas a continuación.

Tabla 1. CSR Establecidas

	CSR_1	CSR_2	CSR_3
t_1	$(s_5, 0)$	$(s_4, 0)$	$(s_2, 0)$
t_2	$(s_5, 0)$	$(s_4, 0)$	$(s_3, 0)$
t_3	$(s_3, 0)$	$(s_6, 0)$	$(s_5, 0)$
t_4	$(s_2, 0)$	$(s_5, 0)$	$(s_6, 0)$

Para valorar al nuevo escenario *Jornadas*, A, se va a contar con la opinión de 3 evaluadores $E = \{e_1, e_2, e_3\}$ que corresponde a un administrador de la red y a dos usuarios de la misma, respectivamente.

4.2 Recogida de las valoraciones individuales

En esta fase son recogidas las valoraciones de los evaluadores sobre el nuevo escenario A, las cuales son mostradas en Tabla 2.

Tabla 2. Valoraciones en 2-tuplas del nuevo escenario A

Escenario A	e_1	e_2	e_3
t_1	$(s_5, 0)$	$(s_5, 0)$	$(s_4, 0)$
t_2	$(s_6, 0)$	$(s_5, 0)$	$(s_5, 0)$
t_3	$(s_4, 0)$	$(s_3, 0)$	$(s_3, 0)$
t_4	$(s_2, 0)$	$(s_1, 0)$	$(s_2, 0)$
ucf	$(s_0, 0)$	$(s_1, 0)$	$(s_1, 0)$

4.3 Elección de la configuración adecuada

En esta fase se realizarán los pasos necesarios para obtener la configuración adecuada para el nuevo escenario.

4.3.1 Agregación de las valoraciones individuales.

Una vez que las valoraciones individuales se han obtenido, se calcula la CSR-C para el escenario A y el *ucf-c*. Para ello, es necesario seleccionar un operador de agregación para 2-tuplas lingüística. En este ejemplo ilustrativo se ha seleccionado el operador de agregación media ponderada para 2-tupla, definición 1, con el objetivo de dotar al administrador de la red e_1 , un mayor peso en la configuración que al resto de evaluadores $\{e_2, e_3\}$. Así, el vector de pesos utilizado es $w = \{0.4, 0.3, 0.3\}$ y las valoraciones colectivas obtenidas son las siguientes:

$$CSR - C = \{(s_5, -0.3)_{a1}, (s_5, 0.4)_{a2}, (s_3, 0.4)_{a3}, (s_2, -0.3)_{a4}\}$$

$$\overline{ucf - c} = (s_1, -0.4)$$

4.3.2 Calculo de la distancia entre CSR-C y CSRs

En esta fase, se calcula la cercanía de las tres CSR desarrolladas por el centro educativo previamente (ver Tabla 1) y la CSR-C obtenida en la fase anterior para el escenario A, usando la ecuación (5). Las distancias para las tres CSRs son las siguientes:

$$\bar{d}_{CSR_1=} \, d(CSR - C, CSR_1) = (s_0, 0.35)$$

$$\bar{d}_{CSR_2=} \, d(CSR - C, CSR_2) = (s_2, 0)$$

$$\bar{d}_{CSR_3=} \, d(CSR - C, CSR_3) = (s_3, -0.25)$$

4.3.3 Selección de la configuración adecuada

En esta última etapa se identifica a CSR_{min} , es decir, la CSR más cercana a la CSR-C. En este ejemplo, siguiendo la ecuación (6), CSR_{min} corresponde a CSR_1 .

$$CSR_{min} = CSR_1 \, ; \bar{d}_{min} \quad = (s_0, 0.35) = Min((s_0, 0.35), (s_2, 0), (s_3, -0.25))$$

A continuación, se compara la distancia de CSR_{min} con el *ucf-c* para conocer si satisface la QoS de red del escenario A. En este ejemplo, la distancia de la CSR_{min} se encuentra por debajo del *ucf-c*, ya que $(s_0, 0.35) \leq (s_1, -0.4)$. Por tanto, la CSR_1 es la propuesta para el escenario *Jornadas*. Dicha configuración satisface la QoS de red de los usuarios e implicará que sean evitados los inconvenientes asociados al desarrollo de una nueva configuración en la organización.

5 Conclusiones

En una configuración de servicios de red (CSR) se asigna a cada tipo de tráfico (ToT) una prioridad en base a las opiniones proporcionadas por un grupo de evaluadores. Dicho proceso envuelve incertidumbre e imprecisión, por lo que es adecuado el uso de etiquetas lingüísticas para afrontar este problema. El desarrollo y las pruebas necesarias para implantar una nueva CSR conllevan una serie de inconvenientes para la compañía, por lo que la decisión de implementar una nueva configuración no es trivial. En esta contribución se ha propuesto un nuevo modelo lingüístico de evaluación de QoS de red basado en distancias que obtiene la configuración adecuada para un nuevo escenario. La novedad de dicho modelo es que solicita un umbral de cercanía factible a cada evaluador que representa el margen de tolerancia de las prioridades proporcionadas por el evaluador para garantizar la QoS de la red en el nuevo escenario. El modelo de evaluación computa un umbral de cercanía factible colectivo (*ucf -c*), el cual es utilizado para escoger entre una nueva configuración para el escenario en base a las opiniones del grupo (CSR-C) o la CSR establecida en la compañía y más próxima a la CSR-C. Esta última es escogida siempre que cubre los aspectos de garantía de QoS de la red marcados por el *ucf-c,* evitando los perjuicios a la compañía de implementar una nueva configuración.

Agradecimientos

Este trabajo está parcialmente financiado por los proyectos de investigación AGR-648 y TIN2012-31263 así como fondos del Santander Universidades.

Referencias

1. Evans, J., Filsfils, C.: Deploying IP and MPLS QOS for Multiservice Networks. Theory and Practice. Elservier (2007)
2. Szigeti, T., Hattingh, C.: End-to-End QoS Network Design: Quality of Service in LANs, WANs, and VPNs. Cisco Press, Indianapolis (2004)
3. Vegesna, S.: IP Quality of Service. Cisco Press, Indianapolis (2001)
4. Gramajo, S., Martínez, L.: A linguistic deci-sion support model for QoS priorities in networking. Knowledge-Based Systems, New Trends on Intelligent Decision Support Systems 32, 65-75 (2012)
5. Zadeh, L.: The concept of a linguistic variable and its application to approximate reasoning. Part I, Part II and Part III. Information Sciences 199–249,301–357,143–180 (1975)
6. Gheorghe, L.: Designing and Implementing Linux Firewalls with QoS using netfilter, iproute2, NAT and L7-filter. Packt Publishing, Birmingham, UK (2006)
7. Martínez, L., Espinilla, M., Pérez, L.G.: A linguistic multi-granular sensory evaluation model for olive oil International Journal of Computer Intelligence Systems 1, 148-158 (2008)
8. Martínez, L.: Sensory Evaluation Based on Linguistic Decision Analysis. International Journal of Aproximated Reasoning 44, 148-164 (2007)
9. Espinilla, M., Andrés, R.d., Martínez, F.J., Martínez, L.: A 360-degree performance appraisal model dealing with heterogeneous information and dependent criteria. Information Sciences 222, 459-471 (2013)
10. Marcomini, A., Suter, G.W., Critto, A.: Decision Support Systems for Risk based Management of Contaminated Sites, New York (2009)
11. Yager, R.R.: An approach to ordinal decision making. International Joumal of Approximate Reasoning 12, 237–261 (1995)
12. Herrera, F., Martínez, L.: A 2-tuple Fuzzy Linguistic Representation Model for Computing with Words. IEEE Transactions on Fuzzy Systems 8, 746-752 (2000)
13. Degani, R., Bortolan, G.: The Problem of Linguistic Approximation in Clinical Decision Making. International Journal of Approximate Reasoning 2, 143-162 (1988)
14. Delgado, M., Verdegay, J.L., Vila, M.A.: Aggregation Operations of Linguistic Labels. International Journal of Intelligent Systems 2, 351-370 (1993)
15. Herrera, F., Martínez, L.: The 2-tuple Linguistic Computational Model, Adventages of its linguistic description, accuracy and consistency. Int.J. of Uncertainly, Fuzziness and Knowledge-Based Systems 9, 33-48 (2001)
16. Wei, G.-W.: Extension of TOPSIS method for 2-tuple linguistic multiple attribute group decision making with incomplete weight information. Knowledge and Information Systems 25, 623-634 (2010)

A Fuzzy Approach to the Weighted Overlap Dominance Model*

Camilo Franco, Jens Leth Hougaard, and Kurt Nielsen

Department of Food and Resource Economics, Faculty of Science, University of Copenhagen, Denmark

Abstract. Decision support models are required to handle the various aspects of multi-criteria decision problems in order to help the individual understand its possible solutions. In this sense, such models have to be capable of aggregating and exploiting different types of measurements and evaluations in an interactive way, where input data can take the form of uniquely-graded or interval-valued information. Here we explore the Weighted Overlap Dominance (WOD) model from a fuzzy perspective and its outranking approach to decision support and multidimensional interval analysis. Firstly, imprecision measures are introduced for characterizing the type of uncertainty being expressed by intervals, examining at the same time how the WOD model handles both non-interval as well as interval data, and secondly, relevance degrees are proposed for obtaining a ranking over the alternatives. Hence, a complete methodology is presented for ordering and identifying the best alternatives under an interactive procedure that takes into account the natural imprecision and relevance of information.

Keywords: Weighted overlap dominance, uncertainty due to imprecision, relevance ranking

1 Introduction

An interactive outranking methodology has been previously presented [8], named the Weighted Overlap Dominance (WOD) model, for ordering a set of alternatives based on mutidimensional interval data. This model proposes a volume-based methodology for dealing with interval information, where the score of an alternative regarding some criterium is represented by an interval of values. Overall, the WOD procedure develops pairwise comparisons between alternatives regarding multiple criteria, and identifies the underlying outranking and indifference relations according to the specification of some user-defined parameters. These parameters can be updated or tuned up according to the individual's beliefs on the outranking thresholds and the trade-off between possible better and worse scores. Hence, the WOD proposal allows exploiting the flexibility of

* Financial support from the Center for Research in the Foundations of Electronic Markets (CFEM), funded by the Danish Council for Strategic Research, is gratefully acknowledged.

an interval representation of knowledge, belonging to the outranking literature (see e.g., [12]) under the important distinction that it is directly built for treating interval information, unlike other outranking procedures that can be extended to do so (see e.g., [3,9]).

Here we explore a fuzzy approach on the WOD model, based on fuzzy set theory [6,14], focusing on the characterization of *uncertainty due to imprecision*. Such imprecision or vagueness refers to a basic type of uncertainty where the quality of the available knowledge justifies the use of intervals (see e.g., [7], but also [1]). Then, dealing with common applied problems, it may be the case that the available knowledge takes the form of one unique value, being necessary to understand how the model deals with both types of precise and imprecise data.

Therefore, in this paper we enhance the capability of the WOD procedure for decision support, offering a solid foundation for examining the type of uncertainty that intervals represent, such that it is able to take into account the various forms of precise and imprecise knowledge. Besides, we explore its results regarding the outranking order that it proposes, pointing out some difficulties that rise if the alternatives are to be ranked or weakly ordered. Such difficulties refer to the *semi-transitivity* of its *outranking* relation and the *non-transitivity* of its *indifference* relation. Then, in order to arrive to a recomended course of action, i.e., a ranking of alternatives, we propose the use of *relevance degrees*. Such degrees are introduced here following the general approach of [5], complementing the WOD procedure in such a way that it is possible to obtain a simple solution every time the decision maker updates his beliefs.

This paper is organized as follows. First, we review the WOD model, presenting its basic definitions and methodology. Then, we explore such methodology under a fuzzy approach, focusing on its manipulation of imprecision by non-interval and interval data. Following this, some properties for the WOD outranking and indifference relations are presented, and relevance measures are proposed for arriving to a ranking of alternatives. We end with some final comments for future research.

2 The WOD outranking methodology

Here we review the WOD methodology as initially presented in [8]. First of all, consider a set of alternatives $N = \{1, ..., n\}$ and a set of criteria $M = \{1, ..., m\}$, such that every alternative $a \in N$ receives a score according to their satisfaction of criterium $j \in M$, given by $x_{aj} = \left[x_{aj}^L, x_{aj}^U\right]$, where $x_{aj}^L \leq x_{aj}^U$. Then, considering the m criteria that are used for evaluating each alternative, the input data consist of m-dimensional cubes $c_a = \left[x_{aj}^L, x_{aj}^U\right]^m$ containing the interval degrees for every alternative in N.

Now, the WOD procedure needs the following definitions in order to identify the outranking and indifference relations for any pair of alternatives $a, b \in N$. First, define a vector of weights $w \in \mathbb{R}_+^m$ according to the importance of the criteria, and label the alternatives such that $w \cdot x_a^U \geq w \cdot x_b^U$, where $x_a^U = \left(x_{a1}^U, ..., x_{am}^U\right)$. Then, regarding the pairwise comparisons between multidimen-

sional interval scores, define the three following sets: one containing the subintervals that are greater than the upper bound of b,

$$\hat{Z}(a,b) = \left\{ i \in c_a | w \cdot i > w \cdot x_b^U \right\}, \tag{1}$$

another one containing the subintervals that are less than the lower bound of b,

$$\check{Z}(a,b) = \left\{ i \in c_a | w \cdot i < w \cdot x_b^L \right\}, \tag{2}$$

and the last one containing the subintervals that completely overlap,

$$\tilde{Z}(a,b) = \left\{ i \in c_b | w \cdot i > w \cdot x_a^L \right\}. \tag{3}$$

Finally, let V be some volume operator such that $V(\emptyset) = 0$.

In this way, in the case that $\tilde{Z} = \emptyset$, it holds that a outranks b ($a \succ b$), given that a strict *dominance* exists of a over b. Here we say that one cube c_a *dominates* another cube c_b if it is true that $x_a^L > x_b^U$.

Else, in the case that $\check{Z} = \emptyset$, the outranking situation of $a \succ b$ holds if and only if the following expression holds,

$$P(a,b) = \frac{V\left(\hat{Z}\right)}{V(c_a)} + \frac{V\left(c_a \setminus \hat{Z}\right)}{V(c_a)} \frac{V\left(c_b \setminus \tilde{Z}\right)}{V(c_b)} > \beta, \tag{4}$$

where the favourable information for preferring a over b, given by the aggregation of the different ratios in (4), is evaluated according to $\beta \in [0,1]$. In this case, both alternatives are considered indifferent ($a \sim b$), if and only if $P(a,b) \leq \beta$.

In the last case, if it holds that $\check{Z} \neq \emptyset$, then the ratio between (1) and (2) is examined,

$$\frac{V\left(\hat{Z}\right)}{V\left(\check{Z}\right)}, \tag{5}$$

where it holds either that $a \succ b$, that $a \sim b$ or that $b \succ a$, if and only if it respectively holds that (5) is greater, equal or less than $\gamma \in \mathbb{R}_+$.

Hence, the alternatives may be ordered according to their outranking and indifference relations, with the interactive updating of the user-defined parameters β and γ. These parameters function as thresholds for assigning the outranking and the indifference situations under the different cases of overlapping, where γ also expresses a certain trade-off between possible better and worse scores. For more details on the WOD procedure see [8].

3 A fuzzy approach to the WOD model

3.1 The fuzzy WOD model

Here we explore the representation of the degree in which an alternative $a \in N$ satisfies a given criterium $j \in M$ by means of fuzzy sets [6,14]. In this way, being N the referential set of objects, a fuzzy set A is given by

$$A = \left\{ \langle a, \mu_A(a) \rangle | a \in N \right\}, \tag{6}$$

where $\mu_A(a) : N \to [0,1]$ is its characteristic function and $\mu_A(a) \in [0,1]$ is its membership degree, verifying up to which extent the object $a \in N$ satisfies the property represented by A.

Following this definition of fuzzy sets, a unique value is used for expressing the degree in which A is verified by some $a \in N$. Nonetheless, if such value is not unique, then interval-valued fuzzy sets can be used instead (see [7,2], but also [1,11]). In this way, an interval-valued fuzzy set Z is characterized by a mapping

$$\mu_Z(a) : N \to L([0,1]), \tag{7}$$

where $L([0,1])$ is the set of all closed sub-intervals in $[0,1]$, such that,

$$L([0,1]) = \left\{ x = \left[x^L, x^U\right] \mid \left(x^L, x^U\right) \in [0,1]^2 \text{ and } x^L \leq x^U \right\}, \tag{8}$$

and its membership degree is given by $\mu_Z(a) \in L([0,1])$.

Taking into account the framework for multi-criteria decision support and the WOD procedure, let Z_j represent the property characterizing some criterium $j \in M$, such that $\mu_{Z_j}(a) = \left[\mu^L_{Z_j}(a), \mu^U_{Z_j}(a)\right]$ is the interval degree in which the predicate "a satisfies the property Z_j" is verified. In this way, considering all of the m criteria $j \in M$, we have that the input data for the fuzzy-WOD model is given by the m-dimensional cubes $c_a = \left[\mu^L_{Z_j}(a), \mu^U_{Z_j}(a)\right]^m$, and the sets $\hat{Z}, \check{Z}$ and $\tilde{Z}$ can be respectively defined, where for any $a \in N$,

$$\mu^L(a) = \left(\mu^L_{Z_1}(a), ..., \mu^L_{Z_m}(a)\right), \tag{9}$$

$$\mu^U(a) = \left(\mu^U_{Z_1}(a), ..., \mu^U_{Z_m}(a)\right), \tag{10}$$

and

$$w \cdot \mu(a) = \left[w \cdot \mu^L(a), w \cdot \mu^U(a)\right]. \tag{11}$$

Now, the WOD model develops its procedure by means of the volume operator V, evaluating the overlap between the multidimensional cubes. In particular, checking how the procedure works in cases where data has no definite volume, like in the case of non-interval degrees, the following examples can be examined.

Example 1. Consider the following non-interval membership degrees, for two alternatives $a, b \in N$ and one criterium $j = 1$, given by

$$\mu_{Z_1}(a) = [0.9, 0.9], \quad \mu_{Z_1}(b) = [1,1].$$

First, the WOD procedure tells us to label the alternatives such that $\mu^U_{Z_1}(a) > \mu^U_{Z_1}(b)$. Hence, we have that

$$\mu_{Z_1}(a) = [1,1], \quad \mu_{Z_1}(b) = [0.9, 0.9].$$

Then, we verify that $\tilde{Z} = \emptyset$, where it holds that $a \succ b$, a result that could be expected.

Example 2. Now consider the following membership degrees for the same alternatives and criterium, but this time given by,

$$\mu_{Z_1}(a) = [0.7, 0.7], \quad \mu_{Z_1}(b) = [0.7, 0.7].$$

In this case, we have that both least and upper bounds are equal, and either way, if we compare a with b or b with a, we arrive again to the case where $\tilde{Z} = \emptyset$, obtaining that $a \succ b$ holds, which is not what would be expected.

It is only natural to expect that if the degrees for the alternatives are equal, then both alternatives should be indifferent. Example 2 suggests that there are some difficulties concerning the WOD procedure when the upper bounds of the membership degrees coincide. The next example takes a closer look at this issue.

Example 3. Consider the following membership degrees for two alternatives $a, b \in N$ and one criterium $j = 1$,

$$\mu_{Z_1}(a) = [0, 1], \quad \mu_{Z_1}(b) = [1, 1].$$

Then we have that $\mu^U_{Z_1}(a) = \mu^U_{Z_1}(b)$ and, following the WOD procedure, we can arrive to two different cases. Maintaining the current labeling of alternatives ($a = a$ and $b = b$), we have that $\tilde{Z} \neq \emptyset$ holds. In such case, we evaluate (5) and we obtain that it is equal to 0, given that $V\left(\hat{Z}\right) = V(\emptyset) = 0$, and as a result it holds that $b \succ a$. On the contrary, if we label the alternatives, such that $a = b$ and $b = a$, then we would be in the case where $\tilde{Z} = \emptyset$ holds, and given that $V\left(\hat{Z}\right) = V(\emptyset) = 0$ and that $V\left(c_a \setminus \hat{Z}\right) = V(\emptyset) = 0$, it is verified that, $P(a, b) = 0 \not> \beta$. Hence, indifference is obtained between a and b, i.e., it holds that $a \sim b$, which contradicts the previous result.

As a consequence, the WOD procedure has to be extended for being capable of dealing with non-interval data and in general, with the particular situations where the upper bounds of different membership degrees coincide. This is a problem also related to the nature of the volume operator V and its use for measuring the magnitude of overlap between the m-dimensional cubes. Hence, a proper theoretical framework has to be given for examining the type of *uncertainty* that the V operator effectively measures.

3.2 Uncertainty due to imprecision

Uncertainty is a common phenomenon of life and applied decision problems. There are many ways of dealing with it, depending on the nature or source of the uncertainty. Here, under this fuzzy approach, two main types of uncertainty can be identified, *uncertainty due to ignorance* and *uncertainty due to imprecision*. The first one refers to *ignorance* or lack of knowledge (see e.g., [2,4], but also [11]), where there is not enough information for verifying the intensity in which a given object fulfills some concept or predicate. Hence, this first type of

uncertainty deals with the quantity of the available information and its sufficiency for estimating a membership degree. On the other hand, the second type of uncertainty refers to the quality of the available knowledge, which takes the form of intervals due to its *imprecise* nature. In this way, ignorance deals with the quantity of knowledge, while imprecision deals with its quality.

Imprecision is naturally present in common measurements and perceptions. Think e.g., in the *Heisenberg principle*, where an undeniable limit is recognized over the precision with which two complementary physical variables, such as position and momentum, can be measured and known simultaneously. In this way, the heuristic says that the more precise you are determining the position of some particle, the less precise you are determining its momentum.

Therefore, we propose to use *imprecision measures* as defined below, with the objective of measuring the magnitude of uncertainty due to imprecision over some membership degree μ_{Z_j}.

Definition 1. *Let μ_{Z_j} be the membership degree of Z_j, given by the respective lower and upper bounds $\mu^L_{Z_j}$ and $\mu^U_{Z_j}$, and let $\varepsilon \in (0,1)$. Then, the function $\delta : [0,1]^2 \rightarrow [\varepsilon, 1+\varepsilon]$ is an imprecision measure if and only if it fulfills the following conditions:*

1. *$\delta\left(\mu_{Z_j}\right) = \varepsilon$ if and only if $\mu^U_{Z_j} = \mu^L_{Z_j}$.*
2. *$\delta\left(\mu_{Z_j}\right) = 1+\varepsilon$ if and only if $\mu^L_{Z_j} = 0$ and $\mu^U_{Z_j} = 1$.*
3. *If $\mu^U_{Z_j} > \mu^L_{Z_j}$ then $\delta\left(\mu_{Z_j}\right) > \varepsilon$.*
4. *For two memebership degrees μ_{Z_j} and ν_{Z_j}, $\delta\left(\mu_{Z_j}\right) > \delta\left(\nu_{Z_j}\right)$ if and only if $\left(\mu^U_{Z_j} - \mu^L_{Z_j}\right) > \left(\nu^U_{Z_j} - \nu^L_{Z_j}\right)$.*

The first condition states that imprecision is minimum, i.e., equal to ε, if and only if the lower and upper bounds are the same, while the second condition states that imprecision will be maximum, i.e., equal to $1+\varepsilon$, if and only if the lower bound is minimum, i.e., equal to 0, and the upper bound is maximum, i.e., equal to 1. Then, the measure for imprecision will be positive and greater than ε every time the upper bound of the interval is greater than its lower bound. Finally, the last condition states that the greater the difference between the lower and upper bounds, the greater it will be the uncertainty due to imprecision.

An example for an imprecision measure is given by,

$$\delta'\left(\mu_{Z_j}\right) = \mu^U_{Z_j} - \mu^L_{Z_j} + \varepsilon. \tag{12}$$

As a result, the volume operator used by the WOD procedure can be understood from a fuzzy perspective by means of imprecision measures. In this sense, the fuzzy-WOD volume opertaor V refers to the magnitude of the uncertainty due to imprecision associated to any membership degree, where for every $a \in N$,

$$V\left(c_a\right) = \delta\left(\mu_{Z_1}(a)\right) \cdot \delta\left(\mu_{Z_2}(a)\right) \cdot ... \cdot \delta\left(\mu_{Z_m}(a)\right). \tag{13}$$

Now, it is possible to solve the particular cases examined in Examples 2-3 in the following way, without compromising the original WOD methodology.

It has been noted that the the WOD procedure requires that for any pair of alternatives $a, b \in N$, they have to be labeled such that for some vector of weights w, it holds that $w \cdot \mu^U(a) \geq w \cdot \mu^U(b)$. This condition can now be reformulated in the fuzzy-WOD model as follows. First, labeling alternatives such that the strict inequality $w \cdot \mu^U(a) > w \cdot \mu^U(b)$ holds. Then, if it is the case that the equality $w \cdot \mu^U(a) = w \cdot \mu^U(b)$ holds, labeling the alternatives according to their imprecision measures, such that $\delta(w \cdot \mu(a)) > \delta(w \cdot \mu(b))$ holds. In any other case, if $w \cdot \mu^U(a) = w \cdot \mu^U(a)$ and $\delta(w \cdot \mu(a)) = \delta(w \cdot \mu(b))$ hold, then it is said that $a \sim b$.

Example 4. Let us revisit Example 2, where we have equal non-interval data (both their lower and upper bounds coincide). Then, examining the imprecision associated to each membership degree, according to δ', we have that both degrees have the same imprecision. Hence, both alternatives are indifferent, i.e., $a \sim b$ holds, which is what was expected in the first place.

Example 5. Revisiting Example 3, where we have equal upper bounds, we can now examine the labeling of the alternatives according to the imprecission associated to their membership degrees. Hence, we have that

$$\delta'(\mu_{Z_1}(a)) = \delta'([0,1]) = 1 + \varepsilon,$$

and

$$\delta'(\mu_{Z_1}(b)) = \delta'([1,1]) = \varepsilon.$$

Then, no relabeling is necessary, and we have that $\check{Z} \neq \emptyset$ holds. In such case, we have that

$$\frac{V\left(\hat{Z}\right)}{V\left(\check{Z}\right)} = 0,$$

given that $V\left(\hat{Z}\right) = V(\emptyset) = 0$. As a result, it is obtained that $b \succ a$, which agrees with what could be expected.

Now that the foundation for representing uncertainty due to imprecision has been given, and the fuzzy-WOD procedure is capable of handling appropriately the particular cases where the upper bounds of different membership degrees coincide, we turn our attention to the ranking of alternatives. In the following section we explore some of the characterisitic properties of the WOD relations $\succ$ and $\sim$.

4 On the outranking and indifference relations

Examining the outcome of the fuzzy-WOD procedure, the alternatives are ordered according to their interval memebership degrees. As a result, two types of binary relations hold, i.e., the outranking, $\succ$, and the indifference, $\sim$, relations. In this section we explore some of their mathematical properties and justify our proposal on relevance measures of Section 5.

As it has been shown in [8], the outranking relation $\succ$ is a semi-transitive relation, such that for every $a, b, c \in N$, if $a \succ b$ and $b \succ c$ hold, then it cannot hold that $c \succ a$. Besides, it can be shown that $\succ$ is *irreflexive*, i.e., it does not hold that $a \succ a$, and *asymmetric*, i.e., if it holds that $a \succ b$, then it does not hold that $b \succ a$.

Proposition 1. *For every $a, b \in N$, the WOD outranking relation $\succ$ is irreflexive and asymmetric.*

In this way, if $\succ$ is semi-transitive, the indifference relation cannot be understood as a standard equivalence relation, i.e., being reflexive, symmetric and transitive, as shown in the following proposition.

Proposition 2. *For every $a, b, c \in N$, the WOD indifference relation $\sim$ is not an equivalence relation, i.e., it's not transitive.*

In order to illustrate Proposition 2, consider three alternatives $a, b, c \in N$, and the semi-tranistivity of $\succ$, such that if $a \succ b$ and $b \succ c$,then it holds either that $a \succ c$ or that $a \sim c$. But if $a \sim c$ holds, then it cannot be that $\sim$ is transitive while both $a \succ b$ and $b \succ c$ are true.

As a result, the WOD procedure assigns an order that is not a ranking of N, unless some type of construction is developed where the *conflicting* or *dependent* alternatives can be considered as *equivalent* objects, as previously done in [8], although they are not mathematically equivalent. In this sense, the WOD characterization of $\succ$ and $\sim$ does not produce a weak order, where for any two alternatives the first one is either ranked higher, equal or lower than the second one. Therefore, we now introduce *relevance measures* for obtaining a ranking over N (based on the general intuition of [5]), explaining the process in which alternatives can be effectively ranked by means of the fuzzy-WOD model.

5 Ranking alternatives according to their relevance

The fuzzy-WOD approach to decision support allows the interaction of the individual with the ordering process, where by means of user-defined parameters the system exploits the flexibility of the interval information. According to its procedure, the model obtains an outranking order of the alternatives, aiding the decision maker to understand the possible solutions of the problem. As it has been pointed out, its raw outcome is not a weak order, and hence, the problem remains to be solved if the alternatives are to be ranked in a descriptively satisfactory way.

Here we explore the ordering process according to the outranking and indifference relations $\succ$ and $\sim$, and observe that every time an alternative dominates or outranks another one, its relevance should increase, but also, that such increment should be higher if the outranked alternative is also an important one. Following this intuition, we propose the use of *relevance degrees* [5], such that each alternative obtains a relevance value according to the number and the importance of the alternatives that it outranks.

In this way, let the importance of an object $a \in N$ be given by the number of objects that it outranks, denoted by d_a, and let D_a be the set of all objects that are outranked by a. Then, we define relevance degrees as follows.

Definition 2. *For every alternative $a \in N$, its relevance degree is defined by,*

$$\sigma_a = d_a + \sum_{b \in D_a} d_b.$$

As a result, based on the WOD outranking ordering, the alternatives can be automatically ranked according to their relevance degrees given by σ. Therefore, each alternative has an associated degree that is either greater, equal or less than any other degree.

Notice that relevance degrees, as presented in Definition 2, fulfill the following intuitive conditions for characterizing the notion of *relevance*, such that for any $a, b, c, d \in N$:

- If an alternative does not outrank any other alternative, it has no relevance at all. That is, it is true that $\sigma_a = 0$ if and only if it holds that $D_a = \emptyset$.
- An alternative gains relevance with the number and the importance of the alternatives that it outranks. That is, if $a \succ b$ and $D_b = \emptyset$ hold, then it is true that $\sigma_a > \sigma_b$.
- If an alternative outranks every other alternative, its relevance is maximum. That is, if $D_a = \{N \setminus \{a\}\}$, then for every $b \neq a \in N$, it is true that $\sigma_a > \sigma_b$.
- One alternative is more relevant than another if it outranks alternatives with a higher degree of importance. That is, if $a \succ b$, $c \succ d$ and $b \succ d$ hold, then $\sigma_a > \sigma_c$ holds true.

Example 6. Consider six alternatives $a, b, c, d, e, f \in N$, such that the following WOD relations hold,

$$a \succ b,\ a \succ d,\ a \succ e,$$
$$b \sim d,\ b \succ f,\ b \succ e,$$
$$c \sim b,\ c \succ d,\ c \succ e,$$
$$d \succ e.$$

Then, alternative a is the clear winner, but if we examine the relation between b and c, we see that they are indifferent although b is indifferent with d and c outranks d. Hence, it is not clear how to rank these alternatives, because if we declare an indifference class between b and c, we have to include also d, but then again, we know that c outranks d.

Applying relevance measures over this problem, we obtain the following results,

$$\sigma_a = 6,\ \sigma_b = 2,\ \sigma_c = 3,\ \sigma_d = 1,\ \sigma_e = 0,\ \sigma_f = 0,$$

from where it can be concluded that a is the first alternative of the ranking, followed by c, then by b, then by d, and finally by e and f which are both equivalent. In this way, alternatives are ranked by their relevance in the fuzzy-WOD outranking order, according to their relative importance regarding the complete set of alternatives.

Notes and Comments. We have explored the WOD model from a fuzzy perspective, characterizing the type of uncertainty due to imprecision that can be represented by means of interval information. As a future lines of research it is observed that an interactive decision support system should allow the user to express his/her attitudes by means of words or linguistic terms. In this way, the individual would be required to use words instead of numbers to express his attitude parameters, extending in this way the decision support model into a more human-friendly environment. Such extension could be done, e.g., following the Computing with Words methodology [10,15] and its approach on the human ability to handle and evaluate perceptions through language. Besides, it should be explored how distinct rankings (resulting from different experts with different beliefs) can be aggregated for identifying a single recommended choice.

References

1. Atanassov, K., Gargov, G. Interval valued intuitionistic fuzzy sets. Fuzzy Sets and Systems 31, 343-349 (1989)
2. Barrenechea, E., Fernandez, A., Herrera, F., Bustince, H.. Preference relations using ignorance functions: interval-valued non dominance criterion. Advances in Intelligent and Soft Computing 107, 243-255 (2012)
3. Chen, T., Tsao, Ch. The interval-valued fuzzy TOPSIS method and experimental analysis. Fuzzy Sets and Systems 159, 1410-1428 (2008)
4. Franco, C., Rodríguez, J.T., Montero, J. Relational structures for measures of ignorance. Advances in Intelligent Systems Research 1, 613-619 (2011)
5. Franco, C., Rodríguez, J.T., Montero, J. Relevance in preference structures. Proceedings of the 2012 International Conference on Intelligent Systems and Knowledge Engineering (ISKE2012). Beijing, China, December 15-17, 2012, paper FIT 4-502.
6. Goguen, J. The logic of inexact concepts. Synthese 19, 325-373 (1969)
7. Grattan-Guinness, I. Fuzzy membership mapped onto interval and many-valued quantities. Mathematical Logic Quarterly 22, 149-160 (1976)
8. Hougaard, J.L., Nielsen, K. Weighted Overlap Dominance - A procedure for interactive selection on multidimensional interval data. Applied Mathematical Modelling 35, 3958-3969 (2011)
9. LeTéno, J.F., Mareschal, B. An interval version of PROMETHEE for the comparison of building products' design with ill-defined data on environmental quality. European Journal of Operational Research 109, 522-529 (1998)
10. Martínez, L., Ruan, D., Herrera, F. Computing with words in decision support systems: an overview on models and applications. International Journal of Computational Intelligence Systems 3, 382-395 (2010)
11. Montero, J., Gómez, D., Bustince, H. On the relevance of some families of fuzzy sets. Fuzzy Sets and Systems 158, 2429-2442 (2007)
12. Roy, B. The outranking approach and the foundations of ELECTRE methods. Theory and Decision 31, 49-73 (1991)
13. Yager, R., Petry, F. A framework for linguistic relevance feedback in content-based image retrieval using fuzzy logic. Information Sciences 173, 337-352 (2005)
14. Zadeh, L. The concept of a linguistic variable and its application to approximate reasoning-I. Information Sciences 8, 199-249 (1975)
15. Zadeh, L. Fuzzy logic = computing with words. IEEE Transactions on Fuzzy Systems 4, 103-111 (1996)

Nuevas Operaciones Binarias sobre los Conjuntos Borrosos de Tipo 2

Pablo Hernández[1], Susana Cubillo[2], Carmen Torres[2]

[1] Departamento de Matemática y Física,
Univ. Nacional Experimental del Táchira (UNET),
San Cristóbal, Táchira, Venezuela
phernandezv@unet.edu.ve
[2] Departamento de Matemática Aplicada,
Univ. Politécnica de Madrid (UPM),
28660 Boadilla del Monte, Madrid, España
{scubillo,ctorres}@fi.upm.es

Resumen Walker et al. ([19], [20]) definieron dos familias de operaciones binarias sobre **M** (conjunto de las funciones de [0,1] en [0,1]), y determinaron que, bajo ciertas condiciones, estas operaciones son t-normas (normas triangulares) o t-conormas sobre **L** (subconjunto de las funciones normales y convexas de **M**). En este trabajo se introducen dos operaciones binarias sobre **M**, diferentes a las dadas por Walker et al., y se estudian varias propiedades de las mismas, con el objeto de deducir nuevas t-normas y t-conormas sobre **L** y sobre **M**.

Palabras Clave: Funciones de [0,1] en [0,1], funciones normales y convexas, t-normas y t-conormas.

1. Introducción

Los conjuntos borrosos de tipo 2 (T2FSs) fueron introducidos por L.A. Zadeh en 1975 [22], como una extensión de los conjuntos borrosos de tipo 1 (FSs). Mientras que en estos últimos el grado de pertenencia de un elemento al conjunto viene determinado por un valor en el intervalo $[0, 1]$, en el caso de los T2FSs el grado de pertenencia de un elemento es un conjunto borroso en [0,1], es decir, un T2FS queda determinado por una función de pertenencia $\mu : X \to \mathbf{M}$, donde $\mathbf{M} = [0, 1]^{[0,1]}$ es el conjunto de las funciones de [0,1] en [0,1] (ver [11,13,14,19]). En este artículo se obtendrán resultados para los T2FSs con grados de pertenencia en **M**, y también en el subconjunto **L** de las funciones normales y convexas de **M**.

Las normas triangulares (t-normas) fueron introducidas por Menger [12]; mas tarde Schweizer y Sklar en [17,16] dieron la axiomática, usada actualmente, para definir t-normas. Debido a la estrecha conexión entre la teoría de los conjuntos borrosos y la teoría del orden (ver, por ejemplo, [5]), distintos autores han estudiado las t-normas sobre conjuntos parcialmente ordenados acotados (posets acotados), como por ejemplo en [3] y en [2]. En [15] se consideró la extensión

de las t-normas sobre retículos acotados exigiendo unos axiomas (en adelante se llamarán "axiomas básicos"), que coinciden con los dados en [3] y [2].

En [4] se extendieron las definiciones de t-norma y t-conorma a los conjuntos borrosos intervalo valorado (IVFSs), pero se agregaron otras restricciones o propiedades a los axiomas básicos, estableciendo los "axiomas restrictivos". Posteriormente, en [19,20] los autores extendieron estos axiomas a los T2FSs, y presentaron dos familias de operaciones binarias sobre $\mathbf{M}$, determinando que, bajo ciertas condiciones, son t_r-normas y t_r-conormas (t-normas y t-conormas según los axiomas restrictivos) sobre $\mathbf{L}$. El objetivo de este artículo es la introducción de dos operaciones binarias sobre M, diferentes a las presentadas en [19,20], y el estudio de sus propiedades, para determinar si son t_r-norma o t_r-conorma sobre $\mathbf{L}$ y sobre $\mathbf{M}$.

El artículo se organiza como sigue: en la Sección 2 se recuerdan algunas definiciones y propiedades de los FSs, IVFs y T2FSs; se exponen los antecedentes relacionados con las t-normas y t-conormas sobre los mismos, y se presentan los axiomas básicos y restrictivos. En la Sección 3 se presentan las operaciones $\circledcirc$ y $\circledast$ (ver Definición 12), y se estudia si satisfacen cada uno de los axiomas de t_r-norma y t_r-conorma, respectivamente, sobre $\mathbf{L}$ y sobre $\mathbf{M}$. Esto nos permitirá proponer nuevas operaciones que resultarán ser t-norma y t-conorma sobre $\mathbf{L}$, y sobre $\mathbf{M}$.

La última sección está dedicada a exponer algunas conclusiones.

2. Preliminares

En todo el trabajo se denotará por X un conjunto no vacío que representará el universo de discurso. Además, $\leq$ denotará la relación de orden en el retículo de los números reales.

2.1. Varios Tipos de Conjuntos Borrosos

Definición 1. ([21]) *Un conjunto borroso de tipo 1 (FS), A, queda caracterizado por una* función de pertenencia μ_A,

$$\mu_A : X \to [0,1],$$

donde $\mu_A(x)$ es el grado de pertenencia de un elemento $x \in X$ al conjunto A.

Definición 2. ([1,18]) *Un conjunto borroso intervalo valorado (IVFS), A, queda caracterizado por una* función de pertenencia σ_A,

$$\sigma_A : X \to I = \{[a,b] : 0 \leq a \leq b \leq 1\}.$$

De esta manera, el grado de pertenencia de un elemento $x \in X$ al conjunto A es un intervalo en $[0,1]$.

Definición 3. ([13,14]) *Un conjunto borroso de tipo 2 (T2FS), A, queda caracterizado por una* función de pertenencia*:*

$$\mu_A : X \to \mathbf{M} = [0,1]^{[0,1]} = Map\,([0,1],[0,1]).$$

Esto es, $\mu_A(x)$ *es un conjunto borroso en el intervalo* $[0,1]$*, y es el* grado *de pertenencia de un elemento* $x \in X$ *al conjunto A. Por tanto,*

$$\mu_A(x) = f_x\ , \quad donde \quad f_x : [0,1] \to [0,1].$$

Se denota por $F_2(X)$ *al conjunto de todos los conjuntos borrosos de tipo 2 sobre X.*

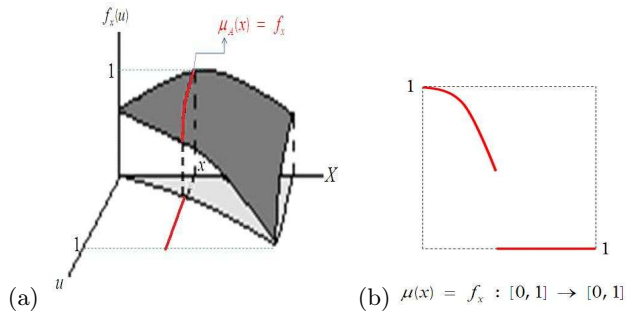


Figura 1. (a) Ejemplo de un T2FS, y (b) Ejemplo de un grado de pertenencia al T2FS.

Definición 4. ([19]) *Sea* $a \in [0,1]$*. La función característica de a es* $\boldsymbol{a} : [0,1] \to [0,1]$*, donde*

$$\boldsymbol{a}(x) = \begin{cases} 1 \ si\ x = a \\ 0 \ si\ x \neq a \end{cases}$$

Sea $\mathbf{J} \subset \mathbf{M}$ el conjunto de todas las funciones características de los elementos en [0,1]. Es decir, $\mathbf{J} = \{\mathbf{a} : [0,1] \to [0,1] : \ a \in [0,1]\}$. Hay un isomorfismo (de orden) entre $(\mathbf{J}, \sqsubseteq)$, ver Definición 7, y el intervalo $([0,1], \leq)$, que es el conjunto de los valores de pertenencia de los conjuntos borrosos.

Definición 5. ([19]) *Sea* $[a,b] \subseteq [0,1]$*. La función característica de* $[a,b]$ *es* **[a,b]** $: [0,1] \to [0,1]$*, donde*

$$\mathbf{[a,b]}(x) = \begin{cases} 1 \ si\ x \in [a,b] \\ 0 \ si\ x \notin [a,b] \end{cases}$$

Sea $\mathbf{K} \subset \mathbf{M}$ el conjunto de todas las funciones características de los subintervalos de $[0,1]$. Además, existe un isomorfismo (de orden) entre $(\mathbf{K}, \sqsubseteq)$ y el conjunto de los valores de pertenencia de los IVFSs.

En [19], se justifica que las operaciones sobre $Map(X, \mathbf{M})$ se pueden definir de forma natural a partir de las operaciones sobre $\mathbf{M}$, verificando las mismas propiedades. Por tanto, en este artículo trabajaremos en $\mathbf{M}$ ya que todos los resultados se pueden extender directa y fácilmente a $Map(X, \mathbf{M})$, conjunto de las funciones de pertenencia de los elementos de $F_2(X)$.

Definición 6. ([7,19]) *En* $\mathbf{M}$ *se definen las operaciones* $\sqcup$ *(unión),* $\sqcap$ *(intersección),* $\neg$ *y los elementos* $\bar{0}$ *y* $\bar{1}$ *de la siguiente forma:*

$$(f \sqcup g)(x) = sup\{f(y) \wedge g(z) : y \vee z = x\},$$

$$(f \sqcap g)(x) = sup\{f(y) \wedge g(z) : y \wedge z = x\},$$

$$\neg f(x) = sup\{f(y) : 1 - y = x\} = f(1 - x),$$

$$\bar{0}(x) = \begin{cases} 1 \ si \ x = 0 \\ 0 \ si \ x \neq 0 \end{cases}, \quad \bar{1}(x) = \begin{cases} 1 \ si \ x = 1 \\ 0 \ si \ x \neq 1. \end{cases},$$

donde $\vee$ y $\wedge$ son las operaciones máximo y mínimo, respectivamente, en el retículo $[0, 1]$. Obsérvese que $\bar{0}$ y $\bar{1}$ son precisamente las funciones características de 0 y 1, respectivamente.

Es fácil probar que $\sqcup$ y $\sqcap$ son idempotentes ($f \sqcap f = f$ y $f \sqcup f = f$, para todo $f \in \mathbf{M}$) y satisfacen las leyes de De Morgan respecto a la operación dada $\neg$, pero $\mathbb{M} = (\mathbf{M}, \sqcup, \sqcap, \neg, \bar{0}, \bar{1})$ no tiene estructura de retículo, ya que no se cumple la ley de absorción [7,19]. Por otra parte, las operaciones $\sqcup$ y $\sqcap$ cumplen las propiedades requeridas para definir, cada una de ellas, un orden parcial sobre $\mathbf{M}$.

Definición 7. ([14,19]) *En* $\mathbf{M}$ *se definen los dos órdenes parciales siguientes:*

$$f \sqsubseteq g \ si \ f \sqcap g = f; \ \ f \preceq g \ si \ f \sqcup g = g.$$

En general, estos dos órdenes no coinciden [14,19]. $f \sqcap \bar{1} = f$, lo cual implica que $f \sqsubseteq \bar{1}$, $\forall f \in \mathbf{M}$ ([19]), es decir, $\bar{1}$ es el mayor elemento del orden parcial $\sqsubseteq$. Por otra parte, $\bar{0} \sqcup f = f$, por tanto $\bar{0} \preceq f$, $\forall f \in \mathbf{M}$ ([19]), esto es $\bar{0}$ es el menor elemento del orden parcial $\preceq$. Además, la función constante $g = 0$ ($g(x) = 0$, $\forall x \in [0, 1]$) es el menor y mayor elemento de $\sqsubseteq$ y $\preceq$, respectivamente.

Con el fin de facilitar las operaciones sobre $\mathbf{M}$, en trabajos anteriores se dieron la Definición y Teorema siguientes:

Definición 8. ([7,19]) *Si* $f \in \mathbf{M}$*, se definen* $f^L, f^R \in \mathbf{M}$ *como*

$$f^L(x) = sup\{f(y) : y \leq x\}, \qquad f^R(x) = sup\{f(y) : y \geq x\}.$$

f^L y f^R son monótonas creciente y decreciente, respectivamente. A partir de f^L y f^R, se estableció el siguiente resultado.

Teorema 1. ([19]) *Para todo* $f, g \in \mathbf{M}$*, se tiene:*

$$f \sqsubseteq g \ \Leftrightarrow \ (f^R \wedge g) \leq f \leq g^R, \qquad f \preceq g \ \Leftrightarrow \ (g^L \wedge f) \leq g \leq f^L.$$

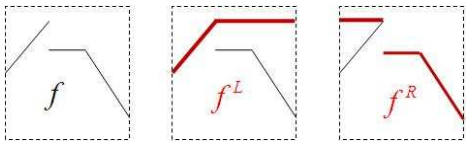


Figura 2. Ejemplos de f^L y f^R.

Es fácil comprobar que la función característica de un intervalo $[a, b]$ es $\mathbf{[a,b]} = \mathbf{a}^L \wedge \mathbf{b}^R$ (ver [19]).

A continuación se considerará $\mathbf{L}$, subconjunto de las funciones normales y convexas de $\mathbf{M}$, ya que en este conjunto se va a tener una estructura de retículo acotado y completo, lo que va a permitir construir t-normas y t-conormas de forma adecuada. Recordemos que:

Definición 9. *Una función $f \in \mathbf{M}$ es normal si $sup\{f(x) : x \in [0,1]\} = 1$.*

Sea $\mathbf{N}$ el conjunto de todas las funciones normales en $\mathbf{M}$. Obsérvese que dada $f \in \mathbf{M}$, se tiene que $f \in \mathbf{N}$ si y sólo si $f^L \vee f^R = 1$, donde 1 es la función tal que $f(x) = 1$, para todo $x \in [0,1]$.

Definición 10. *Una función $f \in \mathbf{M}$ es convexa, si para cualquier $x \leq y \leq z$, se cumple que $f(y) \geq f(x) \wedge f(z)$.*

Sea $\mathbf{C}$ el conjunto de todas las funciones convexas en $\mathbf{M}$. Obsérvese que si $f \in \mathbf{M}$, entonces $f \in \mathbf{C}$ si y sólo si $f = f^L \wedge f^R$.

El conjunto de todas las funciones normales y convexas de $\mathbf{M}$ será denotado por $\mathbf{L}$. El álgebra $\mathbb{L} = (\mathbf{L}, \sqcup, \sqcap, \neg, \bar{0}, \bar{1})$ es un subálgebra de $\mathbb{M}$. En $\mathbf{L}$, los órdenes parciales $\sqsubseteq$ and $\preceq$ coinciden, y $\mathbb{L}$ es un retículo acotado ($\bar{0}$ y $\bar{1}$ son el mínimo y el máximo, respectivamente) y completo (ver [6,7,14,19]). Además, es evidente que $\mathbf{J} \subset \mathbf{K} \subset \mathbf{L} \subset \mathbf{M}$.

La siguiente caracterización nos ayudará a establecer nuevos resultados:

Teorema 2. ([6,7]) *Sean $f, g \in \mathbf{L}$. $f \sqsubseteq g$ si y sólo si*

$$g^L \leq f^L \quad y \quad f^R \leq g^R.$$

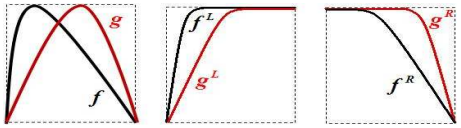


Figura 3. Ejemplo donde $f \sqsubseteq g$.

2.2. T-normas y T-conormas

Recordemos que una t-norma ([10]) es una operación binaria $T : [0,1]^2 \to [0,1]$, conmutativa, asociativa, creciente en cada argumento, y con elemento neutro 1. Además, una t-conorma es una operación binaria $S : [0,1]^2 \to [0,1]$, conmutativa, asociativa, creciente en cada argumento, y con elemento neutro 0. Definiciones similares se aplican para los retículos acotados. En [4,20] se extendió esta definición a los IVFSs y a los T2FSs, añadiendo algunos axiomas que reflejan algunas propiedades deseables. Por ejemplo, como $\mathbf{J} \subset \mathbf{K} \subset \mathbf{L}$, parece rasonable exigir que las t-normas sobre $\mathbf{L}$ sean cerradas sobre $\mathbf{J}$ y sobre $\mathbf{K}$. Además, debido a que las t-normas sobre los IVFSs satisfacen $T([1,1],[a,b]) = [a,b]$ y $T([0,0],[a,b]) = [0,0]$, entonces, por analogía, se requiere que $T([0,1],[a,b]) = [0,b]$. De esta manera se establecieron los siguientes "axiomas restrictivos":

Definición 11. ([20]) *La operación binaria $T : \mathbf{L}^2 \to \mathbf{L}$ es una t_r-norma (según los axiomas restrictivos) sobre $\mathbf{L}$ si:*

1. *T es conmutativa*
2. *T es asociativa.*
3. *$T(f, \bar{1}) = f$ para cualquier $f \in \mathbf{L}$ (elemento neutro).*
4. *Sean $f, g, h \in \mathbf{L}$ tal que $g \sqsubseteq h$, entonces $T(f,g) \sqsubseteq T(f,h)$ (creciente en cada argumento).*
5. *$T(\mathbf{[0,1]}, \mathbf{[a,b]}) = \mathbf{[0,b]}$.*
6. *T es cerrada en $\mathbf{J}$.*
7. *T es cerrada en $\mathbf{K}$.*

De forma similar, una operación binaria $S : \mathbf{L}^2 \to \mathbf{L}$ es una t_r-conorma si satisface: los axiomas 1, 2, 4, 6 y 7 de la t_r-norma; el axioma 3': $S(f, \bar{0}) = f$; y el axioma 5': $S(\mathbf{[0,1]}, \mathbf{[a,b]}) = \mathbf{[a,1]}$. Los axiomas 1, 2, 3 y 4, son llamados "axiomas básicos", y la operación que satisface dichos axiomas será denominada simplemente t-norma o t-conorma, según sea el caso.

3. T-normas y T-conormas sobre L

En [18,19,20] se demostró que las operaciones $\sqcap$ y $\sqcup$ son t_r-norma y t_r-conorma, respectivamente, sobre $\mathbf{L}$. Además se introdujeron dos nuevas familias de operaciones sobre $\mathbf{M}$, $\blacktriangle$ y $\blacktriangledown$, obteniendo que son t_r-norma y t_r-conorma, respectivamente, sobre $\mathbf{L}$.

$$(f \blacktriangle g)(x) = sup\{f(y) \wedge g(z) : y \triangle z = x\},$$

$$(f \blacktriangledown g)(x) = sup\{f(y) \wedge g(z) : y \triangledown z = x\},$$

donde $\triangle$ y $\triangledown$ son, respectivamente, t-norma y t-conorma continuas sobre [0,1].

Los autores no estudiaron si $\blacktriangle$ y $\blacktriangledown$, verifican los axiomas básicos sobre $\mathbf{M}$, respecto a cada órden parcial.
Observemos que si $\triangle = \wedge$ y $\triangledown = \vee$, entonces $\blacktriangle = \sqcap$ y $\blacktriangledown = \sqcup$.

Posteriormente, [8,9] presentaron dos familias de operaciones binarias sobre $\mathbf{M}$, más generales que $\blacktriangle$ y $\blacktriangledown$, ya que por $\wedge$ consideraron cualquier operación binaria sobre [0,1], y determinaron que, bajo ciertas condiciones, una familia de operaciones es t-norma sobre $(\mathbf{M}, \sqsubseteq)$ y la otra es t-conorma sobre $(\mathbf{M}, \preceq)$.

A continuación se introducen dos operaciones sobre $\mathbf{M}$, que no se obtienen (ver Observaciones 1 y 2) con las fórmulas para $\blacktriangle$ y $\blacktriangledown$, y se analiza si satisfacen cada uno de los axiomas restrictivos.

Definición 12. *Para todo $f, g \in \mathbf{M}$ se definen las operaciones binarias*

$$f \circledcirc g = f^R \wedge g^R \ , \qquad f \circledast g = f^L \wedge g^L \ .$$

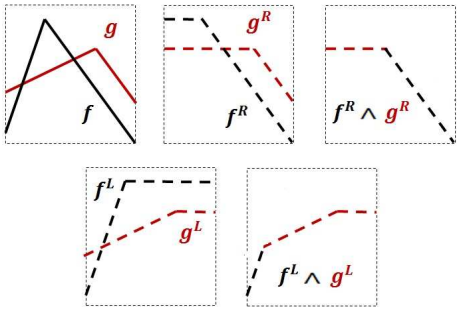


Figura 4. Ejemplos de las operaciones $\circledcirc$ y $\circledast$.

De acuerdo a [19], $f^R \wedge g^R = f^R \sqcap g^R = f^R \sqcap g = f \sqcap g^R = (f \sqcap g)^R$, y $f^L \wedge g^L = f^L \sqcup g^L = f^L \sqcup g = f \sqcup g^L = (f \sqcup g)^L$, por tanto, $\forall\ f, g\ \in \mathbf{M}$

$$f \circledcirc g = f^R \sqcap g^R = f \sqcap g^R = f^R \sqcap g = (f \sqcap g)^R \ ,$$

$$f \circledast g = f^L \sqcup g^L = f \sqcup g^L = f^L \sqcup g = (f \sqcup g)^L \ .$$

Por otra parte, es fácil comprobar que $\circledcirc$ y $\circledast$ son monótonas decreciente y creciente, respectivamente. Por tanto,

$$(f \circledcirc g)^R = (f^R \wedge g^R)^R = (f^R \wedge g^R) = f \circledcirc g,$$

$$(f \circledast g)^L = (f^L \wedge g^L)^L = (f^L \wedge g^L) = f \circledast g.$$

Proposición 1. *$\circledcirc$ y $\circledast$ son conmutativas y asociativas, sobre $\mathbf{M}$.*

El siguiente lema nos permite deducir que $\bar{1}$ y $\bar{0}$ no son elementos neutros de $\circledcirc$ y $\circledast$, respectivamente.

Lema 1. $\forall\ f \in \mathbf{M}$,

$$f \circledcirc \bar{1} = f^R, \qquad f \circledast \bar{0} = f^L.$$

Del siguiente lema se deduce que $\bar{0}$ y $\bar{1}$ no son elementos absorbentes de $\circledcirc$ y $\circledast$, respectivamente, sobre **M**. No obstante, si lo son sobre **N**, como se puede ver en el Corolario 1.

Lema 2. $\forall\ f \in \mathbf{M}$,

$$f \circledcirc \bar{0} = f^R \wedge \bar{0}, \qquad f \circledast \bar{1} = f^L \wedge \bar{1}.$$

Corolario 1. *Si* $f \in \mathbf{N}$, *entonces*

$$f \circledcirc \bar{0} = \bar{0}, \qquad f \circledast \bar{1} = \bar{1}.$$

Observación 1. En [19], se obtuvo que $\bar{1}$ y $\bar{0}$ son los elementos neutros de $\blacktriangle$ y $\blacktriangledown$, respectivamente, en **M**. Sin embargo, considerando el Lema 1 y el Corolario 1, basta tomar un $f \in \mathbf{L}$, tal que $f \neq f^R$ y $f \neq f^L$ (por ejemplo, $f(x) = 1 - 4(x - 0{,}5)^2,\ x \in [0,1]$) para comprobar que $\bar{1}$ y $\bar{0}$ no son elementos neutros para las operaciones $\circledcirc$ y $\circledast$, sobre **L**, y por tanto, tampoco sobre **M**. En consecuencia, esto es suficiente para asegurar que $\circledcirc$ y $\circledast$ no se obtienen con las fórmulas para $\blacktriangle$ y $\blacktriangledown$, sobre **L**, y por tanto, tampoco sobre **M**.

Analicemos las leyes ditributivas respecto a $\sqcap$ y $\sqcup$.

Proposición 2. $\forall\ f, g,\ h\ \in \mathbf{M}$ *se cumple que*

$$f \circledcirc (g \sqcap h) = (f \circledcirc g) \sqcap (f \circledcirc h)\ , \qquad f \circledast (g \sqcup h) = (f \circledast g) \sqcup (f \circledast h)\ .$$

Analicemos si las operaciones $\circledcirc$ y $\circledast$ son crecientes en cada argumento, respecto a los órdenes parciales $\sqsubseteq$ y $\preceq$, respectivamente.

Proposición 3. *Sean* $f, g,\ h\ \in \mathbf{M}$. *Se cumple que*

$$\textit{Si } g \sqsubseteq h \Rightarrow (f \circledcirc g) = (g \circledcirc f) \sqsubseteq (f \circledcirc h) = (h \circledcirc f).$$

$$\textit{Si } g \preceq h \Rightarrow (f \circledast g) = (g \circledast f) \preceq (f \circledast h) = (h \circledast f).$$

Proposición 4. *Sean* $f, g \in \mathbf{M}$, *entonces*

$$(f \circledcirc g) \in \mathbf{C}\ , \qquad (f \circledast g) \in \mathbf{C}\ .$$

Proposición 5. $\circledcirc$ *y* $\circledast$ *son cerradas en* **N**, *y en* **L**.

Corolario 2. $\circledcirc$ *y* $\circledast$ *son crecientes en cada argumento en el retículo* $(\mathbf{L}, \sqsubseteq)$.

Los siguientes resultados proporcionan algunas propiedades de $\circledcirc$ y $\circledast$ en los conjuntos **J** y **K**.

Proposición 6. *Sean* $a, b, c, d \in [0,1]$, *tal que* $a \leq b$ *y* $c \leq d$, *entonces*

$$\mathbf{[a,b]} \circledcirc \mathbf{[c,d]} = \mathbf{[0,e]},\ e = b \wedge d. \qquad \mathbf{[a,b]} \circledast \mathbf{[c,d]} = \mathbf{[f,1]},\ f = a \vee c.$$

Es decir, $\circledcirc$ *y* $\circledast$ *son cerradas en* **K**.

Corolario 3. $\circledcirc$ *y* $\circledast$ *no son cerradas en* $\mathbf{J}$.

Observación 2. $\blacktriangle$ y $\blacktriangledown$ son cerradas en $\mathbf{J}$, por tanto el Corolario 3 también justifica que $\circledcirc$ y $\circledast$ no se obtienen de $\blacktriangle$ y $\blacktriangledown$, sobre $\mathbf{L}$, ni sobre $\mathbf{M}$.

Corolario 4. *Sean* $a, b \in [0,1]$*, tal que* $a \leq b$*, entonces*

$$\mathbf{[0,1]} \circledcirc \mathbf{[a,b]} = \mathbf{[0,b]} \ , \quad \mathbf{[0,1]} \circledast \mathbf{[a,b]} = \mathbf{[a,1]} \ .$$

Observación 3. Las operaciones $\circledcirc$ y $\circledast$, no satisfacen los axiomas correspondientes al elemento neutro y la clausura en $\mathbf{J}$, sin embargo, verifican todos los demás axiomas de t_r-norma y t_r-conorma, respectivamente, sobre $\mathbf{L}$.

Proposición 7. *Sean las operaciones binarias*

$$f \bar{\circledcirc} g = \begin{cases} f & si\ g = \bar{1} \\ g & si\ f = \bar{1} \\ f \circledcirc g\ en\ otro\ caso \end{cases} \ , \quad f \bar{\circledast} g = \begin{cases} f & si\ g = \bar{0} \\ g & si\ f = \bar{0} \\ f \circledast g\ en\ otro\ caso \end{cases} \ ,$$

para todo $f, g \in \mathbf{M}$*. Entonces,* $\bar{\circledcirc}$ *es una t-norma sobre* $(\mathbf{M}, \sqsubseteq)$*. No satisface la clausura en* $\mathbf{J}$*, pero, verifica todos los demás axiomas de* t_r*-norma sobre* $\mathbf{L}$*, por tanto es una t-norma sobre* $\mathbf{L}$*. Además,* $\bar{0}$ *es el elemento absorbente de* $\bar{\circledcirc}$ *en* $\mathbf{L}$*. De forma similar,* $\bar{\circledast}$ *es una t-conorma sobre* $(\mathbf{M}, \preceq)$*. No satisface la clausura en* $\mathbf{J}$*, pero, verifica todos los demás axiomas de* t_r*-conorma sobre* $\mathbf{L}$*, por tanto es una t-conorma sobre* $\mathbf{L}$*. Además,* $\bar{1}$ *es el elemento absorbente de* $\bar{\circledast}$ *en* $\mathbf{L}$*.*

Observación 4. Si $f, g \in \mathbf{J}$, tal que $f \neq \bar{1} \neq g$ y $f \neq \bar{0} \neq g$, entonces $\bar{\circledcirc} = \circledcirc$ y $\bar{\circledast} = \circledast$, y por tanto, de acuerdo a la Observación 2, $\bar{\circledcirc}$ y $\bar{\circledast}$ no se obtienen de las fórmulas de $\blacktriangle$ y $\blacktriangledown$, sobre $\mathbf{L}$, ni sobre $\mathbf{M}$.

4. Conclusiones

En este trabajo se han introducido las operaciones binarias $\circledcirc$ y $\circledast$, diferentes a las definidas en [19]. Se ha analizado si dichas operaciones satisfacen cada uno de los axiomas de t_r-norma y t_r-conorma sobre $\mathbf{L}$ y sobre $\mathbf{M}$.

A partir de $\circledcirc$ y $\circledast$ se han obtenido las operaciones binarias $\bar{\circledcirc}$ y $\bar{\circledast}$, que han resultado ser, respectivamente, t-norma sobre $\mathbf{L}$ y sobre ($\mathbf{M}$,$\sqsubseteq$), y t-conorma sobre $\mathbf{L}$ y sobre ($\mathbf{M}$,$\preceq$). Se ha demostrado que estas Ãºltimas operaciones verifican todos los axiomas de t_r-norma y t_r-conorma, respectivamente, sobre $\mathbf{L}$, excepto la clausura sobre $\mathbf{J}$.

Agradecimientos. Este trabajo ha sido financiado parcialmente por la CICYT (España) proyecto TIN2011-29827-C02-01, UPM-CAM, FONACIT (Venezuela) y UNET (Venezuela).

Referencias

1. Bustince, H., Barrenechea, E., Pagola, M.: Generation of interval-valued fuzzy and Atanassov's intuitionistic fuzzy connectives from fuzzy connectives and from K_α operators. Laws for conjunctions and disjunctions. Amplitude. Internat. J. Intell. Systems, 23, 680–714 (2008).
2. De Baets, B., Mesiar, R.: Triangular norms on product lattices. Fuzzy Sets and Systems, 104, 61–75 (1999).
3. De Cooman, G., Kerre, E.: Order norms on bounded partially ordered sets. Journal Fuzzy Mathematics, 2, 281–310 (1994).
4. Gehrke, M., Walker, C., Walker, E.: Some comments on interval-valued fuzzy sets. Internat. J. Intell. Systems, 11, 751–759 (1996).
5. Goguen, J.: L-Fuzzy Sets. J. Math. Anal. Appli. 18(1), 623–668 (1967).
6. Harding, J., Walker, C., Walker, E.: Convex normal functions revisited. Fuzzy Sets and Systems, 161, 1343–1349 (2010).
7. Harding, J., Walker, C., Walker, E.: Lattices of convex normal functions. Fuzzy Sets and Systems, 159, 1061–1071 (2008).
8. Hernández, P., Cubillo, S., Torres, C.: Negations on type-2 fuzzy sets. Fuzzy Sets and Systems, en revisión.
9. Hernández, P., Cubillo, S., Torres, C.: About t-norms on type-2 fuzzy sets. In: 8th Conference of the European Society for Fuzzy Logic and Technology, EUSFLAT 2013, Aceptado, por aparecer.
10. Klement, P., Mesiar, R., Pap, E.: Triangular Norms. Kluwer Academic Publishers, Dordrecht, The Netherlands (2000).
11. Mendel, J., Jhon, R.: Type-2 fuzzy sets made Simple. IEEE Trans. Fuzzy Systems, 10(2), 117–127 (2002).
12. Menger, K.: Statical metrics. Proc. Nat. Acad. Sci. U.S.A., 37, 535-537 (1942).
13. Mizumoto, M., Tanaka, K.: Fuzzy sets of type-2 under algebraic product and algebraic sum. Fuzzy Sets and Systems, 5, 277–290 (1981).
14. Mizumoto, M., Tanaka, K.: Some properties of fuzzy sets of type-2. Inf. Control, 31, 312–340 (1976).
15. Ray, S.: Representation of a Boolean algebra by its triangular norms. Matheware and Soft Computing, 4, 63–68 (1997).
16. Schweizer, B., Sklar, A.: Associative functions and statistical triangle inequalities. Publ. Math., 8, 169–186 (1961).
17. Schweizer, B., Sklar, A.: Statistical metric spaces. Pacific J. Math., 10, 313–334 (1960). .
18. Walker, C., Walker, E.: Some general comments on fuzzy sets of type-2. Internat. J. Intell. Systems, 24, 62–75 (2009).
19. Walker, C., Walker, E.: The algebra of fuzzy truth values, Fuzzy Sets and Systems, 149, 309–347 (2005).
20. Walker, C., Walker, E.: T-norms for type-2 fuzzy sets. In: Proc. Internat. Conf. on Fuzzy Systems IEEE 2006, pp. 1235–1239, July 16–21, Vancouver (Canadá) (2006).
21. Zadeh, L.: Fuzzy sets. Inf. Control, 20, 301–312 (1965).
22. Zadeh, L.: The concept of a linguistic variable and its application to approximate reasoning. Inf. Sci., 8, 199–249 (1975).

Sistema de Apoyo al Consenso para Problemas de Toma de Decisión en Grandes Grupos basado en una Plataforma Multi-Agente

Iván Palomares*, Francisco J. Quesada, Luis Martínez

Universidad de Jaén, Departamento de Informática,
Campus Las Lagunillas s/n, 23071 Jaén
{ivanp,fqreal,martin}@ujaen.es

Resumen Los procesos de consenso para la resolución de problemas de Toma de Decisión en Grupo buscan alcanzar un alto nivel de acuerdo colectivo antes de tomar una decisión. La participación de grandes grupos en dichos problemas es cada vez más frecuente, lo que hace necesario el empleo de nuevos modelos y técnicas para reducir el elevado coste del proceso de consenso en tales casos, así como el desarrollo de sistemas de apoyo al consenso basados en arquitecturas escalables, que faciliten computacionalmente el manejo de dichos grupos. En esta contribución presentamos un sistema de apoyo al consenso basado en una arquitectura multi-agente para problemas de toma de decisión en grandes grupos, que incorpora un modelo de autonomía semi-supervisada basado en agentes con el objetivo de minimizar el coste invertido en el proceso de consenso.

1. Introducción

En los problemas de Toma de Decisión en Grupo (TDG), dos o más expertos intentan alcanzar una solución a un problema compuesto por un conjunto de alternativas o posibles soluciones a dicho problema [4, 6]. Tradicionalmente, los problemas de TDG se han resuelto aplicando un proceso de selección de alternativas, sin tener en cuenta el nivel de acuerdo entre los expertos, lo que implica que la solución alcanzada puede no ser aceptada por algunos de ellos [1]. Los procesos de alcance de consenso, en los que los expertos discuten y modifican sus preferencias, guiados por un moderador, para alcanzar un alto nivel de acuerdo antes de tomar una decisión [13], surgieron para solventar este inconveniente.

Los procesos de consenso se han convertido en un importante campo de investigación dentro de la TDG en las últimas décadas. Como resultado de ello, se han propuesto diferentes modelos teóricos de consenso en la literatura [10,13,15]. En base a dichos modelos, se han implementado múltiples Sistemas de Apoyo al Consenso (SAC) [9, 16], con el objetivo de facilitar los procesos de consenso. Algunos de estos SAC reemplazan a la figura del moderador humano, lo que supone una automatización de las tareas llevadas a cabo por el mismo.

* corresponding author.

A pesar de los logros alcanzados gracias a los modelos y SAC propuestos, los procesos de consenso aún presentan algunas limitaciones, tales como la necesidad de manejar grandes grupos, cuya existencia es cada vez más frecuente debido a la irrupción de nuevos paradigmas y medios, tales como redes sociales [14] y e-democracia [5], dando así lugar a problemas de TDG con grandes grupos [8]. En dichos problemas, la existencia de expertos o subgrupos con opiniones muy diferentes entre sí es especialmente frecuente. Como consecuencia de ello, el proceso de consenso podría suponer un alto coste por parte de los expertos en revisar y modificar sus preferencias y, por consiguiente, una mayor cantidad de tiempo invertido en alcanzar un acuerdo. Por ello, resulta conveniente dotar a los modelos y SAC actuales de mecanismos que minimicen la necesidad de supervisión de preferencias por parte de los expertos, automatizando dicha supervisión en aquellos casos en los que ésta no implique un cambio importante en sus opiniones. Por otro lado, surge la necesidad de desarrollar SAC basados en una arquitectura altamente escalable y distribuida, tal y como las basadas en el paradigma de los Sistemas Multi-Agente [11], que sea capaz de soportar computacionalmente y de facilitar procesos de consenso con grandes grupos.

En esta contribución presentamos un SAC para la resolución de problemas de TDG con grandes grupos, basado en una arquitectura multi-agente. Dicho sistema incorpora un modelo de autonomía semi-supervisada, que permite a los expertos delegar en agentes software algunas de las supervisiones que éstos deben realizar sobre sus preferencias durante el proceso de consenso, reduciendo así el coste invertido en el mismo. La arquitectura multi-agente está dotada de una interfaz que permite una comunicación con los expertos de forma asíncrona.

Este trabajo se estructura como sigue: en la Sección 2 se introducen los preliminares necesarios. En la Sección 3, presentamos el SAC, mostrando su arquitectura y el modelo de autonomía semi-supervisada que incorpora, así como un ejemplo de su utilización. Por último, concluimos el trabajo en la Sección 4.

2. Preliminares

En esta sección se revisarán algunos conceptos básicos sobre TDG y procesos de alcance de consenso, necesarios para entender las principales características de nuestra propuesta.

2.1. Problemas de Toma de Decisión en Grupo

Los problemas de TDG se caracterizan por la existencia de múltiples expertos, con diferentes puntos de vista, que deben alcanzar una solución común a un problema de toma de decisión [6]. Formalmente, un problema de TDG se compone de los siguientes elementos [1, 4]:

- Un conjunto $X = \{x_1, \ldots, x_n\}$ $(n \geq 2)$ de *alternativas* o posibles soluciones al problema.
- Un conjunto $E = \{e_1, \ldots, e_m\} (m \geq 2)$ de individuos o *expertos*, que dan su opinión sobre las alternativas en X mediante una estructura de preferencia.

Una de las estructuras de preferencia más utilizadas en problemas de TDG bajo incertidumbre es la *relación de preferencia difusa* [7]. Una relación de preferencia difusa P_i asociada al experto e_i se caracteriza por una función de pertenencia $\mu_{P_i} : X \times X \to [0,1]$, y se representa, para X finito, mediante una matriz de dimensión $n \times n$ como:

$$P_i = \begin{pmatrix} p_i^{11} & \dots & p_i^{1n} \\ \vdots & \ddots & \vdots \\ p_i^{n1} & \dots & p_i^{nn} \end{pmatrix}$$

donde cada valoración $p_i^{lk} = \mu_{P_i}(x_l, x_k) \in [0,1]$ representa el grado de preferencia de la alternativa x_l sobre x_k según el experto e_i, de forma que $p_i^{lk} > 0{,}5$ indica preferencia de x_l sobre x_k, $p_i^{lk} < 0{,}5$ indica preferencia de x_k sobre x_l, y $p_i^{lk} = 0{,}5$ indica indiferencia entre ambas alternativas.

Los procesos clásicos de selección para la resolución de problemas de TDG [3] se componen de una *fase de agregación*, en la que se combinan las preferencias de los expertos, y una *fase de explotación*, donde se obtiene una alternativa o subconjunto de alternativas como solución al problema [12].

2.2. Procesos de Alcance de Consenso

Al resolver problemas de TDG puede surgir el inconveniente de que algunos expertos no estén de acuerdo con la decisión tomada, porque piensen que sus opiniones no han sido tenidas en cuenta. Los *procesos de consenso* surgieron para solucionar este inconveniente, mediante la discusión entre expertos y la modificación de sus preferencias, con el objetivo de acercarlas entre sí y conseguir un alto nivel de acuerdo antes de tomar la decisión [13].

Existen diferentes nociones y enfoques del consenso, uno de los más extendidos es el de *soft consensus* [4], propuesto por Kacprzyk. Este enfoque considera diferentes niveles de acuerdo parciales en grupos (que suelen medirse como valores en el intervalo unitario) y está basado en el concepto de mayoría difusa, según el cual existe consenso cuando "la mayoría de expertos participantes coincide en su opinión sobre las alternativas más importantes".

El proceso de consenso es un proceso dinámico e iterativo, normalmente coordinado por una figura humana conocida como *moderador*, encargada de supervisar y guiar a los expertos durante dicho proceso [13]. La Figura 1 muestra un esquema general de los procesos de consenso, basándose en la utilización de relaciones de preferencia difusas [9]:

1. *Expresión y Recopilación de preferencias*: Cada experto e_i proporciona al moderador una relación de preferencia difusa P_i sobre X.
2. *Determinar grado de consenso*: El moderador calcula el nivel de acuerdo alcanzado en el grupo, $cr \in [0,1]$, mediante medidas de consenso, normalmente basadas en diferentes medidas de similitud y operadores de agregación.
3. *Control del consenso*: Se compara cr con un umbral de consenso mínimo deseado por el grupo, $\mu \in [0,1]$. Si es suficiente, se ha alcanzado consenso y el grupo pasa al proceso de selección; en caso contrario, es necesario llevar a

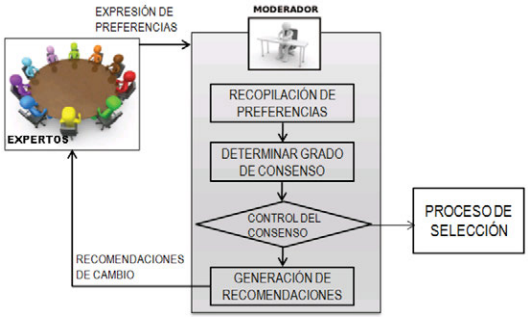


Figura 1: Esquema general de los procesos de consenso

cabo otra ronda de discusión. Para evitar un excesivo número de rondas, se puede utilizar un parámetro, $Maxround \in \mathbb{N}$, que indica el máximo número de rondas de discusión permitidas sin haber alcanzado un acuerdo.

4. *Generación de Recomendaciones*: El moderador determina aquellas valoraciones, p_i^{lk}, de los expertos e_i que se encuentran más alejadas del consenso en la ronda actual, y recomienda a dichos expertos su modificación (aumentar o disminuir el valor de dichas valoraciones) para aumentar el grado de acuerdo en la siguiente ronda.

3. Sistema de Soporte al Consenso para TDG con Grandes Grupos

En esta sección presentamos un SAC basado en una arquitectura multi-agente, para la resolución de problemas de TDG con grandes grupos. Dicho sistema supone una extensión del presentado en [9], e introduce dos novedades con el objetivo de facilitar el manejo de grandes grupos en los procesos de consenso:

- Un *modelo de autonomía semi-supervisada*, que permite a los expertos delegar parte de las tareas de supervisión y modificación de sus preferencias en agentes software, los cuales realizarán dichas tareas de forma autónoma con el fin de minimizar la cantidad de supervisión humana requerida durante el proceso de consenso.
- Una *interfaz Web*, para facilitar la participación de expertos físicamente separados en procesos de consenso. Dicha interfaz permitirá al experto enviar y revisar sus preferencias, además de brindarle la posibilidad de seleccionar un perfil de comportamiento, que será implementado por un agente encargado de modificar sus valoraciones, como se verá en la Sección 3.1.

La descripción del sistema se divide en tres partes: (i) el modelo de autonomía semi-supervisada propuesto, (ii) la arquitectura y principales funcionalidades del sistema y (iii) un ejemplo de su funcionamiento.

3.1. Modelo de Autonomía Semi-supervisada basado en Agentes

La constante supervisión de recomendaciones de cambio por parte de los expertos (fase *Generación de Recomendaciones*, Sec. 2.2) puede causar algunos problemas, tales como un elevado coste temporal y la posibilidad de que algunos expertos pierdan interés y motivación en seguir participando en el proceso de consenso. Para evitar estos problemas, proponemos un modelo de autonomía semi-supervisada basado en un conjunto de *perfiles de cambio* y *reglas de supervisión*, que serán implementados por un conjunto de *agentes expertos* [9], encargados de realizar parte de las tareas propias de los expertos humanos, emulando su comportamiento y modificando sus preferencias de forma semi-autónoma durante el proceso de consenso. Cabe destacar que, a pesar de que existen modelos que proponen una total automatización de las tareas llevadas a cabo por los expertos [15], nuestro interés se centra en permitir al propio experto humano supervisar sus preferencias en ciertos casos en los que las recomendaciones de cambios sugeridas supongan un cambio importante en su opinión, preservando así su soberanía en un cierto grado.

A continuación se describen los principales componentes del modelo de autonomía semi-supervisada propuesto: perfiles de cambio y reglas de supervisión.

Perfiles de cambio. Establecen la estrategia adoptada por el experto para modificar sus valoraciones de forma autónoma, cuando su supervisión no es necesaria. En el modelo de consenso subyacente al SAC que presentamos [9], los expertos expresan sus preferencias mediante relaciones de preferencia difusas, y cada recomendación de cambio generada consiste en una terna de la forma $(e_i, (x_l, x_k)$,*Dirección*$)$, que indica que el experto $e_i \in E$, debe modificar su valoración p_i^{lk} en la dirección dada por *Dirección*$\in \{Aumentar, Disminuir\}$.

En los procesos de consenso, los expertos suelen adoptar diferentes estrategias para modificar sus preferencias. Con el fin de emular dichas estrategias, proponemos definir los tres tipos de perfiles siguientes:

1. *Perfil seguro*: Representa a expertos seguros de sus opiniones iniciales. Aplican cambios pequeños al comienzo del proceso de consenso, aunque dichos cambios son mayores conforme aumenta el número de rondas de discusión.
2. *Perfil inseguro*: Representa a expertos más inseguros de sus opiniones iniciales. Aplican grandes cambios al comienzo del proceso de consenso, que serán menores conforme éste avance.
3. *Perfil Neutral*: Representa a expertos moderadamente seguros de sus opiniones iniciales, que prefieren aplicar cambios sobre ellas de manera uniforme durante todo el proceso.

Cada experto selecciona el perfil que mejor refleje su comportamiento, y se lo proporciona a un agente experto que se encargará de automatizar las tareas de supervisión de dicho experto. Los perfiles de cambio se modelan mediante *funciones de cambio* (basadas en funciones de negociación de agentes como *Kasbah* [2]) que determinan el grado de incremento/decremento $\Delta(r)$ sobre una valoración

p_i^{lk}, en función de la ronda actual de consenso $r \in \mathbb{N}$. Así, un perfil de tipo seguro, inseguro o neutral, puede modelarse mediante una función de cambio creciente, decreciente o constante, respectivamente, como muestra la Figura 2.

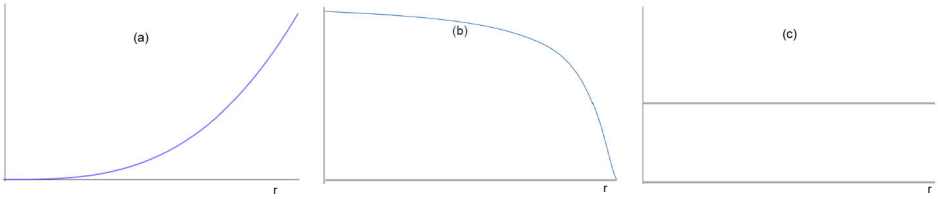


Figura 2: Función de cambio para perfil: (a) seguro, (b) inseguro, (c) neutral.

Reglas de supervisión. Aunque los perfiles de cambio permiten eliminar completamente la necesidad de supervisión humana de los expertos, existen situaciones en las que se proponen cambios significativos en sus valoraciones, por lo que es deseable que el experto supervise dichos cambios para decidir si acepta o no realizarlos. Para detectar este tipo de situaciones, proponemos un mecanismo basado en reglas mediante las cuales los agentes deciden si aplican los cambios sugeridos por el modelo de consenso automáticamente, o bien solicitan la supervisión del experto humano correspondiente.

El SAC propuesto es lo suficientemente flexible para permitir la modificación y/o extensión de las reglas de supervisión a tener en cuenta. En este trabajo consideramos las siguientes reglas de supervisión:

- R.1: SI $p_{i(r+1)}^{lk} > 0{,}5$ Y $p_{i1}^{lk} \leq 0{,}5$ ENTONCES pedir supervisión al experto e_i en su valoración p_{ir}^{lk}.
- R.2: SI $p_{i(r+1)}^{lk} < 0{,}5$ Y $p_{i1}^{lk} \geq 0{,}5$ ENTONCES pedir supervisión al experto e_i en su valoración p_{ir}^{lk}.

donde p_{ir}^{lk} representa la valoración actual de e_i sobre (x_l, x_k) en la ronda r, p_{i1}^{lk} es su valoración inicial, y $p_{i(r+1)}^{lk} = p_{ir}^{lk} \pm \Delta(r)$ representa el nuevo valor que ésta tomaría al aceptar la recomendación, según el perfil de cambio elegido por el experto. En otras palabras, el sistema requerirá la supervisión de los expertos siempre que una recomendación sobre una valoración implique un cambio en la preferencia sobre la alternativa, respecto a su opinión inicial.

3.2. Arquitectura del Sistema

El SAC que presentamos está basado en una arquitectura multi-agente, y en una arquitectura cliente/servidor provista de una interfaz Web de comunicación con el usuario. La Figura 3 muestra un esquema simplificado de la arquitectura del sistema, centrado en el flujo de comunicación con los usuarios.

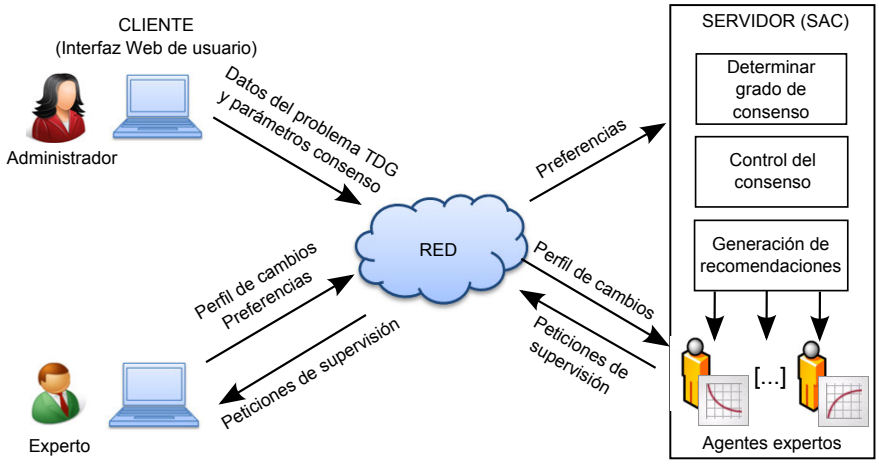


Figura 3: Arquitectura del sistema.

Para la implementación de la plataforma multi-agente, se ha utilizado la plataforma de desarrollo de agentes JADE[1]. Los agentes del SAC automatizan totalmente las tareas del moderador humano [9], lo que facilita llevar a cabo procesos de consenso en problemas de TDG con grandes grupos. La interfaz Web permite una comunicación asíncrona entre los usuarios y agentes del sistema, y su implementación está basada en Servlets y JSP.

El sistema distingue dos tipos de usuario: usuario experto y usuario administrador. A continuación se indican las funcionalidades ofrecidas por el sistema a cada uno de ellos:

Usuario Experto. Las principales funciones que puede realizar un usuario experto son:

- *Introducir preferencias*: El usuario experto introduce sus valoraciones iniciales sobre las alternativas del problema de TDG considerado, y las envía al sistema (Fig. 4). Además, al comienzo del problema, el experto podrá elegir el perfil de cambio deseado en dicho problema.
- *Supervisar recomendaciones de cambio*: El sistema envía al usuario experto peticiones para supervisar aquellas recomendaciones de cambio que impliquen un cambio importante en alguna de sus valoraciones, permitiendo a éste aceptar o ignorar cada recomendación.
- *Modificar perfil de usuario*: El usuario experto dispone de un perfil modificable en el sistema, con información sobre sus datos personales y perfil de cambio seleccionado por defecto.
- *Consultar información del problema*: Durante el proceso de consenso, el usuario experto puede consultar la información sobre el estado actual del problema de TDG, así como la ronda de discusión actual y el grado de consenso alcanzado.

[1] http://jade.tilab.com

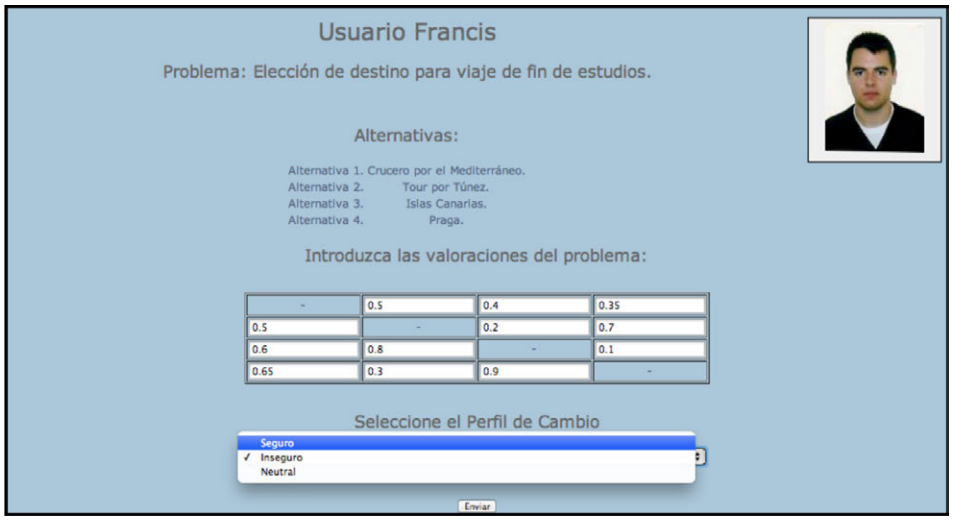


Figura 4: Introduciendo preferencias iniciales y perfil de cambio.

Usuario Administrador. El usuario administrador es responsable de realizar las siguientes acciones:

- *Crear un problema*: Consiste en definir un nuevo problema de TDG y los parámetros de éste, incluyendo: conjunto de alternativas, usuarios expertos invitados a participar y parámetros del proceso de consenso.
- *Gestión de expertos*: El usuario administrador puede crear, modificar o eliminar a los usuarios expertos existentes en el sistema.

3.3. Ejemplo de Aplicación

En este apartado mostramos un ejemplo de utilización del SAC propuesto para resolver un problema de TDG mediante consenso, con el objetivo de mostrar las ventajas del modelo de autonomía semi-supervisada presentado.

El enunciado del problema de TDG es el siguiente: un grupo de 23 estudiantes de Ingeniería Informática, $E = \{e_1, \ldots, e_{23}\}$, desea alcanzar un acuerdo para decidir el destino de su viaje de fin de estudios. Los posibles destinos son, $X = \{x_1$: Crucero por el Mediterráneo, x_2 : Tour en Túnez, x_3 : Islas Canarias y x_4 : Praga$\}$. El grupo desea llegar a un nivel de acuerdo $\mu = 0{,}85$, y el máximo número de rondas de discusión es $Maxround = 10$. Los estudiantes eligieron los siguientes perfiles de cambio: 4 seguros, 9 inseguros y 10 neutrales. En el Cuadro 1 se muestran las funciones de cambio $\Delta(r)$ asociadas a dichos perfiles.

Para la resolución del problema, se reunió a los alumnos en un laboratorio de prácticas, donde accedieron al sistema e introdujeron sus opiniones iniciales (véase Fig. 4). Tras cada ronda de consenso, cada alumno recibe una petición para realizar las supervisiones necesarias sobre sus valoraciones (en su caso). Los agentes expertos envían dichas peticiones si se cumple alguna de las reglas anteriormente definidas en Sec. 3.1, en otro caso realizan los cambios propuestos automáticamente.

Cuadro 1: Funciones de cambio consideradas.

Perfil de cambio	**Función de cambio** $\Delta(r)$
Seguro	$0{,}2\left(\frac{r}{10}\right)^3$
Inseguro	$0{,}2\left(1-\left(\frac{r}{10}\right)^3\right)$
Neutral	0.1

Los resultados obtenidos en cada ronda de consenso, r, se recogen en el Cuadro 2, y consisten en el grado de consenso alcanzado, el número total de recomendaciones de cambio generadas, el número de supervisiones necesarias y el número de expertos que han tenido que realizar alguna supervisión en dicha ronda. Los resultados muestran que el número de supervisiones requeridas es muy inferior al número total de recomendaciones de cambio generadas en cada ronda, por lo que la mayoría de ellas no supone un cambio importante en las valoraciones. Además, solo una pequeña proporción de expertos debe realizar alguna supervisión de sus preferencias durante todo el proceso. Se concluye que el SAC presentado contribuye a reducir el coste temporal invertido en los procesos de consenso para problemas de TDG con grandes grupos.

Cuadro 2: Resultados del proceso de consenso.

r	Grado consenso	#Recomendaciones	#Supervisiones	#Expertos sup.
1	0.6484	108	**0**/108	**0**/23
2	0.7317	82	**5**/82	**4**/23
3	0.7811	65	**25**/65	**6**/23
4	0.8100	55	**14**/55	**6**/23
5	0.8351	13	**3**/13	**2**/23
6	**0.8502**			

4. Comentarios Finales

En este trabajo hemos presentado un sistema de apoyo al consenso basado en una arquitectura multi-agente, para facilitar la resolución de procesos de consenso en problemas de toma de decisión con grandes grupos. Dicho sistema está dotado de una interfaz Web que permite llevar a cabo procesos de consenso no presenciales, y se caracteriza por incorporar un modelo de autonomía semi-supervisada que elimina el problema de la supervisión constante de preferencias por parte de los expertos.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el Proyecto de Investigación TIN-2012-31263 y fondos ERDF.

Referencias

1. C.T.L. Butler and A. Rothstein. *On Conflict and Consensus: A Handbook on Formal Consensus Decision Making.* Takoma Park, 2006.
2. A. Chavez and P. Maes. Kasbah: An agent marketplace for buying and selling goods. *Proceedings of the 1st International Conference on the Practical Appication of Intelligent Agents and Multi-Agent Technology*, pages 75–90, 1996.
3. F. Herrera, E. Herrera-Viedma, and J. Verdegay. A sequential selection process in group decision making with linguistic assessments. *Information Sciences*, 85(1995):223–239, 1995.
4. J. Kacprzyk. Group decision making with a fuzzy linguistic majority. *Fuzzy Sets and Systems*, 18(2):105–118, 1986.
5. J. Kim. A model and case for supporting participatory public decision making in e-democracy. *Group Decision and Negotiation*, 17(3):179–192, 2008.
6. J. Lu, G. Zhang, D. Ruan, and F. Wu. *Multi-Objective Group Decision Making.* Imperial College Press, 2006.
7. S.A. Orlovsky. Decision-making with a fuzzy preference relation. *Fuzzy Sets and Systems*, 1(3):155–167, July 1978.
8. I. Palomares, R.M. Rodríguez, and L. Martínez. An attitude-driven web consensus support system for heterogeneous group decision making. *Expert Systems with Applications*, 40(1):139–149, 2013.
9. I. Palomares, P. Sánchez, F. Quesada, F. Mata, and L. Martínez. *Comas: A Multiagent System for Performing Consensus Processes, in Abraham, Ajith; et al. (Eds.) Advances in Intelligent and Soft Computing*, volume 91, pages Springer, 125–132. 2011.
10. R.O. Parreiras, P. Ekel, J.S.C. Martini, and R.M. Palhares. A flexible consensus scheme for multicriteria group decision making under linguistic assessments. *Information Sciences*, 180(7):1075–1089, 2010.
11. D.L. Poole and A.K. Mackworth. *Artificial Intelligence: Foundations of Computational Agents.* Cambridge University Press, 2010.
12. M. Roubens. Fuzzy sets and decision analysis. *Fuzzy Sets and Systems*, 90(2):199–206, 1997.
13. S. Saint and J. R. Lawson. *Rules for Reaching Consensus. A Modern Approach to Decision Making.* Jossey-Bass, 1994.
14. C. Sueur, J.L. Deneubourg, and O. Petit. From social network (centralized vs. decentralized) to collective decision-making (unshared vs. shared consensus). *PLoS one*, 7(2):1–10, 2012.
15. Z. Xu. An automatic approach to reaching consensus in multiple attribute group decision making. *Computers & Industrial Engineering*, 56(4):1369–1374, 2009.
16. S. Zadrozny and J. Kacprzyk. *An Internet-based group decision and consensus reaching support system, in Applied Decision Support with Soft Computing (Studies in Fuzziness and Soft Computing) 124.* pp. 263-275, Springer, 2003.

Estudio comparativo de modelos de un vehículo aéreo obtenidos mediante técnicas analíticas y basadas en redes neuronales

J. Enrique Sierra[1], Matilde Santos[2]

[1] Universidad de Burgos, Burgos, España
jesiegar@gmail.com
[2] Universidad Complutense de Madrid, Madrid, España
msantos@ucm.es

Resumen. A partir de datos de entrada/salida reales de un vehículo aéreo no tripulado (UAV) se han generado diferentes modelos mediante distintas técnicas de identificación de sistemas, incluyendo las que proporciona el soft computing. El dispositivo que se ha utilizado es un cuatrirotor, por lo tanto un sistema complejo, no lineal, multivariable y con acoplamientos. Se han analizado y comparado los resultados de la aplicación de técnicas paramétricas y de estrategias inteligentes para la obtención de modelos orientados al control.

Palabras clave: Soft computing, Redes neuronales, técnicas paramétricas, modelización, identificación, vehículos aéreos no tripulados (UAV).

1 Introducción

La identificación de sistemas es uno de los pasos más importantes para obtener modelos útiles y realizables de sistemas complejos. Fundamentalmente existen dos vías para alcanzar este objetivo: la primera de ellas consiste en aplicar las ecuaciones físicas conocidas que gobiernan el comportamiento dinámico y emplear este conocimiento para construir las ecuaciones diferenciales que lo representan. Esta metodología resulta muy útil cuando las variables son perfectamente conocidas y las relaciones entre ellas no son demasiado complejas. La otra vía consiste en la medida y el tratamiento de las señales en la entrada y la salida del sistema, y la aplicación de diferentes mecanismos de mapeado para describir las relaciones internas subyacentes entre las entradas y las salidas. En esta categoría se encuentran, entre otras, algunas de las técnicas proporcionadas por el Soft Computing, como las redes neuronales y los algoritmos genéticos [5].

La modelización de un vehículo aéreo no tripulado, UAV (Unmanned Aerial Vehicle), no es una tarea sencilla. La complejidad proviene de diferentes fuentes: la aleatoriedad de los flujos de aire y de las fuerzas exógenas, la inhomogeneidad de la estructura mecánica interna, y la fuerte no linealidad de la dinámica del sistema. Todos estos factores convierten a las técnicas basadas en el control inteligente en un camino idóneo para modelar estos sistemas. Otros trabajos han aplicado estas estra-

tegias para modelar estos sistemas, tales como [9] y [10]. También se pueden encontrar ejemplos de la aplicación de las técnicas inteligentes para modelar otros sistemas complejos no lineales como, por ejemplo, vehículos marinos [11]. Mediante este trabajo se realizan dos aportaciones novedosas al estudio de los UAVs, la aplicación de las redes neuronales de regresión generalizada como técnica de modelización, y la utilización del error cuadrático medio de la respuesta al escalón, estimada mediante análisis de correlación, como herramienta de contraste de la eficacia del modelo.

Empleando datos reales de entrada/salida de un UAV se han obtenido diferentes modelos basados en redes neuronales. También se han utilizado métodos tradicionales de identificación y los resultados se han comparado.

El vehículo a modelar es el ARDRONE 2.0 comercializado por Parrot (figura 1). Este aparato se fabrica en fibra de carbono y plástico PA66. Está equipado con cuatro motores tipo brushless de alta eficiencia, alimentados por una batería de litio de 11.1 V y 1000 mAh que le confiere una autonomía de vuelo de aproximadamente 12 minutos. La masa es de 380 g sin carcasa y 420 g con carcasa. Dispone de un acelerómetro de 3 ejes, un giróscopo de 2 ejes (balanceo y cabeceo), y un giróscopo de precisión para el ángulo de guiñada [7].

Fig. 1. Vehículo aéreo no tripulado seleccionado (Fuente: www.parrot.com)

A continuación se detalla la organización del documento. En la sección 2 se desarrolla un análisis preliminar describiendo las señales examinadas y el análisis de correlación. La sección 3 se destina al estudio de los modelos obtenidos mediante técnicas tradicionales. En la sección 4 se aplican las redes neuronales basadas en los datos experimentales. Los resultados son discutidos y comparados en 5. El documento finaliza con las conclusiones.

2 Análisis preliminar

2.1 Señales empleadas

Para realizar el modelado se parte de un conjunto de datos suministrados por el Grupo de Ingeniería de Control de CEA para el concurso de Ingeniería de Control 2013 [2]. El conjunto recoge los siguientes datos: instante de muestreo, pitch, roll, x, vx, y e vy, con un tiempo de muestreo de 60 ms, tomados durante 9 minutos. Nos

centraremos en el modelado de las velocidades vx y vy, puesto que la posición (x,y) viene determinada por la integración de vx y vy. Las señales de entrada y salida se representan en la figura 2. En azul vx y pitch y en verde vy y pitch. Puede adelantarse ya el importante desacoplo entre las variables que se corroborará mediante la correlación.

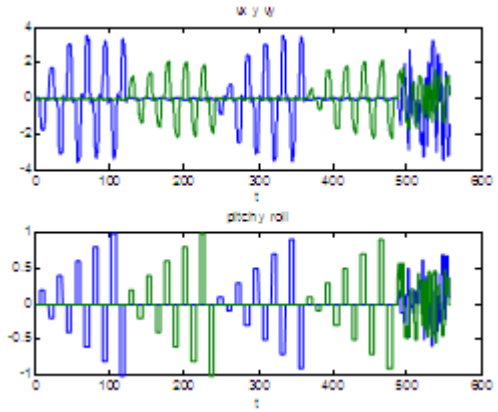


Fig. 2. Señales de entrada y salida

2.2 Análisis de correlación

A partir de la estima de la covarianza entre la entrada y la salida de un sistema y la estima de la varianza de la entrada, puede calcularse la estima de la respuesta al impulso, con las expresiones siguientes:

$$\hat{R}_{yu}(\tau) = \frac{1}{N}\sum y(t)u(t-\tau) \tag{1}$$

$$\hat{\lambda} = \frac{1}{N}\sum u(t)^2 \tag{2}$$

$$g = \frac{\hat{R}_{yu}(\tau)}{\hat{\lambda}} \tag{3}$$

Siendo y la salida del sistema, u la entrada, N el número de datos, R la covarianza, λ la varianza y g la respuesta estimada al impulso. Integrando la respuesta al impulso puede obtenerse la respuesta al escalón, la cual puede ser muy útil para ayudarnos a estimar el grado de acoplamiento entre las variables del sistema. Esta metodología de estudio suele denominarse análisis de correlación [1]. Empleando el método descrito se obtiene la figura 4.

En la respuesta al escalón de la matriz de funciones de transferencia de la figura 4, se aprecia como la ganancia en estado estacionario de los elementos de la diagonal principal, correspondientes al enlace (fila 1, columna 1) y (fila 2, columna 2), son bastante mayores que el resto. Este hecho nos indica que las dinámicas del sistema están parcialmente desacopladas, pudiendo convertir el sistema MIMO (multi input multi output) en dos sistemas SISO (single input single output). Aprovechando este

hecho, los esfuerzos en modelaje en este trabajo se centrarán en la influencia del pitch sobre la velocidad en el eje x.

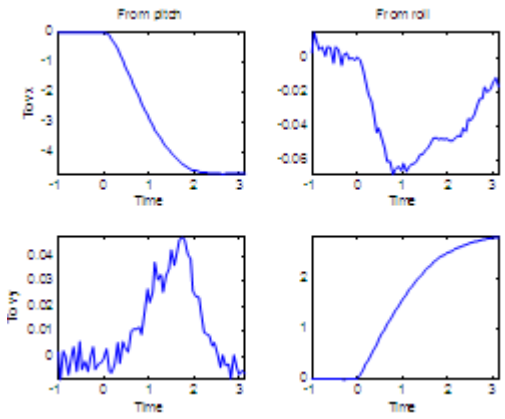


Fig. 3. Respuesta a un escalón estimada del sistema MIMO

3 Modelos paramétricos

3.1 Modelos basados en el espacio de estados

Mediante el modelo en el espacio de estados se describe el comportamiento del sistema a partir de las siguientes expresiones genéricas, [4]:

$$\dot{x}(t) = Ax(t) + Bu(t) + Ke(t) \tag{4}$$

$$y(t) = Cx(t) + Du(t) + e(t) \tag{5}$$

Donde x(t) representa el estado interno del sistema, u(t) las entradas, y(t) las salidas, e(t) una señal de error, y A, B, C, D son polinomios. En las figuras 4 izquierda y derecha se representan respectivamente, la comparación de la respuesta del modelo con los datos reales y la comparación de la estima de la respuesta al escalón respectivamente para orden 4.

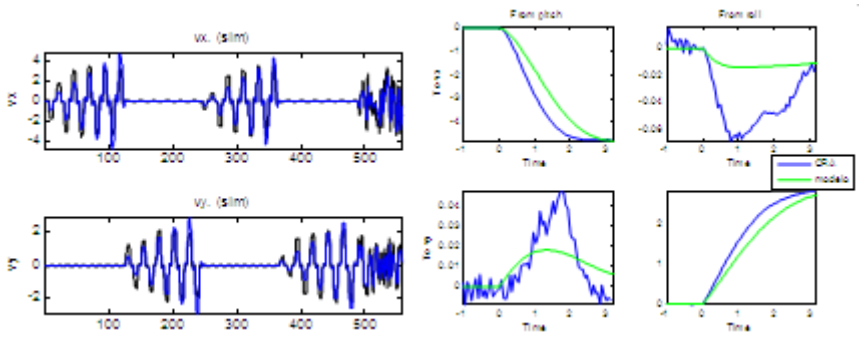


Fig. 4. Simulación del modelo de sistema de estados con orden 4 (izquierda). Comparación respuesta a un escalón del modelo de sistema de estados con orden 4(derecha)

3.2 Modelos ARX

Los modelos ARX (autoregressive exogenous model), descritos en [3], describen el sistema con una ecuación de la siguiente forma:

$$Ay(t) = Bu(t) + e \tag{6}$$

Donde y es la salida, u la entrada y e el error de la estima y A y B son polinomios. Si se restringe el grado máximo de los polinomios a 10, el mejor resultado se consigue con la configuración ARX (10, 10, 1) donde 10 es el grado de A y de B y 1 son las muestras de retardo. En la figura 5 izquierda se representa la comparación de la simulación temporal de los diferentes modelos estudiados con los datos reales y en 5 derecha la comparativa entre la respuesta al escalón estimada mediante análisis de correlación y la proporcionada por los diferentes modelos ARX estudiados

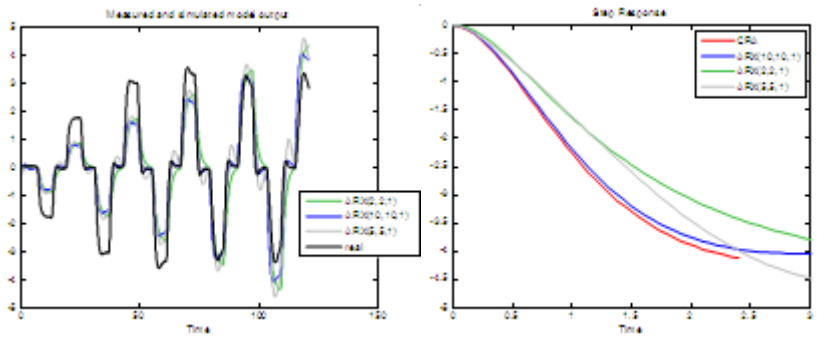


Fig. 5. Comparativa entre los datos reales y simulaciones de los modelos ARX. (izquierda). Comparativa de la respuesta al escalón estimada mediante CRA y los modelos ARX. (derecha)

3.3 Modelos OE

Los modelos OE describen el sistema mediante la expresión:

$$y(t) = \frac{B}{F}u(t)+e \tag{7}$$

Donde B y F son polinomios, e el error de la estima, u la entrada e y la salida del sistema. Si se restringe el grado máximo de los polinomios a 10 el mejor modelo se consigue con el OE (10, 10, 1). Siendo 3 el grado del polinomio B y F y 1 el retardo del sistema

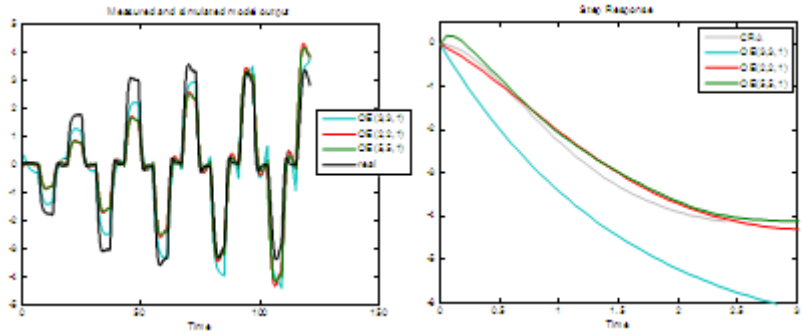


Fig. 6. Comparativa entre los datos reales y simulaciones de los modelos OE. (izquierda). Comparativa de la respuesta al escalón estimada mediante CRA y los modelos OE (derecha)

En la figura 6 izquierda se presenta la comparativa de los datos de salida reales de vx, y el resultado de la simulación temporal para diferentes modelos OE. Mientras que en la figura 6 derecha se presenta la comparación de la estima de la respuesta al escalón obtenida mediante análisis de correlación y la proporcionada por los modelos OE estudiados.

4 Modelos basados en redes neuronales

4.1 Redes neuronales de base radial

Las redes de base radial están formadas por tres capas, una capa de entrada donde se reciben las señales del exterior, una capa oculta conectada con la capa de entrada donde cada elemento tiene asociada una función de base radial y una capa de salida donde se combinan linealmente los resultados de las funciones de activación de la capa oculta [6].

La salida de cada elemento de la capa oculta Zi(n) se calcula como la distancia entre el patrón de entrada al centro del cluster ponderada por di y aplicando este valor a una función de base radial, según la siguiente expresión:

$$z_i(n) = \Phi\left(\frac{\sqrt{\sum(x_j(n)-c_{ij})^2}}{d_i}\right) \tag{8}$$

La combinación lineal de los valores de salida de la capa oculta para generar los valores de salida de la red se realiza mediante la siguiente expresión:

$$y_k(n) = \sum w_{ik} z_i(n) + \mu_k \tag{9}$$

Siendo wik los pesos de la capa de salida y μk los umbrales de activación. Siendo en nuestro caso, la función de base radial Φ:

$$\Phi(n) = e^{-n^2} \tag{10}$$

Para realizar el estudio, el conjunto de datos de entrada de la red se ha formado con la señal de entrada en el instante actual pitch(t) y valores en algunos instantes de muestreo anteriores pitch(t-Ki) donde i = 1…100/K para diferentes valores de K. Y el conjunto de datos objetivo se ha generado con los valores de la señal de salida en el instante actual vx(t).

Se han empleado 5000 muestras de las señales, el 60% se ha empleado para entrenar las redes y el resto para simular.

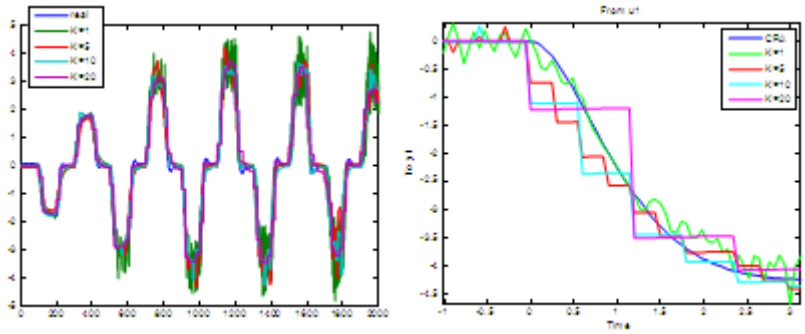


Fig. 6. Comparativa entre los datos reales y simulaciones (izquierda) y de la respuesta al escalón (derecha), para diferentes valores de K de la red de base radial

Como puede apreciarse en las figura 7, donde se comparan los resultados de simulaciones y la respuesta al escalón conforme se aumenta K, la estima de la respuesta al escalón es mejor pero como contrapartida se aumenta el ruido en la simulación temporal.

4.2 Redes neuronales de regresión generalizada

Las redes de regresión generalizada suponen una modificación sobre las redes de base radial. En la capa de salida se realiza una normalización de los valores obtenidos en la capa oculta y una ponderación para obtener los valores de salida de la red. Estas redes resultan especialmente útiles en la aproximación de funciones [8]. Esta característica de especialidad puede observarse en la figura 8 donde la aproximación a los valores reales para diferentes K es prácticamente perfecta. En el caso de la respuesta al impulso de la figura 8, se puede observar como la aproximación es muy buena para todos los valores de K, la diferencia consiste en el tamaño de los escalones empleados, el cual crece conforme K aumenta

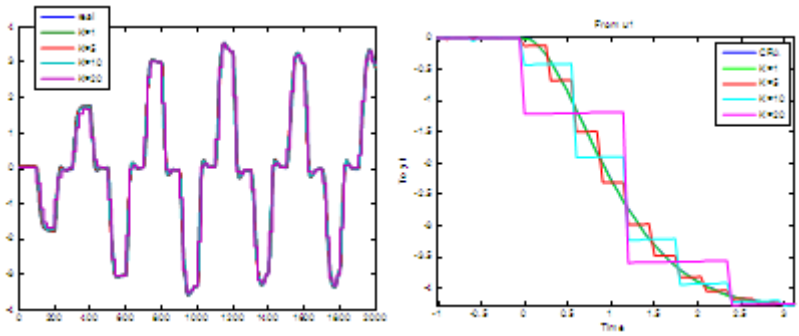


Fig. 7. Comparativa entre los datos reales y simulaciones (izquierda) y de la respuesta al escalón (derecha), para diferentes valores de K de la red de regresión generalizada

4.3 Redes neuronales feed-forward backpropagation

Estas redes neuronales son más simples que las anteriores. Están formadas por varias capas, y cada capa es similar a un perceptrón pero substituyendo la función de activación de tipo escalón por un sigmoide [5]. Para la comparación se ha empleado una red de 5 capas, con un algoritmo de entrenamiento de Levenberg-Marquardt y como función de aprendizaje el gradiente descendiente. En la figura 9 izquierda se presenta la comparativa de las simulaciones obtenidas para diferentes valores de K. Por otra parte, en la figura 9 derecha se presenta la comparativa de la respuesta el escalón obtenida con los modelos ensayados para diferentes valores de K

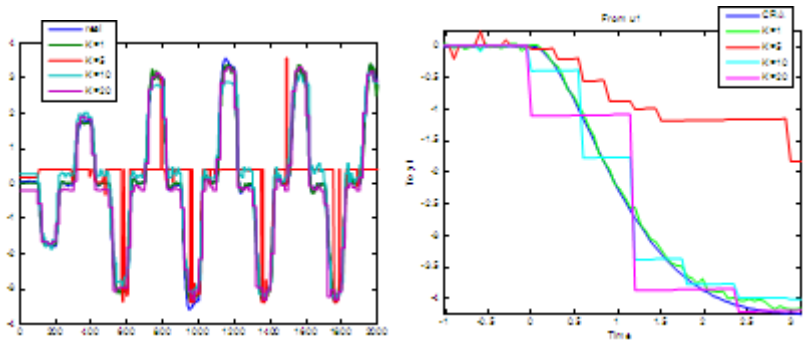


Fig. 8. Comparativa entre los datos reales y simulaciones (izquierda) y de la respuesta al escalón (derecha), para diferentes valores de K de la red feed-forward backpropagation

Se aprecia en las figura 9 y 9 como para K=5 la estima de la respuesta al escalón se aleja bastante de la obtenida mediante el análisis de correlación.

5 Resultados

La tabla 1 muestra los resultados cuantitativos en términos de error cuadrático medio, MSE (Mean Squared Error) para cada técnica estudiada. Tanto los resultados gráficos de las secciones anteriores como los cuantitativos de esta sección, muestran que las redes de regresión generalizada presentan el mejor comportamiento.

	Modelos paramétricos			Modelos basados en redes neuronales		
Technique	Espacio Estados orden 4	ARX 10,10,1	OE 10,10,1	Feedforward Network	Radial Basis Network	Generalized Regression Network
MSE	0.6439	0.5056	0.4675	0.0541	0.1477	2.68E-5
MSE step response	0.1478	0.0122	0.1182	0.0319	0.0602	1.35E-5

Tabla 1. Comparación del MSE para valores en el dominio del tiempo y la respuesta al escalón estimada.

6 Conclusiones

En este trabajo se ha analizado la posibilidad de modelar un vehículo aéreo no tripulado usando diferentes técnicas de identificación de sistemas. Se han comparado técnicas paramétricas y modelos basados en redes neuronales. En el dominio del tiempo, se ha comprobado el error entre las salidas reales y las salidas del modelo, y el error entre la respuesta al escalón estimada obtenida de datos reales y la respuesta al escalón estimada obtenida del modelo.

Los UAVs son sistemas muy complejos de modelar, la complejidad deriva de la aleatoriedad de los flujos de aire y de las fuerzas exógenas y de la dinámica fuertemente no lineal. Estos factores señalan al control inteligente como una estrategia eficiente para modelar estos sistemas. Para a llevar a cabo esta tarea, se han implementado un conjunto de herramientas informáticas que nos permitirán modelar nuevos tipos de UAVs acelerando el proceso de desarrollo, reduciendo por tanto el tiempo y el coste.

Los resultados son altamente satisfactorios: el MSE obtenido para las redes de regresión generalizada indica que el modelo proporciona un comportamiento excelente tanto para la secuencia temporal como para la respuesta al escalón. En última instan-

cia, se ha contrastado la idoneidad de las técnicas basadas en redes neuronales para identificar y modelar estos sistemas.

7 Agradecimientos

A los autores les gustaría agradecer al Grupo de Ingeniería de Control del Comité Español de Automática [2] la aportación del fichero con datos reales empleado para modelar el UAV.

Referencias

1. Bendat, J.S., Piersol, A.G.,: Engineering Applications of Correlation and Spectral Analysis. , Wiley-Interscience, New York (1980)
2. Control Engineering Group of the Spanish Committee of Automatic: www.ceautomatica.es/og/ingenieria-de-control (2013)
3. Isaksson,A.J.: Identification of ARX-models Subject to Missing Data. In: IEEE Transactions on Automatic Control (1993)
4. Kim,C.J., Nelson,C.R.: State-space models with regime switching: classical and Gibbs-sampling approaches with applications. In: MIT Press Books, (2003)
5. Norgard, P.M. et al: Neural Networks for Modelling and Control of Dynamic Systems – A Practitioner's Handbook. Springer,London (2000)
6. Park, J., Sandberg, I.W,: Universal Approximation Using Radial-basis-function Networks. In: Neural Computation (1991)
7. Parrot . www.parrot.com (2013)
8. Rutkowski, L.,: Generalized Regression Neural Networks in Time-varing Environment. In: IEEE Transactions on Neural Networks (2004)
9. San Martin,R. et al: Neural Networks Training Architecture for UAV Modelling. In: World Automation Congress (WAC) , Budapest, Hungary (2006, July 24-26)
10. San Martin, R. et al: Unmanned Aerial Vehicle (UAV) Modelling Based on Supervised Neural Networks. In: Proceedings of the 2006 IEEE International Conference on Robotics and Automation Orlando, Florida (May 2006)
11. Santos, M., López, R., de la Cruz, J.M.:A Neuro-Fuzzy Approach to Fast Ferry Vertical Motion Modelling. In: Engineering Application of Artificial Intelligence, vol. 19, pp.313-321, (2006)

Sobre las interacciones: distancias e integrales

Vicenç Torra[1]

IIIA, Institut d'Investigació en Intel·ligència artificial,
CSIC, Consejo Superior de Investigaciones Científicas,
Campus Universitat Autònoma de Barcelona s/n,
08193 Bellaterra, Catalunya, Spain

Abstract. La definición de independencia de variables aleatorias se basa en la distribución de probabilidad conjunta. La distribución gausiana multivariante es un tipo de distribución conjunta en el que las variables pueden no ser independientes. Sin embargo son concebibles otros tipos de relaciones diferentes a la distribución gausiana. En un trabajo reciente mostramos que las medidas difusas pueden utilizarse para definir distribuciones de probabilidad con interacciones entre variables. En este trabajo se presenta un resumen de nuestros resultados recientes en este tipo de distribuciones.
Palabras clave: Distribuciones de probabilidad, integral de Choquet, distancia de Mahalanobis, operadores de agregación, independencia entre variables aleatorias.

1 Introducción

La diferencia fundamental entre las integrales difusas [8] y las medias ponderadas es que las primeras permiten combinar la información teniendo en cuenta las interacciones entre las fuentes de información. Estas interacciones no pueden tenerse en cuenta en la media ponderada. En las integrales difusas la interacción se expresa mediante las medidas difusas.

En el mundo de las distancias, la cuestión de la interacción entre variables aparece también en la distancia de Mahalanobis [5]. Esta distancia tiene en cuenta la matriz de covariancias y, por ello, la distancia se ve afectada por las interacciones entre variables.

Así pues, tenemos dos objetos matemáticos que permiten tener en cuenta la interacción, uno es la distancia de Mahalanobis y el otro la integral difusa. El primero expresa las interacciones mediante la matriz de covariancias (y, por tanto, en términos de correlaciones) y el segundo mediante las medidas difusas.

Dada esta situación uno puede plantearse si es posible considerar un marco en el que ambos aspectos estén relacionados. En este trabajo se resumen algunas aportaciones en este sentido.

En la sección 2 repasamos algunas definiciones que necesitamos en este trabajo. En particular, revisamos los conceptos de distancia de Mahalanobis, de medida difusa y de integral de Choquet [2]. En la sección 3 presentamos la distribución de probabilidad gaussiana y una distribución de probabilidad basada

en la integral de Choquet. En la sección 4 presentamos una generalización de la distancia de Mahalanobis y la integral de Choquet, que llamamos el operador Choquet-Mahalanobis. Esta definición nos permite definir una nueva distribución de probabilidad. El artículo acaba con unas conclusiones y algunas líneas de trabajo futuro.

2 Algunas definiciones

Empezamos esta sección definiendo la distancia de Mahalanobis [5]. Esta distancia está basada en la matriz de covariancias.

Sean $Y_1, \ldots, Y_n$ variables aleatorias, y sean $\bar{Y}_i = E(X_i)$ para $i = 1, \ldots, n$; entonces, la matriz de covariancias Σ se define como:

$$\Sigma = \begin{bmatrix} \mathrm{E}[(Y_1 - \bar{Y}_1)(Y_1 - \bar{Y}_1)] & \cdots & \mathrm{E}[(Y_1 - \bar{Y}_1)(Y_n - \bar{Y}_n)] \\ \vdots & \ddots & \vdots \\ \mathrm{E}[(Y_n - \bar{Y}_n)(Y_1 - \bar{Y}_1)] & \cdots & \mathrm{E}[(Y_n - \bar{Y}_n)(Y_n - \bar{Y}_n)] \end{bmatrix}.$$

En este trabajo nos interesa más la notación alternativa siguiente:

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \rho_{12}\sigma_{12} & \cdots & \rho_{1n}\sigma_{1n} \\ \rho_{21}\sigma_{21}^2 & \sigma_2^2 & \cdots & \rho_{2n}\sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{n1}\sigma_{n1}^2 & \rho_{n2}\sigma_{n2} & \cdots & \sigma_n^2 \end{bmatrix}.$$

donde σ_i es la desviación estándard de Y_i y $\rho_{i,j}$ es el coeficiente de correlación entre las variables Y_i y Y_j.

Desde un punto de visto matemático, la matriz de covariancias es una matriz semidefinida positiva, y, por tanto, también simétrica. Además, se sabe que cada matriz semi-definida positiva es una matriz de covariancias.

De acuerdo con lo dicho mas arriba, la distancia de Mahalanobis se define en términos de una matriz de covariancias. Damos la definición a continuación.

Definición 1 *Sean $Y_1, \ldots, Y_n$ variables aleatorias; sean a y b dos vectores en $\mathbb{R}^n$, sea Σ la matriz de covariancia de $Y_1, \ldots, Y_n$; entonces, la distancia de Mahalanobis entre a y b se define como:*

$$d_M(a, b) = (a - b)^T \Sigma^{-1} (a - b)$$

A continuación definimos el concepto de medida difusa (medida no aditiva) y de integral de Choquet [2].

Definición 2 *Sea* $X = \{x_1, \ldots, x_n\}$ *un conjunto; entonces, una función* $\mu : 2^X \to [0, \infty)$ *es una medida difusa si satisface los axiomas siguientes:*

(i) $\mu(\emptyset) = 0$ *(condiciones de frontera)*
(ii) $A \subseteq B$ *implica* $\mu(A) \leq \mu(B)$ *(monotonia)*

Definimos a continuación la integral de Choquet, que integra una función respecto de una medida difusa.

Definición 3 *Sea* μ *una medida difusa sobre* X*; entonces, la* integral de Choquet *de una función* $f : X \to \mathbb{R}^+$ *respecto a la medida difusa* μ *se define como*

$$(C)\int f d\mu = \sum_{i=1}^{n} [f(x_{s(i)}) - f(x_{s(i-1)})]\mu(A_{s(i)}), \tag{1}$$

donde $f(x_{s(i)})$ *indica que los índices se han permutado de manera que* $0 \leq f(x_{s(1)}) \leq \cdots \leq f(x_{s(n)}) \leq 1$*, y donde* $f(x_{s(0)}) = 0$ *y* $A_{s(i)} = \{x_{s(i)}, \ldots, x_{s(n)}\}$.

3 Distribuciones de probabilidad: expresando interacciones

La distancia de Mahalanobis está estrechamente relacionada con la distribución gausiana o normal en el caso multivariante. En particular, una distribución gausiana en un espacio multivariante $\mathbb{R}^n$ viene representada mediante un vector de medias $\bar{x}$ y una matriz de covariancias Σ. Entonces, dado cualquier punto $x \in \mathbb{R}^n$ la distribución gausiana se define como

$$P(x) = \frac{1}{(2\pi)^{m/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\bar{x})^T \Sigma^{-1}(x-\bar{x})}$$

Aquí x^T representa la matriz traspuesta de x.

Si consideramos la distancia de Mahalanobis dada en la definición 1 podemos reescribir esta probabilidad como sigue:

$$P_{\bar{x},\Sigma}(x) = \frac{1}{(2\pi)^{1/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}d_\Sigma(x,\bar{x})}$$

Podemos así observar que la distribución normal incluye información sobre la relación entre las variables expresada mediante la matriz de covariancias. De igual forma uno puede considerar si es posible definir una distribución que tenga en cuenta la relación entre variables tal como se representa en una integral de Choquet. En [11, 9] se introdujo una distribución de probabilidad que en lugar de la distancia de Mahalanobis se basaba en la integral de Choquet. Así pues, en este tipo de distribución las interacciones entre variables seran expresadas mediante medidas difusas y no mediante la matriz de covariancias.

Definimos a continuación estas nuevas distribuciones de probabilidad.

$$P_{\bar{x},\mu}(x) = \frac{1}{K} e^{-\frac{1}{2}CI_\mu((x-\bar{x})\otimes(x-\bar{x}))}$$

donde K es una constante que debemos definir de manera que la función sea una probabilidad, y donde $v \otimes w$ denota el vector obtenido como el producto elemento a elemento de los vectores v y w (i.e., $(v \otimes w) = (v_1 w_1 \dots v_n w_n)$).

A esta distribución de probabilidad basada en la integral de Choquet la llamamos *exponential family of Choquet integral based class-conditional probability-density functions*.

En un intento de integrar los dos conceptos de interacción, hemos definido una distribución de probabilidad que incorpora tanto la matriz de *covariancias* Σ como la medida difusa μ. Escribimos aquí covariancias en cursiva porque, como se ha demostrado, la covariancia entre variables aleatorias construidas a partir de la distribución de probabilidad no corresponde a la *covariancia* indicada por la matriz Σ.

4 Integrales y operadores

Como hemos discutido en las secciones precedentes, las medidas difusas y la matriz de covariancias son dos maneras diferentes de expresar interacciones (o dos maneras de expresar interacciones diferentes). En la sección 3 hemos visto como podiamos definir una distribución de probabilidad que incorpora la información de interacción de las medidas difusas.

Dados los dos tipos de interacciones, uno puede plantearse como considerar los dos tipos en un mismo entorno. Hemos definido el operador Choquet-Mahalanobis con este objetivo. Este operador toma un vector de datos y además tanto una medida difusa como una matriz de *covariancias*. Definimos a continuación este operador.

Definición 4 *Sean $Y_1, \dots, Y_n$ variables aleatorias; sea a un vector en $\mathbb{R}^n$, sea Σ la matriz de covariancias de $Y_1, \dots, Y_n$, sea μ una medida difusa sobre el conjunto $\mathcal{Y} = \{Y_1, \dots, Y_n\}$ y sea $\Sigma^{-1} = LL^T$ la descomposición de Cholesky de su inversa. Entonces, definimos el operador de Choquet-Mahalanobis como sigue:*

$$CMI_{\mu,\Sigma}(a) = CI_\mu(v \otimes w)$$

donde v y w son los vectores definidos como $v = a^T L$ y $w = L^T a$ y donde $v \otimes w$ denota el producto elemento a elemento de los vectores v y w (esto es, $(v \otimes w) = (v_1 w_1 \dots v_n w_n)$.

Cuando Σ es una matriz definida positiva, la descomposición de Cholesky es única. Este es el caso cuando utilizamos una matriz de covariancias Σ para generar una distribución de probabilidad. Por tanto, el operador de Choquet-Mahalanobis está bien definido.

En esta definición utilizamos la descomposición de Cholesky porqué es única para matrices definidas positivas.

La definición del operador Choquet-Mahalanobis nos permite introducir una distancia y una distribución de probabilidad. Empezamos con la definición de la distancia.

Definición 5 *Sea μ una medida difusa y sea Σ una matriz definida positiva. Entonces definimos la distancia de Choquet-Mahalanobis como sigue:*

$$d_{\mu,\Sigma}(x,y) = CMI_{\mu,\Sigma}(x-y).$$

La distribución de probabilidad basada en la distancia de Choquet-Mahalanobis sigue:

$$P_{\bar{x},\mu,\Sigma}(x) = \frac{1}{K} e^{-\frac{1}{2} d_{\mu,\Sigma}(x,\bar{x})}$$

donde K es una constante que debemos definir de manera que la función sea una probabilidad.

5 Conclusiones y trabajo futuro

En este trabajo hemos visto algunas aportaciones que permiten integrar en un único marco las interacciones entre variables que encontramos tanto en la matriz de covariancias como en las medidas difusas.

Hemos visto que la construcción de una distribución de probabilidad en términos de la distancia de Mahalanobis puede utilizarse para definir probabilidades en términos de integrales de Choquet y de nuestro nuevo operador.

Recientemente [10] hemos visto como el operador puede generalizarse utilizando las integrales de Choquet que dependen del nivel [3] (*level-dependent Choquet integral*), lo que permite definir nuevas distribuciones de probabilidad.

Hemos estudiado también [10] algunas propiedades de estas distribuciones, como la esperanza y sus covariancias, y si datos siguiendo una distribución basada en la integral de Choquet pasa el test de Mardia [6, 1, 4, 7]. Tenemos previsto estudiar nuevas propiedades de estas distribuciones de probabilidad.

En [9] habiamos motivado este trabajo en relación a las aplicaciones. En particular, vimos como este tipo de distribuciones pueden ser utilizadas para problemas de clasificación. Sin embargo, la aplicación de estos conceptos a problemas reales necesita primero resolver el proceso de identificación de los parámetros de la distribución. Esto es, la medida difusa y, en el caso de la distribución basada en la distancia de Choquet-Mahalanobis, de la matriz Σ. Estas son algunas lineas de trabajo futuro.

Agradecimientos

Se agrade el apoyo de los proyectos del MEC y MINECO ARES (CONSOLIDER INGENIO 2010 CSD2007-00004), eAEGIS (TSI2007-65406-C03-02) y COPRIVACY (TIN2011-27076-C03-03).

References

1. Baringhaus, L.; Henze, N. (1991). Limit distributions for measures of multivariate skewness and kurtosis based on projections. Journal of Multivariate Analysis 38: 51. doi:10.1016/0047-259X(91)90031-V. edit

2. Choquet, G. (1953/54) Theory of capacities, Ann. Inst. Fourier 5 131-295.
3. Greco, S., Matarazzo, B., Giove, S. (2011) The Choquet integral with respect to a level dependent capacity, Fuzzy sets and systems 175 1-35.
4. Kankainen, A., Taskinen, S., Oja, H. (2004). On Mardia's tests of multinormality. In Theory and Applications of Recent Robust Methods. Edited by Hubert, M., Pison, G., Struyf, A., Van Aelst, S. (pp. 153-164). Basel: Birkhäuser.
5. Mahalanobis, P. C. (1936) On the generalised distance in statistics, Proceedings of the National Institute of Sciences of India 2:1 4955.
6. Mardia, K. V. (1970) Measures of multivariate skewness and kurtosis with applications, Biometrika 57:3 519-530.
7. Mecklin, C. J., Mundfrom, D. J. (2004) An appraisal and bibliography of tests for multivariate normality, International Statistical Review 72:1 123-138.
8. Torra, V., Narukawa, Y. (2007) Modeling decisions: information fusion and aggregation operators, Springer.
9. Torra, V., Narukawa, Y. (2012) On a comparison between Mahalanobis distance and Choquet integral: The Choquet–Mahalanobis operator, Information Sciences 190 56–63.
10. Torra, V., Narukawa, Y. (2013) Exponential family of Level dependent Choquet integral based class-conditional probability functions, Proc. of AGOP 2013, in press.
11. Torra, V. (2011) Expressing interactions: Mahalanobis and Choquet operators, Proc. of AGOP 2011.

A new framework in R to support consensus processes in group decision making

Raquel Ureña and Enrique Herrera-Viedma

Department of Computer Science and A.I,
University of Granada, 18071 Granada Spain
raquel@decsai.ugr.es; viedma@decsai.ugr.es

Abstract. Rapid changes in the business environment such us the globalization and the increasing necessity to make complex decisions involving a wide range of alternatives has made that group decision support systems attained a great attention. The major goal of these systems is to obtain a solution accepted by the whole group of people who take part in the decision process. Indeed, in the literature we can find several algorithms to asses multi-person decision making. However there are no tools available to carry out automatically these processes using the opinions provided by the experts.
In this paper we present a new tool implemented in R to support decision making processes with soft consensus. This work is the first step to build a complete system to support multi-expert decision making including the most relevant algorithms in the literature and developing mechanisms to compare and evaluate their performances.

Keywords: Group decision making, consensus, fuzzy preference relation, R

1 Introduction

Group decision making (GDM) consists of multiple individual interacting to choose the best option between all the available ones. Each decision maker (expert) may have its own opinions and background, which enables them to approach the problem from different perspectives, but they share a common interest in achieving agreement on selecting the most suitable option [1]. Rapid changes in the business environment such us the globalization, the international expansion of most firms as well as the increasing necessity of making complex decision in short periods of time or even in real time, have made indispensable the development of techniques to support multi-person decision making. These systems are designed to help final decision makers to deal with the large number of possible alternatives, criteria, opinions and preferences that the experts board may have, which need to be taken into account to make the best decision. GDM algorithms are aimed to deal with all these variability bringing closer the opinions of the individual experts to find a solution which is accepted by the group of experts as a whole. Many of these systems also try to achieve a high degree of agreement among the experts to reach consensus decisions.

Since GDM problems have attracted many attention, in the specialized literature we can find a huge variety of algorithms that combines the preferences provided by the experts, to obtain the best solution. In most of these systems experts have to express their preferences by means of a set of evaluations over a set of alternatives using different representation formats, (fuzzy preference relations, multiplicative preference relations, interval preference relations, intuitionistic preference relations and linguistic preference relations among others)or even various representation formats in the same decision making process. There is also a wide variety of ways of combing the experts' preferences to reach the best decision: Some algorithms use aggregation mechanisms based in different variants of the Yager's ordered weighted averaging operator (OWA), [2] among with iterative negotiation processes to reach consensus [1] whereas others use optimization techniques to reach the best alternative. In spite of having a wide range of different algorithms to support decision making processes, there are no software tools which automatically compute them to calculate the best solution from the experts' preferences. Thus, the lack of such tools hinder the use of GDM algorithms to assist actual decision process. Also quantitatively compare the advantages and weaknesses that new decision making algorithms could provide, becomes a daunting task without such tools.

In this paper we present a new software tool fully implemented in R to carry out consensus guided decision making processes. In this first version we have implemented a simply consensus based decision making system using fuzzy preference relations. However the final objective is to build a complete multi-person decision making platform in which the experts can introduce their preferences and the system provides the best option using well known decision making algorithms.This tool would not be only useful to assist decision making processes, but also it will be helpful to validate and compare the already existing algorithms and develop new ones.

The rest of the paper is set out as follows: In section II we make a brief review of the main concepts of GDM and consensus. In section III we present the new R framework to support decision making. Sections IV shows an example of how the system works and finally our concluding remarks are pointed out among with some future work in section V.

2 Preliminaries

In order to make this paper as self contained as possible in this section we briefly introduce the main concepts used along the paper. Firstly we explain the classical problem of GDM and their main steps and then we introduce the consensus process.

GDM problem A classical GDM problem may be defined as a decision situation where [3]:

- There exists a group of two or more experts, $E = \{e_1, e_2, \ldots, e_m\}$ $(m \geq 2)$, characterized by their own ideas, attitudes, motivations and knowledge.

- There is a problem to solve in which a solution must be chosen among a set of possible alternatives, $X = \{x_1, x_2, \ldots, x_n\}$ $(n \geq 2)$.
- The experts try to achieve a common solution.

In a fuzzy context, the objective is to classify the alternatives from best to worst, associating with them some degrees of preference expressed in the $[0, 1]$ interval.

The solution for a GDM problem is derived either from the individual preferences provided by the experts, without constructing a social opinion, or by computing first a social opinion and then using it to find a solution [4]. These two approaches are respectively called direct and indirect approaches, and within both approaches, the process for reaching a solution of the GDM problem is composed by two steps [3, 5]:

1. *Aggregation phase.* In this phase, all preferences given by the experts must be aggregated into only one preference structure. This aggregation is carried out by means of particular aggregation operators that are usually defined for this purpose. Currently, at least 90 different families of aggregation operators have been studied [6–13, 2]. Among them the Ordered Weighted Averaging (OWA) operator proposed by Yager [2] is the most widely used.
   The aggregation operation by means of a quantifier guided OWA operator, ϕ_Q, is carried out as follows:

$$p_{ij}^c = \phi_Q\left(p_{ij}^1, \ldots, p_{ij}^m\right) = \sum_{k=1}^{m} w_k \cdot p_{ij}^{\sigma(k)}, \tag{1}$$

   where σ is a permutation function such that $p_{ij}^{\sigma(k)} \geq p_{ij}^{\sigma(k+1)}$, $\forall\, k = 1, \cdots, m-1$; Q is a fuzzy linguistic quantifier [14] that represents the concept of fuzzy majority and it is used to calculate the weighting vector of ϕ_Q, $W = (w_1, \ldots, w_n)$ such that, $w_k \in [0, 1]$ and $\sum_{k=1}^{n} w_k = 1$, according to the following expression [2]:

$$w_k = Q\left(k/n\right) - Q\left((k-1)/n\right), \forall k \in \{1, \ldots, n\}. \tag{2}$$

   In figure 1 are depicted some examples of linguistic quantifiers such as *"at least half"*, *"most of"* and *"as many as possible"*, which can be represented by the following function,

$$Q(r) = \begin{cases} 0 & \text{if } 0 \leq r < a \\ \frac{r-a}{b-a} & \text{if } a \leq r \leq b \\ 1 & \text{if } b < r \leq 1 \end{cases} \tag{3}$$

   using the values $(0, 0.5)$, $(0.3, 0.8)$ and $(0.5, 1)$ for (a, b), respectively [15].
   Alternative representations for the concept of fuzzy majority can be found in the literature. For example, Yager in [16] considered the parameterized family of RIM quantifiers $Q(r) = r^a$ $(a \geq 0)$ for such representation. This family of functions guarantees that [9]: (i) all the experts contribute to the final aggregated value (strict monotonicity property), and (ii) associates, when $a \in [0, 1]$, higher weight values to the aggregated values with associated higher importance values (concavity property).

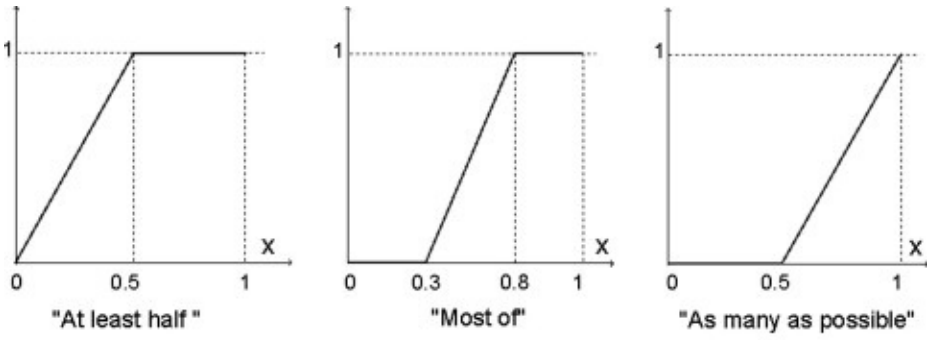


Fig. 1. Linguistic quantifiers at least half, most of, and as many as possible.

2. *Exploitation phase.* This phase uses the information produced in the aggregation phase to identify the solution set of alternatives for the problem.

The above selection process does not require any consensus among the experts. This could led to situations in which some experts do not agree with the final solution, since they can consider that their opinions have not been taken into account [17, 18]. To avoid those situations it is preferable to include mechanism to check the agreement among the experts before obtaining the solution for the GDM problem. These mechanism are widely known as consensus processes, and are carried out before a final solution is given [19, 20]. That way the system guaranties that all the experts agree with the final solution.

In the next subsection we are going to explain briefly the consensus process in GDM.

2.1 Consensus

We can find several approaches in the literature to deal with consensus such as the one in which the time evolution of the experts' opinion along the consensus process is modelled by Markov chains [21]. However the most widely consensus process applied in the literature in GDM problems is an iterative process managed by a moderator. So we are going to focus on this last approach.

A consensus process is defined as a dynamic and iterative group discussion process coordinated by a moderator, helping experts to bring their opinions closer. At the beginning the set of experts could disagree, then the consensus process is applied and, in each step, the degree of existing consensus among experts' opinions is measured. If the consensus degree is lower than a specified threshold, the moderator would ask experts to negotiate their opinions in an effort to bring them closer. Otherwise, the moderator would apply the selection process to obtain the final solution to the GDM problem. The measurement of the level of agreement among experts involves necessarily the calculation of the distance or, equivalently, the similarity between their preference values. According to [24] the formal definition of distance is as follows:

Definition 1 (Distance). *Let A be a set. A function $d: A \times A \longrightarrow \mathbb{R}$ is called a distance (or disimilarity) on A if, for all $x, y \in A$, there holds*

1. $d(x, y) \geq 0$ *(non-negativity)*
2. $d(x, y) = d(y, x)$ *(symmetry)*
3. $d(x, x) = 0$ *(reflexivity)*

Definition 2 (Similarity). *Let A be a set. A function $s: A \times A \longrightarrow \mathbb{R}$ is called a similarity on A if s is non-negative, symmetric, and if $s(x, y) \leq s(x, x)$ holds for all $x, y \in A$, with equality if and only if $x = y$.*

The main transforms between a distance d an a similarity s bounded by 1 are [24]:

$$d = 1 - s;\ d = \frac{1-s}{s};\ d = \sqrt{1-s}; d = \sqrt{2(1-s^2)};\ d = \arccos s;\ d = -\ln s$$

In this paper, we use the first transform to go from a distance function to a similarity function assuming the values of $d \in [0, 1]$.

3 The developed system

In this section we focus on the characteristics of the tool developed in R [22] to automatically assist multi-person decision making processes. We have chosen R to develop our system since it is a well known open source language and environment for statistical computing and graphics. The source code among with an example of how to use the developed tool can be downloaded from [23].

In figure 3 it is depicted an overview of the main elements of the developed system.

In the following subsections we explain the main elements of the tool focusing on their main characteristics and in relevant implementation details.

3.1 Preference relation generation

In this first version the tool works with square matrices of fuzzy preference relation $P = (p_{ij}) \forall i, j = 1 \cdots n$, where n is the number of possible alternatives and m is the number of experts who take part in the decision process. Each value p_{ij} represents the degree towards alternative i is preferred to alternative j. The system can work with any number of experts and any number of alternatives. It is able to read the experts' preferences from a CSV file and including them in a R dataframe to compute the next steps in the decision making process.

If the user wants to use this tool for training and testing purposes instead of working with real preference relations the system is also able to generate preferences automatically. To that aim we only have to set the number of experts, and the number of alternatives. The system can build consistent and non-consistent preference relations.

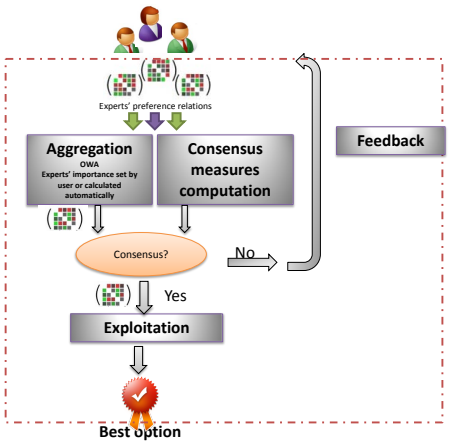


Fig. 2. Overview of the developed system

3.2 Consensus

The mechanism to obtain consensus is based on a minimum threshold that the consensus degree need to surpass in order to continue to the aggregation phase.

Consensus degrees are computed in three different levels:

Level 1. Consensus degree on the pairs of alternatives, cp_{ij}. It measures the agreement among all experts on the pair of alternatives (x_i, x_j) : To compute that, the distance between every pair of experts' preference relation is calculated.

Level 2. Consensus degree on alternatives, ca_i. It measures the agreement between every pair of experts on the alternative x_i. It is obtained by calculating the distance between the vector of preferences for every alternative. That is, we calculate the distance using the rows of the preference relation matrix as input vectors for the distance calculation.

The global consensus on alternatives measures de agreement among all experts on the alternative x_i. It is obtained aggregating the consensus on alternatives obtained for every pair of experts.

Level 3. Consensus degree on the relation, cr. It measures the global agreement among all experts, and it is obtained by aggregating all the consensus degrees at the level of alternatives.

To measure the distance among the experts' preferences, the system can calculate the following distances: the Manhanttan Distance, the Euclidean distance, the Dice distance, the Cosine distance and the Jarcard distance. In [24] it is carried out a comparative study of the application of these distance functions in the measurement of consensus, and also the authors analyse which distance functions are better to speed up the convergence of the consensus process.

3.3 Information aggregation

The aggregation is carried out using the quantifier guided OWA operator in eq. 1 which needs a weighting vector. This weighting vector can be used to set the experts' degree of importance in the aggregation. This way, the user can set the importance of each expert's opinion. This is specially useful in situations where the information handled by experts is not equally relevant [25]. Otherwise the system will calculate the weights for the aggregation using a linguistic quantifier, as it is explained in eq. 1. The user can set parameters of the linguistic quantifier to obtain different types, by default the available ones are: "At least half", "Most of", " As many as possible", and the RIM quantifiers.

3.4 Exploitation phase

This phase transforms the global information about the alternatives into a global ranking of them, from which the set of solution alternatives is obtained. The global ranking is calculated applying two choice degrees of alternatives to the collective fuzzy preference relation [26]: the *quantifier guided dominance degree* and the *quantifier guided non dominance degree.*

Finally, the solution X_{sol} is obtained by applying these two choice degrees and selecting those alternatives with maximum choice degrees.

4 Example

In this section we include an example to show how the developed tool works. We assume a GDM problem with four alternatives $X = \{x_1, x_2, x_3, x_4\}$ (N=4) and three experts $E = \{e_1, e_2, e_3\}$ (M=3). We will be using the OWA operator guided by the linguistic quantifier "most of", with a fixed minimum threshold for the global consensus value of $\gamma = 0.8$. The experts' preferences are provided in the file "preferences.csv" First of all we set the configuration parameters, and we call the tool to carry out the decision process.

Configuration parameters

```
M=3; # Number of experts in the decision process
N=4; #Number of alternatives
prefererecies_file='preferencias.csv';#File with the experts' preferences
distance='cosine';
consensusThreshold=0.85; #mimimun consensus level that needs to be achieved.
```

Once the tool receives the experts' preferences it includes all of them in an unique R data frame as we can see as follows:

Experts' preferences in an R data frame

```
preferences =
    0.5000    0.8143    0.2435    0.9293
```

```
    0.1857    0.5000    0.3500    0.1966
    0.7565    0.6500    0.5000    0.2511
    0.0707    0.8034    0.7489    0.5000
    0.5000    0.6160    0.4733    0.3517
    0.3840    0.5000    0.8308    0.5853
    0.5267    0.1692    0.5000    0.5497
    0.6483    0.4147    0.4503    0.5000
    0.5000    0.9172    0.2858    0.7572
    0.0828    0.5000    0.7537    0.3804
    0.7142    0.2463    0.5000    0.5678
    0.2428    0.6196    0.4322    0.5000
```

Then the tool computes the consensus measures. *Consensus measures provided by the tool*

```
consensusLevel1 =
         0    0.1982    0.2298    0.5776
    0.1982         0    0.4808    0.3887
    0.2298    0.4808         0    0.2986
    0.5776    0.3887    0.2986         0
         0    0.1029    0.0423    0.1721
    0.1029         0    0.4037    0.1839
    0.0423    0.4037         0    0.3167
    0.1721    0.1839    0.3167         0
         0    0.3011    0.1874    0.4055
    0.3011         0    0.0771    0.2048
    0.1874    0.0771         0    0.0181
    0.4055    0.2048    0.0181         0

globalConsesus =0.9140
```

Normally the global consensus value is lower than the accepted minimum threshold previously settled, and an iterative process of negotiation among the experts starts until the global consensus ratio surpasses the threshold. However, in this case the global consensus is higher than the threshold so the the aggregation and exploitation is carried out to obtain the best alternative, which, in this case is the first one.

Aggregation and exploitation phase

```
aggregated =
    0.5000    0.9172    0.2858    0.7572
    0.0828    0.5000    0.7537    0.3804
    0.7142    0.2463    0.5000    0.5678
    0.2428    0.6196    0.4322    0.5000
GDD =0.5815    0.3985    0.5018    0.4404
BestAlternative=1
```

5 Conclusions and future work

In this paper we present a tool to carry out automatically complete consensus based decision making process. This tool, which has been fully implemented in R , is able to select automatically the best possible solution among various alternatives from the experts' preferences fostering the consensus among the experts.

In this version we have implemented a classical decision model using only fuzzy preference relations, however we plan to develop more sophisticated decision making models with different types of preference relations, and with mechanisms to deal with missing information. Moreover more complex consensus models based on trust networks and ontologies will be incorporated.

The final goal is to build a complete decision making platform in which the experts can introduce their preferences and the system provides the best option according to different well known decision making algorithms.

This tool is not only very useful to assist decision making processes, but also it is helpful to validate and objectively compare the already existing algorithms and develop new ones.

Acknowledgments. This research work has been developed with the financing of FEDER funds in FUZZYLING-II Project TIN2010-17876, and the Andalusian Excellence Projects TIC-05299 and TIC-5991. .

References

1. E. Herrera-Viedma, S. Alonso, F. Chiclana, and F. Herrera. A consensus model for group decision making with incomplete fuzzy preference relations. *IEEE Transactions on Fuzzy Systems*, 15(5):863–877, 2007.
2. R.R. Yager. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transactions on Systems, Man and Cybernetics*, 18(1):183–190, 1988.
3. J. C. Fodor and M. Roubens, *Fuzzy Preference Modelling and Multicriteria Decision Support*, Kluwer Academic Publishers, 1994.
4. J. Kacprzyk, "Group decision making with a fuzzy linguistic majority", *Fuzzy Sets and Systems*, vol. 18, pp. 105–118, 1986.
5. M. Roubens, "Fuzzy sets and decision analysis", *Fuzzy Sets and Systems*, vol. 90, pp. 199–206, 1997.
6. S-M. Zhou, F. Chiclana, R.I. John, and J.M. Garibaldi. Type-1 owa operators for aggregating uncertain information with uncertain weights induced by type-2 linguistic quantifiers. *Fuzzy Sets and Systems*, 159(24):3281–3269, 2008.
7. S-M. Zhou, F. Chiclana, R.I. John, and J.M. Garibaldi. Alpha-level aggregation: a practical approach to type-1 owa operation for aggregating uncertain information with applications to breast cancer treatments. *IEEE Transactions on Knowledge and Data Engineering*, 23(10):1455–1468, 2011.
8. F. Chiclana, E. Herrera-Viedma, F. Herrera, and S. Alonso. Induced ordered weighted geometric operators and their use in the aggregation of multiplicative preference relations. *International Journal of Intelligent Systems*, 19(3):233–255, 2004.

9. F. Chiclana, E. Herrera-Viedma, F. Herrera, and S. Alonso. Some induced ordered weighted averaging operators and their use for solving group decision-making problems based on fuzzy preference relations. *European Journal of Operational Research*, 182(1):383–399, 2007.
10. Z. S. Xu and Q. L. Da. An overview of operators for aggregating information. *International Journal of Intelligent Systems*, 18(9):953–969, 2003.
11. G. J. Klir and T. A. Folger. *Fuzzy Sets, Uncertainty and Information*. Prentice Hall, New Jersey, 1988.
12. F. Herrera, E. Herrera-Viedma, and F Chiclana. A study of the origin and uses of the ordered weighted geometric operator in multicriteria decision making. *International Journal of Intelligent Systems*, 18(6):689–707, 2003.
13. János Fodor and Marc Roubens. *Fuzzy Preference Modelling and Multicriteria Decision Support*. Kluwer Academic Publishers, 1994.
14. L.A. Zadeh. A computational approach to fuzzy quantifiers in natural languages. *Computers & Mathematics with Applications*, 9(1):149–184, 1983.
15. J. Kacprzyk. Group decision making with a fuzzy linguistic majority. *Fuzzy Sets and Systems*, 18(2):105–118, 1986
16. R.R. Yager. Quantifier guided aggregation using owa operators. *International Journal of Intelligent Systems*, 11(1):49–73, 1996.
17. C. T. Butler and A. Rothstein, *On Conflict and Consensus: A Handbook on Formal Consensus Decision Making*, Takoma Park, 2006.
18. S. Saint and J. R. Lawson, *Rules for Reaching Consensus: A Modern Approach to Decision Making*, Jossey-Bass, 1994.
19. J. Kacprzyk, M. Fedrizzi and H. Nurmi, "Group decision making and consensus under fuzzy preferences and fuzzy majority", *Fuzzy Sets and Systems*, vol. 49, pp. 21–31, 1992.
20. C. Carlsson, D. Ehrenberg, P. Eklund, M. Fedrizzi, P. Gustafsson, P. Lindholm, G. Merkuryeva, T. Riissanen and A. G. S. Ventre, "Consensus in distributed soft environments", *European Journal of Operational Research*, vol. 61, pp. 165–185, 1992.
21. F. Harary, "On the measurement of structural balance", *Behavioral Science*, vol. 4, pp. 316–323, 1959.
22. The R project, http://www.r-project.org/
23. R framework to support consensus processes in GDM (source code), http://sci2s.ugr.es/cedi2013urena/GDMR.rar
24. Chiclana, F., Tapia Garca, J.M. , del Moral, M.J., Herrera-Viedma. E.: A statistical comparative study of different similarity measures of consensus in group decision making. Inf. Sci. 221, 110–123 (2013)
25. F. Chiclana, E. Herrera-Viedma, F. Herrera and S. Alonso, "Some induced ordered weighted averaging operators and their use for solving group decision making problems based on fuzzy preference relations", *European Journal of Operational Research*, vol. 182, pp. 383–399, 2007.
26. F. Herrera and E. Herrera-Viedma. Choice functions and mechanisms for linguistic preference relations. *European Journal of Operational Research*, 120(1):144–161, 2000.

1296

17-20 sept 2013
MULTICONFERENCIA
CAEPIA'13
Madrid

VI Simposio de Teoría y Aplicaciones de Minería de Datos [TAMIDA 2013]

Editores asociados:

Alicia Troncoso
José C. Riquelme

Simposio organizado por

Red Española de Minería de Datos y Aprendizaje

Entidades Colaboradoras

1298

Prefacio

La Red Española de Minería de Datos y Aprendizaje organiza por cuarta vez en el seno del CEDI 2013 el Simposio TAMIDA, dedicado a la Minería de Datos y el Aprendizaje Automático. Este año además por primera vez todos los congresos relacionados con Inteligencia Artificial forman una federación de congresos denominada Multiconferencia CAEPIA. Asimismo es la sexta vez que nos reunimos desde el Workshop fundacional celebrado en Sevilla durante la conferencia IBERAMIA en 2002. Desde entonces el concepto de Minería de Datos ha pasado de ser un término conocido sólo por iniciados, a ser noticia de actualidad debido a su uso por los gobiernos de EEUU o Reino Unido[1].

La aparición de grandes retos relacionados con paradigmas como Big Data u Open Data debe marcar el camino de nuestra investigación. Por otra parte, la Minería de Datos se ha transformado en un importante negocio que permite explotar la información desde un punto de vista comercial. Hoy en día convivimos con multitud de aplicaciones prácticas, como la publicidad dirigida a determinados perfiles de usuarios que son construidos a partir de datos publicados en internet, sistemas de recomendación para comprar libros o música, detección de fraudes, etc. Todo esto junto al enorme interés de la ciberseguridad, hacen que la Minería de Datos sea ahora mismo unas de las líneas de investigación con mayor futuro en las tecnologías informáticas y una de las principales demandas en muchas empresas del sector tecnológico.

El objetivo de celebrar TAMIDA dentro de la Multiconferencia CAEPIA es aunar toda la investigación relacionada con la Inteligencia Artificial que se realiza en España para establecer un foro en el que se intercambien ideas y experiencias, se propicie la captación de financiación a través de grupos multidisciplinares y se fomente la relación con empresas interesadas en el know how de los investigadores que desarrollan estas líneas.

La Red está formada actualmente por grupos de investigación de la mayoría de universidades españolas y tiene actualmente financiación concedida a través de una acción complementaria del Ministerio de Ciencia e Innovación. Estas acciones complementarias han permitido el networking de los grupos de investigación que trabajan en este campo y la creación de vínculos de trabajo.

En este capítulo se recogen 16 trabajos que constituyen una muestra significativa de la investigación que se está haciendo actualmente en España en este campo. En estos momentos donde la oferta de conferencias, congresos, y workshops es tan elevada, y con el escenario de escasez de financiación para la investigación en España, queremos agradecer a los autores participantes el que

[1] `http://internacional.elpais.com/internacional/2013/06/12/actualidad/1371053720\_613825.html`

hayan querido exponer su trabajo en estas páginas. Esperamos que la estancia en Madrid sea provechosa para todos.

Madrid, Julio 2013

Alicia Troncoso
José C. Riquelme

Organización

EL VI Simposio de Teoría y Aplicaciones de Minería de Datos (TAMIDA 2013) ha sido organizado por la Red Española de Minería de Datos y Aprendizaje.

Comité Organizador

Alicia Troncoso (Track Chair)	Universidad Pablo de Olavide de Sevilla
José C. Riquelme	Universidad de Sevilla

Comité de Programa

Oscar Cordón, U. Granada
Karina Gibert, U. P. de Cataluña
Francisco Herrera, U. Granada
José A. Lozano, U. País Vasco
José Riquelme, U. Sevilla
Sebastián Ventura, U. Córdoba
José Gámez, U. Castilla La Mancha
José Hernández Orallo, U. P. Valencia
César Hervás, U. Córdoba
Héctor Pomares, U. Granada
Alicia Troncoso, U. Pablo de Olavide

Sponsors

- Ministerio de Ciencia e Innovación

1302

Regla k_M-vecinos más cercanos en Bases de Datos de Baja Calidad

José M. Cadenas, M. Carmen Garrido, Raquel Martínez, Alejandro Martínez

Dept.de Ingeniería de la Información y las Comunicaciones
Facultad de Informática, Universidad de Murcia, Spain
{jcadenas,carmengarrido,raquel.m.e,amm26}@um.es

Resumen En dominios y situaciones reales es muy frecuente encontrarse con datos de baja calidad. Estos datos de baja calidad puede aparecer por diferentes razones y pueden expresarse mediante valores missing, fuzzy, intervalares, subset, etc. En este trabajo se propone un método para imputar valores missing a partir de datos de baja calidad. Este método está basado en el modelo predictivo k_M-vecinos más cercanos. También se extiende el método de condensación CNN para trabajar desde datos de baja calidad. Dicho algoritmo mejorará el proceso de imputación.

Keywords: Datos de baja calidad, k-NN, imputación, condensación

1. Introducción

En las áreas del Machine Learning y Minería de Datos, uno de los métodos de imputación de valores missing más conocido y usado es la regla k_M-vecinos más cercanos, conocida como k_M-NN (k_M-Nearest Neighbour), donde k son los vecinos considerados y M el número de ejemplos total de la base de datos [11,12]. Sin embargo, el método sufre de algunos inconvenientes, entre los que podemos destacar tres, [12]: la necesidad de un elevado requerimiento de memoria para almacenar todos los ejemplos que constituyen el conjunto de aprendizaje, la baja eficiencia durante el funcionamiento de la regla de decisión debido al elevado cálculo de similaridades entre los ejemplos test y los ejemplos de aprendizaje, y la poca tolerancia al ruido ya que usa todos los ejemplos como relevantes.

Se han propuesto distintas aproximaciones y estudios para suavizar los problemas anteriores. Sin embargo, hay un grupo de métodos que se enfrentan simultaneamente a los tres problemas anteriores y que están basados en la reducción de ejemplos. El objetivo de estos métodos es obtener un conjunto de ejemplos representativo del conjunto original con un tamaño bastante inferior y con similar o superior precisión de clasificación para los nuevos ejemplos de entrada. En la literatura, son conocidos como métodos de reducción, selección de ejemplos o selección de prototipos. Dependiendo de la estrategia seguida por los métodos, pueden eliminar ruido, redundancia o ambas, [7].

Una clasificación muy extendida de estos métodos de selección distingue tres clases de métodos: edición, condensación y métodos híbridos, [1]. El objetivo de

los métodos de edición es eliminar ruido, el objetivo de los métodos de condensación es eliminar redundancia y los métodos híbridos tratan de eliminar ruido y redundancia. Sin embargo, estos métodos han sido diseñados suponiendo que los ejemplos (que constituyen el clasificador) carecen de imprecisión e incertidumbre.

Cada vez más se está incorporando el trabajo con datos imprecisos e inciertos (es decir, datos de baja calidad, que denotaremos LQD) cuando no se dispone de un conjunto de datos de mejor calidad. En este trabajo extendemos la regla k_M-NN para que pueda llevar a cabo la imputación de valores missing desde LQD. Denominamos a esta regla como k_{M-LQD}-NN. Además, abordamos una de las mejoras extendiendo el método de condensación CNN de forma que pueda trabajar también con LQD (lo denoteremos por CNN_{LQD}). Esta mejora supone un punto de partida para llevar a cabo posteriores mejoras de la regla.

El presente trabajo ha sido estructurado de la siguiente manera: en la Sección 2 presentamos una breve descripción de los tipos de LQD que pueden afectar a los conjuntos de datos, en las Secciones 3 y 4 presentamos la regla k_{M-LQD}-NN y el método CNN_{LQD} desde LQD, respectivamente. En la Sección 5 mostramos algunos resultados experimentales y por último, en la Sección 6 presentamos las conclusiones.

2. Datos de Baja Calidad

En esta sección se presentan los distintos LQD con los que trabaja tanto la regla k_{M-LQD}-NN que presentamos en la Sección 3, como el método de condensación CNN_{LQD} que definimos en la Sección 4.

2.1. Valores Missing

Un factor importante a considerar cuando se trabaja con técnicas de Minería de Datos, es el tratamiento que se realizará a los valores missing. Algunas decisiones que se suelen tomar son:

- Permitir valores missing en el conjunto de datos cuando la técnica es robusta a la existencia de tales valores.
- Eliminar los atributos con valores missing del conjunto de datos.
- Eliminar los ejemplos con valores missing del conjunto de datos.
- Reemplazar los valores missing por un valor que conserva la media o la varianza (global o clases/grupos), en el caso de valores numéricos, o por la moda en el caso de valores nominales. Otra manera de estimar un valor missing es utilizar otros ejemplos y hacer uso de cualquier técnica de predicción.
- Reemplazar los valores missing por un intervalo que abarca todo el dominio de los atributos correspondientes cuando la técnica sea robusta a la existencia de intervalos en el caso de atributos numéricos o por un conjunto compuesto por todos los valores del dominio cuando la técnica es robusta a la existencia de este tipo de valores en el caso de atributos nominales.

2.2. Datos con Ruido

El ruido es un término amplio que se ha interpretado de diferentes maneras. Por ejemplo, se ha definido como un error aleatorio en la medición de un atributo. Otra definición acerca del ruido establece que es cualquier propiedad en un patrón detectado que no se crea por el modelo real subyacente sino por la aleatoriedad en el mundo o en los distintos sensores [10]. En estas definiciones, el ruido es considerado un error al azar. El ruido de los datos puede ocurrir por varias razones: 1) debido a los problemas de los instrumentos de medida y 2) debido al hecho de que grandes conjuntos de datos suelen obtenerse por métodos automatizados. Dependiendo de la técnica de Minería de Datos que usemos podemos, si es posible,:

- detectar los valores con ruido, y realizar un tratamiento similar al de los valores missing.
- conocer los errores de la medición producidos por el instrumento de medida (media y desviación estándar), e incorporar esta información al conjunto de valores de los atributos con ruido por medio de algún tipo de transformación.

2.3. Datos con Incertidumbre

En [6], Dubois describió los conceptos de imprecisión e incertidumbre en términos de incertidumbre estocástica y epistémica. La incertidumbre estocástica surge del azar o variabilidad relacionada con los procesos naturales. La incertidumbre epistémica surge de la naturaleza incompleta/imprecisa de la información disponible. Si bien la incertidumbre estocástica se aborda adecuadamente utilizando la teoría de probabilidades clásica, varias teorías de incertidumbre se han desarrollado con el fin de manejar explícitamente la información incompleta/imprecisa entre las que se encuentran la teoría de la posibilidad.

Algunas representaciones simples de la información con imprecisión/incertidumbre basadas en intervalos, y en su generalización, son:

- El uso de un intervalo $[a, b]$, por lo que suponemos que el valor del atributo está dentro del mismo.
- El uso de un conjunto o intervalo fuzzy que asigna un grado de posibilidad entre 0 y 1 para cada valor del intervalo.

Otra forma de representar la imprecisión/incertidumbre es mediante subconjuntos crisp/fuzzy (que denotaremos crisp/fuzzy subsets) de un conjunto finito de valores:

- En el caso de atributos nominales, usando un crisp subset, como {rojo,azul} si el dominio es {rojo,azul,verde} o mediante un fuzzy subset que asigna grados de posibilidad a los distintos valores {0.2/rojo,0.8/azul}.
- En el caso de atributos numéricos, usando un fuzzy subset de etiquetas linguísticas asociadas a una partición fuzzy del atributo. Por ejemplo, el valor de temperatura crisp de 45 grados puede ser transformado como el fuzzy

subset {0.0/frío,0.2/templado,0.8/calor}. Si la partición es del tipo Ruspini, la suma de los grados de pertenencia es 1. En cualquier otro tipo de partición, la suma puede tomar otro valor como {0.1/frío,0.3/templado,0.9/calor}.

Dependiendo del tratamiento de la imprecisión/incertidumbre en las distintas técnicas de Minería de Datos, se puede:

- Permitir estos valores en el conjunto de datos cuando la técnica es robusta a la existencia de tales valores.
- Eliminar los atributos con imprecisión/incertidumbre del conjunto de datos.
- Eliminar los ejemplos con imprecisión/incertidumbre del conjunto de datos.
- Reemplazar/imputar los valores con imprecisión/incertidumbre.

3. La Regla k_{M-LQD}-NN desde LQD

En esta sección se describe la regla k_{M-LQD}-NN desarrollada para realizar imputación de valores missing desde LQD. Previamente, introducimos algunos aspectos necesarios para su desarrollo.

a) Para atributos numéricos, los diferentes tipos de valores son expresados como una cuádrupla representando funciones de pertenencia (Figura 1).

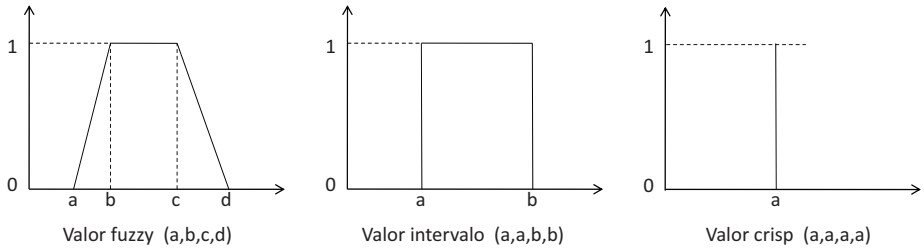


Figura 1. Valores numéricos como cuádruplas

b) En el caso de atributos nominales representamos un valor nominal crisp como un subset formado por un simple valor del dominio (p.e. {rojo}); un crisp subset de valores (p.e. {rojo,verde}); un fuzzy subset de valores (p.e. {0.2/rojo,0.8/azul}).

Como parte de la regla k_{M-LQD}-NN, es necesaria una métrica que nos sirva como medida de distancia entre dos ejemplos $x^{(j)}$ y $x^{(h)}$. Definimos la función distancia como $d_{LQD}(x^{(j)}, x^{(h)}) = \sqrt{\frac{\sum_{i=1}^{n-1} f(x_i^{(j)}, x_i^{(h)})^2}{n-1}}$ donde la función $f(\cdot,\cdot)$ es una medida heterogénea entre dos valores que pueden ser del tipo LQD. Por tanto, la función $f(x_i^{(j)}, x_i^{(h)})$ estará definida a partir de las funciones $f_1(x_i^{(j)}, x_i^{(h)})$ y $f_2(x_i^{(j)}, x_i^{(h)})$ cuando el atributo i es numérico o nominal, respectivamente.

3.1. La Regla k_{M-LQD}-NN

Usando la función $d_{LQD}(\cdot,\cdot)$ obtenemos el conjunto de k vecinos más cercanos a un ejemplo z (a este conjunto lo denotaremos $KLQD$). A partir de este conjunto llevamos a cabo la imputación de los valores missing de z. Dado que la clase de los ejemplos puede estar dada con imprecisión (por un crisp/fuzzy subset), debemos de definir la forma de considerar dicho valor de clase en la imputación. En el caso concreto de la regla k_M-NN desde datos crisp, la clase conocida de z hace que sólo se consideren en la imputación aquellos vecinos que pertenecen a dicha clase. Ahora en la regla k_{M-LQD}-NN desde LQD, se ponderará la aportación de cada vecino en la imputación de forma proporcional a la distancia de su clase con la clase de z. Si la clase de z es desconocida, obtenemos la clase más frecuente entre los vecinos (que podrá ser una clase imprecisa). Dicha clase será el valor imputado a z y el considerado en la ponderación de cada vecino para la imputación de otros valores de z.

El Algoritmo 1 muestra la regla k_{M-LQD}-NN. En el algoritmo, las funciones f_1 y f_2, y por tanto d_{LQD}, podrán ser cualesquiera que se consideren funciones distancias adecuadas entre LQD. El umbral α permite definir el nivel o grado mínimo de creencia media en el que el usuario quiere realizar las imputaciones. Es decir, el valor $\frac{P_{KLQD}}{k}$ se define como la media de los grados de creencia en los que se basa la imputación. Cuando α sea definido como 0 por el usuario se llevarán a cabo todas las imputaciones independientemente de la creencia que las apoya, y cuando α sea 1 sólo se llevarán a cabo imputaciones basadas en ejemplos con el mismo valor de clase que el imputado.

4. El Método CNN_{LQD} para condensación desde LQD

Ante la idea de trasladar a la regla k_{M-LQD}-NN las mejoras que se han llevado a cabo sobre la regla k_M-NN a lo largo de la literatura, comenzamos extendiendo un método de condensación para que se pueda condensar el conjunto de ejemplos a partir de LQD. El método que extendemos es el método CNN (Condensed Nearest Neighbors) [9] que si bien es uno de los primeros métodos de selección de ejemplos que aparece en la literatura, tiene un buen comportamiento frente a nuevas propuestas manteniendo cierta simplicidad. En este trabajo abordamos este primer aspecto de mejora que servirá como punto de partida para llevar a cabo la extensión de otro tipo de mejoras. Para ello, en esta sección describimos brevemente el método de condensación CNN. Una revisión más detallada del método CNN y de otros métodos de condensación podemos encontrarla en [1,13] y una comparativa bastante completa en [7]. Posteriormente, se presenta el método CNN_{LQD} para condensación desde LQD.

4.1. Método de Condensación CNN

El método CNN es uno de los más utilizados y fué desarrollado por Hart en 1966 [9]. Este algoritmo construye un subconjunto S a partir del conjunto de

Algoritmo 1 – k_{M-LQD}-NN desde LQD

Introducir z e inicializar k, $1 \leq k \leq M$;
Obtener el conjunto $KLQD$ de los k ejemplos más cercanos a z según $d_{LQD}(z, x^{(i)})$;
if la clase de z es desconocida **then** $z_n = \{\frac{\sum_{x^{(i)} \in KLQD} \mu(x^{(i)}, \omega_c)}{|KLQD|} / \omega_c\};\ c = 1, \ldots, I$
donde $\mu(x^{(i)}, \omega_c)$ es el grado de creencia del valor de clase ω_c en el i-ésimo ejemplo.
end if
for all ejemplo $x^{(i)}$ de $KLQD$ **do**
Calcular $dclass_i = f_2(x_n^{(i)}, z_n)$ y obtener el valor $p_i = 1 - dclass_i$
end for
Calcular $P_{KLQD} = \sum_{x^{(i)} \in KLQD} p_i$
if $\frac{P_{KLQD}}{k} \geq \alpha$
for all atributo z_j desconocido de z
if z_j numérico **then** $z_j = \frac{\sum_{x^{(i)} \in KLQD} p_i \cdot x_j^{(i)}}{P_{KLQD}}$
else if z_j nominal **then** $z_j = \{\frac{\sum_{x^{(i)} \in KLQD} p_i \cdot \mu(x^{(i)}, v)}{|KLQD|} / v\};\ \forall v \in Dominio_{z_j}$, donde $\mu(x^{(i)}, v)$ es el grado de creencia del valor v del j-ésimo atributo en el ejemplo $x^{(i)}$
end if
end for
else No se realiza la imputación;
end if

datos M tal que todo ejemplo de M estará más cerca a un ejemplo de S de la misma clase que a otro ejemplo de S de distinta clase. El algoritmo comienza seleccionando un ejemplo de cada clase de M y los inserta en S. Cada ejemplo de M se clasifica con 1-NN usando solamente los ejemplos de S. Si un ejemplo no se clasifica bien, se añade a S. Este proceso se repite hasta que no haya ejemplos en M que se clasifiquen incorrectamente. Su mayor desventaja es que es un método altamente dependiente del orden de presentación de los objetos. Sin embargo, tiende a quedarse con los ejemplos cercanos a la frontera de decisión de cada clase, ya que estos son los que tienden a estar mal clasificados.

4.2. El Método de Condensación CNN_{LQD}

En esta sección extendemos el método CNN para que pueda llevar a cabo el condensado desde LQD. En el Algoritmo 2 se muestra de forma completa el proceso llevado a cabo por el método CNN_{LQD}.

El Algoritmo 2 ordena los ejemplos del conjunto inicial de acuerdo a la entropía fuzzy ($EntropiaF(\cdot)$) del atributo clase. De esta forma se consideran en primer lugar aquellos ejemplos con valores de clase menos imprecisos/inciertos.

Como se indica en el Algoritmo 2, para cada ejemplo de M se calcula su vecino más cercano en S y se comparan sus valores de clases mediante una medida de similitud. Si la similitud de las clases está por debajo de un valor de umbral, β, el ejemplo se considera mal clasificado con los ejemplos de S y se añade a dicho conjunto. El valor β es un parámetro externo del algoritmo

Algoritmo 2 – CNN_{LQD} desde LQD

Introducir el conjunto de ejemplos inicial M e inicializar $S = \emptyset$;
Calcular $EntropiaF(\cdot)$ de la clase de cada uno de los ejemplos de M;
Ordenar M en orden creciente de acuerdo a su entropía fuzzy;
Seleccionar de M un ejemplo de cada una de las clases existentes y añadirlos a S;
Eliminar los ejemplos seleccionados en el paso anterior del conjunto M;
while S cambie **do**
  for all ejemplo $x^{(i)}$ de M **do**
    Calcular su vecino más cercano de S de acuerdo a la función $d_{LQD}(\cdot,\cdot)$;
    Calcular $simil_{clase}(clase(x^{(i)}), clase(1\text{NN}_{x^{(i)}})) = 1 - f_2(clase(x^{(i)}), clase(1\text{NN}_{x^{(i)}}))$
    if $simil_{clase} \geq \beta$ **then** No se añade el ejemplo $x^{(i)}$ al conjunto S
      else Añadir $x^{(i)}$ al conjunto S y eliminarlo de M;
    end if
  end for
end while

que permite definir el grado con el que el usuario está dispuesto a considerar que dos ejemplos pertenecen a la misma clase. Cuanto más se acerque su valor a 0, más ejemplos serán considerados como clasificados correctamente y por lo tanto el conjunto final de ejemplos seleccionados será más pequeño. Es decir, valores pequeños de β conducen a bases de datos más condensadas. Finalmente tendremos los ejemplos seleccionados durante el proceso de condensación en S.

5. Experimentos

Para comprobar el funcionamiento de los algoritmos propuestos vamos a mostrar algunos resultados de su aplicación a distintas bases de datos cuando imputamos valores missing. El proceso de imputación se llevará a cabo a partir de bases de datos con LQD y de las bases de datos condensadas usando el método CNN_{LQD}.

5.1. Definición de los Experimentos

La función $d_{LQD}(\cdot,\cdot)$ utilizada es:

$$f_1(x_i^{(j)}, x_i^{(h)}) = \frac{\sqrt{\frac{(a-a')^2+(b-b')^2+(c-c')^2+(d-d')^2}{4}}}{max_i - min_i}$$

donde $x_i^{(j)}, x_i^{(h)}$ son valores numéricos definidos por las cuádruplas (a,b,c,d) y (a',b',c',d') respectivamente y

$$f_2(x_i^{(j)}, x_i^{(h)}) = 1 - \frac{Card(x_i^{(j)} \bigcap x_i^{(h)})}{Card(x_i^{(j)} \bigcup x_i^{(h)})}$$

donde $x_i^{(j)}, x_i^{(h)}$ son valores nominales y $Card(x_i^{(j)} \bigcap x_i^{(h)})$ y $Card(x_i^{(j)} \bigcup x_i^{(h)})$ están definidas como la cardinalidad de los crisp/fuzzy subsets que resultan de la unión e intersección de $x_i^{(j)}$ y $x_i^{(h)}$ respectivamente.

La medida de entropía utilizada es la definida en [4]:

$$EntropiaF(\mu(\cdot)) = -\sum_{j=1}^{n}(\mu(c_j)\log(\mu(c_j)) + (1-\mu(c_j)(\log(1-\mu(c_j)))$$

donde $\mu(\cdot)$ denota un crisp/fuzzy subset y c_i los valores pertenecientes al subset.

Usamos los valores $k = \sqrt{M}$, $\alpha = 0$ y $\beta = 1$ en todos los experimentos. La Tabla 1 muestra las bases de datos utilizadas [2], indicando el número de ejemplos (M), el número de atributos ($|A|$) (entre paréntesis se muestran el número de atributos numéricos y nominales) y el número de clases (I) para cada base de datos. "Abr" indica la abreviatura de la base de datos usada en los experimentos.

Tabla 1. Bases de Datos

Base de Datos	**Abr**	**M**	**|A|**	**I**	**Base de Datos**	**Abr**	**M**	**|A|**	**I**
Iris Plants	IRP	150	5 (4,1)	3	Contraceptive M.	CMC	1473	10 (9,1)	3
W. Breast C. (org)	BCW	683	10 (9,1)	2	Attitude Smok.	SMO	2855	9 (2,7)	3
Pima Indian Diab.	PIM	768	9 (8,1)	2	Waveform	WAV	4999	22 (21,1)	3
German Credit	GER	1000	25 (24,1)	2					

En estas bases de datos hemos introducido LQD[1] usando la herramienta NIP [3]. En concreto hemos introducido un 5 % de valores fuzzy y un 5 % de intervalos en cada atributo numérico y un 5 % de fuzzy subsets en cada atributo nominal incluida la clase. Una vez añadido LQD, la base de datos es particionada en los conjuntos test y train (20 % y 80 % respectivamente). En los experimentos de clasificación obtenemos la clase de todos los ejemplos del test y en los experimentos de regresión estimamos el valor del primer atributo numérico de todos los ejemplos del test. Este proceso se repite tres veces y por lo tanto los resultados muestran siempre la media de las tres repeticiones.

Dado que la clase imputada a un ejemplo $x^{(i)}$ puede ser un fuzzy subset $\{\mu(x^{(i)},\omega_1)/\omega_1, \mu(x^{(i)},\omega_2)/\omega_2, \ldots, \mu(x^{(i)},\omega_I)/\omega_I\}$, dicho valor se transforma en un crisp subset de la siguiente forma: Sea ω_{max} la clase con mayor grado en el fuzzy subset, el crisp subset se obtiene como $clase_{k_{M-LQD}-NN}(x^{(i)}) = \{\omega_c | \frac{\mu(x^{(i)},\omega_{max})-\mu(x^{(i)},\omega_c)}{\mu(x^{(i)},\omega_{max})} < \gamma\} \cup \{\omega_{max}\}$. Por tanto, para obtener los resultados de clasificación, aplicamos el proceso de decisión mostrado en el Algoritmo 3.

A partir del Algoritmo 3, construimos el intervalo de precisión en clasificación como $[min_acierto, max_acierto]$ donde $min_acierto$ se calcula considerando sólo los aciertos (variable $acierto$) y $max_acierto$ es calculado considerando como aciertos la suma de las variables $acierto + (acierto_error)$.

En la imputación de atributos numéricos de un ejemplo $x^{(i)}$, de nuevo, el valor imputado $x_{j\ imp}^{(i)}$ puede ser un LQD. Calculamos el error de imputación como el valor medio de los valores $f_1(x_j^{(i)}, x_{j\ imp}^{(i)})$.

[1] Bases de datos disponibles en http://heurimind.inf.um.es/heurimind/low-quality-repository/LQD.shtm

Algoritmo 3 Decisión en la clasificación

for all ejemplo $x^{(i)}$ de BD_{test} **do**
  if $(clase(x^{(i)}) == clase_{k_{M-LQD-NN}(x^{(i)})})$ **then** acierto++;
  else if $(clase(x^{(i)}) \bigcap clase_{k_{M-LQD-NN}(x^{(i)})} \neq \emptyset)$ **then** acierto_error++;
    else error++;
  end if
end for

5.2. Resultados Experimentales

En la Tabla 2 se muestran los resultados obtenidos en los experimentos. La tabla muestra la imputación a partir de las base de datos de train y la imputación a partir de las bases de datos condensadas. Las columnas "C" y "R" muestran la imputación en clasificación y regresión respectivamente. En "C" utilizamos tres posibles valores de γ del Algoritmo 3. La última columna "TR" muestra la tasa de reducción tras la condensación.

Tabla 2. Resultados experimentales para las bases de datos de la Tabla 1

	Sin Condensación				**Con Condensación**				
	C			**R**	**C**			**R**	**TR**
	$\gamma =$ **5 %**	$\gamma =$**10 %**	$\gamma =$**20 %**		$\gamma =$**5 %**	$\gamma =$**10 %**	$\gamma =$**20 %**		
IRP	[97.8,97.8]	[97.8,97.8]	[95.6,100]	0.30	[97.8,97.8]	[97.8,97.8]	[96.7,97.8]	0.32	64.4
BCW	[97.1,97.1]	[96.8,97.1]	[96.8,97.1]	1.68	[97.1,97.1]	[97.1,97.1]	[96.8,98.0]	1.81	59.1
PIM	[75.4,76.0]	[71.2,78.7]	[70.4,79.3]	2.14	[75.6,75.8]	[67.1,80.4]	[63.8,81.3]	2.08	41.9
GER	[70.8,75.2]	[70.7,75.3]	[66.0,81.2]	1.04	[67.2,76.5]	[66.8,76.5]	[57.7,84.8]	1.07	40.8
CMC	[42.3,51.1]	[37.8,55.8]	[28.8,65.2]	5.80	[42.3,51.1]	[37.7,55.8]	[28.8,65.2]	5.75	22.5
SMO	[68.6,68.7]	[68.6,68.7]	[68.0,68.9]	0.36	[68.6,68.7]	[68.6,68.7]	[68.0,68.9]	0.37	31.6
WAV	[82.9,86.4]	[81.4,87,6]	[77.9,89.9]	0.82	[83.2,86.1]	[79.8,88.6]	[76.1,91.3]	0.82	46.6

Como podemos observar en la Tabla 2, los resultados obtenidos muestran un comportamiento muy estable de los algoritmos propuestos cuando trabajan con LQD. Estos resultados se ajustan a los obtenidos en la literatura para estas mismas bases de datos sin LQD. Además, podemos observar en la imputación del atributo clase cómo al aumentar γ, el intervalo de porcentaje medio de precisión es más ámplio. Esto nos indica la falta de decisión a la hora de elegir un valor para imputar. Comparando los resultados obtenidos "sin" y "con" condensación podemos deducir que la condensación ha funcionado correctamente.

6. Conclusiones

En dominios y situaciones reales es muy frecuente encontrarse con conjuntos de datos que contienen valores de baja calidad. En este trabajo hemos propuesto un

método para imputar valores missing y un método de condensación a partir de LQD que resultan de utilidad cuando no se dispone de otro tipos de datos. Estos métodos están basados en el modelo predictivo k_M-NN. Debido al uso de LQD, los métodos propuestos son adaptables a las distintas decisiones que podrían tomar los usuarios. Estas adaptaciones se realizan mediante diversos valores umbrales y nos permiten reflejar la "claridad" de decisión de los usuarios cuando deben decidir el valor numérico o nominal de un atributo ante los LQD. Los resultados experimentales muestran el buen comportamiento de dichos métodos.

Como trabajo futuro, comentar que el trabajo propone una mejora inicial y simple del método de imputación pero pretendemos extender las distintas mejoras de la regla k_M-NN llevadas a cabo en la literatura.

Agradecimientos

Soportado por el proyecto TIN2011-27696-C02-02 del MINECO de España. Gracias también a la Agencia de Ciencia y Tecnología de la R. de Murcia (España) por el programa 04552/GERM/06 y por la beca FPI de Raquel Martínez.

Referencias

1. Angiulli, F.: Fast Nearest Neighbor Condensation for Large Data Sets Classification. IEEE Trans. on knowledge and data engineering **19** (11) (2007) 1450–1464
2. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository. University of CA, School of ICS, *http://www.ics.uci.edu/∼mlearn*, 2007
3. Cadenas, J.M., Garrido, M.C., Martínez-España, R.: Software tool: NIP tool, Universidad de Murcia, http://heurimind.inf.um.es, 2012
4. De Luca, A., Termini, T.: A definition of non-probabilistic entropy in setting of fuzzy set theory. Information Control **20** (1971) 301–312
5. Devi, F.S., Murty, M.N.: An Incremental Prototype Set Building Technique. Pattern Recognition **35(2)** (2002) 505–513
6. Dubois, D., Prade, H.: Possibility Theory: An approach to Computerized Processing of Uncertainty. New York: Plenum Press, 1988
7. García, S., Derrac, J., Cano, J.R., Herrera, F.: Prototype Selection for Nearest Neighbor Classification. IEEE Trans. Pattern Anal. Mach. **34 (3)** (2012) 417–435
8. Gates, G.W.: The reduced nearest neighbor rule. IEEE Trans. on Information Theory **18** (1972) 431–433
9. Hart, P.E.: The Condensed Nearest Neighbor Rule. IEEE Trans. Information Theory **14 (3)** (1968) 515–516
10. Hickey, R.: Noise Modeling and Evaluating Learning from Examples. Artificial Intelligence **82 (1–2)** (1996) 157–179
11. Keller, J.M., Gray, M.R., Givens, J.A.: A fuzzy K-Nearest Neighbor Algorithm. IEEE Trans. on Systems, Man and Cybernetics **15 (4)** (1985) 580–585
12. Kononenko, I., Kukar, M.: Machine Learning and Data Mining: Introduction to Principles and Algorithms. Horwood Publishing Limited, 2007
13. Wilson, D.R., Martinez, T.R.: Reduction Techniques for Instance-Based Learning Algorithms. Machine Learning **38 (3)** (2000) 257–286

Análisis del virus de la gripe A mediante descubrimientos de subgrupos difusos

C.J. Carmona, C. Chrysostomou, H. Seker, M.J. del Jesus

Department of Computer Science, University of Jaen, Spain [1]
{ccarmona,mjjesus}@ujaen.es
Department of Genetics, University of Leicester, Leicester, United Kingdom [2]
cc390@le.ac.uk
Centre for Computational Intelligence, De Montfort University, Leicester, United Kingdom [3]
hseker@dmu.ac.uk

Resumen El virus de la gripe A está siendo en los últimos años uno de los principales problemas de pandemias mortales de los Siglos XX y XXI. Las diferentes mutaciones que sufre este virus desembocan en una gran dificultad para la creación de vacunas y/o medicinas que puedan combatirlo.
En este estudio experimental se buscan relaciones interesantes y atípicas entre diferentes proteínas del virus de la gripe A. Con estas relaciones, se buscan las propiedades capaces de distinguir y describir los distintos tipos de virus que podrían proporcionar a los expertos información que ayude en el desarrollo de nuevas terapias para este virus. Para ello, se estudian un conjunto de proteínas recogidas en los últimos años. Para cumplir este objetivo, se realizan inicialmente unas transformaciones de las cadenas proteicas del virus para su posterior análisis con el algoritmo de descubrimiento de subgrupos basado en sistemas difusos evolutivos más destacado, el algoritmo NMEEF-SD.

Keywords: Sistemas difusos evolutivos, Virus de la gripe A, Descubrimiento de subgrupos

1. Introducción

El virus de la gripe A pertenece a la familia *Orthomyxoviridae* y afecta principalmente en aves y algunos mamíferos. El genoma de este virus está formado por 8 genes sencillos: el gen *hemagglutinin* (HA), el gen *neuraminidase* (NA), el gen *nucleoprotein* (NP), el gen *matrix proteins* (M), el gen *non-structural proteins* (NS) y los tres genes *RNA polymerase* (PA, PB1, PB2). Raras veces surgen brotes o pandemias cuando el virus de la gripe A se transmite de aves salvajes a aves domésticas.

Durante el Siglo XX se han registrado tres grandes pandemias provocadas por el virus de la gripe A dentro de la raza humana, concretamente causadas por los subtipos de virus H1N1, H2N2 y H3N2. Además de estos tres subtipos, dentro del

virus de la gripe A, el H5N1 se considera como conductor de la pandemia actual. En este análisis, se utilizan estos cuatro subtipos del virus de la gripe A, que son el objetivo principal de estudio para la creación de medicinas o antivirales, que se denominan inhibidores de NA [13].

A lo largo de los años se ha recogido información referente a estos subtipos de virus [2]: para el subtipo H1N1 se han recogido 200 proteínas desde el 2009, para el H2N2 se han recogido 76 entre los años 1957 y 1968; para el subtipo H3N2 se han recogido 200 desde el periodo 1968 hasta el 2000 y para el subtipo H5N1 se han recogido 70 proteínas desde 2005 a 2009. La relación de estos subtipos del virus de la gripe A con respecto al gen NA es la siguiente: El virus H1N1 es el resultado de reordenaciones entre el virus H1N1 euro-asiático del cerdo y el virus H1N2 del cerdo, el virus H2N2 es el resultado de la reordenación entre el virus H1N1 humano y el virus de la gripe aviar H2N2, el virus H3N2 es el resultado de la reordenación entre el virus H2N2 circulante entre humanos y el virus de las aves H3 y el virus H5N1 fue creado mediante diversas combinaciones de subtipos de virus de la gripe A.

Para el análisis del problema, este trabajo se centra en la técnica de minería de datos del Descubrimiento de Subgrupos (SD) [10] cuyo principal objetivo es la obtención de relaciones parciales en los datos con estadísticas inusuales y de interés con respecto a una variable objetivo. Para ello, se va a aplicar el algoritmo NMEEF-SD [3] que es en la actualidad el algoritmo de SD basado en sistemas difusos evolutivos (EFSs) [9] más destacado de la literatura. Los EFSs están basados en lógica difusa y permiten trabajar en entornos con variables continuas sin necesidad de una previa discretización como es el problema que se presenta en este trabajo.

Este trabajo se divide en las siguientes secciones: En la Sección 2, se puede observar la transformación llevada a cabo sobre las proteinas para prepararlas y poder aplicar SD, en la Sección 3 se presenta de forma general SD y las ventajas de aplicar el algoritmo NMEEF-SD y en la Sección 4 se presenta el estudio experimental realizado. Para finalizar se presentan las conclusiones obtenidas en el trabajo.

2. Procesamiento de señal para el análisis de secuencias de proteínas

Recientemente, se han utilizado diversos métodos dentro de la bioinformática para el análisis de secuencias de proteínas, donde algunos de los más comunes son el *Resonant Recognition Model* [5,6] y el *Complex Resonant Recognition Model* [4]. Estudios previos [15] han utilizado los subtipos del virus de la gripe A para analizar el gen HA con el *Resonant Recognition Model* con el objetivo de identificar nuevas terapias que permitan el desarrollo de nuevas medicinas así como comprender cómo interacciona el virus de la gripe con sus receptores.

A diferencia de otros estudios previos, este estudio experimental ha sido realizado directamente mediante un espectro absoluto que se deriva de la aplicación

de la transformación discreta de fourier para cada secuencia proteica con codificación numérica. Para poder aplicar la función de fourier, es necesario utilizar un índice de aminoácido, como el *electron-ion interaction potential* (EIIP) [14]. Mediante este índice, mostrado en la tabla 1, se convierten las secuencias proteicas en secuencias numéricas.

Tabla 1. Valores del índice *electron-ion interaction potential*

Amino	***EIIP***	***Amino***	***EIIP***	***Amino***	***EIIP***	***Amino***	***EIIP***
Leu	0.0000	Tyr	0.0516	Ile	0.0000	Trp	0.0548
Asn	0.0036	Gln	0.0761	Gly	0.0050	Met	0.0823
Glu	0.0057	Ser	0.0829	Val	0.0058	Cys	0.0829
Pro	0.0198	Thr	0.0941	His	0.0242	Phe	0.0946
Lys	0.0371	Arg	0.0959	Ala	0.0373	Asp	0.1263

La transformación discreta de fourier se define mediante la ecuación 1:

$$X(n) = \sum_{m=0}^{N-1} x(m) e^{-j(2/N)nm} \quad n = 1, 2, \ldots, N/2 \tag{1}$$

donde $x(m)$ es el valor de la posición m de la serie numérica, N es el número de puntos en la serie, y $X(n)$ son los coeficientes de la transformada. La máxima frecuencia del espectro viene determinada por la siguiente ecuación:

$$F = \frac{1}{2d} \tag{2}$$

donde F es la frecuencia máxima y d es la distancia entre puntos de la secuencia.

Si se asume que todos los puntos de la secuencia son equidistantes con una distancia $d = 1$ entonces la frecuencia máxima del espectro sería $F = \frac{1}{2(1)} = 0{,}5$. Esto indica que el rango de frecuencia no depende del número de puntos en la secuencia sino de la resolución del espectro. La salida de la transformada de Fourier es una secuencia que se puede representar como indica la ecuación 3.

$$X(n) = (R(n) + I(n)j), \quad n = 1, 2, \ldots, N/2 \tag{3}$$

donde $R(n)$ es la parte real de la secuencia y la función $I(n)j$ la parte imaginaria.

El paso final en el cálculo del espectro absoluto de la transformada se calcula mediante la ecuación 4.

$$S_a(n) = X(n)X*(n) = |X(n)|^2, \quad n = 1, 2, \ldots, N/2 \tag{4}$$

donde S_a es el espectro absoluto para una proteína específica, $X(n)$ son los coeficientes de la transformada en las series de $x(n)$ y $X*(n)$ son los complejos conjugados. Para escalar el espectro absoluto se utiliza la ecuación 5.

$$V = \frac{\sqrt{\sum_{n=0}^{L} C_a(n)}}{L} \tag{5}$$

donde L es el número de puntos en el espectro absoluto (C_a).

Para el análisis de las proteínas del virus de la gripe A, como las secuencias tienen diferentes longitudes, la técnica del relleno vacío (completar con 0 las variables vacías) se emplea para extender las secuencias hasta un valor de $N = 512$, de forma que la salida del espectro absoluto (ec. 4) tenga 256 propiedades.

3. Descubrimiento de subgrupos y su aplicación al problema del virus de la gripe A

En esta sección se describe brevemente por un lado la técnica de SD y por otro las ventajas proporcionadas por el algoritmo NMEEF-SD en este problema.

3.1. Descubrimiento de subgrupos

El SD es un tipo de inducción descriptiva que pretende generar modelos basados en reglas cuya finalidad es descriptiva, empleando una perspectiva predictiva para obtenerlos [11,16]. Se trata por tanto de una tarea con objetivos básicamente descriptivos que incluye características de la inducción predictiva. Este concepto se define como [17]:

> *En SD, asumimos una población de individuos dada (objetos, clientes, ...) y una propiedad de estos individuos en la que estemos interesados. La tarea del SD es entonces descubrir los subgrupos de la población que son estadísticamente "más interesantes", es decir, individuos que sean tan grandes como sea posible y tenga una distribución estadística los más atípica posible, con respecto a una propiedad de interés.*

Así, una regla (R), que consiste de una descripción de un subgrupo inducido, puede ser definida formalmente como [12]:

$$R : Cond \rightarrow VarObj$$

donde $VarObj$ es el valor de la variable de interés o variable objetivo para la tarea de SD (puede aparecer además en la bibliografía específica como $Clase$), y $Cond$ es comúnmente una conjunción de funciones (pares atributo-valor) que es capaz de describir una distribución estadística inusual con respecto a la variable objetivo.

Existen diferentes elementos a especificar en el diseño de un algoritmo de SD [1], donde uno de los más destacados son las medidas de calidad utilizadas para el proceso de búsqueda y/o evaluación de los algoritmos. A continuación, se detallan las medidas de calidad más utilizadas en la literatura y en este trabajo:

- *Atipicidad*: Esta medida se describe como el balance entre la cobertura de la regla y la ganancia de precisión [12]. Se puede calcular como:

$$Atip(R) = \frac{n(Cond)}{n_s}\left(\frac{n(VarObj \cdot Cond)}{n(Cond)} - \frac{n(VarObj)}{n_s}\right) \qquad (6)$$

donde n_s es el número de ejemplos, $n(Cond)$ es el número de ejemplos que satisfacen la condición de la regla, $n(VarObj \cdot Cond)$ es el número de ejemplos que satisfacen la condición y además pertenecen al valor de la variable objetivo en la regla y $n(VarObj)$ son todos los ejemplos del valor de la variable objetivo.

- *Sensibilidad*: Esta medida mide la proporción de ejemplos correctamente descritos [11]. Se puede calcular como:

$$Sens(R) = \frac{n(VarObj \cdot Cond)}{n(VarObj)} \tag{7}$$

Esta medida de calidad se utiliza ara evaluar la calidad de los subgrupos en el espacio ROC (*Receiver Operating Characteristic*). La medida de sensibilidad combina la precisión y generalidad generada para un valor de la variable objetivo.

- *Confianza difusa*: Determina la frecuencia relativa de los ejemplos que satisfacen tanto el antecedente como el consecuente de una regla entre aquellos que satisfacen sólo el antecedente [7]. Se calcula como:

$$CnfD(R) = \frac{\sum\limits_{E^k \in E / E^k \in VarObj} APC(E^k, R)}{\sum\limits_{E^k \in E} APC(E^k, R)} \tag{8}$$

donde APC es el grado de compatibilidad entre un ejemplo y el antecedente de una regla difusa.

3.2. Aplicación al problema del virus de la gripe A del algoritmo NMEEF-SD

Tradicionalmente, el problema del virus de la gripe A se ha resuelto utilizando clasificadores. Sin embargo, el principal inconveniente de la utilización de los clasificadores para resolver problemas de bioinformática es, en general, la falta de interpretabilidad obtenida por los modelos. Esto se debe a que los modelos extraídos tienen la exactitud como principal objetivo, lo que provoca la obtención de modelos de una cierta complejidad, ya que utilizan un amplio número de variables o propiedades para describir diferentes virus del conjunto de datos. De esta forma, es muy difícil para los expertos analizar y comprender el comportamiento de un conjunto de datos con respecto a una variable de interés. Por el contrario, los algoritmos de SD extraen modelos sencillos, con pocas reglas y un bajo número de variables, para una variable objetivo.

La búsqueda de reglas interesantes y atípicas por los algoritmos de SD es una de las ventajas proporcionadas por la aplicación del algoritmo NMEEF-SD. Para este problema, el algoritmo utiliza las medidas de atipicidad (ec. 6) y sensibilidad (ec. 7) como vectores objetivo del enfoque multi-objetivo permitiendo además maximizar, no solo estas medidas, sino también otras medidas de la tarea de SD como la confianza.

Otra de las ventajas de la aplicación de NMEEF-SD es la utilización de lógica difusa [18] para resolver el problema, ya que la obtención de reglas difusas facilita el análisis a los expertos porque se emplean etiquetas lingüísticas en todas las variables del conjunto de datos, lo que proporciona a los expertos un conocimiento más cercano al razonamiento humano, empleando valores del lenguaje natural en vez de intervalos numéricos.

Por todo ello, NMEEF-SD es un algoritmo basado en un sistema multiobjetivo difuso evolutivo [8] que contribuye a extraer conocimiento novedoso y relevante sobre relaciones entre las propiedades del problema y diferentes tipos del virus de la gripe A.

4. Estudio experimental

El problema tiene una alta dimensionalidad y está compuesto por 256 variables y 546 secuencias de proteínas distribuidas de la siguiente forma: 200 secuencias del subtipo H1N1, 76 del H2N2, 200 del H3N2 y 70 del subtipo H5N1. Todas las variables son continuas y toman valores en el dominio de los números reales. El algoritmo NMEEF-SD considera las variables continuas como variables difusas lingüísticas aplicando lógica difusa. Más concretamente, en este problema se emplean funciones de pertenencia triangulares para las variables.

Los parámetros utilizados por el algoritmo NMEEF-SD son: tamaño de la población=50, evaluaciones=10000, probabilidad de cruce=0.6, mutación=0.1, etiquetas={3,5,7,9}, objetivos={atipicidad, sensibilidad} y confianza mínima={0.2, 0.4, 0.6}

Debido a la naturaleza no determinística del algoritmo NMEEF-SD, se ha aplicado un esquema de validación cruzada de 5 particiones, con 5 ejecuciones por partición. De esta forma, los resultados que se muestran son la media de los resultados obtenidos para cada conjunto de datos para las diferentes ejecuciones, es decir la media de las 25 ejecuciones (5 particiones x 5 ejecuciones de cada partición). En cada tabla, se muestran los valores de: número de etiquetas lingüísticas, umbral mínimo de confianza empleado (Min_{Cnf}), número de reglas (n_r), número de variables (n_v), atipicidad ($ATIP$), sensibilidad ($SENS$) y confianza ($CONF$).

El estudio experimental que se presenta a continuación consta de dos partes. Por una parte, en la sección 4.1 se estudian los resultados de la aplicación del algoritmo NMEEF-SD mientras por otro lado, en la sección 4.2 se aplica NMEEF-SD al conjunto de datos completo para obtener información descriptiva acerca de los diferentes tipos de virus estudiados en el problema.

4.1. Análisis de los resultados obtenidos por el algoritmo NMEEF-SD

Debido a la complejidad del problema se han utilizado diferentes número de etiquetas lingüísticas por variable y distintos umbrales de confianza mínima para

Tabla 2. Resultados obtenidos por el algoritmo NMEEF-SD

ELs	Min_{Cnf}	n_r	n_v	*ATIP*	*SENS*	*CONF*
3	0.2	4.60	2.79	0.153	1.000	0.747
	0.4	3.80	2.65	0.174	1.000	0.811
	0.6	2.60	2.73	0.190	1.000	0.849
5	0.2	3.40	2.13	0.125	0.990	0.708
	0.4	3.00	2.17	0.134	0.992	0.767
	0.6	2.20	2.10	0.148	1.000	0.807
7	0.2	3.00	2.28	0.110	0.963	0.760
	0.4	2.40	2.42	0.113	0.939	0.854
	0.6	1.60	2.37	0.127	0.938	0.911
9	0.2	1.60	2.00	0.092	0.952	0.585
	0.4	1.40	2.00	0.099	0.944	0.631
	0.6	0.60	0.80	0.048	0.378	0.394

encontrar la configuración del algoritmo que obtenga los mejores resultados para el mismo. Los resultados se muestran en la tabla 2.

En general, se puede observar que los mejores resultados se obtienen con el uso de 3 etiquetas lingüísticas y con un umbral de confianza de 0.6. Sin embargo, el número de reglas obtenido es inferior al número de virus analizados en el conjunto de datos, lo que indica que el algoritmo no ha obtenido reglas para describir todos los subtipos de virus. Por ello, se debe realizar un análisis de los subgrupos obtenidos por el algoritmo con 3 etiquetas lingüísticas y poder establecer la mejor configuración del algoritmo a este problema. Los resultados de este análisis se presentan en la tabla 3, donde se muestran los resultados de todos los subgrupos obtenidos en cada grupo de la validación cruzada para cada subtipo de virus.

Tabla 3. Resultados para cada subtipo de virus con 3 etiquetas lingüísticas

Min_{Cnf}	*Virus*	n_r	n_v	*ATIP*	*SENS*	*CONF*
0.2	H1N1	8.00	2.88	0.199	1.000	0.849
	H2N2	5.00	3.20	0.101	1.000	0.543
	H3N2	6.00	2.50	0.178	1.000	0.812
	H5N1	5.00	2.60	0.102	1.000	0.717
0.4	H1N1	8.00	2.88	0.199	1.000	0.849
	H2N2	3.00	2.33	0.107	1.000	0.601
	H3N2	5.00	2.40	0.193	1.000	0.835
	H5N1	3.00	3.00	0.104	1.000	0.768
0.6	H1N1	7.00	3.00	0.202	1.000	0.867
	H2N2	0.00	0.00	0.000	0.000	0.000
	H3N2	5.00	2.40	0.193	1.000	0.835
	H5N1	1.00	3.00	0.101	1.000	0.867

Como se ha mencionado anteriormente en el análisis de la tabla 2 y con los resultados mostrados en la tabla 3, los subgrupos obtenidos para un umbral de confianza de 0.6 indica que no hay subgrupos para poder describir todos los subtipos de virus. Esto se debe a que el nivel de confianza es muy alto para obtener buenos resultados en todas las virus. Por ello, los resultados obtenidos en esta configuración deben ser descartados.

Por tanto, los mejores resultados para el algoritmo NMEEF-SD se obtienen con 3 etiquetas lingüísticas y un umbral de confianza mínimo de 0.2 y 0.4. Este estudio se completa con un análisis de los subgrupos obtenidos para cada virus:

- Los subgrupos obtenidos para el virus H1N1 tienen una alta interpretabilidad porque el número de variables es bajo, donde en general los subgrupos obtenidos tienen menos de 3 variables (considerando también la variable objetivo como una variable). Los valores para la medida de atipicidad son los más altos con respecto a los valores obtenidos en el resto de clases. Además, la relación entre sensibilidad y confianza es muy bueno, ya que el algoritmo obtiene subgrupos donde todas las secuencias de proteínas para los virus son cubiertas y la confianza está cercana al 85 %.
- Para el virus H2N2 se obtienen los subgrupos con el menor número de variables por lo que la interpretabilidad en este virus es excelente. Los valores de atipicidad son además altos considerando que este virus tiene un número muy bajo de secuencias en el conjunto de datos. El nivel de sensibilidad de los subgrupos extraídos es el máximo y el valor de la confianza es bueno ya que los subgrupos superan el 60 %.
- En el virus H3N2 se obtienen los mejores subgrupos juntos con el virus H1N1, donde la interpretabilidad y los valores de la atipicidad, sensibilidad y confianza son muy altos.
- El virus H5N1 es el subtipo con menor número de proteínas. A pesar de esto, los resultados de sensibilidad y confianza son muy interesantes porque los subgrupos cubren todos los ejemplos del virus con un buen nivel de confianza (más del 70 %). Los resultados para la relevancia y atipicidad son además aceptables.

4.2. Subgrupos difusos extraídos por el algoritmo NMEEF-SD

Una vez analizado en la etapa anterior la configuración con mejores resultados para el algoritmo, a continuación se realiza un nuevo experimento utilizando el conjunto de datos completo, para analizar los subgrupos obtenidos por el algoritmo que puedan ser de interés para los expertos con un umbral mínimo de confianza de 0.2 y 3 etiquetas lingüísticas.

La tabla 4 muestra los subgrupos obtenidos por el algoritmo NMEEF-SD para cada clase, en la que la propiedad número x se identifica con el nombre $f(x)$. La tabla presenta además los resultados asociados a cada subgrupo.

Tabla 4. Subgrupos obtenidos por el algoritmo NMEEF-SD

Subgrupo	***ATIP***	***SENS***	***CONF***
$SI\ (f44 = Bajo\ Y\ f97 = Bajo) \rightarrow$ H1N1	0.224	1.000	0.966
$SI\ (f9 = Bajo\ Y\ f54 = Bajo\ f153 = Bajo\ Y\ f217 = Bajo) \rightarrow$ H2N2	0.105	1.000	0.600
$SI\ (f8 = Bajo) \rightarrow$ H3N2	0.182	1.000	0.730
$SI\ (f141 = Bajo\ Y\ f207 = Bajo\ Y\ f219 = Bajo) \rightarrow$ H3N2	0.196	0.995	0.966
$SI\ (f115 = Bajo) \rightarrow$ H5N1	0.097	1.000	0.677

Como se puede observar en la tabla 4, los buenos resultados en atipicidad muestran conocimiento novedoso y desconocido del problema. Además, la sensibilidad obtenida para la mayoría de los subgrupos tiene el máximo nivel (100 %) y la confianza es muy alta con valores que están por encima del 60 % y algunos muy cercanos al máximo nivel. Estas buenas relaciones entre los valores de sensibilidad y confianza presentan subgrupos de alta calidad. La interpretabilidad es también excelente, obteniendo subgrupos que en ningún caso superan las 4 variables, es decir se obtienen subgrupos con únicamente 4 variables como máximo de un total de 256.

Otros métodos que utilizan técnicas de procesamiento para extraer propiedades biológicamente relacionadas para caracterizar secuencias de proteínas, como el *Resonant Recognition Model* en el gen HA [15] y *Complex Resonant Recognition* para el gen NA [4], emplean análisis informativos de espectro para caracterizar un tipo de virus específico o compararlo con otras proteínas basadas en picos de frecuencia comunes [4]. Mediante el uso del algoritmo NMEEF-SD, tal y como se muestra en la tabla 4, se pueden extraer reglas sencillas basadas en la recuperación de propiedades del espectro absoluto, con respecto al virus de la gripe A. Con estas propiedades se puede obtener conocimiento que permita mejorar el análisis sobre este tipo de virus, ya que permite a los expertos centrarse en un conjunto reducido de propiedades. Esto se traduciría, por ejemplo, en que para una secuencia de proteína desconocida con este modelo se puede determinar qué tipo de virus es, estudiando su comportamiento en 11 variables en vez de tener que analizar el espectro completo de 256 variables.

5. Conclusiones

La búsqueda de relaciones novedosas y atípicas entre los subtipos del virus de la gripe A, proporciona a los expertos conocimiento novedoso relacionado con este virus que pueda aportar información para ayudarlos en el desarrollo de nuevas terapias o vacunas para este virus.

El conjunto de reglas obtenidas por el algoritmo se podría utilizar por tanto para el desarrollo de nuevas terapias y/o vacunas para mejorar los tratamientos y combatir el virus de la gripe A con un conjunto de solo 11 propiedades.

Acknowledgments.

Este trabajo ha sido subvencionado por el Ministerio de Economía y Competitividad bajo el proyecto TIN201233856, Fondos FEDER, y el por el Plan de Investigación de Andalucía bajo el proyecto TIC-3928, Fondos FEDER.

Referencias

1. M. Atzmueller, F. Puppe, and H. P. Buscher, *Towards Knowledge-Intensive Subgroup Discovery*, Proceedings of the Lernen - Wissensentdeckung - Adaptivität - Fachgruppe Maschinelles Lernen, 2004, pp. 111–117.

2. Y. Bao, P. Bolotov, D. Dernovoy, B. Kiryutin, L. Zaslavsky, T. Tatusova, J. Ostell, and D. Lipman, *The influenza virus resource at the National Center for Biotechnology Information*, Journal of virology **82** (2008), no. 2, 596.
3. C. J. Carmona, P. González, M. J. del Jesus, and F. Herrera, *NMEEF-SD: Non-dominated Multi-objective Evolutionary algorithm for Extracting Fuzzy rules in Subgroup Discovery*, IEEE Transactions on Fuzzy Systems **18** (2010), no. 5, 958–970.
4. C. Chrysostomou, H. Seker, N. Aydin, and P. Haris, *Complex Resonant Recognition Model in Analysing Influenza A Virus Subtype Protein Sequences*, 10th IEEE International Conference on Information Technology and Applications in Biomedicine, 2010.
5. I. Cosic, *Macromolecular bioactivity: is it resonant interaction between macromolecules: Theory and applications*, IEEE transactions on bio-medical engineering **41** (1994), 1101–1114.
6. I. Cosic and E. Pirogova, *Bioactive peptide design using the Resonant Recognition Model*, Nonlinear Biomedical Physics **1** (2007), no. 1, 7.
7. M. J. del Jesus, P. González, F. Herrera, and M. Mesonero, *Evolutionary Fuzzy Rule Induction Process for Subgroup Discovery: A case study in marketing*, IEEE Transactions on Fuzzy Systems **15** (2007), no. 4, 578–592.
8. M. Fazzolari, R. Alcalá, Y. Nojima, H. Ishibuchi, and F. Herrera, *A review of the application of Multi-Objective Evolutionary Systems: Current status and further directions*, IEEE Transactions on Fuzzy Systems **21** (2013), no. 1, 45–65.
9. F. Herrera, *Genetic fuzzy systems: taxomony, current research trends and prospects*, Evolutionary Intelligence **1** (2008), 27–46.
10. F. Herrera, C. J. Carmona, P. González, and M. J. del Jesus, *An overview on Subgroup Discovery: Foundations and Applications*, Knowledge and Information Systems **29** (2011), no. 3, 495–525.
11. W. Kloesgen, *Explora: A Multipattern and Multistrategy Discovery Assistant*, Advances in Knowledge Discovery and Data Mining, American Association for Artificial Intelligence, 1996, pp. 249–271.
12. N. Lavrac, B. Cestnik, D. Gamberger, and P. A. Flach, *Decision Support Through Subgroup Discovery: Three Case Studies and the Lessons Learned*, Machine Learning **57** (2004), no. 1-2, 115–143.
13. A. Moscona, *Neuraminidase inhibitors for influenza*, New England Journal of Medicine **353** (2005), no. 13, 1363.
14. V. Veljkovic, I. Cosic, B. Dimitrijevic, and D. LalovicC, *Is it possible to analyze DNA and protein sequences by the methods of digital signal processing?*, IEEE Transaction on Biomedical Engineering **32** (1985), no. 5, 337–341.
15. V. Veljkovic, N. Veljkovic, C. P. Muller, S. Mueller, S. Glisic, V. Perovic, and H. Koehler, *Characterization of conserved properties of hemagglutinin of H5N1 and human influenza viruses: possible consequences for therapy and infection control*, BMC Structural Biology **9** (2009).
16. S. Wrobel, *An Algorithm for Multi-relational Discovery of Subgroups*, Proceedings of the 1st European Symposium on Principles of Data Mining and Knowledge Discovery, LNAI, vol. 1263, Springer, 1997, pp. 78–87.
17. ———, *Inductive logic programming for knowledge discovery in databases*, ch. Relational Data Mining, pp. 74–101, Springer, 2001.
18. L. A. Zadeh, *Soft Computing and Fuzzy Logic*, IEEE Software **11** (1994), no. 6, 48–56.

Minería de datos para la predicción de retenciones de tráfico

O. Castaño[1], F. Martínez-Álvarez[1], A. Troncoso[1], J. C. Riquelme[2]

[1]Departamento de Lenguajes y Sistemas Informáticos. Universidad Pablo de Olavide.
ocascal@alu.upo.es,fmaralv@upo.es,ali@upo.es
[2]Departamento de Lenguajes y Sistemas Informáticos. Universidad de Sevilla.
riquelme@us.es

Resumen. El objetivo de este trabajo de investigación es la búsqueda de patrones y modelos del comportamiento de las series temporales de datos recogidos por los detectores de la Dirección General de Tráfico en carretera. La extracción de dichos patrones, se utilizará para predecir el comportamiento del sistema de modo tan preciso como sea posible, con el objetivo de poder alertar con suficiente antelación la generación de retenciones en la circulación, minimizando el tiempo de respuesta y dando alternativas al flujo de vehículos.

Palabras clave: Congestión de tráfico, predicción retenciones, minería de datos, series temporales.

1. Introducción

La congestión del tráfico en la Ronda de Circunvalación SE-30, concretamente en el Puente del Centenario, es un problema latente en la ciudad de Sevilla. El volumen de vehículos que soporta esta sección de la vía varía en tramos horarios de un sentido a otro. Este puente está provisto de un carril reversible y desde su comienzo hasta su final es imposible abandonar el mismo por algún camino alternativo. Por ello, es de vital importancia una correcta gestión de dicho carril reversible y aún más la posibilidad de brindar alternativas al tráfico en momentos de máxima afluencia.

El Centro de Gestión del Tráfico del Suroeste (CGTS) maneja un total de 220 estaciones de toma de datos en carretera y 214 cámaras de circuito cerrado de televisión, entre otros recursos y equipamiento. Uno de los objetivos de este organismo es prevenir la congestión y reducir su duración y longitud una vez que se han producido. Para ello, lo más fácil sería incrementar la capacidad de las infraestructuras existentes, pero esto exige un elevado coste y es primordial optimizar recursos.

En este trabajo se muestra una propuesta de propósito general para predecir congestiones de vehículos en vías de circulación. Como caso de estudio, se evalúa en el punto kilométrico anteriormente comentado. Para la elaboración de este estudio se han utilizado datos de tráfico recogidos por detectores en carretera por

medio de las estaciones de toma de datos (ETD), que son transferidos al CGTS de la Dirección General de Tráfico, con cuya ayuda se ha realizado el procesamiento de las variables de entrada. Tras la aplicación de distintos algoritmos, se ha predicho para cuatro escenarios distintos (15, 30, 45 y 60 minutos de antelación) las posibles retenciones, obteniéndose en promedio valores superiores al 85 %.

La saturación y la obstrucción son causas de un atasco. Además, no se debe pasar por alto los efectos sobre la salud humana de los gases emitidos por los vehículos [4]. Un embotellamiento puede estar provocado por dos motivos. En primer lugar por un volumen muy alto de vehículos circulando al mismo tiempo. Cuando éste supera la capacidad para la que está preparada la carretera, la vía se satura y se producen embotellamientos. Este volumen es de 2000 vehículos por carril y hora, unos dos automóviles por segundo aproximadamente, y se calcula para un carril que está en unas condiciones óptimas: ancho de dos a tres metros y pavimento en buen estado. Este tipo de retenciones son las más habituales, se repiten periódicamente cuando la gente va y vuelve del trabajo o con las salidas de vacaciones.

En segundo lugar, el tráfico puede estar provocado por incidentes como averías de vehículos, accidentes, derramamientos de objetos en la calzada, lluvia, nevadas, etc. Estos hechos provocan que disminuya la capacidad de circulación en un punto de la carretera. Los coches se van acumulando, empiezan a hacer adelantamientos para esquivar los obstáculos y se termina formando un embudo.

Una vez que se ha formado un atasco, éste crece muy rápidamente. Los vehículos llegan a la retención circulando a mucha velocidad, tienen que frenar bruscamente, y provocan que los que van detrás también lo hagan. El resultado es una retención que se propaga hacia atrás a mucha velocidad, a un ratio de unos cinco metros por cada vehículo que llega al tapón [10].

Hay una serie de factores que contribuyen a agravar un embotellamiento. Uno de ellos es la propia composición del tráfico [12]. Un carril por el que circulan cuatro camiones en fila no tiene la misma capacidad que uno que sólo está ocupado por coches. En general, los vehículos pesados circulan más lentos y generan largas colas de automóviles. Otro factor que incide negativamente sobre el tráfico es la interacción entre automóviles. Cuando la densidad de circulación es superior a los 1700 vehículos por hora, cualquier adelantamiento o maniobra provoca una reducción de la velocidad de la carretera [8]. Los ramales e incorporaciones también ayudan a complicar más las cosas en un atasco, ya que traen consigo un aporte extra de vehículos a una vía que de por sí ya está saturada.

Reducir la velocidad ayuda a evitar atascos. Permite circular de forma más homogénea. Si todos los vehículos que van por una carretera lo hicieran a poca velocidad, la capacidad de circulación de la vía no se saturaría tan rápidamente. La masa de coches avanzaría lentamente, pero de forma uniforme. Es algo muy complejo y posiblemente la solución más factible sea crear más infraestructura para más coches. El problema es que según se van construyendo más carreteras y autopistas, también se incrementa el número de coches, y al poco tiempo el tráfico queda de la misma manera que antes. Se han intentado tomar ciertas

medidas en varias ciudades del mundo, donde en algunos sitios han funcionado mejor que otros [2].

En algunas autopistas, se han hecho obras para añadir nuevos carriles, ampliándola de forma general, aprovechando el arcén, o disminuyendo el tamaño de los carriles existentes si es factible. Estas series de ajustes son bastante caros y llevan mucho tiempo. Todo depende del crecimiento de la población y la venta de vehículos, por lo que es una solución a medio plazo [1].

Una solución menos costosa sería estudiar como se comporta el tráfico, a partir de información de sensores en carretera [5], en un determinado segmento de la red viaria según sus variables: [9] congestión, intensidad, ocupación, velocidad media de los vehículos, etc. La obtención de un modelo para controlar el tráfico varía dependiendo del área de estudio.

El resto del trabajo se divide como sigue. En la Sección 2, por el contrario, se centra en la metodología seguida para la predicción de la congestión en el punto kilométrico bajo estudio. Los resultados obtenidos se presentan y discuten en la Sección 3. Por último, las conclusiones derivadas de este estudio se resumen en la Sección 4.

2. Metodología

Para poder modelar el comportamiento del tráfico de forma correcta es preciso interpretar los datos recibidos por la ETD de forma coherente para realizar un posterior filtrado. El sistema comunica directamente con el Centro de Gestión del Tráfico, donde se encuentran los servidores de información y datos. Estos equipos son los que procesan en un primer nivel los datos para eliminar valores erróneos, fuera de rango, a la vez que se encargan de mantener en comunicación los equipos de toma de datos con el centro de control. Estos dispositivos son las Estaciones Remotas Universales (ERU) y permiten la manipulación de los equipos de campo como periféricos propios de las ERUs.

2.1. Información proporcionada por las ETD

La ETD utiliza un conjunto de sensores instalados en la calzada para detectar el paso de los vehículos. Una ETD tiene capacidad para controlar hasta un total de 48 puntos de sensorización que podrán ser de tipo inductivo (espiras), piezoeléctricos (sensores de ejes), lazos virtuales de infrarrojos, microondas o láser (con los correspondientes equipos adicionales). En función de las características de los sensores instalados en la calzada, (por ejemplo detectores simples o dobles) se obtienen un conjunto de variables de tráfico u otras.

Los datos sobre los que se basa este estudio son de espiras electromagnéticas, que son unas bobinas constituidas por un solo conductor empotrado en el pavimento que da varias vueltas formando un polígono en la calzada, situado en la zona de paso de los vehículos. El funcionamiento es el siguiente: la ETD está constantemente a la espera de pulsos por las entradas digitales que producen las espiras electromagnéticas colocadas en la calzada. Ante la recepción de uno,

lo guarda y analiza la secuencia completa. La ETD tiene definidas una serie de secuencias que coinciden con el paso de vehículos. En caso de que la secuencia coincida con alguno de los tipos predefinidos, se analiza, se averiguan los datos del vehículo detectado y se produce su conteo. Cada cierto tiempo (periodo de integración), los datos se acumulan y se mantienen listos para servir.

Estos datos están referidos a las siguientes variables de tráfico que se obtienen a partir de todos los vehículos registrados por la entidad inferior del sistema, detector (formado por espiras dobles) ó carril durante el periodo de integración (PI), de 1 minuto:

1. Intensidad (INTENSIDAD). Número de vehículos que han sido registrados por el detector D en el periodo de integración PI de 1 minuto. Se expresa en número vehículos por hora.
2. Ocupación (TOCUP). Media aritmética de los tiempos de ocupación de los vehículos que han sido registrados por el detector D en el periodo de integración. Se expresa en porcentaje % respecto del PI de 1 minuto.
3. Velocidad media (VMED). Media aritmética de las velocidades de los vehículos que han sido registrados por el detector D en el periodo de integración, medida en Km/h.
4. Distancia media (GAP). Media aritmética de las distancias entre vehículos consecutivos registrados por el detector D en el periodo de integración, medida en metros.
5. Longitud media (LONGMED). Media aritmética de las longitudes de los vehículos que han sido registrados por el detector D en el periodo de integración, medida en decímetros.
6. Composición longitud (CLASLONG). Número de vehículos clasificados por longitud, de acuerdo a las categorías establecidas según los umbrales configurados, que han sido registrados por el detector D en el periodo de integración. Se expresa en % de la categoría que corresponda (ligero o pesado).
7. Composición velocidad (CLASVEL). Número de vehículos clasificados por velocidad, de acuerdo a las categorías establecidas según los umbrales configurados, que han sido registrados por el detector D en el periodo de integración. El sistema permite obtener hasta 3 clasificaciones distintas por velocidad. Se expresa en % de la categoría que corresponda (ligero o pesado).
8. Alarma kamikaze (VEH_INV). Variable booleana que indica la existencia de algún vehículo circulando en sentido contrario en el periodo de integración. Se expresa en SÍ=1 ó NO=0.
9. Sentido inverso (SENT_CIR). Variable booleana que indica que la circulación en el periodo de integración ha presentado una modificación de sentido respecto al habitual. Esto sucede en carril reversible, que en determinado instante varía su sentido de circulación, y puede ocurrir en mitad de un periodo de integración. Se expresa en SÍ=1 ó NO=0.
10. Error (ERR_REL). Es el porcentaje de tiempo dentro del periodo de integración en que si hubiesen pasado vehículos, el detector D no los registra. Este parámetro representa la fiabilidad de la medida obtenida en el periodo de integración. Se expresa en %.

11. Congestión (CONGESTION). Variable booleana que indica la existencia de alguna activación de alarma de congestión determinada por el algoritmo HIOCC desarrollado por el Transport and Road Reserch Lab (TRRL) [3] y que viene implementado en el ETD en el periodo de integración. Se expresa en SÍ=1 ó NO=0. Esta variable será la etiqueta de clase y, por tanto, la que se quiere predecir correctamente.

2.2. Algoritmos de minería de datos utilizados

Se van a aplicar tres métodos de clasificación para encontrar el modelo que mejor se adapte a la resolución del problema. Los métodos usados son el algoritmo C4.5 [13] (que será el algoritmo de control), Naive Bayes [6] y vecinos cercanos [7]. Las implementaciones usadas son las que vienen en WEKA [11], herramienta de minería de software libre, y en concreto las implementaciones *J48*, *NaiveBayes* e *IB1*.

El algoritmo J48 es un procedimiento para generar un árbol de decisión. Generar un árbol de decisión consiste en seleccionar un atributo como raíz del árbol y crear una rama con cada uno de los posibles valores de dicho atributo. Con cada rama resultante (nuevo nodo del árbol), se realiza el mismo proceso, se selecciona otro atributo y se genera una nueva rama para cada posible valor del atributo. Este procedimiento continúa hasta que los ejemplos se clasifiquen a través de uno de los caminos del árbol. El nodo final de cada camino será un nodo hoja, al que se le asignará la clase correspondiente.

Así, el objetivo de los árboles de decisión es obtener reglas o relaciones que permitan clasificar a partir de los atributos. Este algoritmo permite el empleo del concepto ganancia de información, construir árboles de decisión cuando algunos de los ejemplos presentan valores desconocidos para algunos de los atributos, trabajar con atributos que presenten valores continuos, la poda de los árboles de decisión y la obtención de reglas de clasificación.

Los clasificadores bayesianos son clasificadores estadísticos que pueden predecir tanto las probabilidades del número de miembros de clase, como la probabilidad de que una muestra dada pertenezca a una clase particular. La clasificación bayesiana, basada en el teorema de Bayes, han demostrado una alta exactitud y velocidad cuando se han aplicado a grandes bases de datos. Naive Bayes consiste en clasificar un nuevo ejemplo de acuerdo con el valor más probable dados los valores de sus atributos.

Por último, el algoritmo de los vecinos cercanos se basa en la cercanía entre la muestra que se desea clasificar y las muestras que forman el conjunto de entrenamiento. Se trata pues de un algoritmo determinista cuyo punto crítico radica en la determinación del número de vecinos que se quieren utilizar. Las estrategias más frecuentes son coger simplemente el más cercano (aquella muestra del conjunto de entrenamiento que presente la menor distancia con respecto a la muestra que se quiere etiquetar) o tomar un número impar de vecinos para asegurar que no haya empate entre sus clases, en caso de que sea una clasificación binaria.

3. Resultados

En esta sección se presentan, en primer lugar, los parámetros utilizados para evaluar la calidad de los resultados. A continuación, se describe el tratamiento al que se han sido sometidos los datos de entrada, así como la generación de la etiqueta de clase en los cuatro escenarios analizados. Por último, se detallarán los resultados de aplicar los algoritmos J48, Naive Bayes e IB1.

3.1. Parámetros de calidad

En esta sección presentamos los parámetros de calidad utilizados para evaluar los diferentes métodos. Se ha optado por utilizar los más comunes en la literatura. Por un lado tenemos un conjunto de indicadores, que cuantifican los aciertos y errores de los clasificadores:

1. Verdaderos positivos, TP (del inglés *true positive*). Se define como el número de veces que el clasificador asigna un 1 a la instancia que está clasificando (predice la ocurrencia de una retención), y ésta, efectivamente, ocurre durante los siguientes minutos.
2. Verdaderos negativos, TN (del inglés, *true negative*). Es el número de veces que se ha predicho que no ocurrirá una retención durante los próximos minutos y, efectivamente, ésta no ocurre.
3. Falso positivo, FP (del inglés, *false positive*). El número de veces que se ha detectado erróneamente una retención en los próximos minutos, es decir, el número de veces que el clasificador asignó una etiqueta con valor 1 cuando en realidad debía asignar un 0.
4. Falso negativo, FN (del inglés, false negative). El número de veces que no se ha conseguido detectar una retención pero sí hubo una retención minutos posteriores.

A partir de estos cuatro indicadores, se calculan los parámetros de calidad, propiamente dicho. En particular:

1. Sensibilidad, S. Es la proporción de retenciones correctamente identificadas, sin tener en cuenta los FP. Matemáticamente se expresa como: $S = TP/(TP + FN)$.
2. Especificidad, E. Ratio de negativos identificados correctamente. Matemáticamente se calcula como: $E = TN/(TN + FP)$.
3. Valor predictivo positivo, PPV (del inglés, *positive predictive value*). Mide la fiabilidad de los TP, es decir, la certeza asociada a cada TP. En otras palabras, la relación entre TP y FP. Matemáticamente, se formula como: $PPV = TP/(TP + FP)$.
4. Valor predictivo negativo, NPV (del inglés *negative predictive value*). Mide la fiabilidad de los TN, es decir, la certeza asociada a cada TN. En otras palabras, es la relación entre los TN y los FN. Matemáticamente: $NPV = TN/(TN + FN)$.

3.2. Selección y transformación de los datos

Para este estudio, se han obtenido datos del detector del kilómetro 12 decreciente de la SE-30 del último trimestre de 2011 utilizando para el análisis un total de 8926 registros. Se han descartado, por escasa relevancia para el estudio, los atributos de VEH_INV, SENT_CIR, LONGMED, CLASLONG, CLASVEL y ERR_REL.

Además, dentro de todo el conjunto de registros se han descartado aquellos cuyo ERR_REL superaban el 5 %, aproximadamente el 0.056 % de los datos. También se han descartado aquellos registros en los cuales el número de muestras de minuto en un cuarto de hora eran menores a 15 y los que presentan valores cero en las muestras en los intervalos nocturnos, si bien estos valores son normales por el hecho de que de madrugada el tráfico es intermitente y suele ser disperso.

Para ver la importancia de los atributos, por medio de nubes de puntos se ha observado que existen relaciones del tipo:

1. A mayor número de vehículos la distancia entre ellos es menor.
2. La distancia entre vehículos disminuye cuando aumenta el porcentaje de ocupación.
3. A mayor intensidad se produce una menor ocupación de la vía.

Por otro lado, como etiqueta de clase se ha tomado el atributo *Congestión,* que toma valor SÍ=1 ó NO=0. Como se han aplicado las técnicas de minería a distintos intervalos de tiempo, se ha desplazado la clase para analizar el comportamiento a 15, 30, 45 y 60 minutos. Esto es, para cada uno de los cuatro escenarios evaluados, la etiqueta de clase corresponde a si habrá o no congestión en los próximos 15, 30, 45 ó 60 minutos.

3.3. Extracción de conocimiento

En esta sección se muestran los resultados obtenidos tras la aplicación de los algoritmos J48, Naive Bayes e IB1 en los cuatro escenarios analizados, así como una discusión de los mismos.

El periodo de entrenamiento comprende los tres primeros trimestres del año 2011, mientras que para evaluar el modelo se utilizaron las muestras registradas en el último trimestre de dicho año.

La Tabla 1 muestra los resultados para la predicción de congestión de tráfico para cuatro escenarios posibles: con 15 minutos, con 30 minutos, 45 minutos y una hora de antelación cuando se aplicó J48, el algoritmo que mejor resultado reportó. Además, es importante resaltar que la elección de este algoritmo fue hecha acorde a las indicaciones recibidas desde la DGT, ya que se querían resultados de fácil interpretación. En este sentido, la estructura mostrada por los árboles son de interpretación inmediata para cualquier profesional que no esté relacionado con la minería de datos.

Se puede apreciar que los resultados fueron en general muy competitivos, ya que ninguno de los cuatro escenarios bajó del 85 % en promedio. No obstante,

Tabla 1. Resultados para la aplicación de J48.

Parámetro	15 min.	30 min.	45 min.	60 min.
TP	440	511	595	654
TN	8443	8453	8502	8543
FP	228	157	73	14
FN	115	104	54	13
Sensibilidad	79.3 %	83.1 %	91.7 %	98.1 %
Especificidad	97.4 %	98.2 %	99.1 %	99.8 %
PPV	65.9 %	76.5 %	89.1 %	97.9 %
NPV	98.7 %	98.8 %	99.4 %	99.8 %
Media	85.3 %	89.2 %	94.8 %	98.9 %

destaca de manera especial los valores obtenidos para una hora de antelación, en la que los resultados rozaron, en promedio, el 99 % de precisión.

En particular, se puede apreciar que, aplicando el método J48 a 60 minutos se han generado correctamente un total de 654 alarmas (TP), lo cual quiere decir que se han predicho un total de 654 retenciones de manera correcta. Especialmente destacable es el hecho de que sólo se generaron 14 falsas alarmas (FP) frente a los más de 9000 intervalos horarios evaluados. Esto hace que de las 668 retenciones registradas, se haya logrado predecir 654.

Por otro lado, se puede apreciar cómo se califican 8543 estados de no alarma (TN) y que sólo ha habido 13 alarmas no avisadas (FN), haciendo que de las 8556 situaciones de tráfico fluido registradas, 8543 fueran correctas.

El resto de los escenarios evaluados también mostraron resultados favorables en promedio. Sin embargo, debe destacarse que las falsas alarmas fueron sensiblemente superiores para los demás intervalos horarios (228 para 15 minutos, frente a las 14 registradas para una hora de antelación). Este hecho debe cuidarse mucho, ya que activar una política de actuación errónea puede causar incluso más perjuicio que el hecho de no aplicarla.

La Figura 1 muestra el árbol construido por J48 para la predicción de retenciones con una hora de antelación. Se puede apreciar la sencillez de las reglas obtenidas: una retención se producirá en una hora si el tiempo de ocupación es mayor de 27 y el gap menor o igual a 7. En el resto de los casos, no se producirá retención alguna. No se muestra aquí el resto de los árboles generados por motivos de espacio. Sin embargo, cabe destacar que las relaciones encontradas no fueron tan directas ni tan sencillas ya que, en promedio, la profundidad de los tres árboles fue de 4 y tuvieron un total de 12 hojas, en promedio.

Para evaluar la calidad de los valores obtenidos por J48, se muestra en la Tabla 2 los resultados de aplicar Naive Bayes e IB1 a los mismos cuatro escenarios antes descritos para los mismos periodos de entrenamiento y test.

Se puede apreciar que el mejor de los resultados para Naive Bayes (30 minutos de antelación) es peor que el peor de los resultados para el J48 (una hora de antelación). Además, se da la circunstancia de que si en algunos casos del J48 las falsas alarmas se consideraban demasiado elevadas, aquí se dispara el número, llegando a alcanzar las 472 para el caso de 15 minutos de antelación.

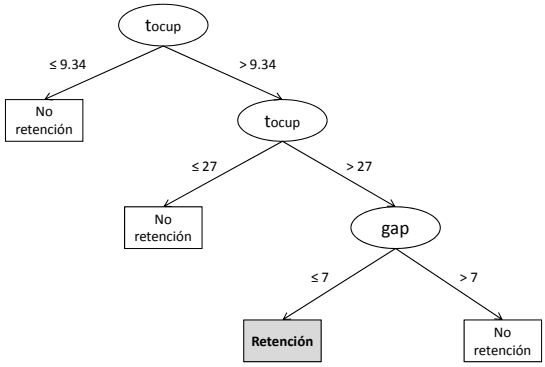


Figura 1. Árbol creado por J48 para el caso de predicción de retención con 60 minutos de antelación

Tabla 2. Resultados para la aplicación de Naive Bayes e IB1.

	Naive Bayes				IB1			
Parámetro	15 min.	30 min.	45 min.	60 min.	15 min.	30 min.	45 min.	60 min.
TP	196	912	340	397	278	220	175	134
TN	7780	7645	7667	7720	8169	8109	8085	8006
FP	472	401	328	271	390	448	471	549
FN	778	267	889	835	389	448	493	534
Sensibilidad	20.1 %	77.4 %	27.7 %	32.2 %	41.7 %	32.9 %	26.2 %	20.1 %
Especificidad	94.3 %	95.0 %	95.9 %	96.6 %	95.4 %	94.8 %	94.5 %	93.6 %
PPV	29.3 %	69.5 %	50.9 %	59.4 %	95.5 %	94.8 %	94.3 %	96.8 %
NPV	90.9 %	96.6 %	89.6 %	90.2 %	41.6 %	32.9 %	27.1 %	19.7 %
Media	55.4 %	84.6 %	66.0 %	69.6 %	68.6 %	63.9 %	60.5 %	56.8 %

Para el caso de IB1, a diferencia de lo ocurrido para J48 y Naive Bayes, la calidad de la predicción va decayendo conforme agrandamos el horizonte temporal. Los resultados de IB1 son los más pobres de todos, ya que su mejor resultado (predicción a 15 minutos) es apenas tres puntos pocentuales superior al peor de Naive Bayes. Llama la atención el elevado número de FP obtenido en los cuatro escenarios evaluados.

4. Conclusiones

En este trabajo se han buscado patrones y modelos del comportamiento de datos recogidos por los detectores de la Dirección General de Tráfico en carretera, en concreto en el punto kilométrico 12 decreciente de la ronda de circunvalación SE-30. La extracción de dichos patrones, se ha utilizado para predecir con suficiente antelación la congestión de dicho punto, con el objetivo de prevenir retenciones, minimizando así el tiempo de respuesta a dicho suceso y dando alternativas a la circulación. Para ello, se ha aplicado de manera satisfactoria el conocido

algoritmo J48 y se han obtenido resultados con un error variando entre el 85 % y el 95 %, dependiendo de la antelación de la predicción realizada.

Agradecimientos

Los autores quieren agradecer a la Dirección General de Tráfico por haber proporcionado los datos y el soporte necesario para realizar este estudio. Igualmente, se quiere agradecer la financiación económica recibida por parte del Ministerio de Ciencia e Innovación, proyecto TIN2007-68084-C-00.

Referencias

1. M. Ahmane, A. Abbas-Turki, F. Perronnet, J. Wu, A. El-Moudni, J. Buisson, and R. Zeo. Modeling and controlling an isolated urban intersection based on cooperative vehicles. *Transportation Research Part C: Emerging Technologies*, 28:44–62, 2013.
2. R. C. Carlson, I. Papamichail, M. Papageorgiou, and A. Messmer. Optimal mainstream traffic flow control of large scale motorway networks. *Transportation Research Part C: Emerging Technologies*, 18:193–212, 2010.
3. J. F. Collins. Automatic incident detection: Experience with TRRL algorithm HIOCC. *TRRL Supplementary Report*, 775:1–6, 1993.
4. J. Currie and R. Walker. Traffic congestion and infant health: Evidence from E-ZPass. *American Economic Journal: Applied Economicsy*, 3(1):65–90, 2011.
5. J. Dunkel, A. Fernandez, R. Ortiz, and S. Ossowski. Event-driven architecture for decision support in traffic management systems. *Expert Systems with Applications*, 38:6530–6539, 2011.
6. D. J. Hand and K. Yu. Idiot's Bayes - not so stupid after all? *International Statistical Review*, 69(3):385–399, 2001.
7. P. E. Hart. The condensed nearest neighbor rule. *IEEE Transactions on Information Theory*, 18:515–516, 1968.
8. J. Z. Hernández, S. Ossowski, and A. García-Serrano. Multiagent architectures for intelligent traffic management system. *Transportation Research Part C: Emerging Technologies*, 10(5):473–506, 2002.
9. W. H. Lee, S. S. Tseng, and W. Y. Shieh. Collaborative real-time traffic information generation and sharing framework for the intelligent transportation system. *Information Sciences*, 180:62–702, 2010.
10. K. Mahmud, K. Gope, and S. M. R. Chowdhury. Possible causes and solutions of traffic jam and their impact on the economy of Dhaka city. *Journal of Management and Sustainability*, 2(2):p112, 2012.
11. WEKA The University of Waikatu. *Data mining with open source machine learning software in Java*. http://www.cs.waikato.ac.nz/ml/weka.
12. Z. Wei Qu, Y. Xing, X. M. Song, Y. Z. Duan, and F. Wei. A study on the coordination of urban traffic control and traffic assignment. *Discrete Dynamics in Nature and Society*, 2012(12):367–368, 2012.
13. J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.

Una debilidad de la estrategia Uno-contra-Uno en clasificación: Potenciando las clases difíciles

Mikel Galar[1], Alberto Fernández[2], Edurne Barrenechea[1], and Francisco Herrera[3]

[1] Departamento de Automática y Computación,
Universidad Pública de Navarra, 31006, Pamplona, España
{mikel.galar,edurne.barrenechea}@unavarra.es
[2] Departamento de Ciencias de la Computación,
Universidad de Jaén, 23071, Jaén, España
alberto.fernandez@ujaen.es
[3] Departamento de Ciencias de la Computación e Inteligencia Artificial,
Universidad de Granada, 18071, Granada, España
herrera@decsai.ugr.es

Resumen La estrategia de binarización Uno-contra-Uno es uno de los métodos más utilizados a la hora de afrontar problemas con múltiples clases. En este trabajo analizamos el hecho de que en esta estrategia se favorece a las clases fáciles en perjuicio de las difíciles (aquellos conceptos del problema que son más difíciles de identificar y que por tanto, obtienen un menor ratio de acierto). Además, presentamos una posible solución para potenciar las clases difíciles mediante una generalización del voto ponderado. Esta propuesta permite obtener una clasificación más balanceada sobre todas las clases, sin acarrear una pérdida de la precisión global.

Palabras clave: multi-clase, uno-contra-uno, descomposición, clases difíciles

1. Introduction

Las estrategias de descomposición [13] permiten afrontar problemas multi-clase mediante clasificadores binarios. Entre ellas, una de las estrategias más utilizadas es la estrategia Uno-contra-Uno (*One-vs-One*, OVO), donde se divide el problema multi-clase original en tantos nuevos subproblemas como posibles pares de clases podamos obtener. Cada nuevo subproblema binario es afrontado por un clasificador independiente, cuya salida es posteriormente combinada con la del resto para obtener la clase final con la que etiquetar una nueva instancia [9].

En los problemas de clasificación, las características que definen a cada una de las clases suelen ser, en general, diferentes: distribución de ejemplos, relación entre las clases, relación entre los ejemplos de la propia clase o solapamiento entre clases. Como consecuencia, algunas de las clases pueden ser más difíciles de distinguir que otras. En este contexto, podemos definir las *clases difíciles*

como aquellas sobre las que se obtiene un ratio de acierto menor (más bajo que el obtenido por el resto de las clases).

En esta contribución nos centramos en aquellos problemas en los cuales todas las clases son igualmente importantes. La precisión global no es capaz de valorar igualmente a todas las clases, ya que los errores sobre las clases difíciles se ven contrarrestados por los aciertos sobre las clases más sencillas, incluso aunque el conjunto de datos esté balanceado. Esto es debido a que esta medida realiza una media sobre todas las instancias, sin tener en cuenta la precisión sobre cada una de las clases independientemente. Por ello, en este trabajo vamos a considerar otras medidas que nos permitan evaluar correctamente la precisión sobre todas las clases de manera global.

Nuestro objetivo en este trabajo es doble: 1) tratar de explicar el porqué de la debilidad de la estrategia OVO cuando tratamos de obtener una buena predicción sobre todas las clases; 2) introducir un nuevo modelo de agregación basado en Funciones de Equivalencia Restringida (*Restricted Equivalence Functions*, REFs) [5], que nos permita modificar modificar las fronteras de decisión de los clasificadores base y potenciar la clasificación de las clases difíciles, sin necesidad de cambiar los clasificadores base entrenados.

Los experimentos que llevamos a cabo en este trabajo incluyen veintiocho conjuntos de datos de los repositorios UCI [3] y KEEL [1]. Utilizamos, además de la precisión global, otras medidas que tienen en cuenta a las clases difíciles y contrastamos los resultados obtenidos mediante test estadísticos [6,10]. Debido a las limitaciones de espacio, utilizaremos únicamente como clasificador base las máquinas vector soporte (*Support Vector Machine*, SVM) [15].

El resto del trabajo está organizado como sigue. En la Sección 2 analizamos el problema de las clases difíciles en la estrategia OVO. Posteriormente, en la Sección 3 mostramos nuestra propuesta para potenciar las clases difíciles. El marco experimental utilizado está explicado en la Sección 4. En la Sección 5 presentamos el estudio experimental que hemos llevado a cabo y finalmente en la Sección 6 presentamos las conclusiones de este trabajo.

2. El problema de las clases difíciles en OVO

Esta sección recuerda el funcionamiento de la estrategia OVO (Subsección 2.1), y explicamos el problema de las clases difíciles (Subsección 2.2).

2.1. Descomposición Uno-contra-Uno

OVO divide un problema de m clases en $m(m-1)/2$ subproblemas binarios (todos los posibles pares de clases) que son afrontados por clasificadores base independientes. Una nueva instancia se clasifica considerando la salida de cada uno de los clasificadores. Cada clasificador que distingue un par de clases $\{C_i, C_j\}$ devuelve un grado de confianza $r_{ij} \in [0,1]$ en favor de la clase C_i ($r_{ji} = 1 - r_{ij}$). Estas salidas pueden almacenarse en lo que se denomina matriz de votos de la

siguiente forma:

$$R = \begin{pmatrix} - & r_{12} & \cdots & r_{1m} \\ r_{21} & - & \cdots & r_{2m} \\ \vdots & & & \vdots \\ r_{m1} & r_{m2} & \cdots & - \end{pmatrix} \tag{1}$$

Para establecer la clase a la que pertenece la instancia puede usarse cualquiera de las agregaciones existentes en la literatura [9] sobre la matriz de votos. La más simple es la estrategia del voto, donde cada clasificador da un voto para la clase que predice y la clase con más votos es con la que se etiqueta la instancia.

2.2. Una debilidad de la estrategia Uno-contra-Uno

La mejora en términos de precisión obtenida por la estrategia OVO frente a los clasificadores base [9] se debe en gran medida a la mejora en la precisión obtenida sobre las clases más fáciles. A continuación, tratamos de mostrar la razón por la que la estrategia OVO tiende a mejorar la precisión sobre las clases fáciles, sin mejorarla sobre las difíciles. Para ello, consideramos el escenario más simple: OVO con la estrategia del voto simple. Recordamos que el ratio de verdaderos positivos (*True Positive Rate*, TPR) de una clase es el número de ejemplos correctamente clasificados de dicha clase entre el total de ejemplos de la misma.

Enunciado del problema y notación:

- Sea un problema con m clases, $\mathbb{C} = \{C_1, \ldots, C_m\}$. Una de ellas ($C_d$) es mucho más difícil de clasificar. El resto de clases $\mathbb{C}_e = \{C_{e_1}, C_{e_2}, \ldots, C_{e_{m-1}}\}$ son igualmente distinguibles.
- Sea TPR_d el TPR sobre la clase difícil y $\mathrm{TPR}_e = \mathrm{TPR}_{e_1} = \ldots = \mathrm{TPR}_{e_{m-1}}$ el TPR sobre cada una de las clases fáciles. Por tanto, $\mathrm{TPR}_d < \mathrm{TPR}_e$.

Suposiciones:

1. Todos los TPR son iguales en todos los clasificadores base que consideran la misma clase (suposiciones similares son consideradas en [12]).
2. Independencia de los clasificadores base (se supone en OVO).
3. Una instancia se clasifica correctamente si todos los clasificadores base competentes (aquellos que consideran la clase real en el entrenamiento) clasifican correctamente la instancia.

En este marco de trabajo, una instancia $\mathbf{x}_{e_1}$ de una clase fáciles (C_{e_1}) va a ser clasificada. La probabilidad de clasificarla correctamente $P(h_{ovo}(\mathbf{x}_{e_1}) = C_{e_1})$ (donde h_{ovo} es el clasificador OVO) viene dada por el TPR de cada uno de los $m-1$ clasificadores competentes para dicha instancia. Por tanto $P(h_{ovo}(\mathbf{x}_{e_1}) = C_{e_1}) = \prod_{i=1}^{m-1} \mathrm{TPR}_{e_1}$. De manera similar, la probabilidad de clasificar correctamente una instancia $\mathbf{x}_d$ de la clase difícil (C_d) es $P(h_{ovo}(\mathbf{x}_d) = C_d) = \prod_{i=1}^{m-1} \mathrm{TPR}_d$.

Problema: Dado que estas probabilidades se calculan como el producto de los TPRs individuales de los clasificadores competentes, según aumenta el número de clases m, la probabilidad de clasificar correctamente la instancia de

la clase difícil decrece más rápidamente, mientras que la probabilidad sobre la clasificación de instancias del resto de clases no decrece tan drásticamente. Como consecuencia, la probabilidad de clasificar correctamente una instancia de la clase difícil es proporcionalmente mucho menor:

$$P(h_{ovo}(\mathbf{x}_{e_1}) = C_{e_1}) = \prod_{i=1}^{m-1} \mathrm{TPR}_{e_1} \gg \prod_{i=1}^{m-1} \mathrm{TPR}_d = P(h_{ovo}(\mathbf{x}_d) = C_d),$$

donde $\gg$ indica que la parte derecha es proporcionalmente mucho menor según aumenta m, a pesar de que las dos partes decrecen. La Tabla 1 muestra claramente este hecho, donde observamos la evolución de la probabilidades de clasificar correctamente una instancia con un TPR específico para la clase en cada clasificador base. Lógicamente, estos porcentajes no son más que cotas inferiores de la probabilidad debido a las suposiciones impuestas.

Tabla 1. Probabilidad de clasificar correctamente una instancia de una clase con un TPR específico y diferentes números de clases.

Clases	TPR 0,5	TPR 0,55	TPR 0,6	TPR 0,65	TPR 0,7	TPR 0,75	TPR 0,8	TPR 0,85	TPR 0,9	TPR 0,95	TPR 1,0
2	0,5	0,55	0,6	0,65	0,7	0,75	0,8	0,85	0,9	0,95	1,0
3	0,250	0,303	0,360	0,423	0,490	0,563	0,640	0,723	0,810	0,903	1,0
4	0,125	0,166	0,216	0,275	0,343	0,422	0,512	0,614	0,729	0,857	1,0
5	0,063	0,092	0,130	0,179	0,240	0,316	0,410	0,522	0,656	0,815	1,0
6	0,031	0,050	0,078	0,116	0,168	0,237	0,328	0,444	0,591	0,774	1,0
7	0,016	0,028	0,047	0,075	0,118	0,178	0,262	0,377	0,531	0,735	1,0
8	0,008	0,015	0,028	0,049	0,082	0,134	0,210	0,321	0,478	0,698	1,0
9	0,004	0,008	0,017	0,032	0,058	0,100	0,168	0,273	0,431	0,663	1,0
10	0,002	0,005	0,010	0,021	0,040	0,075	0,134	0,232	0,387	0,630	1,0
11	0,001	0,003	0,006	0,014	0,028	0,056	0,107	0,197	0,349	0,599	1,0
12	0,0005	0,001	0,004	0,009	0,020	0,042	0,086	0,167	0,314	0,569	1,0

¿Cómo puede aliviarse este problema?

1. Mejorando el TPR_d en cada uno de los clasificadores base.
2. Usando una agregación que tenga en cuenta el problema de las clases difíciles sin necesidad de alterar los clasificadores base que estan debajo.

El primer caso es la solución directa al problema, pero hay que tener en cuenta que, incluso aumentando el TPR_d, mientras los TPRs sobre todas las clases sean diferentes, la dificultad se mantendrá. Además, dicha mejora no es fácil de obtener. Por estas razones, nos centramos en el segundo caso, que de cualquier modo, podría combinarse con el primero.

Desde nuestro punto de vista, las matrices de votos contienen información suficiente como para obtener resultados significativamente diferentes sobre las clases difíciles cambiando únicamente la agregación. Por esta razón, las matrices de votos utilizadas en los experimentos serán las mismas para todas las agregaciones. Por tanto, todas las diferencias mostradas en el apartado experimental serán debido a las agregaciones y no a los clasificadores base, lo cual es de gran importancia para poder evaluar debidamente el rendimiento de la metodología que proponemos.

3. Una agregación basada en REFs

En esta sección introducimos el nuevo modelo de agregación para OVO. Primero recordamos varios conceptos preliminares en la Subsección 3.1, que son

necesarios para poder presentar posteriormente el nuevo modelo en la Subsección 3.2.

3.1. Funciones de Equivalencia Restringidas

Para poder introducir el nuevo modelo de agregación necesitamos recordar varios conceptos provenientes de la teoría de lógica difusa. En la teoría difusa, una negación modela el concepto de complementario u opuesto:

Definición 1 *Una función* $n : [0,1] \to [0,1]$ *con* $n(0) = 1$, $n(1) = 0$, *estrictamente decreciente y continua es una negación estricta. Además, si* n *es involutiva, es decir, si* $n(n(x)) = x$ *para todo* $x \in [0,1]$, *entonces* n *es una negación fuerte.*

Las REFs [5] permiten medir lo próximos que son dos valores puntuales.

Definición 2 *[5] Una función* $\mathrm{REF} : [0,1]^2 \to [0,1]$ *se dice función de equivalencia restringida asociada a una negación fuerte* n *si satisface las siguientes condiciones*

1. $\mathrm{REF}(x,y) = \mathrm{REF}(y,x)$ *para todo* $x, y \in [0,1]$;
2. $\mathrm{REF}(x,y) = 1$ *si y solo si* $x = y$;
3. $\mathrm{REF}(x,y) = 0$ *si y solo si* $x = 1$ *e* $y = 0$ *o* $x = 0$ *e* $y = 1$;
4. $\mathrm{REF}(x,y) = \mathrm{REF}(n(x), n(y))$ *para todo* $x, y \in [0,1]$;
5. *Para todo* $x, y, z \in [0,1]$, *si* $x \leq y \leq z$, *entonces* $\mathrm{REF}(x,y) \geq \mathrm{REF}(x,z)$ *y* $\mathrm{REF}(y,z) \geq \mathrm{REF}(x,z)$.

En este trabajo, el interés de esta medida de proximidad reside en la posibilidad de parametrizarla por medio de automorfismos de la siguiente forma.

Definición 3 *Una función* $\varphi : [a,b] \to [a,b]$ *continua y estrictamente creciente tal que* $\varphi(a) = a$ *y* $\varphi(b) = b$ *se denomina automorfismo del intervalo* $[a,b] \subset \mathbb{R}$.

Proposición 1 *[5] Sean* φ_1, φ_2 *dos automorfismos del intervalo* $[0,1]$. *Entonces* $\mathrm{REF}(x,y) = \varphi_1^{-1}(1 - |\varphi_2(x) - \varphi_2(y)|)$ *es una función de equivalencia restringida asociada a la negación fuerte* $n(x) = \varphi_2^{-1}(1 - \varphi_2(x))$.

Una forma sencilla de parametrizar los automorfismos es mediante un parámetro: sea $\varphi(x) = x^\lambda$, y por tanto, $\varphi^{-1}(x) = x^{1/\lambda}$, con $\lambda \in (0, \infty)$.

3.2. Una generalización del método del voto ponderado

La agregación que proponemos en este trabajo es una generalización del método del voto ponderado (*Weighted Voting*, WV), cuya robustez ha sido tanto teórica como empíricamente probada [11]: $\mathrm{Clase} = \arg \max_{i=1,\ldots,m} \sum_{1 \leq j \neq i \leq m} r_{ij}$.

En nuestro modelo de agregación, en vez de sumar directamente las confianzas dadas por los clasificadores, primero comparamos estas confianzas con el voto seguro (es decir, 1), ya que es el caso en el que debemos dar el voto

más alto. Por tanto, cuanto más similar sea r_{ij} al 1, más importancia tendrá el voto, tal y como ocurría hasta ahora. Sin embargo, nuestra intención es que la importancia del voto dado por cada clasificador sea diferente. Por ello, comparamos ambos valores mediante una REF, $\text{REF}(r_{ij}, 1)$, que nos proporciona una medida de cómo de cerca está r_{ij} del voto seguro. Partiendo de la Proposición 1 y considerando $\varphi_1 = x^{\lambda_1}$, $\varphi_2 = x^{\lambda_2}$, las operaciones y parámetros necesarios para llevar a cabo la comparación pueden reducirse:

$$\text{REF}(x, 1) = (1 - |x^{\lambda_2} - 1^{\lambda_2}|)^{1/\lambda_1} = (x)^{\lambda_2/\lambda_1} = x^{\lambda} \tag{2}$$

La forma en la que podemos alterar la importancia de los votos puede verse en la Figura 1. Esta figura muestra la influencia de λ en la aplicación de la REF en la comparativa frente al voto seguro. Es interesante observar que $\lambda = 1$ no modifica el voto del clasificador, ya que $\text{REF}(r_{ij}, 1) = r_{ij}$, mientras que valores de $\lambda < 1$ potencian el voto ($\text{REF}(r_{ij}, 1) > r_{ij}$) y ocurre lo contrario con $\lambda > 1$ ($\text{REF}(r_{ij}, 1) < r_{ij}$).

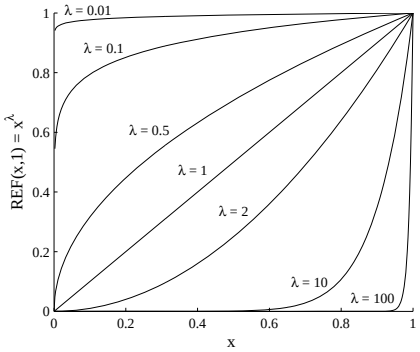


Figura 1. Influence of the parameter λ in $\text{REF}(x, 1)$.

Nota *Por tanto, atendiendo a la Figura 1, podemos observar que las confianzas en favor de las clases difíciles deberían utilizar una REF con un valor de λ pequeño, mientras que aquellas correspondientes a clases más fáciles deberían considerar valores mayores de λ para poder encontrar un buen balance entre sus predicciones.*

En definitiva, tras ver cómo podemos modificar las confianzas dadas por los clasificadores base, de manera similar al WV, agregamos los votos en cada fila:

$$\text{Clase} = \arg \max_{i=1,\ldots,m} \sum_{1 \le j \ne i \le m} \text{REF}_{ij}(r_{ij}, 1) = \arg \max_{i=1,\ldots,m} \sum_{1 \le j \ne i \le m} (r_{ij})^{\lambda_{ij}} \tag{3}$$

donde λ_{ij} es el parámetro que corresponde a $\text{REF}_{ij}(r_{ij}, 1)$. Hacer notar que el caso del WV se recupera cuando $\lambda_{ij} = 1$ para todo $i, j = 1, \ldots, m$ e $i \ne j$.

3.3. Ajustando la agregación para potenciar las clases difíciles

El ajuste de los parámetros es necesario para poder adaptar la agregación a cada clase en cada problema. En este trabajo consideramos el uso del algoritmo

evolutivo de codificación real CHC [7] para optimizar la función objetivo propuesta. A continuación presentamos los dos factores clave de nuestra propuesta como son la propia función objetivo y la codificación de los parámetros.

Función objetivo. Nuestro objetivo debe ser diferente al de la precisión global, ya que esta no tiene en cuenta las clases difíciles. Por ello, proponemos la siguiente función para estimar la calidad de los parámetros $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_{m(m-1)})$ (tenemos un parámetro por cada posición de la matriz de votos):

$$\text{Fitness}(\boldsymbol{\lambda}) = \text{Margen}(\boldsymbol{\lambda}) + 0{,}5 \cdot \text{GM}(\boldsymbol{\lambda}) + 0{,}5 \cdot \text{AvgAcc}(\boldsymbol{\lambda}), \tag{4}$$

donde GM es la media geométrica [4] de las precisiones por clase, AvgAcc es la media aritmética [8] de las precisiones por clase y Margen cuantifica la calidad de la separación de la clases y se calcula como sigue.

$$\text{Margen} = \arg \min_{c=1,\ldots,n_c} \frac{V_i^c - V_j^c}{n_{Tr} \cdot m} \tag{5}$$

donde C_i es la clase predicha y C_j es la segunda clase con más confianza (Eq. (3)); V_i^c, V_j^c son los valores obtenidos en Eq. (3) para cada clase (por la instancia c), respectivamente. n_c es el número de instancias clasificadas correctamente, ya que solo estas son utilizadas. El margen es normalizado para reducir su influencia en la función objetivo. Tomamos el menor de los márgenes ya que es el que mejor representa la separación entre las clases más conflictivas.

Representación de parámetros. Para codificar el conjunto de parámetros reales ($\boldsymbol{\lambda}$) debemos de trasladar el rango de λ que va de 0 a ∞ a unidades codificables en el cromosoma. Utilizamos la siguiente codificación que nos permite buscar en todo el espacio homogéneamente (Figura 1):

$$\boldsymbol{\Phi}(\boldsymbol{\lambda}) = (\phi(\lambda_1), \phi(\lambda_2), \ldots, \phi(\lambda_{m(m-1)})) = (c_{\lambda_1}, c_{\lambda_2}, \ldots, c_{\lambda_{m(m-1)}}) \tag{6}$$

donde cada gen $c_{\lambda_i} \in (0, 1)$, $i \in \{1, \ldots, m(m-1)\}$ y el valor del parámetro se recupera como sigue

$$\lambda_i = \phi^{-1}(c_{\lambda_i}) = \begin{cases} (2 \cdot c_{\lambda_i})^2 & \text{si } c_{\lambda_i} \leq 0{,}5 \\ \dfrac{1}{(2 \cdot (c_{\lambda_i} - 0{,}5))^2} & \text{en otro caso.} \end{cases} \tag{7}$$

4. Marco Experimental

4.1. Clasificadores base y parámetros

Utilizamos SVMs [15] como clasificador base para estudiar la validez de la propuesta. Las confianzas las obtenemos a partir de la estimación de probabilidades del modelo logístico SVM [14]. La Tabla 2 describe los parámetros utilizados en los experimentos que hemos llevado a cabo con la herramienta KEEL [2]. Utilizamos dos configuraciones diferentes con diferentes valores de C y de kernel para estudiar la robustez de la propuesta. Como agregación base para llevar a cabo la comparativa consideramos el método de estimación de probabilidades de Wu *et al.* [17] (PE) cuya precisión ha sido probada [9].

Tabla 2. Especificación de parámetros.

Algoritmo	*Parámetros*
SVM_{Poly}	C = 1.0, Tolerancia = 0.001, Epsilon = 1.0E-12 Kernel = Polinomial, Grado del polinomio = 1
SVM_{Puk}	C = 100.0, Tolerancia = 0.001, Epsilon = 1.0E-12 Kernel = Puk, PukKernel ω = 1.0, PukKernel σ = 1.0
CHC	Tamaño población = 50, Evaluaciones = $1000 \cdot m^2$ Bist por gen (codificación Gray) = 30 Reinicios sin mejora = 3

4.2. Conjuntos de datos y evaluación

Consideramos veintiocho conjuntos de datos de los repositorios UCI [3] y KEEL [1], mostrados en la tabla Tabla 3. Para llevar a cabo la evaluación de los resultados utilizamos la precisión global, la GM y la AvgAcc. En el caso de las clases difíciles, la GM es la que mejor va a reflejar este problema y por tanto la más válida a la hora de evaluarlo. Utilizamos un modelo de validación cruzada de 5 particiones para obtener los resultados. Para llevar a cabo una evaluación

Tabla 3. Descripción de los conjuntos de datos.

C. datos	#Ej.	#Atr.	#Num.	#Nom.	#Cl.	C. datos	#Ej.	#Atr.	#Num.	#Nom.	#Cl.
Balance	625	4	4	0	3	Page-blocks	548	10	10	0	5
Contraceptive	1473	9	9	0	3	Shuttle	2175	9	9	0	5
Hayes-roth	132	4	4	0	3	Autos	159	25	15	10	6
Iris	150	4	4	0	3	Dermatology	358	34	1	33	6
NewThyroid	215	5	5	0	3	Flare	1066	11	0	11	6
Splice	319	60	0	60	3	Glass	214	9	9	0	7
Tae	151	5	5	0	3	Satimage	643	36	36	0	7
Thyroid	720	21	21	0	3	Segment	2310	19	19	0	7
Wine	178	13	13	0	3	Zoo	101	16	0	16	7
Car	1728	6	0	6	4	Ecoli	336	7	7	0	8
Lymphography	148	18	3	15	4	Led7digit	500	7	0	7	10
Vehicle	846	18	18	0	4	Penbased	1100	16	16	0	10
Cleveland	297	13	13	0	5	Yeast	1484	8	8	0	10
Nursery	1296	8	0	8	5	Vowel	990	13	13	0	11

honesta de los resultados, utilizamos test estadísticos no paramétricos tal y como se recomienda en [6,10]. En este caso, dado que realizamos comparativas por pares, utilizaremos el test de Wilcoxon [16].

5. Estudio experimental

En esta sección mostramos la validez de nuestra propuesta basada en REFs (denotada como RA) para potenciar las clases difíciles. Para ello, consideramos dos configuraciones diferentes de las SVM. Los resultados obtenidos con SVM_{Poly} como clasificador base pueden verse en la parte izquierda de la Tabla 4, mientras que los correspondientes a SVM_{Puk} pueden verse en la parte derecha. Los resultados de precisión y AvgAcc aparecen como porcentajes tal y como es habitual. Podemos observar en las tablas como la precisión global se mantiene, mientras que la GM y la AvgAcc han sido aumentadas mediante RA, siendo la mejora en GM notable. En cualquier caso, para obtener conclusiones significativas estos hechos deben ser contrastados con el estudio estadístico adecuado mediante el test de Wilcoxon, cuyos resultados se muestran en la Tabla 5.

Atendiendo a los resultados de los test, las conclusiones obtenidas son similares. Ambos métodos (PE y RA) obtienen precisiones globales equivalentes, pero

Tabla 4. Resultados de precisión, GM y AvgAcc para OVO con PE y RA usando las dos configuraciones de SVM.

	SVM_{Poly}						SVM_{Puk}					
	Accuracy		GM		AvgAcc		Accuracy		GM		AvgAcc	
C. datos	PE	RA	PE	RA	PE	RA	PE	RA	PE	RA	PE	RA
Autos	**74.80**	74,17	**.5479**	,5463	**72.69**	71,73	**68.53**	64,76	**.2544**	,2279	**65.06**	60,19
Balance	90,40	**91.52**	,8310	**.9072**	85,35	**91.12**	88,00	**88.96**	**.8660**	,8514	**86.93**	85,89
Car	92,71	**93.58**	,8651	**.9390**	87,18	**93.97**	63,60	**67.01**	,7452	**.7606**	77,58	**78.88**
Cleveland	**58.25**	46,78	,0000	**.0776**	**30.88**	29,94	45,09	**45.76**	**.0000**	**.0000**	29,78	**30.27**
Contraceptive	49,83	**51.93**	,4604	**.5280**	47,34	**53.19**	**48.41**	45,82	,4406	**.4667**	45,70	**47.29**
Dermatology	**94.13**	**94.13**	**.9408**	**.9408**	**94.58**	**94.58**	**96.09**	94,14	**.9574**	,9238	**96.03**	93,15
Ecoli	**77.69**	74,71	**.1544**	,1418	**68.18**	66,52	**75.31**	**75.31**	,1381	**.1550**	67,35	**68.34**
Flare	**74.67**	73,45	,4517	**.6008**	61,02	**66.10**	**69.42**	65,39	,3277	**.5177**	59,43	**60.42**
Glass	61,26	**61.68**	,2045	**.4834**	55,40	**65.07**	70,60	**72.95**	,5372	**.5598**	68,04	**69.81**
Hayes-Roth	52,22	**71.94**	,4985	**.7015**	55,05	**72.46**	79,54	**81.82**	,8072	**.8277**	82,30	**84.24**
Iris	**96.00**	**96.00**	,9580	**.9583**	**96.00**	**96.00**	94,00	**94.67**	,9375	**.9442**	94,00	**94.67**
Led7digit	**73.00**	**73.00**	,7110	**.7145**	73,01	**73.07**	70,20	**71.00**	,6840	**.6959**	70,32	**71.22**
Lymphography	81,68	**83.06**	,3348	**.3449**	64,87	**66.54**	**80.34**	**80.34**	**.1557**	**.1557**	**54.98**	**54.98**
NewThyroid	**97.21**	96,74	,9599	**.9667**	96,16	**96.83**	**97.67**	**97.67**	**.9811**	**.9811**	**98.16**	**98.16**
Nursery	**91.90**	91,74	,6529	**.7237**	82,22	**87.84**	**81.33**	80,86	**.6793**	,6679	**82.28**	81,82
Pageblocks	**94.70**	87,40	,3042	**.6602**	68,23	**80.39**	94,16	**94.34**	,2757	**.2837**	67,40	**68.66**
Penbased	95,27	**95.36**	,9513	**.9525**	95,29	**95.40**	**97.82**	**97.82**	**.9781**	**.9781**	**97.85**	**97.85**
Satimage	**84.14**	82,89	,7703	**.7874**	79,55	**80.14**	84,92	**85.39**	,8315	**.8368**	84,16	**84.69**
Segment	92,55	**94.63**	,9197	**.9444**	92,55	**94.63**	97,10	**97.14**	,9704	**.9708**	97,10	**97.14**
Shuttle	**96.37**	95,36	**.3477**	,3439	80,67	**80.87**	**99.72**	98,62	**.7650**	,7615	**93.14**	92,76
Splice	79,59	**80.22**	,8325	**.8374**	84,29	**84.69**	**64.56**	58,02	,3787	**.5586**	51,44	**66.42**
Tae	51,72	**54.41**	,4869	**.5220**	51,91	**54.24**	56,30	**58.28**	,5513	**.5707**	56,24	**58.18**
Thyroid	95,69	**97.22**	,4445	**.8816**	67,88	**89.39**	**92.64**	92,50	,4971	**.5559**	62,44	**68.47**
Vehicle	72,46	**73.40**	**.6970**	,6956	72,82	**73.84**	80,49	**80.61**	,7873	**.7887**	80,71	**80.83**
Vowel	69,90	**71.11**	,6822	**.6921**	69,90	**71.11**	**99.39**	**99.39**	**.9936**	**.9936**	**99.39**	**99.39**
Wine	**97.16**	**97.16**	**.9684**	**.9684**	**96.99**	**96.99**	**98.30**	**98.30**	**.9857**	**.9857**	**98.60**	**98.60**
Yeast	**59.10**	54,65	**.0000**	**.0000**	**56.74**	55,46	**56.54**	55,12	,0000	**.0905**	55,37	**55.73**
Zoo	95,05	**97.00**	,0000	**.4000**	85,24	**91.43**	84,19	**90.10**	**.0000**	**.0000**	64,05	**72.14**
Mean	80,34	**80.54**	,5706	**.6521**	74,00	**77.63**	**79.80**	79,72	,5902	**.6111**	74,49	**75.72**

Tabla 5. Wilcoxon test para comparar RA y PE. R^+ corresponde a la suma de rangos para RA y R^- para PE.

	SVM_{Poly}				SVM_{Puk}			
Medida	R^+	R^-	Hipótesis ($\alpha = 0{,}05$)	p-valor	R^+	R^-	Hipótesis ($\alpha = 0{,}05$)	p-value
Precisión	230,0	176,0	No rechazada	0,511089	219,5	186,5	No rechazada	0,883838
GM	367,0	39,0	Rechazada para RA	0,000194	305,0	101,0	Rechazada para RA	0,027306
AvgAcc	353,0	53,0	Rechazada para RA	0,000808	313,5	92,5	Rechazada para RA	0,017668

RA obtiene resultados significativamente mejores en términos de GM y AvgAcc con p-valores muy bajos para ambas configuraciones de SVM. Por tanto, además de mejorar las precisiones sobre las clases difíciles podemos decir que estamos ante un método robusto ante las diferentes configuraciones utilizadas. Únicamente destacar que la mejora obtenida es mayor en SVM_{Poly} debido a que las confianzas dadas por esta configuración son mejores (están mejor repartidas en el rango por la configuración del parámetro C) y por tanto dan más información sobre el proceso de clasificación.

6. Conclusiones

En este trabajo hemos puesto de manifiesto el problema de las clases difíciles en la estrategia OVO. Para tratar de mejorar la clasificación sobre estas clases hemos propuesto una nueva metodología de agregación que generaliza al voto ponderado.

Esta metodología permite aprender los parámetros de las REFs para diferenciar mejor las clases difíciles, lo que ha quedado empíricamente probado con las diferencias estadísticas encontradas en términos de GM.

Agradecimientos Este trabajo ha sido parcialmente subvencionado por el Ministerio de Educación y Ciencia bajo los proyectos TIN2010-15055 y TIN2011-28488 y el plan Andaluz de investigación P10-TIC-6858.

Referencias

1. Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F.: KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. J. Mult.-Valued Logic Soft Comput. 17, 255 – 287 (2011)
2. Alcalá-Fdez, J., Sánchez, L., García, S., del Jesus, M.J., Ventura, S., Garrell, J.M., Otero, J., Romero, C., Bacardit, J., Rivas, V.M., Fernández, J., Herrera, F.: KEEL: a software tool to assess evolutionary algorithms for data mining problems. Soft Comput. 13(3), 307–318 (2009)
3. Asuncion, A., Newman, D.J.: UCI machine learning repository (2007), `http://www.ics.uci.edu/~mlearn/MLRepository.html`
4. Barandela, R., Sánchez, J.S., García, V., Rangel, E.: Strategies for learning in class imbalance problems. Pattern Recogn. 36(3), 849–851 (2003)
5. Bustince, H., Barrenechea, E., Pagola, M.: Restricted equivalence functions. Fuzzy Sets Syst. 157(17), 2333–2346 (2006)
6. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. 7, 1–30 (2006)
7. Eshelman, L.J., Schaffer, J.D.: Real-coded genetic algorithms and interval-schemata. In: Whitley, D.L. (ed.) Foundation of Genetic Algorithms 2. pp. 187–202. Morgan Kaufmann, San Mateo, CA (1993)
8. Ferri, C., Hernández-Orallo, J., Modroiu, R.: An experimental comparison of performance measures for classification. Pattern Recogn. Lett. 30(1), 27–38 (2009)
9. Galar, M., Fernández, A., Barrenechea, E., Bustince, H., Herrera, F.: An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes. Pattern Recogn. 44(8), 1761 – 1776 (2011)
10. García, S., Herrera, F.: An extension on "statistical comparisons of classifiers over multiple data sets"for all pairwise comparisons. J. Mach. Learn. Res. 9, 2677–2694 (2008)
11. Hüllermeier, E., Vanderlooy, S.: Combining predictions in pairwise classification: An optimal adaptive voting strategy and its relation to weighted voting. Pattern Recogn. 43(1), 128–142 (2010)
12. Kuncheva, L., Whitaker, C., Shipp, C., Duin, R.: Limits on the majority vote accuracy in classifier fusion. Pattern Anal. App. 6, 22–31 (2003)
13. Lorena, A.C., Carvalho, A.C., Gama, J.M.: A review on the combination of binary classifiers in multiclass problems. Artif. Intell. Rev. 30(1-4), 19–37 (2008)
14. Platt, J.C.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: Advances in Large Margin Classifiers. pp. 61–74. MIT Press (1999)
15. Vapnik, V.: Statistical Learning Theory. New York: Wiley (1998)
16. Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics Bull. 1(6), 80–83 (1945)
17. Wu, T.F., Lin, C.J., Weng, R.C.: Probability estimates for multi-class classification by pairwise coupling. J. Mach. Learn. Res. 5, 975–1005 (2004)

Asignación de órganos en trasplante hepático mediante regresión ordinal

C. Hervás-Martínez[1], M. Pérez-Ortiz[1], M. Cruz-Ramírez[1] y P.A. Gutiérrez[1] *

Universidad de Córdoba, Dept. de Informática y Análisis Numérico
Campus de Rabanales, Edificio Albert Einstein, 14071 - Córdoba, Spain

Resumen En este trabajo se presenta una metodología novedosa de asignación donante-receptor en el contexto de trasplante hepático, utilizando para ello técnicas de reconocimiento de patrones. Más concretamente, se construye un modelo para predecir la supervivencia del injerto a distintos niveles temporales a partir de la información contenida en una base de datos de pares de donantes-receptores pertenecientes a siete hospitales españoles y uno inglés. La metodología final de asignación aúna el mejor modelo predictivo obtenido junto con una de la metodologías más habituales en trasplante hepático (conocida como *Model for End-stage Liver Disease score* o MELD). El modelo de clasificación desarrollado hace uso de un paradigma de aprendizaje conocido como regresión ordinal (ya que las clases del problema presentan un orden natural entre ellas) desarrollando para ello un método de descomposición binaria en cascada. Los resultados obtenidos muestran que dicha metodología obtiene resultados muy prometedores para resolver este difícil problema de asignación donante-receptor y que la unión de dicho modelo con el método MELD conduce a un compromiso entre el principio de supervivencia (proporcionado por el modelo predictivo) y la gravedad del paciente (proporcionada por el MELD).

1. Introducción

Durante las dos últimas décadas, nuevas tendencias en biomedicina han considerado la utilización de técnicas de aprendizaje de máquinas, como son entre otros, las redes neuronales, las máquinas de soporte vectorial, las redes bayesianas o los árboles de decisión, como una necesidad para mejorar los resultados que las técnicas estadísticas clásicas proporcionaban. El trasplante de órganos es hoy día una técnica ampliamente utilizada como tratamiento para pacientes que presentan problemas hepáticos graves. Sin embargo, dichos trasplantes están seriamente limitados por la disponibilidad de suficientes y apropiados órganos, de tal forma que existe un grave desequilibrio entre la oferta y la demanda, lo que produce por desgracia como resultado la muerte de pacientes en lista de espera. En España, a pesar del aumento de donantes en los últimos años (con una tasa

* Este trabajo ha sido subvencionado en parte por el proyecto TIN2011-22794 del Ministerio Español, Comisión Ciencia y Tecnología (MICYT), fondos FEDER y el proyecto P2011-TIC-7508 de la "Junta de Andalucía" (España).

que alcanza el 34,4 pmp en 2009, según datos del Registro Español de Trasplante Hepático RETH [1]), se ha asistido también de forma paralela al crecimiento del número de pacientes que precisan trasplante hepático, sobrepasando así los 2000 candidatos en los últimos años. Y, aún cuando el número de trasplantes realizados ha crecido, es inevitable el consiguiente incremento de los pacientes en lista de espera y su mortalidad en la misma [2]. Debido a dicho desequilibrio entre oferta y demanda, se han desarrollado distintos métodos para optimizar la asignación, priorizando a aquellos pacientes que podrían ser más adecuados para recibir un determinado órgano. Para el caso específico de trasplante hepático, se han propuesto distintas metodologías como el *Donor Risk Index* (DRI) [3], definido para establecer un indicador cuantitativo asociado al riesgo de trasplante pero usando para ello sólo características del donante. Otra metodología de asignación ampliamente usada es el MELD [4], basado en el principio del "más enfermo primero", pero que sólo considera la información del receptor. En este sentido, estos métodos pueden considerarse como insuficientes para el problema tratado, dado que sólo consideran características de donantes o de receptores, pero no de forma conjunta, cuando en teoría, a la hora de tratar de predecir el éxito de un trasplante sería lógico considerar características asociados a ambos y al trasplante a realizar en sí. De esta forma, recientemente se han desarrollado nuevos sistemas de asignación basados en puntuaciones (Rana et al [5], P. Dutkowski et al. [6], M. Cruz et al. [7] y M. Cruz et al. [8]) donde se tienen en cuenta características provenientes de estas tres fuentes de información.

La Figura 1 representa gráficamente el proceso de asignación de órganos.

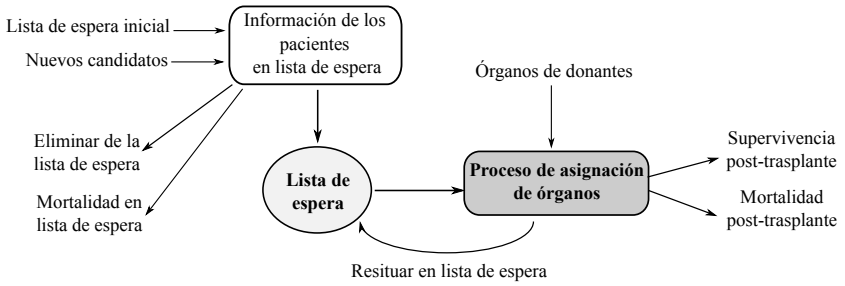


Figura 1. Representación del proceso de trasplante de órganos.

En numerosos lugares de España y del mundo, la asignación del órgano se realiza teniendo en cuenta el riesgo de muerte del receptor utilizando como criterio de asignación la puntuación MELD. Sin embargo, esta política podría ser considerada como polémica o discutible, ya que no permite la asignación a aquel receptor que presente una mayor probabilidad de supervivencia siguiendo los criterios de beneficio y justicia. En este sentido, este trabajo propone una metodología de asignación de órganos basado en características de donantes, receptores y del trasplante en sí capaz de suplir las deficiencias del MELD. Para ello, se

han obtenido datos pertenecientes a siete hospitales españoles y al hospital inglés King's College, con la idea de construir un sistema de asignación supranacional para predecir la supervivencia del órgano transplantado mediante técnicas de aprendizaje de máquinas. Más concretamente, se hará uso de un paradigma conocido como regresión ordinal, ya que las clases del problema tienen en cuenta el tiempo de supervivencia del injerto, presentando entonces un orden natural entre ellas. Estas clases son: 1) fallo del injerto antes de los 15 días de haberse realizado el trasplante; 2) el fallo ha ocurrido entre 15 días y 3 meses; 3) el fallo ha ocurrido entre 3 meses y un año; 4) no se ha presentado fallo del injerto una vez que ha transcurrido un año del trasplante.

La primera cuestión a tener en cuenta en el paradigma de regresión ordinal se basa en la hipótesis de que estos métodos serán capaces de aprovechar la información adicional que se nos proporciona del problema (la estructura ordinal del espacio de salida), y que ello nos servirá para obtener un modelo más robusto y con mejores tasas de acierto que las obtenidas mediante un clasificador nominal. Sin embargo, existe una segunda cuestión, no menos importante: la necesidad de diseñar y utilizar nuevas medidas de evaluación de un clasificador ordinal, dado que en este caso, la función de pérdida cero-uno normalmente usada en clasificación nominal no es capaz de reflejar la bondad de un clasificador ordinal, ya que ésta no distingue entre los distintos posibles errores de clasificación, como podría ser clasificar un patrón de la clase "fallo del injerto antes de los 15 días" en la clase "el fallo ha ocurrido entre 15 días y 3 meses" (un error a tener en cuenta aunque no grave debido a que las clases se presentan contiguas en la escala ordinal) o en la clase "no se ha presentado fallo del injerto" (un error que debería ser claramente más penalizado).

En este trabajo, debido a la complejidad del problema de clasificación que presenta una naturaleza ordinal y desbalanceada (la distribución de patrones por clase es de 76, 76, 62 y 1223) se hace uso de un método ordinal basado en descomposiciones binarias denominado *OneVsPrevious* que simplificará y balanceará en la medida de lo posible el problema de clasificación. Además, se hará uso de las máquinas de soporte vectorial (SVM), junto con dos métodos de combinación de las salidas binarias. Esta metodología muestra resultados muy competitivos para este problema especifico con respecto a otros clasificadores tanto nominales como ordinales, de forma que será usada para determinar un sistema de apoyo a la decisión a los profesionales médicos acerca del mejor receptor a ser trasplantado, el cuál además tendrá en cuenta la valoración del sistema MELD.

El resto del trabajo se organiza como sigue: en la Sección 2 se presenta el método de clasificación ordinal utilizado, en la Sección 3 se explica el proceso de obtención de datos y los resultados obtenidos. Finalmente, en la Sección 4 se concluye el trabajo aportando posibles líneas de ampliación del mismo.

2. Metodología

La meta de un clasificador ordinal es asignar un patrón representado por un vector de variables independientes $\mathbf{x}$ a una de las clases $\mathcal{C}_q$ donde $q \in \{1, \ldots, Q\}$,

y $\mathcal{X} \in \mathbb{R}^d$, siendo d la dimensión del espacio de las variables independientes. Para ello definimos una variable dependiente asociada a la etiqueta de clase, que en este caso es una variable ordinal $\mathcal{Y} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_Q\}$, donde las etiquetas de clase mantienen un orden de la forma $\mathcal{C}_1 \prec \mathcal{C}_2 \prec \cdots \prec \mathcal{C}_Q$, y el operador $\prec$ indica el orden de las etiquetas. Si consideramos que tenemos una m.a.s de patrones, en este caso, variables independientes del donante, del trasplante y del receptor y la variable de clasificación, en la forma $D = \{\mathbf{x}_i, y_i\}_{i=1}^N \in \mathcal{X} \times \mathcal{Y}$ siendo N el tamaño de la muestra de entrenamiento; el objetivo sería determinar una función $f : \mathcal{X} \to \mathcal{Y}$.

2.1. Máquinas de soporte vectorial

El paradigma de máquinas de soporte vectorial o SVM [9,10] es sin duda el método de aprendizaje basado en el truco del *kernel* más utilizado en reconocimiento de patrones dada su buena capacidad de generalización y su determinismo. Su idea básica consiste en la separación de las clases mediante un hiperplano especificado mediante una proyección $\mathbf{w}$ y un umbral b. El hiperplano de separación óptima se define como aquel que maximiza la distancia (el margen) entre el hiperplano y los patrones (o vectores soporte) más cercanos de ambas clases.

Además de la aplicación del truco del *kernel* para permitir funciones discriminantes no lineales en SVM, se hizo otra generalización introduciendo lo que se conoce como variables *slack* ξ_i para evitar la no separabilidad de los datos, relajar las restricciones propias de SVM y manejar datos con ruido. El objetivo final es encontrar un clasificador $f : \mathbb{R}^d \to \mathbb{R}$, de la forma $f(\mathbf{x}) = \mathbf{w} \cdot \Phi(\mathbf{x}) + b$ (siendo Φ la función de mapeo inducida por el *kernel*) que minimice la función:

$$\tfrac{1}{2}||\mathbf{w}||^2 + C \sum_{i=1}^N \xi_i, \tag{1}$$

para un determinado parámetro C y sujeto a las restricciones:

$$y_i((\mathbf{w} \cdot \Phi(\mathbf{x}_i)) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall_i \in \{1, \ldots, N\}.$$

2.2. Metodología propuesta: descomposición binaria en cascada de un problema ordinal

En este trabajo se hará uso de una metodología de descomposición binaria en cascada [11]. Este procedimiento considera $Q-1$ modelos $\mathcal{D}$ (en nuestro caso de tipo SVM, donde cada modelo D_q está determinado por una proyección $\mathbf{w}_q$ y un umbral b_q), de tal forma que el modelo D_q será entrenado para separar las clases $\mathcal{C}_1 \vee \ldots \vee \mathcal{C}_{Q-q-1}$ de la clase $\mathcal{C}_{Q-q}$, no considerando entonces todas las clases en cada submodelo de clasificación (con el objetivo de balancear en la medida de lo posible los conjuntos de datos a usar en cada submodelo).

El conjunto de entrenamiento para el clasificador $D_q = \{\mathbf{w}_q, b_q\}$ vendrá entonces determinado por $\{\mathbf{X}_{(i|i<k)}, \mathbf{X}_{(i|i=k)}\}$. Así, se podría considerar una matriz $\mathbf{M}_{(Q-1\times Q)}$ asociada a las $Q-1$ descomposiciones binarias realizadas como sigue:

$$\mathbf{M} = \begin{pmatrix} -1 & -1 & -1 & +1 \\ -1 & -1 & +1 & 0 \\ -1 & +1 & 0 & 0 \end{pmatrix},$$

donde la etiqueta -1 se asigna a patrones correspondientes a la clase negativa, la etiqueta 1 a patrones pertenecientes a la positiva, y finalmente, los patrones con la etiqueta 0 se excluyen para el proceso de formación del decisor. De esta forma, una vez que hemos entrenado los $Q-1$ modelos, se obtienen para cada patrón $\mathbf{x}$ un conjunto de $Q-1$ valores de decisión $f(\mathbf{x})$, a partir de las cuales habremos de obtener su etiqueta final. Para ello, se han considerado dos alternativas:

- Aproximación jerárquica: Este método fue propuesto originalmente para el modelo de descomposición en cascada [11]. La fase de predicción comienza evaluando el primer submodelo y continuando hasta que uno de los submodelos prediga una clase que no haya sido agrupada con otras clases.
- Aproximación basado en la metodología *Error Correcting Output Codes* o ECOC: Aunque el enfoque jerárquico puede funcionar bien en un gran número de problemas, la predicción depende de las proyecciones evaluadas en la primera etapa, y por lo tanto estará sesgada hacia esas clases. Debido a esto, en este trabajo se considerará otra propuesta para la fase de predicción, propuesta que consistirá en asociar cada clase $\mathcal{C}_q \in \mathcal{Y}$ con una columna de la matriz de codificación $\mathbf{M} \in \{-1,+1\}^{l\times Q}$ previamente definida para un determinado valor de l, (en nuestro caso l será preasignado a $Q-1$, esto es, el número de descomposiciones binarias a realizar). El algoritmo binario se ejecutará una vez para cada fila de la matriz, obteniendo así tantas salidas f como modelos. La predicción se hará entonces usando las salidas f obtenidas, eligiendo la columna de M más cercana al conjunto de valores de decisión $\mathbf{f}(\cdot) = f_1(\cdot),\ldots,f_l(\cdot)$. En este caso, debido a la elección de la norma L1 para la penalización del error en SVM, se ha elegido la función de pérdida de tipo bisagra (también conocida como *hinge-loss*) para el método ECOC, ya que es la misma usada en el problema de optimización de SVM. Para clarificar esto, reformulamos el término de error de (1) en la forma:

$$\underbrace{\frac{1}{2}||\mathbf{w}||^2}_{\texttt{función a minimizar}} + \underbrace{C\sum_{i=1}^{N}\text{loss}(y_i, f(\mathbf{x}_i))}_{\texttt{error}}, \tag{2}$$

donde la función de error que se elige es la función de pérdida de tipo bisagra (para L1-SVM) o su cuadrado (para L2-SVM):

$$H(\mathbf{x}_i) = (1 - y_i \cdot f(\mathbf{x}_i))_+ = \text{máx}(0, 1 - y_i(\mathbf{w}\cdot\Phi(\mathbf{x}_i) + b)) = \xi_i.$$

Las diferentes posibilidades que puede darse para $y_i(\mathbf{w}\cdot\Phi(\mathbf{x}_i)+b)$ son:

- $y_i(\mathbf{w}\cdot\Phi(\mathbf{x}_i)+b) > 1$: el punto está bien clasificado y fuera del margen.
- $y_i(\mathbf{w}\cdot\Phi(\mathbf{x}_i)+b) = 1$: el punto está en el margen.
- $y_i(\mathbf{w}\cdot\Phi(\mathbf{x}_i)+b) < 1$: el punto está dentro del margen o mal clasificado.

La Figura 2 muestra un resumen de las dos propuestas para la unión de las predicciones para la descomposición de tipo cascada.

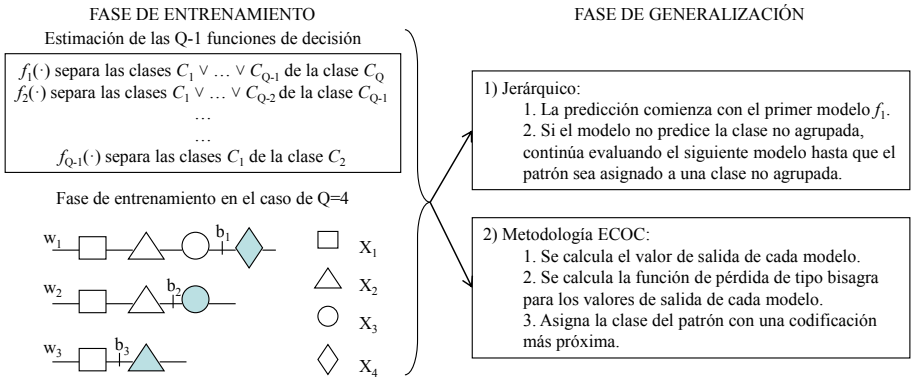


Figura 2. Resumen de las dos propuestas del método de entrenamiento basado en descomposición en cascada.

3. Experimentación

Para obtener la información asociada a los donantes, las características del trasplante realizado y los receptores, se hizo un análisis retrospectivo multicéntrico en 7 unidades de trasplante hepático del estado español y en el hospital King's College de Londres. Todos los pacientes fueron seguidos desde la fecha del trasplante hasta la muerte, la pérdida del injerto o un año después de dicho trasplante. Se recogieron 634 patrones (parejas donante-receptor) correspondientes a los años 2007 y 2008 de los hospitales españoles; mientras que para obtener un número similar de patrones del hospital inglés se recogieron las características de 858 pares de donantes receptores entre enero de 2002 y diciembre de 2010. Con el fin de poder combinar ambas bases de datos se tomaron 16 variables del receptor, 17 del donante y 5 relacionadas con el trasplante en sí. Fue necesario utilizar técnicas estándar de imputación de datos dado que había entre un 1 % y un 10 % de datos perdidos. Aquellos patrones con más de un 10 % de datos perdidos fueron eliminados del estudio, de forma tal que hubo que eliminar 19 y 36 patrones respectivamente de ambas bases de datos, quedando finalmente 615 patrones españoles y 822 ingleses, esto es, 1437 patrones.

3.1. Comparación entre métodos de clasificación

Los métodos de descomposición binaria en cascada que hemos considerado en este trabajo son: JC(SVM): Aproximación jerárquica al modelo en cascada usando SVM y EC(SVM): Aproximación ECOC para el modelo en cascada usando SVM, los cuáles han sido comparados con métodos de clasificación tanto ordinal como nominal: SVM multiclase (aproximación uno contra uno o SVM(1v1)) y SVM multiclase (aproximación uno contra todos o SVM(1vA)) [12]. SVM reformuladas para regresión ordinal [13], aplicando restricciones implícitas (SVORIM)

y con restricciones explícitas (SVOREX). El modelo de probabilidades proporcionales (POM) [14], el cuál fue el primer método de umbral específicamente diseñado para problemas de regresión ordinal. Para los algoritmos SVORIM y SVOREX se ha usado el software de acceso libre proporcionado por los autores[1] y para las diferentes versiones de SVM se ha utilizado el software libsvm[2].

3.2. Medidas de evaluación de un clasificador ordinal

Existen diferentes medidas para evaluar el rendimiento de un clasificador, aunque la más utilizada es el porcentaje de patrones correctamente clasificados (Acc), que como ya se ha indicado, no es la mejor medida para evaluar un clasificador ordinal, ni cuando existe un alto desbalanceo en las clases. Esta medida se define como $Acc = \frac{1}{N}\sum_{i=1}^{N}(I(y_i^* = y_i))$, donde y_i^* es la predicción del modelo e $I(\cdot)$ es la función de pérdida 0/1. Aunque se hará uso de esta medida en este trabajo debido a su importancia, se usará también la media geométrica de las sensibilidades por clase (GMS), la cual está especialmente diseñada para bases de datos no balanceadas como es nuestro caso. Esta medida se define como: $GMS = \sqrt[Q]{\prod_{q=1}^{Q} S_q}$, donde $S_q = \frac{1}{N_q}\sum_{i=1}^{N_q}(I(y_i^* = y_i))$ es la sensibilidad de la clase q-ésima. Finalmente, también se hará uso de la media del error absoluto medio por clase o $AMAE$ [15], la cuál se propuso para problemas ordinales desbalanceados y se define como sigue: $AMAE = \frac{1}{Q}\sum_{q=1}^{Q} MAE_q$, siendo $MAE_q = \frac{1}{N_q}\sum_{i=1}^{N_q}|\mathcal{O}(y_i) - \mathcal{O}(y_i^*)|$, $1 \leq q \leq Q$, donde $\mathcal{O}(\mathcal{C}_q) = q$, $1 \leq q \leq Q$, es decir, $\mathcal{O}(y_i)$ es el orden de la etiqueta y_i.

3.3. Evaluación y selección del modelo

Para poder realizar la comparación del rendimiento de las metodologías presentadas se hizo un diseño experimental de tipo k-fold con $k = 4$ estratificando por centro y clase. Este procedimiento se ha repetido de forma aleatoria 10 veces de forma tal que contamos con 40 pares train/test para cada una de la metodologías. Los parámetros de cada uno de los algoritmos se han optimizado mediante validación cruzada usando un 5-fold en el conjunto de entrenamiento y la métrica GMS. La función de *kernel* elegida para los métodos basados en SVM es la Gaussiana. Para el ancho del *kernel* y el parámetro de coste C asociado a SVM se usó el siguiente conjunto de valores: $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$.

3.4. Resultados

Los resultados obtenidos en los conjuntos de test, se muestran en el Cuadro 1 mediante los estadísticos media y desviación típica, teniendo en cuenta que, en nuestro caso, no sólo nos interesa tener un alto valor en Acc, sino sobre todo, en GMS, y un valor bajo en $AMAE$, dado que necesitamos tener unos buenos

[1] `http://www.gatsby.ucl.ac.uk/~chuwei/svor.htm`
[2] `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

porcentajes de clasificación en las tres primeras clases, donde se pierde el órgano trasplantado. Así, aunque los métodos $JC(SVM)$ y $EC(SVM)$ no presentan valores de Acc elevados, si observamos que los demás procedimientos tienen valores de GMS nulos o cercanos a 0, lo que indica que al menos una de las clases está siendo totalmente desclasificada por el método. Por otra parte, los valores obtenidos con la metodología $EC(SVM)$ podrían considerarse como aceptables, ya que obtiene los mejores valores en GMS y $AMAE$, indicando esto que el clasificador no se centra únicamente en la clase mayoritaria y que en términos de errores ordinales proporciona una clasificación más justa y robusta, por lo que "a priori" podría ser utilizado para el problema tratado. Se incluye además en el Cuadro 1, los resultados del mejor modelo obtenido con esta metodología.

Cuadro 1. Medias y desviaciones típicas ($Media_{DT}$) para los distintos métodos seleccionados en los conjuntos de test.

	Métodos nominales			Métodos ordinales				Mejor modelo
	SVM(1V1)	SVM(1VA)	SVORIM	SVOREX	POM	JC(SVM)	EC(SVM)	EC(SVM)
Acc	$\mathbf{84,82_{1,32}}$	$\mathit{84,43_{1,95}}$	$79,99_{8,51}$	$79,99_{8,51}$	$83,35_{0,63}$	$78,30_{5,57}$	$64,18_{2,94}$	65,46
GMS	$0,00_{0,00}$	$0,00_{0,00}$	$1,03_{4,64}$	$1,03_{4,64}$	$0,00_{0,00}$	$\mathit{3,80_{6,36}}$	$\mathbf{7,46_{9,47}}$	24,69
$AMAE$	$1,497_{0,017}$	$1,494_{0,022}$	$1,445_{0,096}$	$1,445_{0,096}$	$1,504_{0,025}$	$\mathit{1,435_{0,058}}$	$\mathbf{1,345_{0,083}}$	1,154

Los mejores resultados están en **negrita** y los segundos mejores en *cursiva*.

Para comparar la significación estadística de las diferencias obtenidas por los métodos comparados, se han realizado test estadísticos para cada métrica seleccionada. En primer lugar, se analizó si el resultado de las distintas métricas para los métodos seleccionados seguía una distribución normal. En ninguno de los casos se puso asumir una distribución normal usando el test de Kolmogorov-Smirnov con un nivel de significación de $\alpha = 0{,}05$. Como consecuencia de ello, se aplicó el test no paramétrico de Friedman para comprobar si existían diferencias significativas en los resultados. El test concluyó que las diferencias en *ranking* eran significantes para un p-value=0,00, por lo tanto, en el paso siguiente se aplicó el test de rangos con signo de Wilcoxon para pares de algoritmos. Para este test, se ajustó el nivel de significación dividiendo el valor de α por el número de comparaciones menos uno. Los resultados de este test se pueden ver en el Cuadro 2. Para cada algoritmo, estos resultados incluyen el número de algoritmos a los que se gana significativamente (victorias o V), el número de algoritmos sin diferencias significativas en los estadísticos de localización (empates o E) y el número de algoritmos con los que se pierde significativamente (derrotas o D).

3.5. Sistema para asignación donante-receptor final

Para suplir las deficiencias del método de asignación MELD (el cuál no hace uso de las características de los donantes), éste se ha complementado con el modelo predictivo diseñado en este trabajo. En primer lugar, la asignación donante-receptor se realizará en el instante en que haya un órgano disponible de

Cuadro 2. Número de victorias (V), empates (E) y derrotas (D) cuando se comparan los diferentes métodos usando el test de Wilcoxon para $\alpha = 0{,}05$.

	Test de Wilcoxon (V/E/D)						
	SVM(1V1)	SVM(1VA)	SVORIM	SVOREX	POM	JC(SVM)	EC(SVM)
Acc	**5/1/0**	*3/3/0*	1/4/1	1/4/1	2/2/2	1/2/3	0/0/6
GMS	0/4/2	0/4/2	*0/6/0*	*0/6/0*	0/4/2	**3/3/0**	**3/3/0**
AMAE	0/2/4	0/4/2	2/3/1	2/3/1	0/2/4	*3/2/1*	**6/0/0**

Los mejores resultados están en **negrita** y los segundos mejores en *cursiva*.

un donante. Si se hace únicamente mediante la puntuación MELD, esta asignación se realiza teniendo en cuenta la gravedad del receptor y el tiempo en lista de espera, lo que no garantiza que sea la mejor asignación dado que es posible que este emparejamiento tenga una escasa probabilidad de supervivencia. Sin embargo, en nuestra asignación, primero se seleccionarán los k pacientes más graves y con mayor tiempo en lista de espera (siendo k un parámetro a fijar), de forma que se les de mayor probabilidad a dichos pacientes, y después se utilizará el mejor modelo obtenido con la metodología EC(SVM). De esta forma, el órgano se asignará a aquel receptor de entre los seleccionados que sea asignado a la clase mayor en la escala ordinal, es decir, a aquel emparejamiento que tenga una esperanza de supervivencia mayor. Si existiesen varios emparejamientos con esta característica se le asignaría al receptor con mayor MELD. Una simulación de este sistema final mostró que aunque existen algunos factores determinantes tanto en las características del donante como del receptor para la supervivencia o fallo del injerto, en algunos casos existen incompatibilidades entre donantes y receptores, las cuáles sería importante tener en cuenta a la hora de desarrollar un modelo de asignación donante-receptor.

4. Conclusiones

Este trabajo propone una metodología novedosa para asignación donante-receptor en trasplante de hígado basada en técnicas de aprendizaje de máquinas (más concretamente, basada en el paradigma de regresión ordinal). Se construye un modelo ordinal mediante sucesivas descomposiciones binarias en cascada, simplificando así el problema original de clasificación en subproblemas más simples y balanceados, el cuál se usará para predecir la supervivencia del injerto a distintos niveles temporales. Se utiliza para ello un conjunto de datos correspondiente a pares de donante-receptores correspondientes a siete hospitales españoles y uno inglés utilizando características del donante, receptor y del trasplante a realizar. Se obtiene de esta forma, un modelo predictivo capaz de capturar información de compatibilidad/incompatibilidad entre donantes y receptores, el cuál será usado en conjunto con el método MELD, para tener en cuenta además la urgencia del trasplante. De esta forma, el sistema final esencialmente considerará la gravedad del receptor (mediante el método MELD usado a nivel mundial), pero proporcionará oportunidades a receptores más sanos pero que presentan una compatibilidad mayor con el órgano (principio de supervivencia). Como traba-

jo futuro, se podría realizar un análisis de sensibilidad del mejor modelo para determinar las variables que más influyen en la supervivencia del injerto.

Referencias

1. Registro Español de Trasplante Hepático: Sociedad Española de trasplante Hepático, Organización Nacional de Trasplantes (1984–2009)
2. Rosen, H., Prieto, M., Casanovas-Taltavull, T., Cuervas-Mons, V., Guckelberger, O., Muiesan, P., Strong, R., Bechstein, W., O'grady, J., Zaman, A., Chan, B., Berenguer, J., Williams, R., Heaton, N., Neuhaus, P.: Validation and refinement of survival models for liver retransplantation. Hepatology **38**(2) (2003) 460–9
3. Feng, S., Goodrich, N., Bragg-Gresham, J., Dykstra, D., Punch, J., DebRoy, M., Greenstein, S., Merion, R.: Characteristics associated with liver graft failure: the concept of a donor risk index. Am J Trans **6**(4) (2006) 783–90
4. Kamath, P., Kim, W.: The Model for End-stage Liver Disease (MELD). Hepatology **45**(3) (2007) 797–805
5. Rana, A., Hardy, M.A., Halazun, K.J., Woodland, D.C., Ratner, L.E., Samstein, B., Guarrera, J.V., Brown, R.S., Emond, J.C.: Survival outcomes following liver transplantation (SOFT) score: a novel method to predict patient survival following liver transplantation. American Journal of Transplantation **8**(12) (2008) 2537–46
6. Dutkowski, P., Oberkofler, C., Slankamenac, K., Puhan, M., Schadde, E., Müllhaupt, B., Geier, A., Clavien, P.: Are there better guidelines for allocation in liver transplantation? A novel score targeting justice and utility in the model for end-stage liver disease era. Annals of Surgery **254**(5) (2011) 745–753
7. Cruz-Ramírez, M., Hervás-Martínez, C., Fernandez-Caballero, J.C., Briceño, J., de la Mata, M.: Multi-Objective Evolutionary Algorithm for Donor-Recipient Decision System in Liver Transplants. European Journal of Operational Research **222**(2) (2012) 317–327
8. Cruz-Ramírez, M., Hervás-Martínez, C., Gutiérrez, P.A., Pérez-Ortiz, M., Briceño, J., de la Mata, M.: Memetic pareto differential evolutionary neural network used to solve an unbalanced liver transplantation problem. Soft Computing **17**(2) (2013) 275–284
9. Boser, B., Guyon, I., Vapnik, V.: A training algorithm for optimal margin classifiers. In Haussler, D., ed.: Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory, Pittsburgh, PA, ACM Press (1992) 144–152
10. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning **20**(3) (1995) 273–297
11. Wu, H., Lu, H., Ma, S.: A practical svm-based algorithm for ordinal regression in image retrieval. In: Proceedings of the eleventh ACM international conference on Multimedia (Multimedia2003). (2003) 612–621
12. Hsu, C.W., Lin, C.J.: A comparison of methods for multi-class support vector machines. IEEE Transaction on Neural Networks **13**(2) (2002) 415—-425
13. Chu, W., Keerthi, S.S.: Support vector ordinal regression. Neural Computation **19** (2007) 792–815
14. McCullagh, P.: Regression models for ordinal data. Journal of the Royal Statistical Society, Series B (Methodological) **42**(2) (1980) 109–142
15. Baccianella, S., Esuli, A., Sebastiani, F.: Evaluation measures for ordinal regression. In: Proccedings of the Ninth International Conference on Intelligent Systems Design and Applications, ISDA'09. (2009) 283–287

EDM para la detección precoz del fracaso escolar en secundaria

M. Jiménez, J. M. Luna y S. Ventura

Departamento de Informática y Análisis Numérico,
Universidad de Córdoba, Campus de Rabanales, Edificio Albert Einstein,
14071 Cordoba, España. Tel.:+34957212218. Fax:+34957218630
{mjimenez,jmluna,sventura}@uco.es

Resumen El uso de técnicas de minería de datos sobre conjuntos de datos producidos en contextos educativos ayudan a encontrar nuevo conocimiento sobre el proceso de enseñanza-aprendizaje y la gestión de los recursos. El uso de estas técnicas para la predicción del fracaso escolar es de gran utilidad a fin de poder aplicar medidas correctoras. Con este objetivo, el presente estudio trata de determinar la etapa más temprana en la que la calidad de la predicción permite esclarecer la posibilidad de fracaso escolar. Para ello, se ha tratado con información real de un instituto andaluz de educación secundaria, disponiendo dicha información en conjuntos de datos incrementales que representan cómo se va engrosando el expediente académico de un alumno. Los resultados han revelado una detección precoz y robusta del fracaso escolar al final del primer curso de los cuatro que tiene esta etapa.

Palabras clave: EDM, minería de datos, clasificación, fracaso escolar, abandono, secundaria, educación.

1. Introducción

La aplicación de técnicas de Minería de Datos (DM) sobre la información producida en contextos educativos ha originado una disciplina emergente llamada Minería de Datos Educativos (EDM). Esta nueva perspectiva consiste en el desarrollo y aplicación de métodos que exploren datos educativos, así como el uso de técnicas que ayuden a comprender mejor a los estudiantes y los entornos donde aprenden [1].

Entre las principales tareas abordadas por la EDM, la mejora del rendimiento academíco juega un papel fundamental, pudiéndose descomponer en subtareas más sencillas como el como es el descubrimiento de las claves del abandono escolar [2] o la predicción de los resultados académicos [3].

Con independencia de las técnicas utilizadas, la mayoría de los trabajos de EDM existentes en la bibliografía parten de conjuntos de datos generados en entornos de e-learning como LMS (Learning Management System) o sistemas adaptativos [4], y casi siempre, en el ámbito universitario [5]. Son escasos los estudios sobre la anticipación, o la cantidad de información necesaria, para una

predicción robusta del resultado académico. También es frecuente encontrar trabajos que complementan sus datos con resultados de encuestas, cuestionarios o información recabada por otras fuentes [3].

Bajo este marco, el objetivo de este trabajo es la predicción del fracaso escolar en una etapa pre-universitaria a fin de poder aplicar medidas correctoras. De este modo, el grado de anticipación juega un papel de gran relevacia para poder combatir el fracaso escolar. La información considerada en este trabajo es exclusivamente institucional, cedida por un instituto andaluz de educación secundaria, entre la que destacan datos personales del alumnado, sus calificaciones numéricas por evaluaciones, así como sus faltas de asistencia.

Dada la periodicidad trimestral con la que se obtienen nuevos datos, este estudio pretende resolver con qué anticipación es posible determinar si un alumno aprobará el curso y, así mismo, si titulará en secundaria. Para ello se han construido distintos conjuntos de datos que representan, de manera incremental en el tiempo, cada expediente académico. Sobre dichos conjuntos de datos, se han aplicado técnicas de clasificación para determinar en qué momento del proceso de enseñanza-aprendizaje se alcanza el compromiso entre una buena predicción del fracaso escolar y una anticipación lo suficientemente amplia como para poder aplicar propuestas de mejora.

Los resultados obtenidos en el estudio experimental han revelado una detección precoz y robusta del fracaso escolar. Durante el primer curso de los cuatro que tiene esta etapa educativa, se producen resultados muy vinculantes al futuro del estudiante y, por tanto, se cuenta con tiempo suficiente para adoptar medidas correctoras. De este modo, se puede saber qué cambiar y con qué estudiantes hacerlo, aportando información para la mejora del proceso enseñanza-aprendizaje y la optimización de los recursos humanos y materiales.

El presente trabajo se estructura de la siguiente manera: la Sección 2 presenta una descripción del problema, donde se tiene en cuenta la naturaleza de la información tratada, la selección de los datos y los objetivos del estudio; la Sección 3 describe la limpieza, anonimización y preparación de los datos. El estudio experimental se presenta en la Sección 4. Por último, los resultados obtenidos son analizados, así como las conclusiones y futuras mejoras.

2. Descripción del problema

La Consejería de Educación de la Junta de Andalucía almacena desde hace más de una década, y de forma centralizada, información que se genera en los centros educativos (colegios e institutos) a través de la plataforma Séneca[1]. Desde entonces, la puesta en funcionamiento de esta plataforma ha sido paulatina con la incorporación de nuevas funcionalidades, por lo que cursos anteriores al año 2005 poseen muchos valores perdidos.

Por este motivo, sólo las promociones que comenzaron en los cursos 2005/2006 y 2006/2007 son realmente interesantes, integradas por alumnos que en su inmensa mayoría ya han finalizado o abandonado el sistema educativo. Conocer

[1] `https://www.juntadeandalucia.es/educacion/portalseneca/`

el resultado de antemano permite abordar la tarea de la predicción del fracaso escolar mediante técnicas de aprendizaje supervisado, como la clasificación [1], con la que podemos obtener modelos predictivos interpretables.

El centro educativo objeto de estudio se encuentra situado en la provincia de Córdoba, dentro de la comarca de la Campiña Sur. Se trata de un centro típico en cuanto a número de alumnos y profesores, oferta educativa, y contexto socio-económico. Las puntuaciones que obtiene según los indicadores estadísticos de la AGAEVE[2] lo sitúan en la media andaluza.

La gestión y organización educativa ha podido cambiar en algunos matices durante la ventana temporal considerada, por lo que el presente estudio tiene sentido en aquellas conclusiones robustas y generales que no se ven afectadas por dichos cambios.

La información o los principales factores [6] que componen la realidad de cada estudiante pueden ser de muy diversa índole (ver Figura 1). La conducta de cada estudiante frente a su curso y asignaturas vendrá dada por factores intrínsecos a su persona, como su edad, sexo o nacionalidad, sus conocimientos y experiencias previas o background, y sus capacidades, así como sus handicaps, hacia una determinada materia: la capacidad de atención, la motivación [2] o las facilidades que encuentre hacia el entendimiento y el interés prestado.

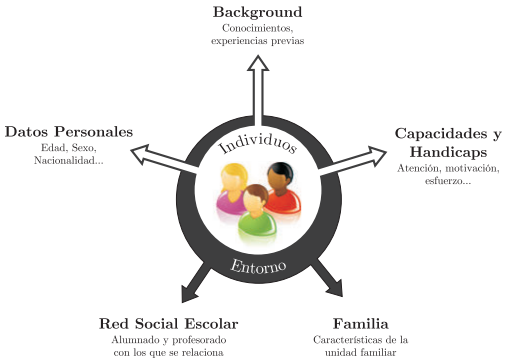


Figura 1. Factores incidentes en el éxito académico

Sin embargo, para el conjunto de alumnos del centro, en Séneca, sólo se almacenan los datos que se exponen:

- **Datos personales**. Nombre y apellidos, DNI, fecha de nacimiento, localidad de residencia, nacionalidad, sexo y tutores legales.
- **NEE**. Necesidades Educativas Específicas. En el supuesto de que un alumno pueda presentar dificultades en su aprendizaje (retrasos, minúsvalías...) se

[2] La Agencia Andaluza de Evaluación Educativa valora y compara anualmente los resultados de los centros educativos de esta comunidad

almacenan informes psicopedagógicos y médicos. Así como las adaptaciones significativas en el curriculum llevadas a cabo de forma individual.

- **Conductas contrarias a la convivencia**. Reflejan un incidente cometido por el alumno y que ha de ser sancionado. Cada entrada contiene una descripción del incidente, las conductas negativas, las correciones aplicadas, el profesor que lo amonesta, la fecha y hora, así como si se trata de una amonestación individual o colectiva.
- **Para cada curso académico**.
  - **Ausencias**. Jornadas y franjas horarias en los que el alumno no asiste a clase. Estas pueden ser justificadas, injustificadas o simplemente faltas de puntualidad o retrasos.
  - **Notas**. Calificaciones obtenidas por el alumno por evaluaciones de cada materia. Las notas son de tipo numérico entero, en el rango [0,10].
  - **Grupo** de alumnos al que pertenece.
  - **Numéro de matrículas**. Nos indica si se está repitiendo curso.

La Figura 2 muestra el flujo del experimento realizado, y explicita las fases en las que se ha descompuesto el problema aquí descrito. Los siguientes apartados tratan al detalle cada una de estas fases.

Figura 2. Flujo del Experimento

3. Preprocesado

La etapa de preprocesado, la cual es independiente a la tarea a realizar a posteriori, ha incluido la limpieza y preparación de los datos. Entendiendo por limpieza la eliminación de ruidos e inconsistencias; y por preparación la anonimización de la información disponible [7] y tratamiento de los datos huérfanos o valores perdidos.

Por lo general, los datos generados en contextos educativos tienen valores discretos, ya sean numéricos o categóricos. Los valores numéricos con los que contamos en este caso son enteros, y los categóricos se han sustituido también por valores enteros. Por su parte, la anonimización es un requisito imprescindible para dar un uso legítimo a información sensible. En España, los usos de esas fuentes de información están limitados por la protección necesaria de la privacidad o de la confidencialidad impuestas por la Ley[3][4][5]. Los datos de carácter personal han sido eliminados y los identificadores completamente disociados. Al tratarse de datos estructurados se ha alcanzado una precisión absoluta en este tratamiento.

4. Diseño Experimental

La extracción de conocimiento para la mejora del rendimiento académico [4] se puede descomponer en sub-tareas o preguntas a contestar a través de la información encerrada en los datos de partida. Todo el dataset puede resultar, a priori, interesante para su resolución, y el preprocesado específico que necesitemos de las fuentes depende de la técnica a emplear. Con el fin de obtener modelos predictivos interpretables se ha considerado el uso de modelos de clasificación para la predicción de la promoción de un curso y la titulación al final de la etapa de secundaria. Para la selección de características se ha tenido en cuenta el ranking obtenido con el método Relief [8], un evaluador basado en instancias.

El diseño experimental ha tenido como base un dataset incremental (ver Figura 3), incluyendo, en una primera etapa, datos previos y atemporales, como el sexo, la edad, nacionalidad, etc. Este dataset irá creciendo evaluación tras evaluación, deteniéndose cuando la bondad de los clasificadores obtenidos no suponga una mejora significativa, logrando una anticipación suficiente para la aplicación de actuaciones correctoras.

Así, se han generado siete conjuntos de datos, el primero correspondiente a la información que se tiene inicialmente: Dataset #1; en la primera evaluación de primero: Dataset #2; en la segunda evaluación de primero: Dataset #3; al final del primer curso: Dataset #4; en la primera evaluación de segundo: Dataset #5; en la segunda evaluación de segundo: Dataset #6; y por último, al final del primer ciclo: Dataset #7.

El entorno escogido para la tarea de clasificación ha sido WEKA [8]. Se han utilizado y comparado diez técnicas clásicas que abordan el problema desde distintas perspectivas. Así, se ha usado un clasificador probabilístico bayesiano (NaiveBayes), una máquina vector-soporte (SMO), una red neuronal de función de base radial (RBFNetwork), el Perceptrón Multicapa (MLP), el algoritmo de

[3] Ley Orgánica 15/1999 de 13 de diciembre de Protección de Datos de Carácter Personal

[4] Real Decreto 994/1999 de Medidas de Seguridad de los ficheros automatizados que contengan datos de carácter personal de 11 de junio de 1999

[5] Real Decreto 1720/2007, de 21 de diciembre de desarrollo de la Ley Orgánica de Protección de Datos

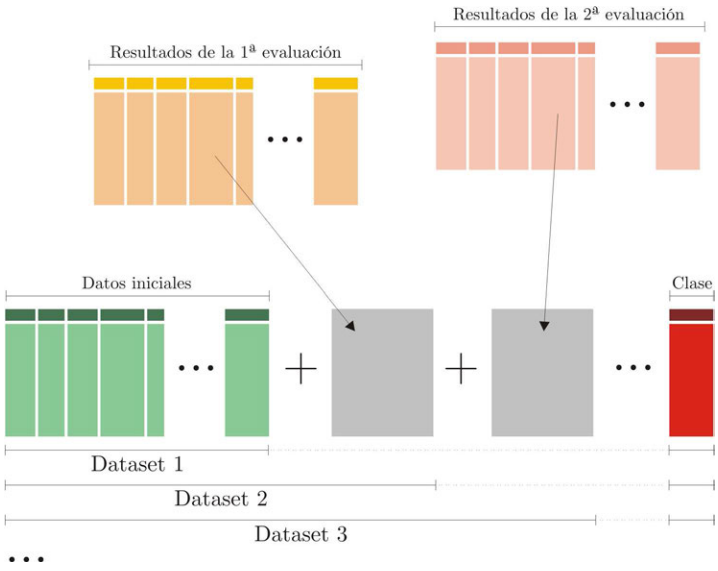


Figura 3. Dataset incremental por evaluaciones

los K vecinos más cercanos (IBk), técnicas basadas en reglas y árboles de decisión (JRip, OneR, J48 y PART) y un clasificador de clase nominal basado en boosting (AdaBoostM1)

El porcentaje de patrones bien clasificados (*accuracy*), y la *kappa* de Cohen son los estadísticos con que evaluaremos la bondad de los clasificadores y las diferencias entre los distintos conjuntos de datos para cada algoritmo.

5. Resultados

La bondad de los clasificadores obtenidos para resolver la tarea de predecir si un estudiante titulará al final de la etapa se presenta en las Tablas 1 y 2. Con el fin de determinar, de manera estadística, si existen diferencias significativas en la predicción del fracaso escolar en las diferentes etapas observadas, se han realizado una serie de tests estadísticos [9].

Centrándonos en los resultados obtenidos para la medida de *accuracy* (ver Tabla 1), el test de Friedman obtiene un valor de 7.452, el cual no pertenece al intervalo crítico de [0, 2.508] para un $\alpha = 0{,}05$. Por tanto, se puede determinar que, con una exactitud del 95 %, existen diferencias significativas en la predicción del fracaso escolar en los diferentes momentos observados.

Analizando los rankings obtenidos para la medida de *accuracy*, se comprueba cómo los mejores resultados se han obtenido en la evaluación final de primero de ESO. Con el fin de determinar si existen diferencias significativas entre esta etapa y el resto, se ha realizado el test de Wilcoxon, cuyos resultados son los que

Tabla 1. Exactitud (Accuracy %) de cada clasificador con cada conjunto de datos

	Dataset						
Algoritmos	#1	#2	#3	#4	#5	#6	#7
NaiveBayes	79.28	88.17	88.59	**90.69**	90.68	89.85	88.18
SMO	79.29	88.64	86.09	89.44	88.11	87.27	**89.77**
MLP	80.13	84.42	85.67	**89.91**	88.03	88.03	84.70
RBFNetwork	79.71	87.30	87.28	**91.11**	87.20	89.02	87.35
IBk	75.53	84.35	86.94	**87.77**	83.03	85.53	87.12
JRip	79.31	87.74	83.12	85.69	85.61	**88.86**	84.70
OneR	80.14	81.87	86.96	86.09	**90.53**	88.03	**90.53**
J48	81.41	**88.55**	86.07	86.52	88.11	83.94	85.53
PART	80.58	82.25	84.35	86.94	87.20	85.53	**88.11**
AdaBoostM1	80.14	88.15	89.93	**90.29**	88.11	85.68	87.20
Ranking	7.00	4.20	4.30	**2.10**	3.40	3.65	3.35

se muestran en la Tabla 3. El test revela que los resultados obtenidos en la evaluación final de primero de ESO son estadísticamente mejores que los obtenidos en etapas previas, con un $\alpha = 0{,}05$. Sin embargo, no se puede demostrar de manera estadística que, a partir de este momento, existan diferencias en la predicción del fracaso escolar. Por tanto, se demuestra que el mejor momento para predecir el fracaso escolar en ESO es el final del primer curso.

Centrándonos en la medida de *Kappa* (Tabla 2), el estadístico de Friedman determina que, con un 95 % de probabilidad, no se puede afirmar que los resultados obtenidos en las distintas evaluaciones se comporten de manera parecida. Este estadístico obtiene un valor de 7.789, el cual no pertenece al intervalo crítico de [0, 2.508] para un $\alpha = 0{,}05$.

De manera similar al análisis de los resultados de la medida de *accuracy*, el mejor ranking es obtenido en la evaluación final de primero de ESO. Así, se uti-

Tabla 2. *Kappa* de Cohen de cada clasificador con cada conjunto de datos

	Dataset						
Algoritmos	#1	#2	#3	#4	#5	#6	#7
NaiveBayes	0.59	0.76	0.77	**0.81**	**0.81**	0.80	0.77
SMO	0.59	0.77	0.72	0.79	0.76	0.75	**0.80**
MLP	0.61	0.69	0.71	**0.80**	0.76	0.76	0.70
RBFNetwork	0.60	0.75	0.75	**0.82**	0.75	0.78	0.75
IBk	0.51	0.69	0.74	**0.76**	0.67	0.72	0.75
JRip	0.59	0.76	0.66	0.71	0.72	**0.78**	0.70
OneR	0.61	0.64	0.74	0.72	**0.81**	0.76	**0.81**
J48	0.63	**0.77**	0.72	0.73	0.76	0.68	0.71
PART	0.61	0.64	0.69	0.74	0.75	0.71	**0.76**
AdaBoostM1	0.61	0.76	0.80	**0.81**	0.76	0.71	0.75
Ranking	7.00	4.30	4.30	**2.25**	3.05	3.65	3.45

Tabla 3. Resultados del test de Wilcoxon para las medidas de *accuracy* y *kappa*

	Accuracy		
Datos	W+	W-	*p*-value
4 vs 2	50	5	0.011
4 vs 3	51	4	0.008
4 vs 5	38	17	0.858
4 vs 6	42	13	0.070
4 vs 7	40	15	0.101

	Kappa		
Datos	W+	W-	*p*-value
4 vs 2	49	6	0.014
4 vs 3	51.5	3.5	0.007
4 vs 5	29.5	15.5	0.203
4 vs 6	40.5	14.5	0.091
4 vs 7	39.5	15.5	0.110

liza el test de Wilcoxon para establecer si existen diferencias significativas entre esta etapa y el resto (Tabla 3). El test revela que, con un 95 % de probabilidad, los resultados obtenidos en la evaluación final de primero de ESO son estadísticamente mejores que los obtenidos en cualquier momento previo. Además, los resultados del test de Wilcoxon determinan que, a partir de la evaluación final de primero de ESO, no existen diferencias significativas en la predicción, por lo que se demuestra que la etapa más temprana para predecir el fracaso escolar es el final del primer curso, y que predicciones posteriores, en perjuicio del grado de anticipación, no mejoran la bondad de la clasificación (ver Figura 4).

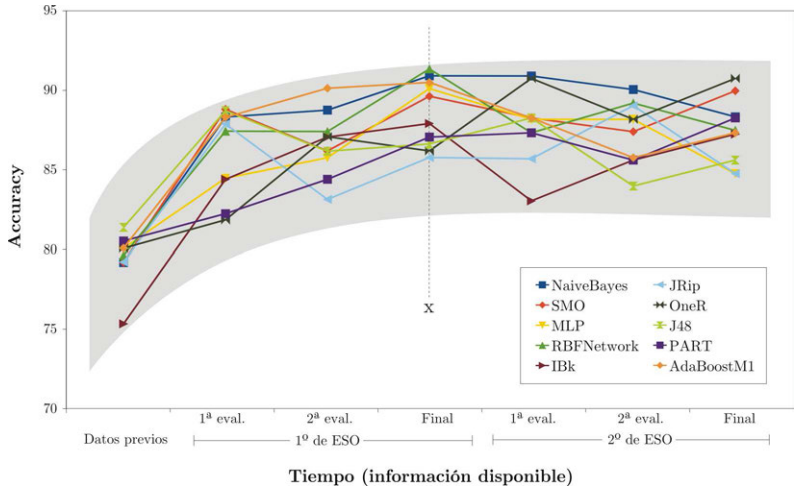


Figura 4. Bondad de la predicción frente al grado de anticipación obtenido.

La cantidad de datos tratada para resolver la predicción de la promoción de cada curso, de manera individual, no permite un estudio estadístico como el anterior, ya que sólo se consideran válidos dos datasets por curso: los datos iniciales y el resultado de la primera evaluación; más tarde no se obtiene un grado

de anticipación suficiente ni interesante desde el punto de vista pedagógico. No obstante, si que resultan interesantes las reglas obtenidas por algunos métodos al resolver esta tarea, y aportan información valiosa que se expone y discute en este apartado.

Se puede comprobar que los algoritmos más interpretables revelan reglas (ver Figura 5) en las que subyace la mayor relevancia de algunas asignaturas y su vinculación con los resultados finales.

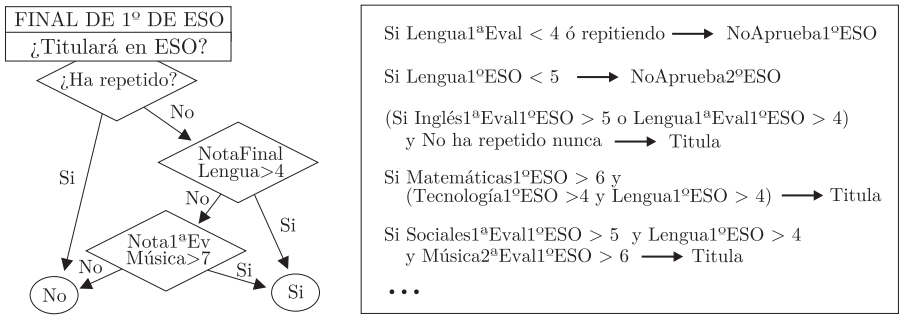


Figura 5. Árbol de decisión y reglas producidas por los algoritmos J48 y JRip

Las destrezas comunicativas toman el papel más importante, y es lógico, pues la falta de comprensión lectora o problemas de expresión constituyen un lastre importante a la hora de adquirir otros conocimientos. La idoneidad curso-edad también es vinculante, el alumnado que ha repetido algún curso presenta mayores dificultades en el aprendizaje.

La detección precoz de estas dificultades, acompañada con la adopción consecuente de medidas tempranas, pueden ser factores clave que ayuden a mejorar los resultados académicos y optimicen la gestión de los recursos disponibles.

6. Conclusiones y trabajos futuros

En el presente trabajo se ha realizado un estudio sobre la detección de la etapa más temprana donde se puede predecir el fracaso escolar en secundaria. Esta predicción tiene como finalidad la actuación precoz, de manera correctora, sobre el alumnado de mayor riesgo. Así, se puede dar un mejor uso a los recursos disponibles como clases de apoyo, desdobles o refuerzos. Se trata de cerrar el flujo del experimento, donde el nuevo conocimiento que se ha descubierto vuelve a revertirse sobre el sistema educativo y volverá a producir datos que serán de nuevo objeto de estudio.

El presente trabajo, es un primer estudio experimental realizado sobre un centro de educación secundaria, por lo que es susceptible de muchas ampliaciones, ya sean en cuanto al volumen de datos, teniendo en cuenta otros centros

y promociones, o en técnicas a emplear sobre estos datos. En trabajos sucesivos, se pretenden aplicar técnicas descriptivas para un análisis más profundo del conocimiento que encierran y con el objetivo general de ayudar a comprender mejor a los estudiantes de educación secundaria y los entornos donde aprenden.

Agradecimientos

El presente trabajo ha sido financiado por la Junta de Andalucía bajo el proyecto P08-TIC-3720. El tratamiento de los datos ha sido posible gracias a la cesión del equipo directivo y administrativo del IES Profesor Tierno Galván de La Rambla, en Córdoba.

Referencias

1. Romero, C., Ventura, S., Pechenizkiy, M., Baker, R.S.J.: Handbook of Educational Data Mining. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series. Taylor & Francis (2011)
2. Finn, J.D., Rock, D.A.: Academic success among students at risk for school failure. Journal of Applied Psychology **82**(2) (1997) 221–234
3. Cortez, P., Silva, A.: Using data mining to predict secondary school student performance. In: Proceeding of the 15th European Concurrent Engineering Conference/5th Future Business Technology Conference, Porto, Portugal (2008) 5–12
4. Romero, C., Ventura, S.: Educational data mining: A review of the state of the art. IEEE Transactions on Systems Man and Cybernetics Part C-Applications and Reviews **40**(6) (2010) 601–618 Times Cited: 20.
5. Barker, K., Trafalis, T., Rhoads, T.R.: Learning from student data. 2004 IEEE Systems & Information Engineering Design Symposium. (2004)
6. Gardner, H.: Intelligence Reframed: Multiple Intelligences for the Twenty-first Century. BASIC BOOKS (NY) (1999)
7. Pyle, D.: Data preparation for data mining. The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann Publishers (1999)
8. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. SIGKDD Explor. Newsl. **11**(1) (November 2009) 10–18
9. García, S., Molina, D., Lozano, M., Herrera, F.: A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: A case study on the cec'2005 special session on real parameter optimization. Journal of Heuristics **15**(6) (2009) 617–644

Optimización evolutiva de la votación para los k vecinos más cercanos

Daniel Mateos-García, José C. Riquelme-Santos y Jorge García-Gutiérrez

Departamento de Lenguajes y Sistemas Informáticos,
Avda. Reina Mercedes S/N, 41012 Sevilla (España)
{mateosg,riquelme,jorgarcia}@us.es
http://www.lsi.us.es

Resumen Este trabajo presenta un algoritmo evolutivo para modificar el sistema de votos en el algoritmo de clasificación de los k vecinos más cercanos. La principal novedad de este enfoque consiste en calcular un conjunto de pesos que se aplica al sistema de votación con independencia de la distancia de cada vecino. El vector de valores calculados puede ser visto como la aportación relativa de cada vecino en la votación para seleccionar la etiqueta de un ejemplo a clasificar. Para demostrar la validez de nuestra propuesta de votación, ésta ha sido probada con 30 datasets del repositorio UCI contra otras 6 variantes del clasificador kNN, resultando en una mejora objetiva que ha sido respaldada estadísticamente.

Key words: votación kNN, computación evolutiva, fuzzy kNN

1. Introducción

En aprendizaje automático y más concretamente en problemas de clasificación, es muy común la utilización de modelos ponderados. Tanto es así, que un ajuste apropiado de pesos durante la fase de entrenamiento, puede mejorar la tasa de predicción del modelo estudiado. Las redes neuronales artificiales (RNAs) y las máquinas de vectores de soporte (MVSs), son quizás los ejemplos más evidentes de la utilización de pesos en modelos de aprendizaje, pero también es frecuente en la regla de los vecinos más cercanos (en inglés k Nearest Neighbours o kNN). En cualquier caso, el objetivo principal de los sistemas de ponderación, es el de ajustar al máximo un conjunto de pesos en la fase de entrenamiento, para posteriormente, obtener una tasa de clasificación lo más alta posible en la fase de prueba. Además de lo anterior, es necesario establecer los mecanismos necesarios para evitar el sobreajuste del modelo obtenido sobre los datos estudiados.

Centrándonos en los métodos de pesado en kNN, la mayoría de las propuestas se centran en el pesado de los atributos o de los ejemplos, y se pueden clasificar como globales o locales, atendiendo al ámbito de aplicación de los datos en consideración. Como ejemplo de métodos globales, Raymer et al. presentan una propuesta para seleccionar y eliminar características mediante un algoritmo genético basado en el vecino más cercano [1]. Este algoritmo optimiza un vector

de pesos que es usado para escalar las características originales. Además de esto, también utilizan un vector de bits para seleccionar atributos simultáneamente. En un trabajo posterior, muestran un algoritmo evolutivo híbrido basado en la función discriminante de Bayes [2]. La finalidad de esta propuesta, es la de aislar características pertenecientes a grandes datasets de origen biomédico seleccionando y extrayendo atributos relevantes.

Además de los métodos basados en computación evolutiva, también existen otras heurísticas en la literatura. Así, Tahir et al. presentan una propuesta híbrida para simultáneamente seleccionar atributos y aplicarles pesos mediante búsqueda tabú, y de esa manera mejorar la tasa de clasificación de los k vecinos más cercanos [3]. Si nos centramos en el pesado de regiones de separación, Fernández et al. proponen un sistema de ponderación local en conjunción con un clasificador basado en prototipos [4]. Los pesos se calculan iterativamente después de normalizar los datos. Esta normalización se basa en la posición de las instancias con respecto al prototipo (o región) al que pertenecen. Continuando con métodos locales, Alsukker et al. usan evolución diferencial como técnica de optimización para buscar pesos que afectan a diferentes elementos de los datos de entrenamiento [5]. En su trabajo se describen cuatro enfoques: pesado de atributos, pesado de vecinos, pesado de clases y pesado mixto (atributos y clases). Es en éste último donde se obtienen los resultados más satisfactorios. Así, el método de pesado mixto obtiene un vector de pesos mediante la concatenación de un vector que contiene un peso por cada atributo, y otro que contiene un peso por cada clase. Mohemmed et al. presentan un clasificador bajo el paradigma del centroide más cercano. La base de este método, es la búsqueda de instancias prototípicas. Éstas son calculadas a partir de las instancias de los datos de entrenamiento, y tomando como medida de concentración la media aritmética. Cuando una instancia de etiqueta desconocida debe ser clasificada, se calcula la distancia de ésta a cada prototipo [6]. Para buscar aquellos centroides que minimizan el error de clasificación, se utiliza un enjambre de partículas como técnica de optimización. Por otro lado, Paredes et al. utilizan distintas funciones de similaridad modificadas con el fin de mejorar el comportamiento del vecino más cercano [7]. Una primera aproximación de esta propuesta, considera un peso por cada característica e instancia para los datos de entrenamiento, lo que hace poco viable el número de parámetros a tener en consideración en el proceso de aprendizaje. De esta manera, los autores presentan tres tipos de reducción: un peso por clase y característica (dependencia de clase), un peso por prototipo (dependencia de prototipo) y una combinación de los dos anteriores. La fase de optimización se lleva a cabo mediante gradiente descendiente.

Otra propuesta basada en la dependencia de las clases y los vecinos, la encontramos en [8]. En este trabajo, se describe un algoritmo evolutivo para buscar una matriz de pesos (un peso por cada atributo y clase) junto a un número óptimo de vecinos, que aumenten la eficacia de la regla kNN.

Existen también referencias al procesamiento de datos desbalanceados [9]. Liu et al. definen una medida llamada Class Confidence Weight (CCW) que mide la probabilidad de que el valor de un atributo pertenezca a una clase. La estimación

de CCW se realiza mediante modelos de mezcla para atributos numéricos, y con redes bayesianas para los categóricos.
Otra perspectiva en el uso de pesos, la encontramos en la aplicación de la teoría de conjuntos difusos para la regla kNN. Las bases de esta idea se centran en la modulación del grado de pertenencia a una clase (por parte de las instancias vecinas) y en la adaptación del sistema predictivo de votación. Este enfoque se denomina Fuzzy k Nearest Neighbour (Fuzzy kNN) y ha presentado buenos resultados en muchos problemas de clasificación [10]. El principal problema de esta propuesta, se centra en la definición de pertenencia difusa a una clase. A pesar de que ésta puede ser establecida por el experto, o incluso ser deducida mediante el análisis *ad hoc* de los datos, la asignación de los valores difusos sigue siendo un problema abierto objeto de investigación [11,12].
Tal y como se ha descrito en los diversos métodos mencionados, podemos resumir que para mejorar la regla del vecino más cercano, se puede aplicar un conjunto de pesos, ya sea de manera global o de forma local, a una función de similaridad. La otra opción analizada consiste en relativizar la pertenencia a una clase por parte de las instancias de entrenamiento. En nuestro trabajo, hemos analizado la posibilidad de que la aportación de los k vecinos en el sistema de votación tenga características distintas en función de un conjunto de pesos calculados. Para ello se ha diseñado un algoritmo evolutivo que calcula la aportación óptima de cada vecino a partir de los datos de entrenamiento.
En las siguientes secciones vamos a mostrar cómo el cálculo de aportaciones relativas para los vecinos, se traduce en una mejora objetiva de la regla kNN. Para ello, el resto del artículo se organiza como sigue. En la sección 2, se muestran los elementos del algoritmo evolutivo utilizado para el cálculo de las aportaciones de la vecindad (los k pesos). Los resultados y tests estadísticos de validación, son detallados en la sección 3. Y finalmente, la sección 4 presenta las conclusiones y trabajo futuro.

2. Método

En esta sección, describiremos nuestro método de votación de vecinos llamado *Evolutionary Votation of Neigbours* (EVoN). Para ello, en la subsección 2.1 presentaremos el propósito de esta propuesta y el uso de los parámetros obtenidos en el proceso de aprendizaje (aportaciones óptimas de la vecindad). Por otro lado, en la subsección 2.2 se explicará el algoritmo de búsqueda en detalle.

2.1. Propósito y funcionalidad

Tal y como se ha descrito previamente, el objetivo perseguido por nuestra propuesta es buscar un conjunto de pesos para modificar la influencia de cada vecino en la votación de la clase a predecir, y de esa manera optimizar la regla de los vecinos más cercanos. Si bien existen referencias al uso de métodos de votación ponderada, podemos decir que normalmente, el foco del estudio se concentra en el cálculo de pesos basados en la distancia entre ejemplos. De esta manera, los

vecinos más cercanos «pesan» más que los más lejanos e influyen más en la votación. Sin embargo, por lo general este sistema no mejora significativamente la bondad de la regla kNN. En nuestro caso, los pesos son calculados por un algoritmo evolutivo con independencia de la distancia. La obtención de un vector de pesos de dominio real, permitirá en la fase de clasificación transformar la influencia de cada vecino sobre la decisión de la clase a devolver, o dicho de otra forma, la votación de un vecino con respecto a la clase a la que pertenece, será un valor real (en vez del clásico valor categórico 1). De esta manera, la decisión de la clase que clasifica a un punto, se toma mediante el máximo de las sumas de los pesos calculados para las etiquetas presentes entre los k vecinos.
Para ilustrar el proceso de aprendizaje, asumimos que el conjunto de clases o etiquetas puede ser representado por el conjunto de números enteros entre 1 y el número de etiquetas b. Así, sea $D = \{(e, l) \mid e \in \mathbb{R}^f \text{ y } l \in \{1, 2, ..., b\}\}$ el conjunto de datos objeto de estudio, siendo f el número de caracterísitcas, y b el número de etiquetas. Sea *label* una aplicación que asigna a cada elemento e, la clase a la que pertenece. Supongamos que D se divide en dos conjuntos TR y TS, siendo cada uno de ellos el conjunto de entrenamiento y el conjunto de prueba respectivamente, tal que $D = TR \cup TS$ y $TR \cap TS = \emptyset$. De esta manera, los ejemplos de TS (conjunto de prueba) serán usados para comprobar la bondad de EVoN, y por tanto, no formarán parte del cálculo de los pesos. Tal y como se detallará en la subsección 2.2, la obtención de un vector $W = (\omega_1, \omega_2, ..., \omega_k)$ se realizará sólo a partir de los ejemplos de TR. Para clasificar la instancia y de TS, se calculan las k instancias más cercanas a y en TR que notaremos x_i. La etiqueta que se asigna a la instancia y viene dada por la formulación:

$$label(y) = \arg \max_{l \in \{1..b\}} \sum_{i=1}^{k} \omega_i \delta(l, label(x_i)) \tag{1}$$

donde $\delta(l, label(x_i))$ vale 1 si $label(x_i) = l$ y 0 en caso contrario.

2.2. Optimización de votos

En esta subsección detallaremos el algoritmo de búsqueda que calcula el vector de aportaciones óptimas para los k vecinos más cercanos. Como se ha mencionado previamente, esta tarea es llevada a cabo por un algoritmo evolutivo, y como tal, es necesario definir sus características generales, esto es, la codificación de individuos, los operadores genéticos, la función de bondad y la política de reemplazo utilizada.

Codificación de individuos Un individuo está formado por un vector de valores reales que simbolizan la aportación relativa de cada vecino sobre el sistema de votación en la regla kNN. En este diseño, las aportaciones están ordenadas en relación a la distancia de cada vecino, siendo el peso de menor posición en el vector, el del vecino más cercano, y el de posición i, el del vecino i-ésimo. De esta

forma, las aportaciones de los vecinos más próximos tendrán mayor importancia, $\omega_1 \geq \omega_2 \geq ...\omega_k$.
En cuanto a la población inicial, ésta estará formada por individuos aleatorios formados por un vector de k valores ordenados comprendidos entre 0 y 1. Para que el clasificador kNN clásico este incluído en la propuesta, hemos añadido a la población inicial un conjunto de vectores con los k primeros valores iniciados a 1, y el resto con un valor 0, esto es, $(1.0, 0.0, ..., 0.0)$ para $k = 1$, $(1.0, 1.0, ..., 0.0)$ para $k = 2$, y así sucesivamente.
El valor máximo inicial de 1 para un peso, podrá ser superado durante el proceso evolutivo.

Cruce y mutación Tal y como se ha mencionado en la subsección 2.2, los genes de cada individuo deben mantener una restricción de orden en todo momento e independientemente de su valor. Además, la función principal del operador de cruce es la de recibir dos individuos progenitores $padre1$ y $padre2$, y devolver un nuevo individuo $hijo$ con características genotípicas de ambos operandos. Para conseguir ambos objetivos, el operador de cruce en el gen i del $hijo$, se define como sigue :

$$hijo(i) = \begin{cases} BLX - \alpha & \text{si } i = 1 \\ (max - min) * \gamma + min & \text{en otro caso} \end{cases}$$

Donde:

$BLX - \alpha$ es el operador de cruce para dominio real definido en [13]
γ es un valor aleatorio entre 0 y 1
$max = hijo(i-1)$
$min = minimo(padre1(i), padre2(i), hijo(i-1))$

En cuanto al operador de mutación y considerando el individuo $indiv$, cada gen i puede ser cambiado según la siguiente ecuación:

$$indiv'(i) = \begin{cases} indiv(i) + indiv(i) * \delta & \text{si } i = 1 \\ indiv(i) - indiv(i) * \delta & \text{si } i = k \\ (indiv(i-1) - indiv(i+1)) * \gamma + indiv(i+1) & \text{en otro caso} \end{cases}$$

Donde δ es un valor aleatorio entre 0 y 1 al principio, pero para un mejor ajuste, cada g generaciones la cota superior es reducida en g/G, siendo G el número de generaciones del algoritmo evolutivo. Así, para $G = 100$ y $g = 10$, la cota superior de δ es 1 en las primeras diez generaciones. En las siguientes diez será de 0.9. Después de otras diez será 0.8 y así sucesivamente.

Función de fitness Durante la fase de entrenamiento, el algoritmo evolutivo utilizará exclusivamente $TR \subset D$ para la obtención de las aportaciones por vecino. Al conocerse la etiqueta de los ejemplos de TR, la función de evaluación estará basada en la tasa de error sobre validación cruzada, utilizando para ello la regla kNN y el sistema ponderado de votación.

```
1: Fitness(W, k, TR) : error
2: error = 0
3: for i = 1 to m do
4:    Dividir TR en s bolsas: B_1...B_s
5:    partialError = 0
6:    for j = 1 to s do
7:       partialError = partialError + Evaluate(W, k, TR − B_j, B_j)
8:    end for
9:    partialError = partialError/s
10:   error = error + partialError
11: end for
12: error = error/m
13: return error

14: Evaluate(W, k, Train, Test) : error
15: error = 0
16: for each instTest in Test do
17:    lab = NearestN(W, k, Train, instTest)
18:    if lab ≠ label(insTest) then
19:       error = error + 1
20:    end if
21: end for
22: error = error/size(Test)
23: return error

24: NearestN(W, k, Train, y) : labY
25: sortedInst y kNeighbours son conjuntos ordenados vacíos
26: for each x in Train do
27:    insertar x en sortedInst ordenado por d(x, y)
28: end for
29: kNeighbours = sortedInst.get(k)
30: labY = majorityLabel(kneighbours, W)
31: return labY
```

Figura 1: **Función de evaluación**

En la Figura 1 se muestra el cálculo de la bondad usando $m \times s$ validaciones cruzadas, donde m es el número de veces que se repite el proceso de validación (línea 3) y s, es el número de particiones del conjunto de entrenamiento TR (línea 4). Así, para cada validación, el conjunto TR es dividido en las s bolsas B_1, B_2... B_s. Después, para cada bolsa B_j, se evalúa el rendimiento de una clasificación, donde B_j actúa como conjunto de prueba, y el resto de TR como conjunto de entrenamiento. Esta evaluación es conducida por la función *Evaluate*, que será descrita posteriormente. El error de clasificación sobre cada B_j es acumulado en promedio por la variable *partialError* (líneas 7 y 9), y por *error* en cada validación (línea 10). Finalmente, el valor de bondad es el resultado de calcular el promedio de todas las validaciones (línea 12).
La función *Evaluate* toma como entrada el vector de pesos W, el valor de k, el subconjunto que actúa como de entrenamiento $TR - B_j$ y el subconjunto que actúa como conjunto de prueba B_j (línea 7). Por tanto, el resultado de esta función es la precisión sobre B_j tomando como referencia $TR - B_j$ para el cálculo de los vecinos.
Para cada instancia del conjunto utilizado para medir la bondad (línea 16), la etiqueta devuelta por la función *NearestN* es la mayoritaria de acuerdo a los k ejemplos más cercanos pertenecientes al conjunto que actúa como training (línea 17). Si la etiqueta devuelta no se corresponde con la etiqueta real de la instancia de prueba, el error se incrementa en 1 (línea 19). Seguidamente, el error resultante es normalizado con el tamaño del conjunto que actúa como de prueba (línea 22). Por tanto, el valor devuelto por *Evaluate* es un número real comprendido entre 0 (todos los ejemplos clasificados) y 1 (ningún ejemplo clasificado).
La función *NearestN* es la encargada de calcular las instancias más cercanas pertenecientes al conjunto de referencia, con respecto a cada instancia y del conjunto sometido a evaluación (línea 24 y siguientes). Cada ejemplo del conjunto de referencia, es insertado en un conjunto ordenado de acuerdo a su distancia a y, de tal manera que el primer ejemplo del conjunto ordenado resultante, se correspondería con el vecino más cercano a y, y el último al más lejano (línea 27). Una vez seleccionados los k vecinos más cercanos del conjunto ordenado (línea 29), se devuelve la etiqueta mayoritaria, de acuerdo a la aportación relativa de cada vecino expresada en el vector W y aplicando la ecuación 1 (lineas 30 y 31).

Política de reemplazo En cuanto al reemplazo generacional de los individuos de una generación a los de la siguiente, hemos elegido un diseño elitista, donde el mejor de los individuos pasa de una generación a otra sin ser afectado por el operador de mutación. El resto de la nueva población está formado de la siguiente manera: Si el número de individuos es N, $C - 1$ son generados por clonación del mejor de los individuos de la generación anterior, y los siguientes $N - C$ individuos son creados mediante el operador de cruce. Todos los individuos excepto el primero, están sujetos al operador de mutación con una probabilidad p. Para seleccionar los individuos que participarán en la operación de cruce, se ha utilizado el método de torneo.

3. Resultados

	EVoN	**IB1**	**IB3**	**IB5**	**FNN1**	**FNN3**	**FNN5**
australian	**85.6 ± 1.3**	80.2 ± 2.2	83.5 ± 1.9	84.3 ± 1.2	80.2 ± 2.2	83.8 ± 1.9	84.3 ± 1.1
balance s.	**89.6 ± 0.6**	86.8 ± 0.9	86.9 ± 1.1	88.2 ± 0.9	78.2 ± 3.5	82.3 ± 2.4	84.6 ± 1.3
breast t.	66.5 ± 5.3	68.2 ± 5.0	63.9 ± 5.9	65.3 ± 7.6	**68.5 ± 5.0**	65.9 ± 5.3	68.2 ± 5.5
breast w.	96.9 ± 1.1	95.6 ± 1.0	96.6 ± 0.9	97.1 ± 1.0	95.9 ± 1.0	96.6 ± 0.9	**97.3 ± 1.0**
car	**93.4 ± 0.5**	93.1 ± 0.5	93.1 ± 0.5	93.1 ± 0.5	76.9 ± 1.5	82.2 ± 1.5	85.9 ± 1.4
cmc	46.6 ± 1.5	44.3 ± 1.3	**47.0 ± 1.6**	45.9 ± 1.5	43.8 ± 1.3	45.4 ± 1.8	45.8 ± 1.4
diabetes	**75.1 ± 1.6**	70.9 ± 2.1	74.3 ± 2.2	74.7 ± 1.5	71.0 ± 1.8	74.2 ± 2.3	74.6 ± 1.5
ecoli	**87.1 ± 2.1**	80.2 ± 2.8	84.8 ± 2.1	86.4 ± 1.9	80.2 ± 2.8	84.8 ± 2.0	87.0 ± 1.9
glass	69.1 ± 2.8	70.0 ± 3.3	68.6 ± 3.4	66.1 ± 4.6	**70.0 ± 3.4**	68.9 ± 2.9	68.3 ± 3.5
haberman	70.9 ± 1.9	67.0 ± 2.5	**71.5 ± 2.7**	71.0 ± 1.7	66.2 ± 2.3	71.4 ± 2.3	70.5 ± 1.8
heart s.	**80.6 ± 2.5**	75.2 ± 3.1	78.5 ± 3.3	78.3 ± 3.1	75.4 ± 3.2	78.5 ± 3.3	78.4 ± 3.0
hill v.	49.0 ± 1.4	50.3 ± 1.5	51.1 ± 2.4	51.3 ± 2.5	50.2 ± 1.4	50.9 ± 2.2	**51.4 ± 2.6**
ionosphere	86.0 ± 2.4	**86.8 ± 2.4**	86.1 ± 1.7	85.6 ± 1.5	**86.8 ± 2.4**	86.1 ± 1.7	85.6 ± 1.5
liver d.	**63.2 ± 5.2**	59.3 ± 3.9	61.8 ± 3.8	58.3 ± 3.7	59.7 ± 4.0	62.1 ± 4.0	58.7 ± 3.9
lymphoma	**83.3 ± 2.7**	81.7 ± 3.0	78.7 ± 4.1	78.5 ± 4.3	82.1 ± 3.3	80.4 ± 4.0	79.4 ± 3.6
mammogr.	**82.4 ± 1.6**	76.8 ± 1.8	80.9 ± 1.8	82.2 ± 1.5	76.1 ± 2.0	80.6 ± 1.9	81.4 ± 1.4
mfeat m.	**73.2 ± 1.3**	65.8 ± 1.4	69.6 ± 1.3	71.0 ± 1.1	66.0 ± 1.6	69.7 ± 1.2	71.6 ± 1
ozone	**94.1 ± 0.2**	92.2 ± 0.9	93.9 ± 0.3	94.0 ± 0.3	92.2 ± 0.9	93.9 ± 0.3	94.0 ± 0.3
pendigits	99.3 ± 0.1	99.3 ± 0.1	99.3 ± 0.0	99.2 ± 0.0	99.3 ± 0.1	**99.4 ± 0.0**	99.2 ± 0.0
postoper.	**72.6 ± 3.7**	62.8 ± 3.0	69.2 ± 4.3	72.4 ± 3.8	56.5 ± 6.9	63.0 ± 5.6	64.9 ± 5.0
sonar	84.3 ± 3.6	84.8 ± 3.4	83.0 ± 4.9	82.5 ± 3.6	**85.2 ± 3.2**	82.3 ± 5.0	82.8 ± 3.7
sponge	88.7 ± 1.7	**92.3 ± 3.0**	88.7 ± 1.7	88.7 ± 1.7	91.7 ± 3.6	90.2 ± 2.2	88.7 ± 1.7
tae	**63.3 ± 4.8**	60.9 ± 6.1	50.3 ± 7.7	53.6 ± 5.8	62.4 ± 5.5	57.1 ± 5.2	54.4 ± 5.3
transfusion	**78.3 ± 1.3**	69.4 ± 2.0	73.8 ± 1.5	75.9 ± 1.6	68.9 ± 1.6	73.0 ± 1.3	75.6 ± 1.7
vehicle	71.3 ± 1.6	70.0 ± 1.4	70.5 ± 1.5	70.9 ± 1.5	69.8 ± 1.5	71.2 ± 1.7	**72.3 ± 1.4**
vote	93.1 ± 1.6	93.0 ± 1.5	93.9 ± 1.8	94.0 ± 2.3	93.1 ± 1.2	93.1 ± 1.7	**94.0 ± 2.3**
vowel	99.0 ± 0.3	**99.0 ± 0.3**	96.4 ± 1.4	92.7 ± 1.3	**99.0 ± 0.3**	96.4 ± 1.4	93.2 ± 1.3
wine	**96.6 ± 1.6**	94.5 ± 1.8	95.6 ± 2.2	95.4 ± 2.3	94.4 ± 1.8	95.7 ± 2.2	95.3 ± 2.3
yeast	**60.4 ± 1.1**	52.9 ± 1.4	55.2 ± 1.1	57.5 ± 1.3	53.0 ± 1.4	55.9 ± 1.3	57.6 ± 1.1
zoo	94.6 ± 2.2	95.3 ± 2.8	92.7 ± 2.7	94.6 ± 2.1	**96.2 ± 2.1**	92.7 ± 2.7	94.6 ± 2.1
	79.8 ± 2.0	**77.3 ± 2.2**	**78.0 ± 2.4**	**78.3 ± 2.3**	**76.3 ± 2.4**	**77.6 ± 2.4**	**78.0 ± 2.2**

Cuadro 1: **Precisión de cada algoritmo estudiado**

Para comprobar la calidad de nuestra propuesta, ésta se ha comparado con el algoritmo IBk, que es la implementación en el framework WEKA[14] para k=1,

3 y 5. También se ha utilizado una implementación del algoritmo Fuzzy kNN que puede ser descargado en [15].
Para los experimentos se han utilizado 30 datasets del repositorio UCI[16] de muy diversas características en cuanto a tipo de atributos y número de clases. Además, todos ellos han sido preprocesados de la misma manera, esto es, binarizando los atributos nominales, normalizando los datos y tratando los valores perdidos con la media. Con respecto a la configuración del algoritmo, se ha utilizado una población de 100 individuos, 100 generaciones, un elitismo del 10 % y una probabilidad de mutación de 0.1. En cuanto a los parámetros α y g, sus valores han sido de 0.5 y 20 respectivamente. En el Cuadro 1 podemos ver los resultados de los algoritmos analizados, y a pie de tabla podemos ver la precisión media de cada uno. Cada dataset ha sido evaluado mediante 10CV utilizando 5 semillas distintas (50 ejecuciones en total). Podemos ver que nuestro algoritmo se comporta mejor en 16 de los 30 datasets, y es el segundo en 4 de los 14 restantes.
Por otro lado, aunque nuestra propuesta destaca sobre las demás, hemos realizado el test no paramétrico Friedman y el método post hoc de Holm para validar los resultados. El motivo de utilizar tests no paramétricos, radica en la alta vulnerabilidad de las condiciones necesarias en los tests paramétricos, especialmente para la condición de esfericidad [17,18].
Para el test de Friedman se ha obtenido la primera posición del ránking, con un $p - value$ de $2.679E - 4$. y rechazándose por tanto la hipótesis nula con un $\alpha = 0.05$.
Para el método post hoc de Holm, que es un procedimiento a posteriori que permite comparar un algoritmo de control (en este caso EVoN) con el resto, también se han rechazado todas las hipótesis de igualdad.
Podemos decir por tanto, que nuestro algoritmo es significativamente mejor que los comparados, desde un punto de vista estadístico.

4. Conclusiones

En este trabajo se ha presentado un algoritmo evolutivo capaz de calcular la aportación óptima de los k vecinos más cercanos. La principal novedad consiste en que, hasta donde sabemos, no hay referencias en la literatura sobre el hecho de que el conjunto de pesos del sistema de votación, sea calculado con independencia de la distancia de cada vecino. Nuestra propuesta de votación, ha sido probada con 30 datasets del repositorio UCI contra otras 6 propuestas basadas en la regla kNN, resultando en una mejora objetiva respaldada estadísticamente.
En trabajos futuros, nos proponemos mejorar la función de fitness haciendo que la curva de aprendizaje, no dependa sólamente de la tasa de error.

Referencias

1. Raymer, M., Punch, W., Goodman, E., Kuhn, L., Jain, A.: Dimensionality reduction using genetic algorithms. Evolutionary Computation, IEEE Transactions on **4**(2) (July 2000) 164–171

2. Raymer, M., Doom, T., Kuhn, L., Punch, W.: Knowledge discovery in medical and biological datasets using a hybrid bayes classifier/evolutionary algorithm. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on **33**(5) (2003) 802–813
3. Tahir, M.A., Bouridane, A., Kurugollu, F.: Simultaneous feature selection and feature weighting using hybrid tabu search/k-nearest neighbor classifier. Pattern Recognition Letters **28**(4) (2007) 438–446
4. Fernandez, F., Isasi, P.: Local feature weighting in nearest prototype classification. Neural Networks, IEEE Transactions on **19**(1) (2008) 40 –53
5. AlSukker, A., Khushaba, R., Al-Ani, A.: Optimizing the k-nn metric weights using differential evolution. In: Multimedia Computing and Information Technology (MCIT), 2010 International Conference on. (2010) 89 –92
6. Mohemmed, A.W., Zhang, M.: Evaluation of particle swarm optimization based centroid classifier with different distance metrics. In: IEEE Congress on Evolutionary Computation'08. (2008) 2929–2932
7. Paredes, R., Vidal, E.: Learning weighted metrics to minimize nearest-neighbor classification error. Pattern Analysis and Machine Intelligence, IEEE Transactions on **28**(7) (2006) 1100–1110
8. Mateos-García, D., García-Gutiérrez, J., Riquelme-Santos, J.C.: On the evolutionary optimization of k-nn by label-dependent feature weighting. Pattern Recognition Letters **33**(16) (2012) 2232 – 2238
9. Liu, W., Chawla, S.: Class confidence weighted knn algorithms for imbalanced data sets. In: Proceedings of the 15th Pacific-Asia conference on Advances in knowledge discovery and data mining - Volume Part II. PAKDD'11, Berlin, Heidelberg, Springer-Verlag (2011) 345–356
10. Keller, J.M., Gray, M.R., Jr: A fuzzy k-nearest neighbor algorithm. IEEE Transactions on Systems, Man, and Cybernetics **15** (1985) 580–585
11. Mendel, J.M.: Advances in type-2 fuzzy sets and systems. Information Sciences **177**(1) (2007) 84 – 110
12. Sanz, J., Fernández, A., Bustince, H., Herrera, F.: A genetic tuning to improve the performance of fuzzy rule-based classification systems with interval-valued fuzzy sets: Degree of ignorance and lateral position. International Journal of Approximate Reasoning **52**(6) (2011) 751 – 766
13. Eshelman, L.J., Schaffer, J.D.: Real-coded genetic algorithms and interval-schemata. In Whitley, D.L., ed.: Foundation of Genetic Algorithms 2, San Mateo, CA, Morgan Kaufmann. (1993) 187–202
14. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: An update. SIGKDD Explorations **11**(1) (2009)
15. Jensen, R., Shen, Q.: Computational intelligence and feature selection: Rough and fuzzy approaches. `http://users.aber.ac.uk/rkj/book/programs.php` (2008)
16. A. Asuncion, D.N.: UCI machine learning repository (2007)
17. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. **7** (2006) 1–30
18. García, S., Herrera, F.: An Extension on "Statistical Comparisons of Classifiers over Multiple Data Sets"for all Pairwise Comparisons. Journal of Machine Learning Research **9** (2008) 2677–2694

Análisis de la Expresión Génica en Tratamientos con Vitamina D mediante Biclustering

Sara Martín-Villanueva[1], Raúl Giráldez[2] y Beatriz Pontes[3]

[1] Facultad de Ciencias Experimentales, Universidad Pablo de Olavide
Ctra. Utrera s/n, 41013, Sevilla
[2] Escuela Politécnica Superior, Universidad Pablo de Olavide
Ctra. Utrera s/n, 41013, Sevilla
giraldez@upo.es
[3] Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla
Avda. Reina Mercedes s/n, 41012, Sevilla
bepontes@us.es

Resumen El estudio a nivel de regulación de la expresión génica de los efectos de ciertos tratamientos médicos tiene un gran interés para la comunidad científica especializada. La tecnología microarray permite realizar esta labor mediante la aplicación de técnicas avanzadas de análisis automático de datos como el biclustering. Este trabajo presenta un estudio de los tratamientos basados en vitamina D mediante la aplicación de la herramienta de biclustering basado en patrones Evo-Bexpa a un microarray de expresión real. Los resultados obtenidos junto a la validación biológica de los mismos demuestran la eficacia de Evo-Bexpa en un contexto real. Asimismo, destaca el interés de los resultados desde el punto de vista biomédico, al descubrirse posibles relaciones entre los genes antes desconocidas y que podrían abrir nuevas líneas investigación en el contexto de los tratamientos basados en vitamina D.

Keywords: Microarrays, Biclustering, Patrones de Expresión, Vitamina D, Validación Biológica.

1. Introducción

La vitamina D se ha utilizada tradicionalmente en tratamiento para algunas enfermedades en pacientes con deficiencia en esta vitamina. Estos tratamientos pueden tener efectos secundarios cuyo análisis es de gran interés para la comunidad científica especializada. Por otro lado, el estudio de los procesos de regulación de la expresión génica ha tenido gran auge en los últimos años, ya que aportan importantes resultados que no sería posible obtener mediante el uso de otras técnicas analíticas. En este contexto, Shalhoub et al [9] presentan un estudio de la influencia de tratamientos basados en vitamina D en seres humanos mediante el uso de tecnología microarray.

La regulación de la expresión génica es un proceso complejo que permite a las células responder a los cambios en las condiciones de su entorno. La tecnología

microarray permite estudiar la expresión génica de un elevado grupo de genes frente a diversas condiciones experimentales, generando una matriz numérica bidimensional en la que cada elemento se corresponde con el nivel de expresión de un determinado gen bajo una condición específica [10]. Estas matrices numéricas permiten el análisis computacional de dichos datos. Existen numerosas técnicas para obtener conocimiento a partir de los datos de un microarray, dependiendo de la aplicación concreta en estudio. En el contexto de este trabajo, el objetivo es agrupar los genes que siguen un patrón de expresión similar que podría derivar de una vía reguladora común [4]. Para ello, se han utilizado tradicionalmente técnicas de clustering [11] agrupando genes según su comportamiento en todas las condiciones experimentales. Sin embargo, generalmente genes relacionados no siguen el mismo comportamiento en todas las condiciones debido a la participación de complejas redes reguladoras que implican múltiples factores. Por esta razón, se han desarrollado técnicas de biclustering [8], una variante del clustering, donde la búsqueda se realiza simultáneamente en las dos dimensiones del microarray. De esta manera, el biclustering nos permite encontrar genes que se comportan de manera similar sólo bajo una serie de condiciones, lo que se corresponde con el comportamiento biológico de los genes: un subgrupo de estos pueden actuar juntos en una condición determinada pero separados en otras condiciones.

Generalmente, los genes implicados en el mismo proceso biológico se comportan de manera similar bajo ciertas condiciones. Por tanto, es interesante encontrar biclusters que agrupen genes cuyos valores de expresión siguen una tendencia similar con respecto a las condiciones que forman parte del mismo. Desde este punto de vista, es posible identificar biclusters en matrices de expresión haciendo uso de medidas basadas en patrones de comportamiento entre genes. En [1] se presenta un estudio en profundidad sobre las principales características inherentes a los biclusters basado en relaciones numéricas entre los valores de expresión de los genes, defiendo formalmente dos tipos de patrones: desplazamiento y escalado.

Debido a la complejidad de las vías reguladoras de los procesos biológicos, el comportamiento biológico de los genes suele seguir un patrón combinado de desplazamiento y escalado. Ello es debido a que no todos estos genes tendrán la misma implicación o influencia en el proceso, ni necesitan alcanzar el mismo nivel de expresión para actuar eficazmente. Por otro lado, es importante destacar que es muy frecuente que se den comportamientos inversos en la tendencia de los genes, ya que a menudo, cuando un gen se expresa bajo una determinada condición otro gen que realiza la función opuesta se reprime porque, en caso contrario, el producto de este gen podría neutralizar el efecto del primero evitando que realizara su función correctamente. Este tipo de comportamiento de regulación inversa es un caso particular de patrón combinado anteriormente mencionado. Los algoritmos de biclustering tradicionales [6] no son capaces de encontrar biclusters cuyos genes sigan patrones de comportamiento combinados ni de regulación inversa. Sin embargo, el avance que las técnicas de biclustering han experimentado en los últimos años han resuelto, total o parcialmente, este

problema. Es el caso de la herramienta Evo-Bexpa [7], que aplica una heurística evolutiva guiada por una métrica propia capaz de conducir la búsqueda hacia la obtención de biclusters que siguen patrones de desplazamiento y escalado simultáneamente, incluidos aquellos que presentan regulación inversa.

En este trabajo se realiza un estudio a nivel de regulación de la expresión génica de los efectos de los tratamientos basados en vitamina D mediante la aplicación de la herramienta Evo-Bexpa a microarray de expresión real correspondiente a la referencia GSE11917 del NCBI. El objetivo de este estudio es doble: por un lado demostrar la eficacia de Evo-Bexpa en un contexto real mediante una validación biológica de los resultados obtenidos; y por otro lado, la obtención de resultados interesantes desde el punto de vista biomédico.

2. Biclustering basado en Patrones

Los patrones de comportamiento de genes pueden definirse formalmente de la siguiente forma [1]:

- **Patrón de desplazamiento**. Un bicluster B sigue un patrón de desplazamiento cuando sus valores b_{ij} se pueden obtener sumando un cierto valor β_j, que será constante para la condición j-ésima, a un valor típico (π_i) para el gen i-ésimo. Formalmente, un bicluster presenta un patrón de desplazamiento cuando sus valores se rigen por la siguiente expresión: $b_{ij} = \pi_i + \beta_j$.
- **Patrón de escalado**. La definición de patrón de escalado es análoga a la del de desplazamiento, sustituyendo el valor aditivo β_j por un factor multiplicativo α_j, tal y como se muestra en la expresión: $b_{ij} = \pi_i \times \alpha_j$.
- **Patrón combinado**. Este patrón es la combinación de las dos expresiones de los patrones de desplazamiento y de escalado, agrupando el error en una sola variable: $b_{ij} = \pi_i \times \alpha_j + \beta_j$.

Desde un punto de vista biológico, los genes implicados en una misma ruta biológica normalmente siguen el mismo patrón de comportamiento bajo determinadas condiciones, ya que se regularán por la misma vía. Sin embargo, generalmente los genes no siguen un patrón de desplazamiento o escalado, sino que presentan un patrón combinado de ambos. Esto es debido a que la complejidad de los procesos biológicos hace necesaria una regulación muy eficiente para que dichos procesos se desarrollen correctamente y estén bien controlados. Así, no todos los genes pertenecientes a una misma ruta se regularán de la misma manera en el sentido de que no todos necesitarán el mismo nivel de expresión para actuar. Asimismo, hay que tener en cuenta que según las condiciones experimentales en que nos encontremos, no todos los genes tendrán la misma implicación en el proceso.

Por otro lado, en las rutas biológicas encontramos a menudo que genes implicados en la misma vía presentan un patrón de comportamiento inverso. Por ejemplo, un gen implicado en la síntesis de glucosa comenzará a expresarse cuando en el cuerpo humano se necesite glucosa para algún proceso determinado, pero simultáneo al aumento de expresión de este gen, será necesario que disminuya la

expresión de un gen implicado en la degradación de la glucosa, ya que si ambos genes actúan a la vez, la tasa de glucosa será menor debido a que se degradaría mientras se sintetiza. Ambos genes tendrán un patrón de expresión inverso, aunque participen en la misma ruta biológica (en este caso el metabolismo de la glucosa). Este fenómeno es muy común y suele estar bien regulado. Sin embargo, un fallo en la regulación de la expresión génica puede tener consecuencias negativas para el individuo en cuestión. De ahí la importancia de encontrar este tipo de patrones para obtener una mayor comprensión de las vías reguladoras y poder detectar irregularidades en las mismas.

2.1. Algoritmo Evo-Bexpa

Evo-Bexpa (*Evolutionary Biclustering based in Expression Patterns*) [7] es una herramienta de biclustering evolutivo basada en la optimización de la métrica denominada VE^t (*Trasnposed Virtual Error*), que cuantifica la calidad de un bicluster basándose en los patrones de expresión seguidos por el mismo. En este sentido, el VE^t de un bicluster es cero cuando éste presenta un patrón perfecto de desplazamiento, escalado o ambos de manera simultánea.

Evo-Bexpa aplica un algoritmo evolutivo cuya función de *fitness* está compuesta por cuatro criterios distintos que se combinan mediante la suma ponderada de los términos que optimizan objetivos diferentes: la medida VE^t, el volumen del bicluster (número de genes y condiciones), la varianza media de genes (cambios en los niveles de expresión) y el solapamiento entre los bicluster obtenidos. Una característica notable que diferencia a Evo-Bexpa otras herramientas es que permite al usuarios cambiar la configuración dependiendo del tipo de biclusters que quiera obtener. Para ello debe establecer los valores de los parámetros: ω_g (número de genes), ω_c (número de condiciones), ω_s (volumen total), ω_{var} (varianza de genes) y ω_{ov} (solapamiento). Evo-Bexpa optimiza el valor de VE^t respetando las restricciones establecida por la configuración de usuario. La descripción completa de Evo-Bexpa así como las instrucciones para su configuración pueden ser consultada en [7].

3. Vitamina D

Como se ha introducido anteriormente, la vitamina D se utiliza como tratamiento para algunas enfermedades relacionadas con la deficiencia en esta vitamina. Un efecto secundario de estos tratamientos puede ser el incremento la concentración de calcio y fósforo, lo que se asocia con un incremento de la calcificación de los vasos sanguíneos, pudiendo provocar trastornos como la arteriosclerosis. Esta calcificación parece ser debida a que las células del músculo liso que forman los vasos sanguíneos se transforman, se transdiferencian a condro u osteoblastos, células que generalmente forman parte de los huesos. Algunos estudios indican que en presencia de fósforo, la vitamina D regula la expresión génica asociada a esta transdiferenciación pero aún no se conoce en profundidad qué rutas biológicas participan en este proceso [9].

El microarray de expresión analizado en este trabajo se corresponde con la referencia GSE11917 del NCBI[4]. Se trata de un estudio en humanos realizado con el Microarray Human Plus 2.0 de Affymetrix[5]. Mediante este microarray, se pretende estudiar la modulación de genes asociados a trastornos cardiovasculares y condro/osteoblastos analizando la expresión de estos genes al añadir fosfato y vitamina D al medio de cultivo, así como otros compuestos. De esta manera, es posible determinar los genes implicados en la calcificación de los vasos sanguíneos y encontrar así un tratamiento alternativo para los pacientes que deban incluir la Vitamina D en su medicación, ya que estos pacientes a menudo presentan una gran calcificación vascular, con riesgo de muerte.

Para ello se realizaron cultivos de células del músculo liso de la arteria coronaria en distintos medios de cultivo. Los medios utilizados se describen a continuación:

- Medio de expansión (ME): D-MEM/15 % FBS
- Medio básico (MB): ME + Dexametasona[6] (10^{-9}M) + Ácido ascórbico[7] (50mg/ml)
- Medio de diferenciación (MD): MB + β-glicerofosfato (10mM)
- Medio de diferenciación y Vitamina D: MD + Vitamina D (10^{-8}M)
- Medio de diferenciación y Vitamina D: MD + Vitamina D (10^{-11}M)

Para cada medio se analizó la expresión de los genes a distintos tiempos de cultivo: a las 4 horas y a 1, 3 o 7 días. También se tomaron los datos a tiempo cero como control para el microarray teniendo finalmente un total de 21 condiciones y 3320 genes. Además, cada condición experimental se analizó por triplicado para poder realizar un análisis estadístico posterior que minimizara la probabilidad de ruido en el conjunto de datos por la posible inclusión de valores erróneos. La experimentación realizada se detalla en [9].

4. Experimentación

En esta sección detallamos la experimentación llevada a cabo sobre el microarray anteriormente descrito. Dicha experimentación incluye tres fases diferenciadas: preprocesado estadístico de los datos, la aplicación de la herramienta de biclustering Evo-Bexpa bajo distintas configuraciones, y finalmente un postprocesado que nos permitió filtrar los biclusters más interesantes para su posterior estudio.

4.1. Preprocesado

Antes de aplicar el algoritmo es necesario realizar un preprocesado de los datos incluidos en el microarray. Para ello, en primer lugar se normalizaron los

[4] http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE11917
[5] http://www.affymetrix.com
[6] La dexametasona acelera la calcificación de CASMC inducida por fosfato.
[7] El ácido ascórbico actúa como cofactor en la formación de la matriz extracelular.

Cuadro 1. Resumen de resultados.

	ω_g	ω_c	ω_s	ω_{ov}	ω_{var}	VE^t	Genes	Conds.
Exp1	0.25	0.5	5	5	0	0.0191	128.17	18.53
Exp2	0.25	0.5	1	10	0	0.0039	11.46	7.07
Exp3	0.25	0.5	1	5	0	0.0039	12.54	7.42
Exp4	0.25	0.5	1.5	5	0	0.0068	18.24	9.77
Exp5	0.25	0.5	2	5	0	0.0095	30.23	10.89
Exp6	0.25	0.5	2.5	5	0	0.0120	48.25	13.11
Exp7	0.25	0.5	3	5	0	0.0143	59.93	15.28
Exp8	0.25	0.5	2	5	0.1	0.0069	8	11.93
Exp9	0.25	0.5	2	5	0.25	0.0072	4.43	8.21
Exp10	0.25	0.5	2	5	0.5	0.0093	3.68	5.82
Exp11	0.25	0.5	5	5	0.1	0.0224	104.87	19.96
Exp12	0.25	0.5	5	5	0.25	0.0219	38.58	17.32
Exp13	0.25	0.5	5	5	0.5	0.0220	13.77	12.61
Exp14	0.25	0.5	5	10	0.1	0.0235	80.2	19.38
Exp15	0.25	0.5	5	10	0.25	0.0224	36.37	17.01
Exp16	0.25	0.5	5	10	0.5	0.0228	13.8	12.9
Exp17	0.25	0.5	5	10	0.75	0.0240	8.31	9.53
Exp18	0.25	0.5	5	10	1	0.0236	6.06	7.5

datos por RMA y se eliminaron los controles de Affymetrix. A continuación, mediante un análisis de varianza ANOVA para cada gen, se estudió cuáles son los genes que más cambian entre las distintas condiciones. Finalmente, tras el análisis ANOVA, se filtraron los datos por un $p - value < 0{,}05$ para cuáles son estadísticamente significativos y no debidos a eventos azarosos. Una vez preparados los datos para la obtención de los biclusters, se aplicó Evo-Bexpa.

4.2. Evo-Bexpa

Se realizaron un total de 18 experimentos aplicando el algoritmo Evo-Bexpa con diferentes configuraciones sobre los datos preprocesados del microarray de expresión. En los distintos experimentos se modificó el tamaño del bicluster, el solapamiento y la varianza modificando los valores de los parámetros ω_g, ω_c, ω_s, ω_{var} y ω_{ov}. Los valores utilizados para los distintos parámetros así como los resultados obtenidos para VE^t y la media de genes y condiciones incluidos en los biclusters se muestran en la Tabla 1.

Evo-Bexpa obtuvo total de 100 biclusters para cada experimento, lo que hace un total de 1800 biclusters. Ello hace necesario realizar un post-procesado para descartar aquellos biclusters no significativos desde el punto de vista biológico.

4.3. Postprocesado

La mayoría de métodos de validación biológica se basan en el uso de repositorios de información biológica y biomédica. Actualmente, dichos datos biológicos vienen dados en forma de ontologías, como la proporcionada por Gene Ontology (GO) [2], o por bases de datos biomédicas, entre las que cabe citar Kyoto Encyclopedia of Genes and Genomes (KEGG) [5]. El post-procesado realizado sobre los 1800 biclusters obtenidos por Evo-Bexpa consistió en la realización de un test de significación estadística (test de Fisher y corrección de Bonferroni) mediante

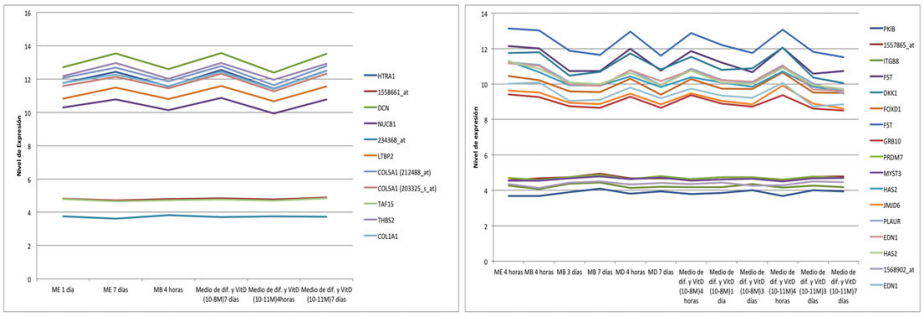
la herramienta Ontologizer [3], utilizando términos incluidos entre los niveles 3 y 6 de la jerarquía GO. Esta selección de niveles fue realizada debido a que, en los últimos niveles se encuentran términos muy generales que están compartidos por un número demasiado elevado de genes; y por otro lado, en los primeros niveles se encuentran términos demasiado específicos que serán muy restrictivos y que son compartidos por un número muy reducido de genes. Se filtraron los datos que presentaron un *p-value*<0.05 para rechazar los biclusters no significativos y, además, se exigió que al menos un 30 % de los genes incluidos en el bicluster compartiesen un determinado término de GO. De esta manera, reducimos la posibilidad de que un término compartido por varios genes del bicluster pueda ser compartido por genes escogidos al azar. Tras este filtrado, obtuvimos finalmente 104 biclusters significativos y un total de 352 términos GO anotados significativamente a los biclusters, entre los niveles 3 y 6 de la jerarquía GO, y que contenían al menos un 30 % de los genes en el bicluster.

5. Interpretación Biológica de los Resultados

Para la validación biológica de los biclusters obtenidos utilizamos la herramienta web DAVID (*The Database for Annotation, Visualization and Integrated Discovery*)[8]. Esta base de datos integrada proporciona un conjunto de herramientas que nos pueden ayudar a comprender el significado biológico de una larga lista de genes, como es el caso de un bicluster. Algunos de estos análisis incluyen mapas de rutas bioquímica, clasificación funcional de genes o la conservación de dominios entre proteínas, entre otros. Además, mediante esta herramienta es posible identificar los términos biológicos más relevantes asociados a una lista de genes. En nuestro caso, utilizaremos la herramienta KEGG Pathway, incluida en DAVID, que incluye una colección de mapas de rutas que representan el conocimiento disponible sobre la interacción molecular en diferentes procesos celulares. De los 3320 genes del microarray, DAVID sólo reconoce 3097. Estos genes pueden estar implicados en una o varias de las rutas que incluye KEGG Pathway. Sin embargo, la mayoría de los genes no tienen una ruta asociada ya que el conocimiento actual disponible es aún muy reducido. De hecho, sólo 776 genes (el 29 % del total) de los incluidos en el microarray completo forman parte de alguna de las rutas que registradas en KEGG Pathway. Los genes analizados se encuentran en 24 rutas, algunas de ellas relacionadas con enfermedades como el Alzheimer, cáncer o diabetes, con rutas metabólicas y con varios procesos celulares.

Para validar biológicamente los biclusters, hemos analizado algunas de las rutas obtenidas en el análisis, comprobando si los genes incluidos en ellas se encuentran agrupados en el mismo bicluster. El resultado obtenido fue que al menos un 80 % de los genes incluidos en algún bicluster estaba agrupado con algún gen de la misma ruta. Este es un resultado muy interesante en cuanto a la eficacia de Evo-Bexpa, ya que, sin ningún conocimiento biológico previo, éste es capaz de agrupar en un mismo bicluster genes implicados en la misma

[8] http://david.abcc.ncifcrf.gov/



a) Bicluster 77 del Experimento 9. b) Bicluster 30 del Experimento 16.

Figura 1. Biclusters.

ruta biológica. También es importante destacar que muchos de estos biclusters incluían un número relativamente elevado de genes implicados en la misma ruta KEGG en relación con el número total de genes del bicluster. Este es el caso, por ejemplo, del bicluster 77 generado en el experimento 9, ilustrado por la Figura 1a, en el que de los 11 genes que contiene, 4 de ellos (COL5A1-212488_at, COL5A1-203325_s_at, COL1A1 y THBS2) están implicados en las mismas rutas (adhesión focal e interacción receptor-matriz extracelular). Sería interesante estudiar los 8 genes restantes para ver si hay alguna relación entre estos y los 4 genes anteriores. Estudios de este tipo pueden ser realizados como base para descubrir nuevas relaciones entre genes que hasta el momento no se habían encontrado. Para estudiar estos 8 genes, en primer lugar comprobamos si participan en alguna ruta conocida y descubrimos que el gen DCN participa en la ruta de señalización TGF-beta, junto con el gen THBS2, de forma que habría una relación directa entre ambos genes y una indirecta entre DCN y el resto de genes mencionados anteriormente.

Dado que KEGG Pathway no nos proporciona ninguna información sobre el resto de genes del bicluster, recurrimos a la otra herramienta mencionada anteriormente, Gene Ontology, para continuar el análisis. Como ya hemos comentado, los términos GO permiten descubrir la relación existente entre los distintos genes. Encontramos una relación directa que no encontramos mediante KEGG Pathway entre DCN, COL5A1 y COL1A1, ya que los tres genes participan en la organización de la matriz extracelular así como en la morfogénesis de órganos. LTBP2 y HTRA1, de los que no disponíamos de información, también comparten términos GO con COL5A1 y COL1A1. Además, NUCB1 tiene términos GO en común con LTBP2 y THBS2, lo que indica que prácticamente todos los genes incluidos en el bicluster están interrelacionados entre sí. En resumen, en un bicluster que incluye 11 genes de los cuales en un principio parecían tener solo relación entre sí 4 de ellos, finalmente, aplicando distintos análisis se ha encontrado relación entre 8 de los genes que el algoritmo ha sido capaz de agrupar. Ni en KEGG Pathway ni en Gene Ontology hemos encontrado información sobre los 3 genes restantes por lo que no podemos afirmar ni descartar que estén relacionados con el resto de genes del bicluster. De hecho, debido a que todos los

genes incluidos en el bicluster presentan el mismo comportamiento, es posible que estos 3 genes participen en alguna de las rutas en las que están implicados el resto de genes.

5.1. Regulación Inversa

Al aplicar el algoritmo Evo-Bexpa al microarray analizado, se han encontrados biclusters que incluyen genes estos comportamientos inversos. Es necesario comprobar que esos genes están relacionados entre sí y que no se trata de genes que participen en rutas distintas. Para ello analizamos el bicluster 30 obtenido en el experimento 16. Este bicluster se representa en la Figura 1b, donde se distinguen dos grupos de genes: un grupo con mayor nivel de expresión (de 9 a 13 aproximadamente) en todas las condiciones y cuya expresión es más elevada a las 4 horas y va disminuyendo conforme pasa el tiempo; y otro grupo de genes con un nivel de expresión mucho menor (de 3,5 a 5 aproximadamente) y cuya expresión disminuye a las 4 horas y va aumentando conforme pasa el tiempo. A continuación realizamos la misma validación en base a los términos GO realizada anteriormente, arrojando resultados muy interesantes, ya que encontramos términos GO que incluyen genes de los dos grupos. En concreto, los genes MYST3, ITGB8 y PDRM7 (que disminuyen su expresión a las 4 horas) tienen asociado términos que comparten con genes del otro grupo. Por ejemplo, los genes MYST3, PDRM7 y JMJD6 participan en la modificación de la cromatina (conjunto de ADN y las proteínas asociadas a éste) actuando sobre residuos del aminoácido lisina, aunque de manera distinta: MYST3 acetila lisinas, PDRM7 metila lisinas y JMJD6 hidroxila lisinas. Para que uno de los genes actúe correctamente, sería necesario que el resto dejasen de actuar. En la Figura 1b se observa que, a diferencia de lo que ocurre con los genes MYST3 y PDRM7, JMJD6 aumenta su expresión a las 4 horas. Esto podría indicar que es necesaria una disminución de la expresión de los otros dos genes para que JMJD6 pueda actuar correctamente ya que, de caso contrario, todos podrían actuar sobre los residuos de lisina a la vez.

6. Conclusiones

Los resultados obtenidos y el estudio de validación biológica de los mismos confirman la eficacia de la herramienta Evo-Bexpa para encontrar biclusters potencialmente interesantes desde el punto de vista biológico, utilizando únicamente información numérica de los niveles de expresión de los genes. Asimismo, cabe destacar la capacidad de Evo-Bexpa de hallar patrones de expresión inversos, cuyo interés biológico también se demuestra en este trabajo y lo que supone una ventaja frente a otros métodos de biclusters que no identifican este tipo de co-expresión en los genes.

Desde el punto de vista biomédico, los resultados también son prometedores, ya que se han identificado posibles relaciones entre genes que, hasta el momento, no se conocían y que podrían abrir nuevas líneas de investigación en el contexto

de los tratamientos basados en vitamina D. Como trabajo futuro, además de realizar estudios similares en otros contextos, se pretende facilitar todos los resultados obtenidos a expertos en el campo biológico que ratifiquen el interés de las relaciones halladas.

Agradecimientos

Este trabajo ha sido subvencionado por el Ministerio de Economía y Competitividad con el proyecto TIN2011-28956-C02. Los autores agradecen a la Dra. Mónica Pérez y a la Dra. Eloisa Andújar, del Centro Andaluz de Biología Molecular y Medicina Regenerativa (CABIMER), su colaboración en el desarrollo de esta investigación.

Referencias

1. J. S. Aguilar-Ruiz. Shifting and scaling patterns from gene expression data. *Bioinformatics*, 21:3840–3845, 2005.
2. M. Ashburner, C. Ball, J. Blake, D. Botstein, H. Butler, J. Cherry, A. Davis, K. Dolinski, S. Dwight, J. Eppig, and et al. Gene ontology: tool for the unification of biology. The Gene Ontology. *Nature Genetics*, 25:25–29, 2000.
3. S. Bauer, S. Grossmann, M. Vingron, and P. Robinson. Ontologizer 2.0-a multifunctional tool for go term enrichment analysis and data exploration. *Bioinformatics*, 24(14):1650–1651, 2008.
4. Y. Cheng and G. M. Church. Biclustering of expression data. In *Proceedings of the 8th International Conference on Intellingent Systemns for Molecular Biology*, pages 93–103, La Jolla, CA, 2000.
5. M. Kanehisa and S. Goto. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res.*, 28:27–30, 2000.
6. S. C. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis: A survey. *IEEE Transactions on Computational Biology and Bioinformatics*, 1:24–25, 2004.
7. B. Pontes, R. Giráldez, and J. S. Aguilar-Ruiz. Configurable pattern-based evolutionary biclustering of gene expression data. *Algorithms for Molecular Biology*, 8(1):4, 2013.
8. A. Prelić, S. Bleuler, P. Zimmermann, and et al. A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics*, 22:1122–1129, 2006.
9. V. Shalhoub, E. Shatzen, S. Ward, J. Young, M. Boedigheimer, L. Twehues, J. McNinch, S. Scully, B. Twomey, D. Baker, P. Kiaei, M. Damore, Z. Pan, K. Haas, and D. Martin. Chondro/osteoblastic and cardiovascular gene modulation in human artery smooth muscle cells that calcify in the presence of phosphate and calcitriol or paricalcitol. *J Cell Biochem*, 111(4):911–21, 2010.
10. C. Tilstone. Dna microarrays: Vital statistics. *Nature*, 424:610–612, 2003.
11. H. Wang, W. Wang, J. Yang, and P. Yu. Clustering by pattern similarity in large data sets. In *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, pages 394–405. ACM, 2002.

Accelerating learning by reusing abstract policies in gErl

Fernando Martínez-Plumed, Cèsar Ferri, José Hernández-Orallo, and María José Ramírez-Quintana

DSIC, Universitat Politècnica de València, Camí de Vera s/n, 46022 València, Spain.
{fmartinez,cferri,jorallo,mramirez}@dsic.upv.es

Abstract. In this paper, we present gErl [7], a general learning system for which the operators can be modified and finetuned for each problem, and its intrinsic ability to transfer learning between related tasks (as q values or models). gErl, as an Erlang-coded general learning system, allows the users write (or adapt) their own operators, according to the problem, data representation and the way the information should be navigated. Since changing operators affect how the search space needs to be explored, heuristics are learnt as a result of a decision process based on reinforcement learning where each action is defined as a choice of operator and rule. The knowledge acquired solving an specific task may be reused to accelerate the learning of new more large related tasks due to the abstraction of states and actions.
Keywords: machine learning operators, complex data, reusing policies, transfer learning, inductive programming, reinforcement learning, Erlang.

1 Introduction

The reuse of knowledge, in an RL setting, tipically attempts to decrease training time by learning a source task before learning the target task [12]. While previous transfer work has mainly focused on reducing training time on closely related tasks by transferring from a simple to complex task in a single domain, we show a (potentially) more powerful way of simplifying this transfer task formulating it as an abstraction of states and actions.

In this paper we present and explore gErl [7], a general rule-based learning system where operators can be defined and customised for each kind of problem. While one particular problem may require generalisation operators, another problem may require operators which add recursive transformations to explore the structure of the data. A right choice of operators can embed transformations on the data but can also determine the way in which rules are generated and transformed, so leading to (apparently) different learning systems. Making the user of the problem adapt its own operators is significantly more difficult than the use of feature transformations or specific background knowledge, since operators can be very complex things and usually embed the essence of a machine learning system.

From here, we devise a flexible architecture which works with populations of rules and programs, which *evolve* as in an evolutionary programming setting or a learning classifier system [3]. Operators are applied to rules generating new rules, which are combined with existing or new programs. With appropriate operators and using some optimality criteria (based on coverage and simplicity) we will eventually find some good solutions to the learning problem. However, without heuristics, the number of required iterations gets astronomically high. This issue is addressed with a reinforcement learning (RL) approach, where the application of an operator over a rule is seen as a decision problem, for which learning also takes place, guided by the optimality criteria which feed a rewarding module. Interestingly, the system is able to reuse knowledge acquired in the current learning process in order to improve or accelerate the learning of future tasks in a totally transparent way due to the feature space used to describing states and actions.

The paper is organised as follows. Section 2 makes a short account of the related and previosu work. Section 3 introduces the gErl system and the RL-based heuristics used to guide the learning process. Section 4 discusses how to transfer knowledge between tasks. Section 5 includes an empirical study. Section 6 closes the paper.

2 Previous work

Although the system we use in this paper is related to different areas of machine learning: learning from complex data, reinforcement learning (*RL*), Learning Classifiers Systems, evolutionary techniques, meta-learning, etc., in this section we will only focus in its ability to transfer previous acquired knowledge between tasks and summarise some of the previous works in this field (Transfer Learning (TL) in RL frameworks) somehow related to our proposal. See [12] for a survey.

Several methods improves learning by introducing additional knowledge into the exploration choices. The main idea considers breaking down a single large target task into a series of simpler source tasks, where transfer learning can be performed by initializing the Q-values of a new episode with previously learned Q-values [4, 6]. Other algorithms introduce previous knowledge throught macro-actions or *options* [11] (somewhat more general actions) to learn (with less data) new action policies in Semi-Markov Decision Processes.

Hierarchical RL uses different abstraction levels to organize subtasks in a hierarchy [2]. The methods for learning hierarchies or options capture the structure of the domain (SKILL [13]).

Another way to transfer learning can be performed by initializing the Q-values of a new episode with previously learned Q-values. However, in almost all cases it can not directly applied because explicit mappings are needed in order to transfer between tasks with different actions and state representations. This inter-task mapping may be provided to the learner or may be autonomously learned. In the first subgroup we found methods as the *advice* rules [8] where the user *advise* appropriate actions to be preferred in different sets of states or by composing of solutions of elemental sequential tasks [9]. In the second subgroup we can found methods which work with graph mapping techniques that can

automatically find a mapping between actions and state variables in two task as in *qualitative dynamic Bayes networks* [5].

Although TL in RL between related problems has made significant progress in recent years, our work aims to be a fully transparent to the system, where this one takes care of everything (without the need of human experts or the use of inter-tasks mappings) using an abstract description of the knowledge transferred. In addition, our work also may follow the idea of *learning from easy missions*, where the learning of a hard problem could be solved in an incrementally way.

3 The gErl system

As we have mentioned in Section 1, in this paper we use the gErl system [7], a general learning system which can be configured with different (possibly user-defined) operators. As a long-term goal, this can be roughly seen as a general system for designing customised systems for applications with complex data. It can be described as a flexible architecture which works with populations of rules (expressed as unconditional / conditional equations) and programs in the functional language Erlang, which *evolve* as in an evolutionary programming setting or a learning classifier system [3]. Operators are applied to rules and generate new rules, which are combined with existing or new programs. With appropriate operators, using some optimality criteria (based on coverage and simplicity) and using a reinforcement learning-based heuristic (where the application of an operator over a rule is seen as a decision problem fed by the optimally criteria) many complex problems can be solved. Figure 1 shows the general architecture of gErl. In the rest of this section we will resume the main characteristics of the system. We refer the reader to [7] for more details.

.

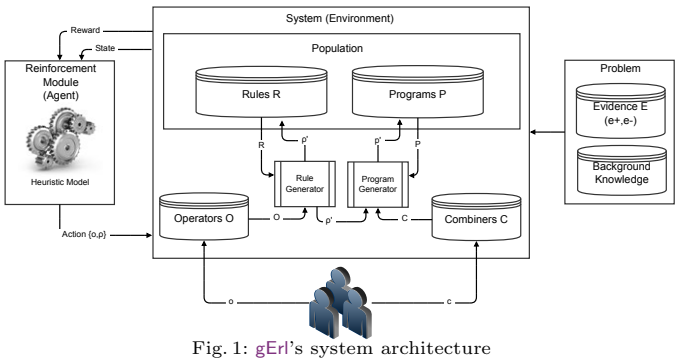


Fig. 1: gErl's system architecture

3.1 Data and model representation and learning problem statement

gErl can be considered and inductive (functional) programming system, as data and rules (and hence solutions) are represented in a functional programming language, Erlang. Let us see how these elements are represented and how the learning problem is stated.

An equation is an expression of the form $l = r$ where l and r are terms. l is called the left hand side (*lhs*) of the equation and r is the right hand side

(*rhs*). $\mathcal{R}$ denotes the space of all (conditional) functional rules ρ of the way l [when G] $\rightarrow T, r$ where l and r are the lhs and the rhs of ρ (respectively), $G = \{g_1, g_2, \ldots g_m \mid m \geq 0\}$) is a set of conditions or Boolean expressions called guards, and $T = b_1, \ldots, b_n$, the tail of ρ, is a sequence of equations. If $G = \varnothing$, then ρ is said to be an unconditional rule. An expresion (term, equation or rule) is *ground* if it does not contain any variable symbol. As usual, variables are denoted by a sequence of letters and digits starting with a capital letter. An example e is a ground rule $l \rightarrow r$ (that is, without condition nor tail), being r in normal form. Let $\mathcal{P} = 2^{\mathcal{R}}$ be the space of all possible functional programs formed by sets of rules $\rho \in \mathcal{R}$. We say that an example $l \rightarrow r$ is covered by a program p (denoted by $p \models \{l \rightarrow r\}$) if l and r have the same normal form with respect to p. A functional program $p \in \mathcal{P}$ is a solution of a learning problem defined by a set of positive examples E^+, a (possibly empty) set of negative examples E^- and a background theory B if it covers all positive examples, $B \cup p \models E^+$ (completeness), and does not cover any negative example, $B \cup p \not\models E^-$ (consistency). Our system has the aim of obtaining complete solutions, but their consistency is not a mandatory property, so approximate solutions are possible.

As we can see in Figure 1, our system works with two sets: a set of rules $R \subseteq \mathcal{R}$ and a set of programs $P \subseteq \mathcal{P}$, where each program $p \in P$ is composed by rules belonging to R. Initially, the set of rules R is populated with the positive evidence E^+ and the set of programs P is populated with as many unitary programs as there are rules in R.

3.2 Operators and learning process

The definition of customised operators is one of the key concepts of our proposal (unlike others systems which are based on the use of different fixed operators [1]). The main idea is that, when the user wants to deal with a new problem, he/she can define his/her own set of *operators* $O \in \mathcal{O}$ especially suited for the data structures of the problem.

An operator can be seen as a piece of code which performs modifications over the *lhs* or *rhs* of a rule and which is written in the same functional language as the system (Erlang) to take advantage of its high-order and reflection capabilities.

As the process progresses, new rules and programs will be generated. First, the *Rule Generator* process (Figure 1) gets the operator o and the rule ρ returned as an action $a = \langle o, \rho \rangle$ by the *Reinforcement Learning Module* (policy) we will explain in the next section. This process applies the operator over the rule obtaining a new rule ρ' (if the operator is not suitable for the rule selected, the process returns the same rule) which is added to R.

3.3 The Reinforment Learning process

In order to give the users the freedom to define their own operators, the search space heuristics in gErl has been conceived as decisions about the operator that must be used at each particular state of the learning process. For this, a model-based reinforcement learning approach has been developed guided by the optimality criteria which feed a rewarding module.

RL problem statement. We model the decision process as a typical reinforcement learning task. Formally, our decision problem is a four-tuple $\langle \mathcal{S}, \mathcal{A}, \tau, \omega \rangle$ where: $\mathcal{S}$ is the state space, where a state at each iteration t of the system is represented as a tuple $\sigma_t = \langle R, P \rangle$ being R and P the population of rules and programs in t, respectively; $\mathcal{A}$ is a finite actions space ($\mathcal{A} = \mathcal{O} \times \mathcal{R}$); $\tau : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is a transition function between states and $\omega : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function. These components are defined below:

- **States.** As we want to find a good solution to the learning problem, we describe each state σ_t by a tuple of features $s_t = \langle \phi_1, \phi_2, \phi_3 \rangle$ from which to extract relevant information in t:
  Global optimality (ϕ_1): This feature shows the average optimality of all programs in P_t. In turn, the optimality of each program p is a weighted combination of three simpler heuristics (according to its importance):

$$Opt(p) = \beta_1 \cdot CRD(p) - \beta_2 \cdot OU(p) + \beta_3 \cdot RP(p) \tag{1}$$

  where CRD (coverage rate difference) is a normalised measure of coverage difference between positive examples and negative examples covered, OU (operator usage) is a normalised measure of how many operators have been used to derive the rule and RP is a measure of the expressiveness of the programs in terms of number of variables, functions and constants. As for the current implementation of gErl, the weights are $\beta_1 = 1$, $\beta_2 = 0.2$ and $\beta_3 = 0.1$. Finally, the *Global optimality* factor is then calculated as the average of the optimalities of all programs in the system:

$$OptGlobal(P_t) = \frac{1}{Card(P_t)} \sum_{p \in P_t} Opt(p) \tag{2}$$

  Average Size of Rules (ϕ_2): measures the average complexity of all rules in R, using the RP measure seen above.
  Average Size of programs (ϕ_3): measures the average cardinality of all the programs in P_t in terms of the number of rules.
- **Actions.** An action is a tuple $\langle o, \rho \rangle$ with $o \in \mathcal{O}$ and $\rho \in \mathcal{R}$, that represents the operator o to be applied to the rule ρ. Each rule is described by a tuple of features $\rho = \langle \varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5, \varphi_6, \varphi_7, \varphi_8 \rangle$, that represents some relevant information of rule ρ: the size (φ_1), the positive coverage rate (φ_2), the negative coverage rate (φ_3), the number of variables of ρ (φ_4), the number of constants (funtors with arity 0) of ρ (φ_5), the number of functors with arity greater than 0 of ρ (φ_6), the number of structures (lists, graphs, ...) of ρ (φ_7) and, finally, φ_8 indicates if the rule ρ is recursive or not.
  We use a natural index as the only feature for operators. As an action consists of a choice of operator and rule, an action is finally a tuple of *nine* features.
- **Transitions.** Transitions are deterministic. A transition τ evolves the current sets of rules and programs by applying the operators selected (together with the rule) and the combiners.
- **Rewards.** The optimality criteria seen above is used to feed the rewards. In particular, we use the result returned by equation (1) as reward.

At each point in time, the reinforcement learning policy can be in one of the states $s_t \in \mathcal{S}$ and may select an action $a_t = \pi(s_t) \in \mathcal{A}$ to execute. Executing such action a_t in s_t will change the state into $s_{t+1} = \tau(s_t, a_t)$, and the policy receives a reward $w_t = \omega(s_t, a_t)$. But as the policy does not know τ and ω, they need to be learned. This is the typical formulation of RL [10] but using features to represent the states and the actions. Therefore, the aim of our decision process is to find a policy $\pi : \mathcal{S} \to \mathcal{A}$ that maximises:

$$V^{\pi}(s_t) = \sum_{i=0}^{\infty} \gamma^i w_{t+i} \tag{3}$$

for all s_t, where $\gamma \in [0, 1]$ is the *discount parameter* which determines the importance of the future rewards ($\gamma = 0$ only considers current rewards, while $\gamma = 1$ strives for a high long-term reward).

Modelling the state-value function by using a regression model. For the RL module in gErl, we use a hybrid between model-free value-function methods (which update a state-value matrix) and model-based methods (which learn models for τ and ω) [10]. In particular, our approach uses the *state-value* function ($Q(s, a)$, which returns quality (q) *values*, $q \in \mathbb{R}$, as in Q-learning [14]) which is generalised by a regression model $M : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ that calculates the optimality or q value for each state and action. By using $a_t = \arg\max_{a \in \mathcal{A}} \{M(s_t, a_i)\}$ we get the best action for state s_t.

In order to train the model we use a 'matrix' Q (which is actually a table), whose rows are in $\mathcal{S} \times \mathcal{A} \times \mathbb{R}$ where $\mathcal{S}$ is a tuple of state features, $\mathcal{A}$ is the tuple of actions, and $\mathbb{R}$ is a real value for q. So, each row in Q records the state and action for each time step in the system and its q-value. Table 1 shows an example of Q. Abusing notation in order to work with Q as a function (like the original Q-matrix in many RL method), we will denote by $Q[s, a]$ the value of q in the row of Q that corresponds to state s and action a.

state (s)			**action** (a)								**q**
ϕ_1	ϕ_2	ϕ_3	o	φ_1	φ_2	φ_3	φ_4	φ_5	φ_6	φ_7	φ_8	
1.223	1.473	6.431	2	17	3	0	4	1	2	0	0	**0.78**
1.301	1.511	6.253	2	16	3	1	3	2	1	0	0	**0.65**
...												

Table 1: Example of a Q matrix (represented as a table).

A supervised model M (using q as the output) is retrained periodically from table Q (gErl uses linear regression by default). Once the system has started, at each step, Q is updated using the following formula, as in Q-learning:

$$Q[s_t, a_t] \leftarrow \alpha \left[w_{t+1} + \gamma \max_{a_{t+1}} M(s_{t+1}, a_{t+1}) \right] + (1 - \alpha) Q[s_t, a_t] \tag{4}$$

where the maximum future value is obtained by the model, $\gamma \in [0, 1]$ is the discount parameter and $\alpha \in [0, 1]$ is the *learning rate* which determines to what extent the newly acquired information will override the old information ($\alpha = 0$ makes the agent not to propagate anything, while $\alpha = 1$ makes the agent consider only the most recent information). By default, $\alpha = 0.5$ and $\gamma = 0.5$.

4 Reusing past policies

In this section, we describe how to reuse policies and how they are applied. The main reason why we can use the past policies ("matrix" Q) learned in order to accelerate the learning of different new tasks is due abstract representation of states and actions (ϕ and φ parameters) which allows the system does not start from the scratch and reuse the optimal information, namely, actions successfully applied from certain states (from the previous task) when it reaches a similar (with similar features) new state.

In our system, when it reaches the solution of a given problem (or it executes a maximum number of steps), Q and the model M can be viewed as knowledge acquired that can be transferred to the new situation. This knowledge acquired, concretely Q, is passed to the new problem to be solved in order to bias the learning. When gErl starts learning the new task, Q is used to train a new model which is used from the begining (exploiting it) and it is updated with the new information acquired in each time step of the system (in order to retrain the new model M). Figure 2 briefly shows the process.

Furthermore, our system also allows consider breaking down a task into a series of smaller task in order to be easier for the system to learn in isolation than in the context of the full task. For instance, many problems can consist of a large evidence, which cannot fit in memory or need lot of computational resources. This approach can also be considered a type of transfer in that a single large target task can be incremental treated as a a series of simpler source tasks, where, if the hypothesis founded in a simple source task is stable (covers part or all positive evidence), the time required to solve the large task will be substantially reduced if the system reuse the knowledge previously acquired.

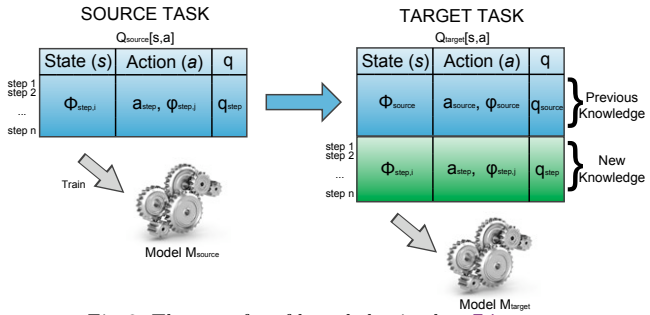


Fig. 2: The transfer of knowledge in the gErl system.

5 Empirical study

In this section, we describe the experiment performed to demonstrate the usefulness of the policy reuse strategy used in the gErl system and how a problem may be broken down in a series of smaller tasks in order to accelerate the learning. We also show more details about our system and how it solves these problems[1].

[1] gErl is available at `http://users.dsic.upv.es/~fmartinez/gerl.html`

Positive Examples E+:
e_1: s([an, unknown, alien, hits, the, house]).
e_2: s([a, small, boy, walks, a, dog]).
e_3: s([a, dog, walks, into, the, house]).
...

Negative Examples E-:
e_4: ¬s([dog, hits, a, boy]).
e_5: ¬s([the, house, a, boy]).
e_6: ¬s([dog, walks,]). hits]).
...

Hypothesis $\mathcal{H}$:
h_1: s([S_1,S_2,S_3,S_4,S_5]) → np(S_1,S_2), vp(S_3), np(S_4,S_5).
h_1: s([S_1,S_2,S_3,S_4,S_5,S_6]) → np(S_1,S_2), vp(S_3,S_4), np(S_5,S_6).
h_3: s([S_1,S_2,S_3,S_4,S_5,S_6]) → np(S_1,S_2), vp(S_3), np(S_4,S_5,S_6).
h_4: s([S_1,S_2,S_3,S_4,S_5,S_6,S_7]) → np(S_1,S_2,S_3), vp(S_4,S_5), np(S_6,S_7).
h_5: s(S_1,S_2,S_3,S_4,S_5,S_6) → np(S_1,S_2,S_3), vp(S_4), np(S_5,S_6).

Background Knowledge B:
b_1: np(S_1,S_2) → det (S_1), noun(S_2).
b_2: np(S_1,S_2,S_3) → det (S_1), adj(S_2), noun(S_3).
b_3: vp(S_1) → verb (S_1).
b_4: vp(S_1,S_2) → verb (S_1), prep(S_2).
b_5: det(a). b_6: det(an). b_7: det(the).
b_8: noun(dog). b_9: noun(boy).
b_{10}: noun(house). b_{11}: noun(alien).
b_{12}: verb(dog). b_{13}: verb(walks).
b_{14}: adj(small). b_{15}: adj(unknown).
b_{16}: prep(into).
...

(a) (b)

Fig. 3: Grammar Learning Example

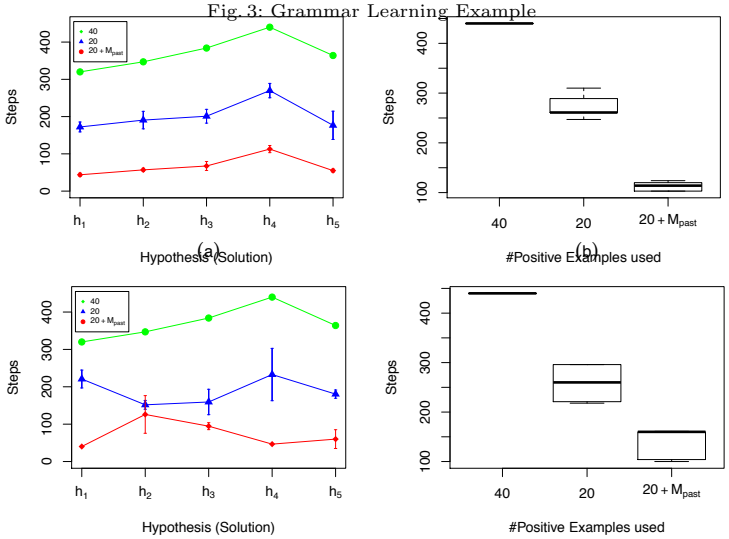


Fig. 4: Top: Preserving the distribution of the original problem. Bottom: Random distribution of examples. Left: Step in which each hypothesis h_i is found (mean). Segments shows the standard deviation. Right: The number of steps needed to learn all hypothesis in each execution

Fig. 3 shows part of a simplified version of the English language phrase grammar problem which is used in our study. The problem consists of 40 positive examples E^+ and 63 negative examples E^-. It is necessary to use a large background knowledge B in order to deal with every kind of word. As we can see in Fig 3, the hypothesis $\mathcal{H}$ is composed by 5 general rules which cover all the positive evidence (each h_i $(1 \leq i \leq 5)$ covers the same number of positive examples).

Since the examples are represented as lists of words, we need to define appropriate operators to navigate this structure and apply local or global changes to it. The first two operators, $nounPhrase_2$ and $nounPhrase_3$, take the input list and generalise its 2 or 3 first components (respectively) by replacing them by variable symbols and adding to the *rhs* of the rule the noun phrase syntax structure constructed with these generalised components. The definition of these operators is written in Erlang, but it can be informally de-

fined using an example as follows: $nounPhrase_2(s([a, small, boy, walks, a, dog])) \Rightarrow s([S_1, S_2, boy, walks, a, dog]) \rightarrow np(S_1, S_2)$; $nounPhrase_3(s([a, small, boy, walks, a, dog])) \Rightarrow s([S_1, S_2, S_3, walks, a, dog]) \rightarrow np(S_1, S_2, S_3)$. As we can see in the previous example, only the second operator generates a useful more general rule for the input string given. The third and fourth operators, $verbPhrase_1$ and $verbPhrase_2$, work in a similar way to the first two operators but, in this case, for constructing verb phrase syntax structures. In order to see how they work, the operators can be informally defined using an example as follow: $verbPhrase_1(s([S_1, S_2, S_3, walks, a, dog]) \rightarrow np(S_1, S_2, S_3)) \Rightarrow s([S_1, S_2, S_3, S_4, a, dog]) \rightarrow np(S_1, S_2, S_3), vp(S_4)$; $verbPhrase_2(s([S_1, S_2, S_3, walks, a, dog]) \rightarrow np(S_1, S_2, S_3)) \Rightarrow s([S_1, S_2, S_3, S_4, S_5, dog]) \rightarrow np(S_1, S_2, S_3), vp(S_4, S_5)$. As before, only one operator generates a useful rule for the input string given, in this case, the third operator.

With this four operators the gErl system is able to solve the previous problem and to find the correct order of application of the operators to obtain each one of the five hypothesis in Fig. 3. For instance, the sequence for the instance $s([an, unknown, alien, hits, the, house]$ is:

$$\begin{aligned} nounPhrase_3(s([an, unknown, alien, hits, the, house].) &\Rightarrow \\ s([S_1, S_2, S_3, hits, the, house] &\rightarrow np(S_1, S_2, S_3) \\ verbPhrase_1(s([S_1, S_2, S_3, hits, the, house] &\rightarrow np(S_1, S_2, S_3)) \Rightarrow \\ s([S_1, S_2, S_3, S_4, the, house] &\rightarrow np(S_1, S_2, S_3), vp(S_4) \\ nounPhrase_2(s([S_1, S_2, S_3, S_4, the, house] &\rightarrow np(S_1, S_2, S_3), vp(S_4)) \Rightarrow \\ s([S_1, S_2, S_3, S_4, S_5, S_6] &\rightarrow np(S_1, S_2, S_3), vp(S_4), np(S_5, S_6) \end{aligned}$$

This latter equation $s([S_1, S_2, S_3, S_4, S_5, S_6] \rightarrow np(S_1, S_2, S_3), vp(S_4), np(S_5, S_6)$ (which corresponds with the hypothesis h_5) is the solution for this example.

To analyse the ability of the system to improve the learning when reusing past policies, we perform two experiments. The first experiment consist in taking 5 random samples of 20 examples from the original problem but keeping the same distribution of the original problem, and learning from them a policy. Each policy learned is used to learn with the rest of 20 positive instances not selected previously in each sample. Fig 4 (top) shows the results. A clear improvement is observed when we reuse knowledge ("$20 + M_{past}$" sample) regarding when we do not ("20" sample). The results also show how the learning has been accelerated using a smaller sample ("20" sample) compared to the original problem ("40" sample). The second experiment also consists in taking 5 random samples of 20 examples from the original problem, but now without taking into account the distribution of examples and learning from them a policy. As in the first experiment, we use these policies learned to bias the learning of the rest of 20 positive instances not selected in each sample. Figure 4 (bottom) shows the results where, as in the previous experiment, there exists a clear improvement when we use previous acquired knowledge to solve future related tasks.

6 Conclusions and future work

One of the problems for reusing in new problems the knowledge acquired from previous problems is how to represent this knowledge. In this paper we have presented the main features of the gErl system, specially its reusing policy which is based on a representation of knowledge as a tuple of abstract characteristics.

We have used our system to solve a simple problem showing that the total learning time is reduced when policy reuse is applied. But, in order to make more conclusive claims, more experiments should be done on other domains.

There are many other things to explore in the context of gErl. We would like to include features for the operators as well. In more general terms, we think that a way (or measure) of similarity between problems would help us to better understand when the system is able to detect these similarities, from the point of view of better assessing the achievements of the system. Finally, while we have focussed on the system gErl, we think that many of the ideas in this paper could also be applied to other kinds of systems, most especially learning classifier systems, reinforcement learning and other evolutionary techniques.

Acknowledgements. This work was supported by the MEC projects CONSOLIDER-INGENIO 26706 and TIN 2010-21062-C02-02, GVA project PROMETEO/2008/051, and the REFRAME project granted by the European Coordinated Research on Long-term Challenges in Information and Communication Sciences & Technologies ERA-Net (CHIST-ERA), and funded by the Ministerio de Economía y Competitividad in Spain. Also, F. Martínez-Plumed is supported by FPI-ME grant BES-2011-045099.

References

1. C. Ferri, J. Hernández, and M.J. Ramírez. Incremental learning of functional logic programs. In *FLOPS*, pages 233–247, 2001.
2. B. Hengst. Discovering hierarchy in RL. Technical report, 2003.
3. J. Holland and Booker. What is a learning classifier system? In *Learning Classifier Systems*, volume 1813 of *LNCS*, pages 3–32. 2000.
4. J.Carroll. Fixed vs Dynamic Sub-transfer in Reinforcement Learning. In *ICMLA'02*. CSREA Press, 2002.
5. Y. Liu and P. Stone. Value-function-based transfer for reinforcement learning using structure mapping. AAAI, pages 415–20, July 2006.
6. M. Madden and T. Howley. Transfer of experience between Reinforcement Learning environments with progressive difficulty. *Artif. Intell. Rev*, 21:375–398, 2004.
7. F. Martínez-Plumed, C. Ferri, J. Hernández-Orallo, and M. J. Ramírez-Quintana. Learning with configurable operators and rl-based heuristics. In A. Appice et al., editor, *New Frontiers in Mining Complex Patterns*, volume 7765 of *LNCS*, pages 1–16. Springer-Verlag, 2013.
8. B. Price and C. Boutilier. Accelerating Reinforcement Learning through implicit imitation. *Journal of Artificial Intelligence Research*, 19:2003, 2003.
9. S. Singh. Transfer of learning by composing solutions of elemental sequential tasks. In *Machine Learning*, pages 323–339, 1992.
10. R. Sutton. *Reinforcement Learning: An Introduction*. MIT Press, 1998.
11. R. Sutton. Between MDPs and semi-MDPs: A framework for temporal abstraction in Reinforcement Learning. *Artificial Intelligence*, 112:181–211, 1999.
12. M. Taylor and P. Stone. Transfer learning for Reinforcement Learning domains: A survey. *Journal of Machine Learning Research*, 10(1):1633–1685, 2009.
13. S. Thru. Finding structure in reinforcement learning. In *NIPS*, 1995.
14. C. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8:279–292, 1992.

A methodology to improve single-experiment k-fold Cross-Validation by reducing the impact of related dataset shift

Jose G. Moreno-Torres, José A. Sáez, Francisco Herrera

Department of Computer Science and Artificial Intelligence,
Universidad de Granada, 18071 Granada, Spain
(jose.garcia.mt,smja,herrera)@decsai.ugr.es

Abstract. Cross-validation is a very commonly employed technique to evaluate classifier performance. However, it can potentially introduce dataset shift, a harmful factor that is often not taken into account, and which can result in inaccurate performance estimation. This works analyzes the impact of partition-induced covariate shift on k-fold cross-validation, and presents an alternative means of partition creation that takes dataset shift into account and avoids it by design.

1 Introduction

In order to evaluate the expected performance of a classifier over a dataset, k-fold cross-validation schemes are commonly used in the classification literature [1]. Also, when comparing classifiers, it is common to compare them according to their performances averaged over a number of iterations of cross-validation. Even though it has been proved that these schemes asymptotically converge to a stable value that allows realistic comparisons between classifiers [2, 3], in practice a very low number of iterations are often used. The most common variations are 2x5, 5x2 and 10x1, with this notation meaning 2 folds iterated 5 times, 5 folds iterated 2 times and 10 folds iterated one time respectively. Note that when more than one iteration takes place, the partitions are assumed to be constructed independently.

This paper studies the intrinsic variability present in k-fold cross-validation schemes from the point of view of dataset shift, [4, 5], which is the situation where the data the classifier is trained on and the data the classifier is going to be used on do not follow the same distribution. This work analyzes how different partitioning methods can introduce dataset shift, and what effect does it have over both the reliability of an experiment with a low number of iterations of k-fold cross-validation and the number of iterations needed to reach a stable value.

To best analyze the impact of dataset shift, we use four different strategies to create the partitioning. One of them is Stratified Cross Validation, which is the most commonly employed method in the literature. Two are methods designed

to minimize the covariate shift introduced, while the last one is designed to check the worst case scenario by introducing as much shift as possible.

The goal of this work is to analyze the accuracy of classifier performance prediction when a single instance of k-fold cross-validation is employed.

The remainder of the paper is organized as follows: Section 2 provides a background on cross-validation and dataset shift. In Section 3 the different partitioning methods used for the experimentation in this work are detailed. Section 4 shows the strategy employed to test the suitability of each partitioning method, while section 5 explains the experimental study performed, including datasets and methods applied and results obtained. The paper is then closed with a few concluding remarks and recommendations in Section 6.

2 Background

This section presents a brief introduction to classifier evaluation through cross-validation in Subsection 2.1, and to dataset shift in Subsection 2.2; introducing the concepts relevant to this work.

2.1 Cross-validation for classifier evaluation

Cross-validation is a technique used for assessing how a classifier will perform when classifying new instances of the task at hand. One iteration of cross-validation involves partitioning a sample of data into two complementary subsets, training the classifier on one subset (called the training set), and testing its performance on the other subset (test set).

In k-fold cross-validation, the original sample is randomly partitioned into k subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the classifier, and the remaining $k-1$ subsamples are used as training data. The cross-validation process is then repeated k times, with each of the k subsamples used exactly once as the test data. The k results from the folds are then averaged to produce a single performance estimation.

2.2 Dataset shift

The term dataset shift refers to the issue where training and test data follow different data distributions [4, 5]. It can happen due to the intrinsic nature of the problem (for example, a classifier trained over financial data from the past five years and used to predict future market changes); or it can be introduced in cross-validation schemes without noticing.

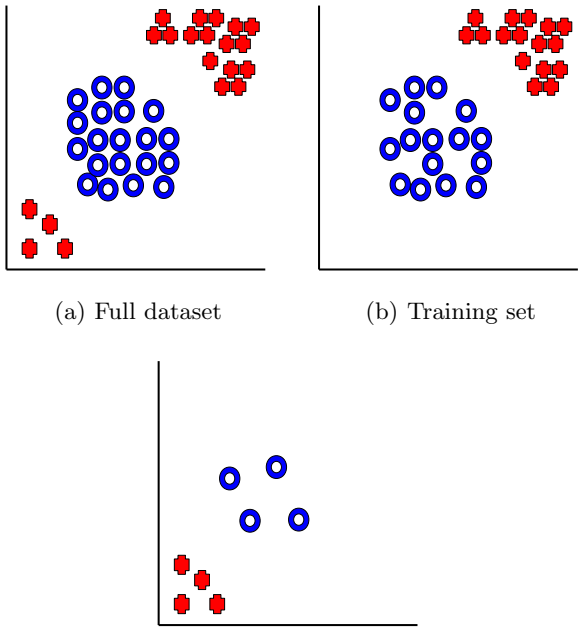
This work focuses on the latter, studying k-fold cross-validation strategies and the types and impact of dataset shift in them. There are two potential types of dataset shift:

1. **Prior probability shift**: Happens when the class distribution is different between the training and test sets. In the most extreme example, the training

set would not have a single example of a class, leading to a degenerate classifier. The problems caused by this kind of shift have already studied, and it is commonly prevented by applying a stratified cross-validation (SCV) scheme.

2. **Covariate shift**: In this case, it is the inputs that have a different distributions between the training and test sets. Figure 1 depicts an extreme example of this type of shift that can also lead to extremely poor classifier performance. This type of shift is often ignored in the literature, and the analysis of its prevalence and potential impact is the main contribution of this work.

Fig. 1: Extreme example of partition-based covariate shift. Note how the examples on the bottom left of the "cross" class will be wrongly classified due to covariate shift.



(a) Full dataset (b) Training set

(c) Test set

3 Partitioning methods

This section presents a detailed explanation of the different partitioning methods used for testing in this work, including pseudo-code to make the replication of our experiments easier. Some assumptions made throughout the pseudocodes:

- The number of folds in a given cross-validation implementation is denoted as k.
- Folds are named $F_i (i = 0, ..., k-1)$. They are treated as a set of examples, initially empty.
- D is another set of examples, initially containing all the examples in the dataset.
- There is a set of classes $C = \{c_1, ..., c_m\}$, where m is the number of classes.
- There is a function $count(c_i)$ that returns the number of examples of class c_i in D.
- Lastly, these methods detail the way to construct the test sets, the training sets are simply the remainder of the dataset.

3.1 Stratified cross-validation

This is the standard method most authors in the field of classification apply (for an example of use, see [6]). A pseudo-code explaining how it works can be seen in Algorithm 1. It is easy to see how this method avoids prior probability

Algorithm 1 Stratified cross-validation partitioning method

```
for each class c_j ∈ C do
  n ← count(c_j)/k
  for each fold F_i(i = 0, ..., k − 1) do
    E ← randomly select n examples of class c_j from D
    F_i ← F_i ∪ E
    D ← D \ E
  end for
end for
```

shift by having an equal distribution class-wise on each fold, but does nothing to prevent covariate shift happening.

3.2 Distribution-balanced stratified cross-validation

Proposed in [7], this method adds an extra consideration to the partitioning strategy as an attempt to reduce covariate shift on top of preventing prior probability shift. The method follows the steps detailed in Algorithm 2.

The idea is that by assigning close-by examples to different folds, each fold will end up with enough representatives of every region, thus avoiding covariate shift.

3.3 Distribution-optimally balanced stratified cross-validation

This method includes a variation over the one above, and is an original contribution of this work. The basic difference with DB-SCV lies in the order the examples are picked to be assigned to each fold. The specifics about this method can be found in Algorithm 3

Algorithm 2 Distribution-balanced stratified cross-validation partitioning method

for each class $c_j \in C$ **do**
  $e \leftarrow$ randomly select an example of class c_j from D
  $i = 0$
  while $count(c_j) > 0$ **do**
    $F_i \leftarrow F_i \cup \{e\}$
    $D \leftarrow D \setminus \{e\}$
    $i = (i + 1)\ mod\ k$
    $e \leftarrow$ closest example to e of class c_j from D
  end while
end for

Algorithm 3 Distribution optimally-balanced stratified cross-validation partitioning method

for each class $c_j \in C$ **do**
  while $count(c_j) > 0$ **do**
    $e_0 \leftarrow$ randomly select an example of class c_j from D
    $e_i \leftarrow i$-th closest example to e_0 of class c_j from D $(i = 1 \ldots k-1)$
    $F_i \leftarrow F_i \cup \{e_i\}$ $(i = 0 \ldots k-1)$
    $D \leftarrow D \setminus \{e_i\}$ $(i = 0 \ldots k-1)$
  end while
end for

3.4 Maximally-shifted stratified cross-validation

This method is basically the opposite of the previous one in terms of covariate shift, trying to maximize it while keeping prior probability shift at a minimum. Its pseudo-code is shown in Algorithm 4.

In this case, the assignation of all close examples to the same fold creates incidences of severe covariate shift, since entire regions are kept without a single example representing them in some folds.

4 Comparing partitioning methods

We performed 3 independent experiments using the same procedure, where the only difference was the type of cross-validation scheme used. We tested the 2x5, 5x2 and 10x1 cross-validation schemes.

For each of the above validation schemes, we created 100 independent experiments using each of the methods described in Section 3 to test single-experiment accuracy.

To evaluate single-experiment accuracy, a "truth" is needed in terms of classifier comparison. To obtain this "truth", we created an extra 200 independent partitions using SCV. For each dataset, we averaged the performance of each classifier over the 200 cross-validation experiments, and then performed a

Algorithm 4 Maximally-shifted stratified cross-validation partitioning method

for each class $c_j \in C$ **do**
  $n \leftarrow count(c_j)/k$
  $e \leftarrow$ randomly select an example of class c_j from D
  for each fold $F_i (i = 0, ..., k-1)$ **do**
    for $s = 1 \rightarrow n$ **do**
      $F_i \leftarrow F_i \cup \{e\}$
      $D \leftarrow D \setminus \{e\}$
      $e \leftarrow$ closest example to e of class c_j from D
    end for
  end for
end for

Wilcoxon signed-ranks test [8] with a significance level of 0.1 to test whether there existed significant differences between their performances. We considered this Wilcoxon test to be the true comparison between each classifier pair.

We chose to focus only on the pairwise comparisons between classifiers where the true comparison showed a significant difference between classifiers, since it is harder to reach relevant conclusions from the cases where a significant difference could not be found. We then tested how often a single cross-validation experiment would yield the same result, setting the significance level equal to the p-value obtained by the true comparison.

Since the procedure is not trivial to understand, we include an example here with the case of 5x2 experiments. 10x1 and 2x5 are done analogously:

1. Average the performance of each classifier on each dataset over the 200 partitions destined to obtain a "true" comparison.
2. Perform a Wilcoxon signed-ranks test with the averaged data for each classifier pair. The results can be seen in Table 1, where the numbers on each cell should be read as R+/R-/p-value. Discard the classifier pairs where p-value$>$ 0.1. In the table, the cases with p-value under 0.1 are marked in bold.

Table 1: Wilcoxon signed-ranks test. R+ corresponds to row algorithm, R- to column. Results are presented as R+/R-/p-value. For better readability, only 5 of the 9 algorithms studied are presented in this example table.

	1NN	3NN	C45	FURIA	LDA
1NN	-	120.5/257.5/0.1029	150/228/$>$ 0.2	120.5/257.5/0.1029	143/235/$>$ 0.2
3NN	120.5/257.5/0.1029	-	186/192/$>$ 0.2	143.5/234.5/$>$ 0.2	**261/117/0.0859**
C45	150/228/$>$ 0.2	186/192/$>$ 0.2	-	120.5/257.5/0.1029	135/243/$>$ 0.2
FURIA	120.5/257.5/0.1029	143.5/234.5/$>$ 0.2	120.5/257.5/0.1029	-	**280/98/0.0280**
LDA	143/235/$>$ 0.2	**117/261/0.0859**	135/243/$>$ 0.2	**98/280/0.0280**	-

3. For each of the 100 instances of 10x1 cross-validation created with each partitioning method, perform a Wilcoxon signed-ranks test for each classifier

pair that showed a significant difference in Step 2 (those marked with bold font), using the p-values obtained in that step as the significance threshold. Count how many of the 100 instances achieve the same results. Table 2 shows an example table of results for DOB-SCV. A similar table is constructed for each of the other partitioning methods studied. The number on each cell is the number of DOB-SCV partitions where the Wilcoxon signed-ranks test declared a significant difference between the compared classifiers' performance.

Table 2: Number of times Wilcoxon signed-ranks obtained the same result in DOB-SCV as it did in the "true" run. Again, only 5 of the 9 methods are presented

	1NN	3NN	C45	FURIA	LDA
1NN	p-value> 0.1	p-value> 0.1	p-value> 0.1	p-value> 0.1	p-value> 0.1
3NN	p-value> 0.1	p-value> 0.1	p-value> 0.1	p-value> 0.1	**96**
C45	p-value> 0.1	p-value> 0.1	p-value> 0.1	p-value> 0.1	p-value> 0.1
FURIA	p-value> 0.1	p-value> 0.1	p-value> 0.1	p-value> 0.1	**66**
LDA	p-value> 0.1	**96**	p-value> 0.1	**66**	p-value> 0.1

4. Average the results of each cell to obtain an aggregated estimation of how close a given partitioning method is to the "true" estimation. In the example of DOB-SCV for 5x2, that average turns out to be 52.50. This is how Table 4 is constructed.

All the experiments were conducted using the KEEL software tool [9].

5 Experimental study

In this section the experimental study is presented. In order to achieve relevant results, 27 datasets and 9 classifiers were used. They can be seen in Subsections 5.1 and 5.2 respectively. All classifiers were compared against each other (resulting in 36 unique pairs) over their performance in the test set. The performance metric chosen was AUC [10]. The results can be found in Subsection 5.3.

5.1 Datasets

All the datasets used are binary classification problems, and were obtained from the KEEL dataset repository [11]. When there were missing values, the whole example was eliminated. A list of the datasets used can be seen in Table 3, where "(R/I/N)" refers to real, integer and nominal attributes.

5.2 Classification algorithms

The following classification algorithms were used in this study:

Table 3: Datasets used for the experimental study

Dataset	# attributes (R/I/N)	# examples	Dataset	# attributes (R/I/N)	# examples
appendicitis	7 (7/0/0)	106	australian	14 (3/5/6)	690
banana	2 (2/0/0)	5300	bands	19 (13/6/0)	365
breast	9 (0/0/9)	277	bupa	6 (1/5/0)	345
chess	36 (0/0/36)	3196	crx	15 (3/3/9)	653
german	20 (0/7/13)	1000	haberman	3 (0/3/0)	306
heart	13 (1/12/0)	270	hepatitis	19 (2/17/0)	80
housevotes	16 (0/0/16)	232	ionosphere	33 (32/1/0)	351
mammographic	5 (0/5/0)	830	monk-2	6 (0/6/0)	432
mushroom	22 (0/0/22)	5644	phoneme	5 (5/0/0)	5404
pima	8 (8/0/0)	768	saheart	9 (5/3/1)	462
sonar	3 (60/0/0)	208	spambase	57 (57/0/0)	4597
spectfheart	44 (0/44/0)	267	tic-tac-toe	9 (0/0/9)	958
titanic	3 (3/0/0)	2201	wdbc	30 (30/0/0)	569
wisconsin	9 (0/9/0)	683			

- Nearest Neighbor, with k=1 (1NN) and with k=3 (3NN) [12].
- Decision tree (C4.5) [13].
- Fuzzy Unordered Rule Induction Algorithm (FURIA) [14].
- Linear Discriminant Analysis (LDA) [15].
- Partial Decision Tree (PART) [16].
- Positive Definite Fuzzy Classifier (PDFC) [17].
- Repeated Incremental Pruning to Produce Error Reduction (RIPPER) [18].
- Support Vector Machine (SVM) [19].

They were chosen to provide a wide range of classifiers for this study. The parameters used were the default ones present in the KEEL tool [9], which are the ones suggested by the original authors of the methods.

5.3 Results

This section presents a summary of the results obtained by the experiments run following the above framework. Due to space concerns, only a brief summary of the results were included, for a more detailed analysis check http://sci2s.ugr.es/covariate-shift_cross-validation.

Table 4 shows a summary of the results regarding single-experiment reliability. The data in said table represents the percentage of the time a single cross-validation experiment comparing two datasets using a Wilcoxon signed-ranks test obtained the right result, which is understood to be the "true" one as defined in Subsection 4. Remember that the significance level is set to the same the "true" data achieved, and that only comparisons between classifiers that showed a significant difference in performance are considered.

Some interesting conclusions can be extracted by looking at these results:

Table 4: Single cross-validation experiment, average accuracy of the Wilcoxon signed-ranks test

Cross-validation scheme	DOB-SCV	DB-SCV	SCV	MS-SCV
2x5	52.313	44.526	51.250	2.474
5x2	52.500	50.000	39.350	2.227
10x1	49.095	49.810	44.048	1.762
Average	51.303	48.112	44.883	2.154

- Partition-induced covariate shift can significantly hamper the reliability of running a single experiment. MS-SCV produces a much worse accuracy than all other partitioning strategies.
- Randomly distributing the examples of a dataset can sometimes induce covariate shift, as can be deduced from SCV having a lower accuracy than both DB-SCV and DOB-SCV.
- DOB-SCV and DB-SCV obtain practically equivalent results, with a non-significant advantage in favor of DOB-SCV.

6 Concluding remarks

We have presented an experimental analysis on the impact partition-based covariate shift can have on the reliability of classifier performance through cross-validation.

We have studied four different partitioning methods and shown that, when covariate shift is introduced, single-experiment reliability is diminished.

We have found that cross-validation approaches that try and limit the impact of partition-induced covariate shift are more reliable when running a single experiment, and thus recommend cross-validation users to use DOB-SCV (which slightly outperforms DB-SCV) as the partitioning method in order to avoid covariate-shift related problems.

Acknowledgments

Jose G. Moreno-Torres and José A. Sáez are currently supported by FPU grants from the Ministerio de Educación y Ciencia of the Spanish Government. This work was supported in part by projects TIN2011-28488 and P10-TIC-6858.

References

1. Kohavi, R.: A study of Cross-Validation and bootstrap for accuracy estimation and model selection. In: International Joint Conferences on Artificial Intelligence. (1995) 1137–1145
2. Stone, M.: Asymptotics For and Against Cross-Validation. Biometrika **64**(1) (1977) 29–35

3. Efron, B., Gong, G.: A leisurely look at the bootstrap, the jackknife, and Cross-Validation. The American Statistician **37**(1) (1983) 36–48
4. Quiñonero Candela, J., Sugiyama, M., Schwaighofer, A., Lawrence, N.D.: Dataset Shift in Machine Learning. The MIT Press (2009)
5. Moreno-Torres, J.G., Raeder, T., Aláiz-Rodríguez, R., Chawla, N.V., Herrera, F.: A unifying view on dataset shift in classification. Pattern Recognition **45**(1) (2012) 521 – 530
6. Cano, J.R., Herrera, F., Lozano, M.: Evolutionary stratified training set selection for extracting classification rules with trade off precision-interpretability. Data & Knowledge Engineering **60** (January 2007) 90–108
7. Zeng, X., Martinez, T.R.: Distribution-balanced stratified cross-validation for accuracy estimation. Journal of Experimental & Theoretical Artificiall Intelligence **12**(1) (2000) 1–12
8. García, S., Fernández, A., Luengo, J., Herrera, F.: Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Information Sciences **180**(10) (2010) 2044–2064
9. Alcalá-Fdez, J., Sánchez, L., García, S., Jesus, M.J.D., Ventura, S., Garrell, J.M., Otero, J., Bacardit, J., Rivas, V.M., Fernández, J.C., Herrera, F.: Keel: A software tool to assess evolutionary algorithms for data mining problems. Soft Computing - A Fusion of Foundations, Methodologies and Applications **13**(3) (2009) 307–318
10. Huang, J., Ling, C.X.: Using auc and accuracy in evaluating learning algorithms. IEEE Transactions on Knowledge and Data Engineering **17** (2005) 299–310
11. Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F.: KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. Journal of Multiple-Valued Logic and Soft Computing, In Press (2010)
12. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Transactions on Information Theory **13** (1967) 21–27
13. Quinlan, J.R.: C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1993)
14. Hühn, J., Hüllermeier, E.: FURIA: an algorithm for unordered fuzzy rule induction. Data Mining and Knowledge Discovery **19**(3) (2009) 293–319
15. Fisher, R.: The use of multiple measurements in taxonomic problems. Annals of Eugenics **7** (1936) 179–188
16. Frank, E., Witten, I.: Generating accurate rule sets without global optimization. In: Proceedings of the Fifteenth International Conference on Machine Learning. (1998) 144–151
17. Chen, Y., Wang, J.: Support vector learning for fuzzy rule-based classification systems. IEEE Transactions on Fuzzy Systems **11**(6) (2003) 716–728
18. Cohen, W.: Fast effective rule induction. In: Machine Learning: Proceedings of the Twelfth International Conference. (1995) 1–10
19. Cortes, C., Vapnik, V.: Support vector networks. Machine Learning **20** (1995) 273–297

Time Series Database Characterization and Similarity Measure Selection

Usue Mori, Alexander Mendiburu and Jose A. Lozano

Intelligent Systems Group, University of the Basque Country UPV/EHU,
Manuel de Lardizabal 1, 20018 San Sebastián, Spain
usue.mori@.ehu.es, alexander.mendiburu@ehu.es, ja.lozano@ehu.es

Abstract. In the past few years, mining time series data has become one of the focal points of research in data mining due mainly to the increasing availability of this type of data and its special characteristics. Mining temporal data frequently requires the definition of a distance measure between different time series, and the inability of common distance measures to correctly capture the similarity between temporal sequences has triggered the creation of many distance estimation methods for this specific type of data. However, the election of a suitable similarity measure for a given time series database is not trivial and no methodology has been published that will support this decision. For this reason, in this paper, a method for the automatic selection of the most adequate distance measure for each time series database will be presented.

Keywords: Time Series, Distance measures, Data Mining

1 Introduction

In recent years, the increase in data collecting technologies and sensors has enabled access to a large amount of temporal data, also denominated time series. This has generated a new type of database where each instance or data point consists of an entire time series. The main characteristics of this type of data are its high dimensionality, its dynamism, its auto-correlation and its noisy nature, all which complicate the study and pattern extraction to a large extent. In view of this, many researchers have focused on finding new methods and on adapting the existing data mining algorithms to obtain useful information from these databases and, so, tasks such as classification, clustering or segmentation have been extended and modified successfully [1].

Many of these tasks require the definition of a distance measure which will estimate the level of similarity or dissimilarity between time series. However, Euclidean distance and other common measures used for regular data are not always the most adequate means to evaluate the similarity between time series. Because of this, the scientific community has focused on elaborating new and more suitable measures for this specific type of data and the fruit of this work, a vast portfolio of distance measures have been published.

Nevertheless, experiments carried out in recent years show that not all of these distance measures are appropriate for all time series databases. It has been suggested previously in [2], that this is probably due to the specific characteristics of each database that make some distance measures more suitable than others on each occasion. However, the choice of a similarity measure for a specific database is not a trivial issue because it is necessary to find a correlation between the characteristics of the time series databases and the properties of the different distance measures. To the extent of our knowledge, this issue has not yet been tackled in the literature and consequently, the first goal of this paper is to identify and quantify a set of relevant features that describe each time series database and will support the choice of one distance measure over another.

Furthermore, making use of these features, the second objective of this paper is to create an automatic distance measure selection method for time series databases. This will be done by training a classifier that, given a specific time series database, will choose the most suitable between two different measures.

The rest of the paper is organized as follows. In Section 2 a set of distance measures for time series and their main characteristics will be presented. In Section 3, the set of novel features that will be used to describe the time series database will be presented. Section 4 introduces our classification framework and some experiment are completed in Section 5. Finally, in Section 6, the main conclusions will be extracted and some insights on future work will be proposed.

2 Similarity Measures

In this paper, five of the most promising or commonly used similarity measures from the literature have been selected for study and comparison.

To begin with, **Euclidean distance (ED)**, is one of the most common measures applied in data mining and consequently, many researchers have directly extended its use to time series. Although ED has been widely used in many application fields, many researchers have pointed out that it is not an adequate measure of time series similarity. First, it is only able to deal with series of equal length. Moreover, it is based on the comparison between points collected at the same time interval and this is a big inconvenience because in many cases time series may be shifted or warped while still maintaining a similar shape. Finally, ED is highly susceptible to noise and outliers [2].

In order to deal with these inconveniences, many similarity measures have been specifically designed for time series data and among them the most common is **Dynamic Time Warping (DTW)**. This distance measure permits warping, shifting, stretching and compressing in order to find the best alignment between two series. This optimal alignment and the resulting DTW distance are calculated using dynamic programming and searching for the minimal path in a distance matrix that defines a mapping between the two series [2, 3].

Apart from DTW, some other distances, which have proven to be effective when working with time series, are measures based on the Edit distance. This measure was initially presented to calculate the similarity between two sequences

of strings but, in the past few years, several adaptations for real sequences have been published. All these measures are based on the idea of counting the minimum number of edit operations (delete, insert and substitute) that are necessary to transform one sequence into the other. Similar to DTW, they are able to deal with shifting and warping in the time axis but, furthermore, by using the delete operation, these distances allow gaps or regions in the sequence that do not match with any other point. This is a big advantage when working with noisy data obtained from real environments [2].

The first edit based distance that will be presented is **Longest Common Subsequence distance (LCSS)** [4]. In this case, the similarity between two time series or real vectors is quantified in terms of the longest common subsequence. However, the fact that gaps or unmatched regions are permitted and do not influence the length of the common sub-sequence must be taken into account. The problem with working with real numbers is that it is difficult to find exact matching points in two different sequences and therefore common sub-sequences will be very short or even non-existent, even if the time series are similar. The solution to this issue is to convert the distance between the points in the time series into 0 or 1. If two points x_i and y_j are closer to each other in the absolute sense than a user specified ϵ, they will be considered equal. On the contrary, if they are farther apart, they will be considered different from each other and the distance between them will be considered 1.

The second edit based distance is **Edit Distance on Real Sequences (EDR)** [5]. This measure is similar to LCSS because it also reduces the distance between two points in the time series to 0 or 1. However, the main difference with LCSS is that EDR penalizes the presence of gaps or unmatched regions with a value proportional to their length. By easing the matching condition, LCSS and EDR are more capable of dealing with noisy data or outliers than other distance measures, because the real values of the points are not taken into account.

The third adaptation to the edit distance is **Edit Distance with Real Penalty (EDR)** [6] which is a combination of DTW and EDR. In this case, given two time series, the optimal alignment between the series is sought as in DTW but, furthermore, this measure incorporates the virtues of EDR by introducing a user defined reference point g that will be used to permit and penalize gaps. If two points are matched, the Euclidean distance between them is added to the total distance, and on the contrary, if a gap is introduced, the euclidean distance between the unmatched point and g is added to the total distance. Taking this into account, the optimal configuration of matching points and gaps will be found by using dynamic programming.

3 Characterization of Time Series Databases

In this section we will introduce a novel set of features that will be used to describe time series databases and that will be relevant for choosing a suitable distance measure. It is important to note that the characterization is not carried out for a single time series but for a whole database of series.

3.1 Dimension of the Database

The dimension of the database is defined in terms of the number of time series that it contains and the average length of these sequences. It has been proven in the past years that these two features have an impact on the accuracy of some similarity measures. On the one hand, the accuracy of rigid distance measures such as ED improves as the number of time series in the database increases [2]. On the other hand, not all distance measures give good results when working with very long time series and therefore the average length of the time series in the database is also an issue that should be taken into account [3].

3.2 Warp

It is very common that time series in a database are shifted and warped in the time axis as can be seen in Figure 1. Visually, anyone would confirm that the two time series in the figure are very similar to each other but not all distance measures are able to deal with this type of similarity. Because of this, quantifying the level of "warp" in a time series database can be highly relevant.

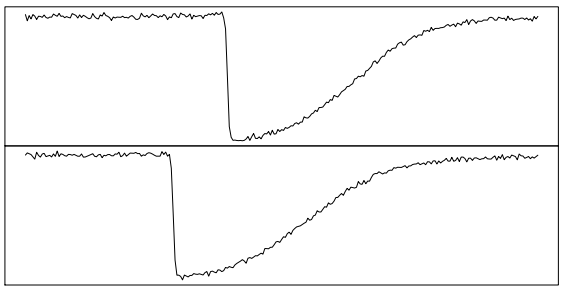


Fig. 1. Warped time series

Given a set of time series $\{X_1, X_2, ...X_n\}$, the solution we propose is based on calculating the cross-correlation of each pair of time series in the database and selecting the time-lag ($lag^*(X_i, X_j)$) in which this correlation is the highest ($CC^*(X_i, X_j)$). This time-lag value, which describes how shifted the two series are from each other, will be used to characterize the warp between each pair of series. Then, in order to give a unique value for the whole database, we will calculate a weighted mean of these lags using the maximum cross-correlation values as coefficients:

$$\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{i-1} CC^*(X_i, X_j) \cdot lag^*(X_i, X_j)$$

In this manner, the pairs of series that are dissimilar at all lags and have a low correlation level will not contribute much to the total warp level and contrarily, the series that have a high correlation will account for most part of the total warp value.

3.3 Trend and Seasonality

It is well known that trend and seasonality are basic characteristics of time series and therefore it may be interesting to take them into account. The methodology used in this paper to quantify these characteristics into a unique number is based on the framework presented in [7].

The first step is to find out whether the time series contain a seasonal component or not. In many cases, when the collection interval of the data is known, we might have some prior information about the seasonal component of the series but in some cases the frequency of the series is unknown and must be found beforehand. This is done by using the spectral density function and finding the frequency which has the greatest contribution to the variance of the series.

If this maximum frequency exists and is bigger than a user specified threshold (10, in this case), then the series has a seasonal component in that frequency and the series will be decomposed into three parts using the STL descomposition method [8]: seasonal part, trend and residuals. If no such frequency is found, the seasonal component of the series will be null and only the trend will have to be extracted using a penalized regression spline. Once this is done, the quantification will be carried out using the variances of each component:

$$\text{Trend level} = 1 - \frac{Var(X_t - S_t - T_t)}{Var(X_t - S_t)},$$

$$\text{Seasonality level} = 1 - \frac{Var(X_t - S_t - T_t)}{Var(X_t - T_t)},$$

where X_t is the true time series and S_t and T_t are the seasonal and trend component of the series respectively. Finally, to characterize a whole database and not a single time series, after calculating these values for all the series in the database, the mean, maximum value and standard deviation will be chosen.

3.4 Noise

The amount of noise in the time series database may be relevant when choosing one distance over another, because some similarity distances have a more robust behavior with this type of data [3]. The quantification we propose consists in applying a moving average filter to the series, with a window equal in size to 10% of their length, in order to remove the noise from each of them. This window size has been selected after experimenting with several values and trying to find a balance between eliminating local fluctuations and over-smoothing the series. Next, calculating the mean and standard deviation of the residual time series in each case will result in a vector of means and a vector of standard deviations. Finally, in order to give an overall solution for the whole database, the mean, maximum and standard deviation of these vectors will be saved.

3.5 Outliers

Outliers, or data points that deviate notably from the rest of the data, greatly affect some similarity measures [3]. The detection of outliers in time series must

be performed after the series is detrended and deseasonalized, which will be carried out using the methodology explained in previous Section 3.3. Once a stationary time series is obtained, the proportion of outliers is calculated using a box-plot method and by choosing the data points that lay outside the whiskers of the plot. The outlier level of the whole database is finally quantified by the mean, maximum and standard deviation as in previous cases.

3.6 Auto-Correlation

The level of auto or self-correlation of a single series is calculated as proposed in [7], where a modification of the Box-Pierce Statistic is used:

$$\frac{1}{h}\sum_{i=1}^{h} {r_k}^2$$

where r_k corresponds to the auto-correlation coefficient with lag k, and h is the maximum lag that is considered, which is usually 20. Once this value has been calculated for all the series in the database, we will calculate their mean, maximum value and standard deviation.

3.7 Skewness and Kurtosis

These two characteristics are related to the shape of the probability distribution of the data. Skewness is a measure of the asymmetry of the data around the mean value and, on the contrary, kurtosis is a measure of how peaked or flat the probability distribution is. As in other cases, the mean, maximum value and standard deviation are used to characterize the whole data set.

4 Automatic Selection of the Most Adequate Similarity Measure

The objective of this section is to use the features proposed in Section 3 to train a classifier that will be used to automatically select a suitable distance measure for each time series database. For this purpose, the features introduced in Section 3 will have to be calculated for each time series database. This will result in a new data set where each sample will correspond to the meta-characteristics of a complete time series database.

In order to complete the data set, the class variable must be defined and in accordance with the objective of this section, it will represent the most adequate distance measure for each time series database. However, since the number of available time series databases is not very large, the comparison of distances will be performed pairwise and consequently, a different classifier will be trained for each pair of distances. This being so, the most adequate measure between a pair will be chosen for each time series database to completely define the value of the class variable.

Nevertheless, choosing the most adequate distance between a pair is not trivial, and different methodologies can be applied for this election [9]. A first methodology is the visual approach, which consists in choosing the most suitable similarity measure by plotting the different time series in the database and visually assessing and evaluating the results obtained by each measure. A more objective and automatic methodology is used in [2] but it requires some a priori knowledge about the true similarity between the series. More specifically, in order to apply this methodology, the series in the database must be separated into clusters based on their true similarity, which can be assessed visually or by comparing the generating processes of the series. If this information is available, the methodology consists in using a 1NN classifier and predicting the cluster to which a set of time series belong. It has been proven that the accuracy of the 1NN classifier depends mostly on the definition of the distance between data points and, therefore, the goodness of different similarity measures can be evaluated by directly using the prediction accuracy of this classifier.

Once the class variable has been defined, the data set will be complete and, finally, the training of the classifiers will be carried out. The methodology used to build the classifiers and obtain a set of predictive models is logistic regression.

4.1 Feature Selection

It is possible that not all the characteristics presented in Section 3 are relevant for discriminating between every pair of distances. Because of this, variable selection has been carried out separately for each classifier by using two different methodologies. The objective is to choose the most relevant variables for each pair of distance measures and to discard any irrelevant information.

The first is Correlation based Feature Selection (CFS) [10], which is a variable selection method based on finding variables that are highly correlated with the class variable but uncorrelated with each other. In the case of discrete class variables, the correlations are calculated using entropy, and therefore the continuous variables must be discretized in advance. The discretization method applied to our data set is equal frequency binning with 3 bins.

The second feature selection method is based on mutual information (MI). First, the MI-s between the input variables and the true class variable have been calculated (MI_T). Second, the class variable has been randomly assigned 1000 times, always respecting the proportions present in the true class variable, and the mean of the mutual information values between the input variables and this random class variable (MI_R) have been calculated. The selected variables are those that fulfill the following condition:

$$\frac{MI_T - MI_R}{MI_R} > \delta$$

where δ is a user specified threshold set. In our case, we have set it to $0,5$ after some preliminary experiments.

5 Experiments and Conclusions

The experiments have been carried out using 36 databases from the UCR time series database archive [11]. For each one, the features described in Section 3 have been calculated and, so, a data set with 36 samples has been obtained.

Moreover, in each of the databases available in the UCR archive, the time series are further grouped into several clusters. The series that correspond to the same group are similar to each other and different from the series belonging to other groups. This prior specification of the similarity between the series enables the use of the second methodology, introduced in Section 4, to define the class variable. Indeed, in [2], an extensive comparative study has been carried out using this second methodology with the objective of evaluating different distance measures. The main procedure of this work consists in training a 1NN classifier for each distance measure and each time series database from the UCR archive and then evaluating these classifiers using the similarity group labels. The errors obtained from this study will be directly applied in our paper and, for each pair of distances d_1 and d_2 and each time series database, the measure with the minimum error rate will be chosen to define the class variable.

Once this is done, a classifier has been trained for each pair of distances using all the features and also applying CFS feature selection and MI based feature selection. Given the small size of the data set, the evaluation of the classifiers has been carried out using a leave one out cross validation scheme. Furthermore, in many cases, when comparing two distances, the class variable is unbalanced and therefore the common accuracy measures usually overestimate the quality of the classifiers. Because of this, Matthews Correlation Coefficient (MCC) has been used to assess the classification power of the introduced models. This measure uses the true positive, true negative, false positive and false negative values to quantify the quality of binary classifiers and it gives a value between -1 and 1. A value of 1 corresponds to perfect classification and a value of 0 corresponds to random classification. The results of this evaluation are presented in Table 1:

Distance Measures	**All Features**	**CFS Feature Selection**	**MI Feature Selection**
ED & DTW	0,04	0,13	0,26
ED & LCSS	0,16	0,14	0,37
ED & EDR	-0,08	0,47	0,35
ED & ERP	0,21	0,13	0,17
DTW & LCSS	0,00	0,33	0,22
DTW & EDR	-0,18	0,52	0,30
DTW & ERP	0,089	0,06	0,06
LCSS & EDR	-0,32	0,15	0,20
LCSS & ERP	-0,17	0,17	0,17
EDR & ERP	0,10	0,33	0,19

Table 1. Matthews Correlation Coefficient values of the Pairwise Classifiers using Different Sets of Input Features

As can be seen, in most cases, the worst accuracy results are obtained when feature selection methods are not applied. This is an expected result, because not all the features are relevant for discriminating between every pair of distances. Furthermore, the number of samples in the data set is very small and therefore the classifiers are not capable of dealing with a very large number of features. On the contrary, in the cases where feature selection has been applied, all the MCC are larger than 0, which indicates that all the classifiers are better than the random classifier. Although the results are not outstanding, given the small size of the database, they prove that the presented methodology is promising and should be further elaborated and pursued. Furthermore, the lowest values are obtained when comparing DTW with ERP, which is logical, since ERP is based on DTW and therefore these two distances have very similar characteristics.

Finally, it is important to note that the feature selection methods choose the most relevant features in order to discriminate the class variable. In this manner, a study of the set of selected variables by these procedures proves the relevancy of the characteristics defined in Section 3, because the properties of the different distance measures are clearly reflected. For example, when comparing LCSS or EDR with ERP or DTW, most of the variables related to noise and outliers have been chosen by the feature selection methods, which is what was expected beforehand. Similarly, when choosing between flexible measures and ED, the variable that quantifies shifting in the time axis frequently appears along with variables related to noise and outliers, which is also an expected result. Other variables such as those related to trend and skewness also appear when comparing several pairs of distances which may be related with the ability of the distance measures to deal with time series that vary a lot.

6 Future Work

Time series data mining is a topic that has captured the interest of a wide section of the scientific community in the past few years. However, many time series mining tasks are strongly influenced by the distance measure used to characterize the similarity between time series. The large number of available distances, together with the little information available concerning their strengths and weaknesses, makes it difficult to choose an adequate distance in each case.

Although some recommendations are present in the literature that propose certain distance measures for certain cases [3], there is no complete guide or system that will decide which distance is the most adequate given a time series database. Because of this, in this work, a promising method that can be used as a baseline to automatize this process has been presented. This novel decision support system will choose the best distance measure between a pair, based on a classifier and an innovative set of characteristics of time series databases.

As we can see, only five similarity measures have been studied, mostly due to the reduced number of available time series databases. Nevertheless, the objective is to further extend this study in order to include other state-of-the-art distances that may improve the results of these measures. Furthermore, the comparison of

more than two distances at the same time is another future research direction but the acquisition of new time series databases will be required for this purpose.

Furthermore, although the characteristics used to define the time series data bases have proven to be relevant, they should be refined and improved so as to obtain new representations that may describe the data sets more accurately and might provide a better discrimination between classes.

7 Aknowledgements

This work has been partially supported by the Saiotek and IT-609-13 programs (Basque Government), TIN2010-14931 (Spanish Ministry of Science and Innovation), COMBIOMED network in computational biomedicine (Carlos III Health Institute), 2011-CIEN-000060-01 (Diputación Foral de Gipuzkoa) and CRC-Biomarkers project (6-12-TK-2011-014, Diputación Foral de Bizkaia). Usue Mori holds a grant from the University of the Basque Country UPV/EHU. Thanks to Prof. Keogh for providing the data used in the study.

References

1. Fu, T.C.: A review on time series data mining. Engineering Applications of Artificial Intelligence **24**(1) (February 2011) 164–181
2. Wang, X., Mueen, A., Ding, H., Trajcevski, G., Scheuermann, P., Keogh, E.: Experimental comparison of representation methods and distance measures for time series data. Data Mining and Knowledge Discovery **26**(2) (February 2012) 275–309
3. Esling, P., Agon, C.: Time-series data mining. ACM Computing Surveys **45**(1) (November 2012) 1–34
4. Vlachos, M., Kollios, G., Gunopulos, D.: Discovering similar multidimensional trajectories. In: Proceedings 18th International Conference on Data Engineering, IEEE Comput. Soc (2002) 673–684
5. Chen, L., Ozsu, M.T., Oria, V.: Robust and Fast Similarity Search for Moving Object Trajectories. In: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data. (2005) 491–502
6. Chen, L., Ng, R.: On The Marriage of Lp-norms and Edit Distance. In: Proceedings of the Thirtieth International Conference on Very Large Data Bases. (2004) 792–803
7. Wang, X., Smith, K., Hyndman, R.: Characteristic-Based Clustering for Time Series Data. Data Mining and Knowledge Discovery **13**(3) (May 2006) 335–364
8. Cleveland, R.B., Cleveland, W.S., McRae, J.E., Terpenning, I.: STL: A Seasonal-Trend Decomposition Procedure Based on Loess. Journal of Official Statistics **6**(1) (1990) 3–73
9. Keogh, E., Kasetty, S.: On the Need for Time Series Data Mining Benchmarks : A Survey and Empirical Demonstration. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (2002) 102–111
10. Hall, M.A.: Correlation-based Feature Selection for Machine Learning. PhD thesis, University of Waikato (1999)
11. Keogh, E., Zhu, Q., Hu, B., Y., H., Xi, X., Wei, L. Ratanamahatana, C.A.: The UCR Time Series Classification/Clustering Homepage (2011)

Large scale part-to-part video matching by a likelihood function using featured based video representation

Silvia Espinosa[1], Joaquín Ordieres-Meré[1], and Antonio Bello-García[2]

[1] PMQ research group, Escuela Técnica Superior de Ingenieros Industriales, Universidad Politécnica de Madrid, 28006 Madrid, Spain.
j.ordieres@upm.es
[2] Departamento de Construcción e Ingeniería de Fabricación, Universidad de Oviedo, 33203 Gijón, Spain

Abstract. A part-to-part video matching algorithm, based on the cross correlation of the similarity-invariantly transformed intensity of motion curves is proposed, including a likelihood estimation for video similarity. Applications are growing as the use of videos for communications is increasing. This algorithm works with featured based video descriptors instead of using videos themselves. Selected features for video representation were the motion curve as well as several hashes for specific key frames. This configuration makes the algorithm suitable for automatic searching in large scale video collections.

Keywords: video retrieval, pattern recognition, motion, distortion, hashing, data mining.

1 Introduction

Video databases have become more and more important in our lives. Decreasing prices in storage devices and the availability of higher communications bandwidth have made it possible to store very large video archives, but in order to maintain such amount of information organized, tools are needed.

There is a great number of tools that manipulate video information under human supervision for storage, playing and editing purposes. However, organization tools are needed allowing to maintain the information efficiently. One of the most common techniques is the video indexing. The main objective of video indexing techniques is to get key frames from video streams in order to create video indexes. One of the most used techniques in video indexing is scene change detection. This technique allows to split the video into structural elements like shots, scenes, etc. By using these techniques, it is possible to enable the following search processes to determine whether a video or a part of it is already contained in other video or on a database. Applications like originality recognition, video database cleansing, video tagging, video result re-ranking and cross-modal divergence detection are in direct relationship with such processes.

Distortion	Parameters	Values
blur	R [-]	1, 5, 10, 15, 20
	σ [-]	1, 2, 3, 4, 5, 6, 7, 8
resize	Square image size [pixels]	32, 64, 128, 256, 512
rotate	Rotation [degrees]	10° to 180°, each 10°

Table 1. Distortion parameter values applied.

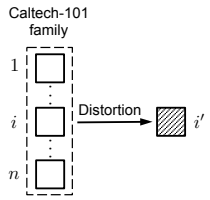


Fig. 1. Notation.

Much research effort have been made to near duplicate video retrieval [12–14, 17, 19]. Nevertheless, as the video editing software evolves and becomes accessible to the general public, the number of videos created by concatenation of video fragments is increasing and part-to-part video matching has not been well addressed yet.

Videos are composed of images, usually taken at a rate of 24 frames per second. Since motion in the shot is usually minuscule, videos can be strongly compressed temporally by retaining only distinct visual appearances [14]. The similarity between video clips is typically calculated by comparing their key-frames. The most popular algorithms for key-frame selection are based on shot boundary detection [15] or time sampling, that can be uniform or based on the extrema of the intensity of motion [13, 14]. Once the key-frames have been extracted, they are represented by their visual features, such as local points and color histogram, etc. Finally, these features are indexed with methods such as dimensionality reduction [18], tree-structure [3] or hashing [7]. The latter is the most accurate to represent video content [17] and, in particular, perceptual hashing is in addition robust in the sense that little perturbations in the original features slightly affect the result [22]. Although many perceptual hashing methods have been proposed in the literature [5,7,20–22], there is not still a comparison that allows to predict the optimum range of application and how they could be combined to achieve a part-to-part video matching algorithm invariant against distortions such as blur, compression and rotation.

In this research work, we propose a part-to-part video matching algorithm where videos are represented by a set of features and stored into a non relational non SQL database, allowing to deal with large-scale matching problems. We have selected a schema free database, the CouchDB with ElasticSearch as the indexer tool. The set of records per video are stored using REST/Json format and they include one record with parameters related to video container itself, like date, time, video formats, path, etc. A second record stores intensity of motion values and color structure per video frame. Last part of the record-set is a variable number of records, each of them describing one specific video-frame by means of a set of hashes. The most representative hashes [21] were implemented: discrete cosine transformation (`dct`), Marr-Hildreth wavelets (`mw`) and a strip of data -strada- (`str`) [16]. We have combined the image hash distances in order for the matching algorithm to be invariant against blur, compression and rotation distortions.

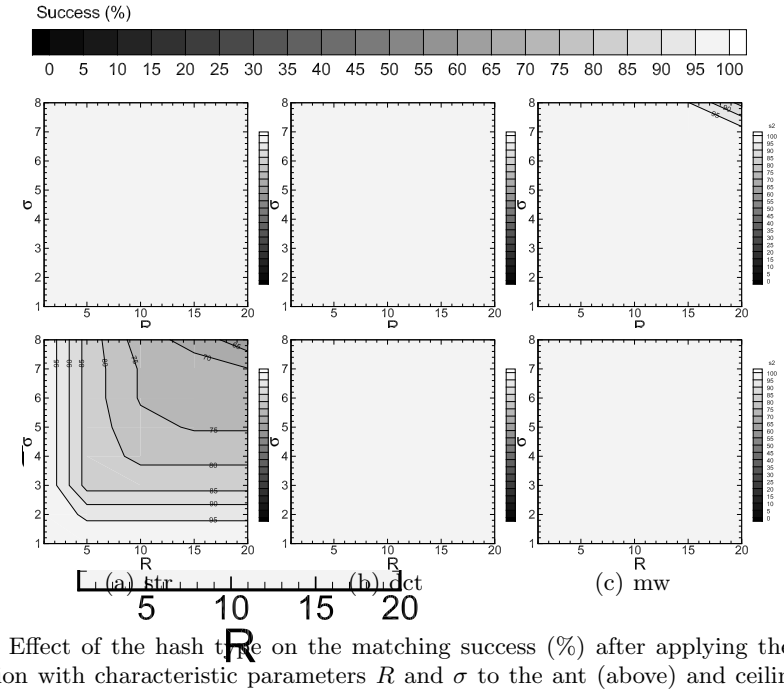


Fig. 2. Effect of the hash type on the matching success (%) after applying the *blur* distortion with characteristic parameters R and σ to the ant (above) and ceiling fan (below) families of the set Caltech-101 [8].

2 Methodology

2.1 Intensity of motion

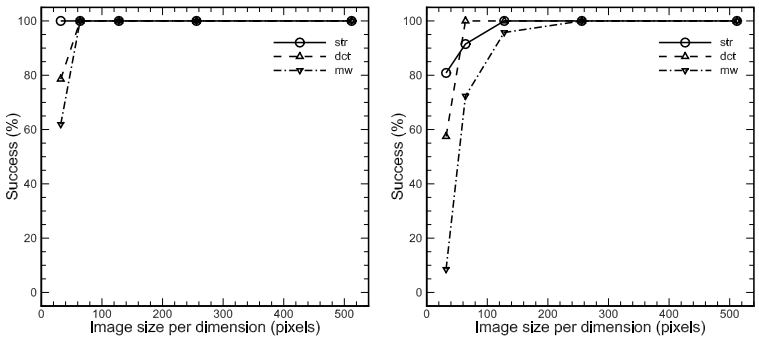
The part-to-part video matching algorithm proposed is based on the cross correlation of the intensity of motion curves. Intensity of motion is defined as the mean of consecutive frame differences [14]:

$$m(t) = \frac{1}{A} \sum_{\mathbf{x}} |L(\mathbf{x}, t+1) - L(\mathbf{x}, t)| \tag{1}$$

where A is the area of the video and $L(\mathbf{x}, t)$ denotes the luminance value of pixel $\mathbf{x}$ of a frame at time t. The motion curves are extracted by outputting intermediate results from the shot detecting tool [1].

2.2 Part-to-part video matching

The state of the art algorithms for 2D part-to-part curve matching are generally $O(N^4)$. For instance, hash-based methods [9,10] can do the comparison in $O(N)$, but they require pre-processing steps that are of $O(N^4)$ asymptotic complexity. Nevertheless, the algorithm proposed in [6], where the input curves can differ by a similarity transform, performs the part-to-part matching in $O(N^3)$. Hence, the latter is utilized for the intensity of motion matching.



(a) Ant (b) Ceiling fan

Fig. 3. Effect of the hash type on the matching success (%) after compressing the images of several families of the set Caltech-101 [8].

First, the absolute curvature $|\kappa(t)|$ of the intensity of motion $\mathbf{s}(t)$ is calculated by using second order central differences for discretizing the second derivative.

$$|\kappa(t)| = \|\ddot{\mathbf{s}}(t)\| \tag{2}$$

Second, the integral of the absolute curvature from an arbitrary start point t_1 on the curve with respect to another fixed point t_2, $K(t_1 : t_2)$, which is invariant under similarity transform [6], is defined as follows:

$$K(t_1 : t_2) = \int_{t_1}^{t_2} |\kappa(t)| \, \mathrm{d}t \tag{3}$$

and discretized in this case by utilizing trapezoidal integration. The curve $t - K$ is sampled at equal intervals along the integral of the absolute curvature K, so as for the interpolation point density to be higher where the absolute curvature gradient magnitude is larger. The mean t is selected when the correspondent t is not unique and linear interpolation is recommended to avoid spurious oscillations. Afterwards, a second interpolation is carried out to obtain the *signed* curvature values that correspond to the previously interpolated points in t [6].

Finally, the curve that should be matched is the *signed curvature*, κ, parametrized by the integral of curvature, K, since is invariant under similarity transform. In order for the matching not to be influenced by the scale, the normalized cross correlation [23] is adopted:

$$\nu(u, \Omega, r, f) = \frac{\sum_{i \in \Omega} \left[r(i) - \bar{r_w} \right] \left[f(i-u) - \bar{f} \right]}{\sqrt{\sum_{i \in \Omega} \left[r(i) - \bar{r_w} \right]^2 \sum_{i \in \Omega} \left[f(i-u) - \bar{f} \right]^2}} \tag{4}$$

where u is the offset of one curve to the other, f is the first curve working as a template window sliding along the second curve r, $\bar{f}$ is the mean value over the whole template and $\bar{r_w}$ is the mean value *of the sliding window* in the second

curve. Nevertheless, in part-to-part matching an infinitesimally small part of one curve will match many parts of another curve, as noted in [6]. To favor longer matches, the following strategies are used:

1. Thresholding on the length of the matching.
2. The score is given by the cross correlation ν, which lies in $[0, 1]$, multiplied by the length of the match L.

After the start points and end points on both curves have been determined, the similarity transform between them can be estimated using the method proposed in [11].

2.3 Similarity between videos

Part-to-part video similarity is determined according to the following algorithm:

1. Determine the biggest common motion segment in both videos as well as their relative position (u_0, Ω_0).
2. Identify the existing image hashes available from those video segments and producing two set of potentially related images ($Img(u_0, \Omega_0, \text{video})$).
3. For each of the existing images into the smaller set, look for the maximum agreement ($1 - distance_{normalized}$) against images into the other set. This will establish the image pairs relationship.
4. Calculate likelihood for video similarity L_{vs}.

The likelihood for similarity between videos is defined as follows:

$$\begin{aligned} L_{vs}(\mathrm{v}_1, \mathrm{v}_2) = \nu(u_0, \Omega_0, Ext(\mathrm{v}_1), Ext(\mathrm{v}_2))) \cdot C_f(u_0, \Omega_0, \mathrm{v}_1, \mathrm{v}_2) \\ \cdot \ (1 - \sum_{i=1}^{p} \frac{Dist(i, Img(u_0, \Omega_0, \mathrm{v}_1), Img(u_0, \Omega_0, \mathrm{v}_2))}{p}) \end{aligned} \tag{5}$$

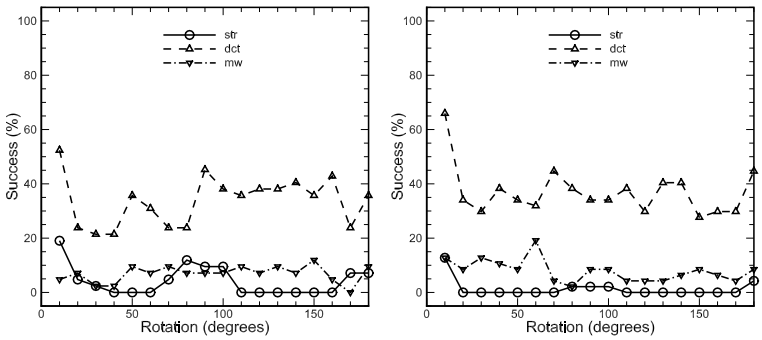
where p is the size of the pairing image relationship, $Ext(.)$ is the function returning the video motion vector for its argument, $Dist(.)$ is the normalized distance function between paired images based on their hashes over the worse distance in these sets [21], $\nu(.)$ was defined at equation 4 and

$$(u_0, \Omega_0) = \arg\max_{u, \Omega} \nu(u, \Omega, Ext(\mathrm{v}_1), Ext(\mathrm{v}_2))) \tag{6}$$

C_f is a function with value of 1 if any inversion in the sequence of relevant images happens for the two videos being examined in the common segment. Its value will be 0 otherwise.

3 Results and discussion

To take care of distance properties against distorsions the ant and ceiling fan families of the Caltech-101 benchmark [8] are distorted utilizing ImageMagick mogrify with the parameter value ranges indicated in Table 1. The library pHash [21]



(a) Ant (b) Ceiling fan

Fig. 4. Effect of the hash type on the matching success (%) after rotating the images of several families of the set Caltech-101 [8].

is utilized for determining the `dct` and `mw` hashes. Using the notation described in Figure 1, two images are considered to be correctly matched if

$$d_{ii'} \leq d_{ij}, \qquad \forall j = 1, \ldots, n, \quad j \neq i \tag{7}$$

where d_{ab} denotes the distance between images a and b. Based on the previous convention, the success of a hashing distance is defined as follows:

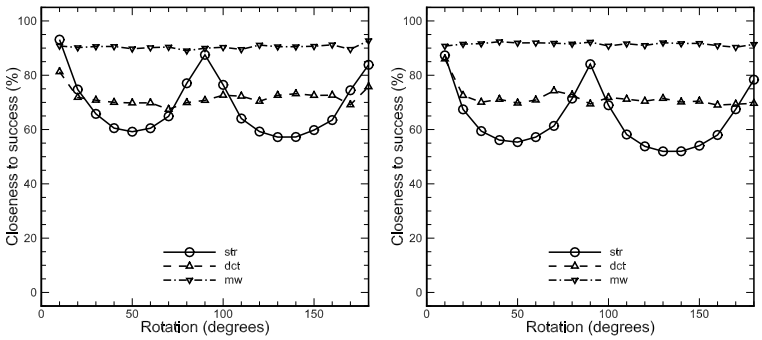
$$\text{Success } (\%) = \frac{n_{(1)}}{n} \cdot 100\% \tag{8}$$

$$\text{Closeness to success } (\%) = \left\langle \frac{\min_{j \in \{1,\ldots,i-1,i',i+1,\ldots,n\}} d_{ij}}{d_{ii'}} \right\rangle_{\text{n}} \cdot 100\% \tag{9}$$

where $n_(1)$ is the number of images that verifies Eq.1 and $\langle\rangle$ denotes average on the images of the family set. In addition, if the distance to the distorted image is 0, the closeness to success is 100% by convention.

In Figure 2, the effect of the hash distance on the matching success after applying the *blur* distortion are presented. As the distortion parameter values are increased, the success decreases. The only hash that is invariant against blur distortion in the range analyzed is the `dct`. Like any Fourier-related transform, `dct` expresses a function or signal in terms of a sum of cosine functions with different frequencies and amplitudes [21]. Low-frequency `dct` coefficients of an image are mostly stable under image manipulations, since most of the signal information tends to be concentrated in a few low-frequency components of the `dct`. This property is also utilized by the JPEG image compression standard [4].

The `mw` hash uses edge detectors for feature extraction. Since the derivative operator acts as a high pass filter, edge detectors based on it are sensitive to noise [4]. While for the ceiling fan family the algorithm is invariant to noise in the whole range under study, for the ant family the complete success is only guaranteed for $\sigma \leq 7$ and $R \leq 15$. Ultimately, the performance of the hash



(a) Ant (b) Ceiling fan

Fig. 5. Effect of the hash type on the closeness to matching success (%) after rotating the images of several families of the set Caltech-101 [8].

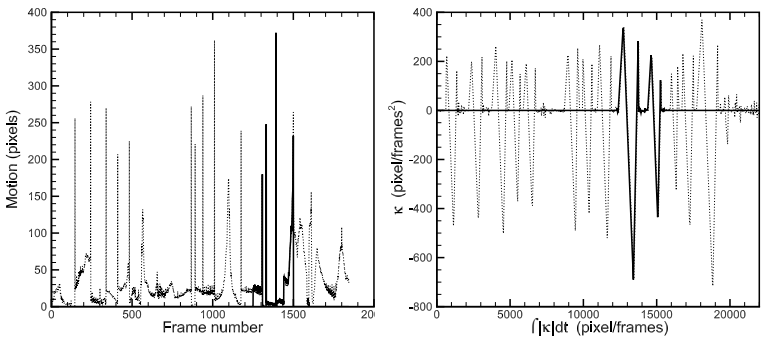
`str` distance function strongly depends on the shape of the object. As this hash compares the luminosity of each pair of pixels, the `str` hash is sensible to the thin details, such as the blades of the ceiling fan, which are highly affected by the blur. As a consequence, while it is invariant against blur distortion with the ant family, with the ceiling fan family a 100% success is only guaranteed when for $\sigma \leq 2$ and $R \leq 5$.

Figure 3 shows the effect of the hash distance on the matching success after applying the *compression* distortion. When compressing the image, the percentage of success remains constant until a critical value is reached. Afterwards, it decreases monotonously. Thus, the hash functions are invariant against blur until reaching a compression of 64x64 and 256x256 pixels^2 for the ant and ceiling fan families respectively. The best overall performance is achieved with the `str` distance function, being invariant against compression distortion for the ant family and with a percentage of success over 80% for the ceiling fan family.

In Figure 4, the effect of the hash distance on the matching success after applying the *rotation* distortion are presented. First, the success significantly decreases in the present of rotation, with a success lower than 70% in all the cases. The `dct` distance is clearly recommendable since the percentage of success achieved is in $(20, 70)$ %, while for the other distances the success is under 20%.

Figure 5 shows the effect of the hash type on the closeness to matching success. The closeness to success of the `dct` distance function is around 70%. Regarding the `str` hash, the best results are obtained around 0°, 90° and 180°, with closeness to success near 90%. However, as the rotation angles move away from these values, the closeness to success decreases up to a 60%. Fortunately, for the Marr-Hildreth hash is around 90%, independently of the rotation angle.

To sum up, in the range under study, the `dct` hash is invariant against blur since the signal information is concentrated in a few low-frequency components of the discrete cosine transformation which are most stable against this distortion. Furthermore, its range of invariance against compression expands until 64x64



(a) Motion (b) Signature

Fig. 6. Motion and curvature part-to-part video matching. The motion of the container video is plotted in dotted thin line; while the fragment, situated in the predicted optimum position, in solid thick line.

pixels^2, independently of the image itself, which would allow to predict robustly the performance with general images. Moreover, the performance of the `dct` hash distance is also the best, with a success up to 70%. Nevertheless, it is highly recommendable to complement this information with the `mw` distance function, with which the image selected by the `dct` hash should be at a distance lower than 1.15 times the `mw` minimum distance.

As a matter of example illustrating the part-to-part video matching, several fragments of the synthetic movie [2] are matched with the original video. A hundred interpolation points are used in the fragment, utilizing the same interpolation point distance in K in the other curve. Moreover, the length of the matching contains at least fifty interpolation points.

Figure 6 illustrates the successful results of part-to-part matching algorithm of the original video with the latter, even in the cases with a intensity of motion range two orders of magnitude smaller and of the order of that of the original video.

4 Conclusions

The part-to-part video matching algorithm proposed, based on the normalized cross correlation of the similarity-invariantly transformed intensity of motion curves, is successful even with a intensity of motion range two orders of magnitude smaller than that of the original video. Major strength for this algorithm is that it never deals with videos themselves but with their extracted features, which are much lighter than videos themselves.

In addition, it is suggested how the most representative hashes and distance functions should be integrated in order for the matching algorithm to be invariant against blur, compression and rotation distortions on shot captured video-frames. The optimum Distance function is the `dct`, together with checking that

the image selected by the `dct` hash should be at a distance lower than 1.15 times the `mw` minimum distance.

Quasi-metric properties were verified for L_{vs} after algorithm implementation. Nevertheless, in the near future extensive tests will be performed to determine singularities and sensitivity to disagreements between metrics and to the shot extracted image densities. Then robustness properties as well as metric properties for the likelihood property will be derived.

Acknowledgments. The authors acknowledge to INTECO organization and the EU DG Home for the funding and support provided throughout the project *ASASEC*: HOME/2010/ ISEC/AG/043.

References

1. (March 2013), `http://johmathe.name/shotdetect.html`
2. (March 2013), `http://www.anc.ed.ac.uk/~amos/movies/afreightc.avi`
3. Böhm, C., Berchtold, S., Keim, D.A.: Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. ACM Comput. Surv. 33(3), 322–373 (Sep 2001), `http://doi.acm.org/10.1145/502807.502809`
4. Bovik, A.C.: The Essential Guide to Image Processing. Academic Press (2009)
5. Bozkaya, T., Bozkaya, T., Ozsoyoglu, M.: Indexing large metric spaces for similarity search queries. ACM Transactions on Database Systems 24, 361–404 (2002)
6. Cui, M., Femiani, J., Hu, J., Wonka, P., Razdan, A.: Curve matching for open 2d curves. Pattern Recogn. Lett. 30(1), 1–10 (Jan 2009), `http://dx.doi.org/10.1016/j.patrec.2008.08.013`
7. Datar, M., Immorlica, N., Indyk, P., Mirrokni, V.S.: Locality-sensitive hashing scheme based on p-stable distributions. In: Proceedings of the twentieth annual symposium on Computational geometry. pp. 253–262. SCG '04, ACM, New York, NY, USA (2004), `http://doi.acm.org/10.1145/997817.997857`
8. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories (2004)
9. Frenkel, M., Frenkel, M., Basri, R.: Curve matching using the fast marching method. IN EMMCVPR pp. 35–51 (2003)
10. Grauman, K., Fergus, R.: Learning binary hash codes for large-scale image search. Cipolla, Roberto (ed.) et al., Machine learning for computer vision. Selected papers based on the presentations at the international computer vision summer school (ICVSS 2012), Sicily, Italy, July 7–15, 2012. Berlin: Springer. Studies in Computational Intelligence 411, 49-87 (2013). (2013)
11. Horn, B.K.P., Horn, B.K.P.: Closed-form solution of absolute orientation using unit quaternions. Journal of the Optical Society of America A 4(4), 629–642 (1987)
12. Huang, Z., Shen, H.T., Shao, J., Cui, B., Zhou, X.: Practical online near-duplicate subsequence detection for continuous video streams. IEEE Transactions on Multimedia 12(5), 386–398 (2010)
13. Joly, A., Frélicot, C., Buisson, O.: Robust content-based video copy identification in a large reference database. In: Proceedings of the 2nd international conference on Image and video retrieval. pp. 414–424. CIVR'03, Springer-Verlag, Berlin, Heidelberg (2003), `http://dl.acm.org/citation.cfm?id=1760167.1760219`

14. Karpenko, A., Aarabi, P.: Tiny videos: A large data set for nonparametric video retrieval and frame classification. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(3), 618–630 (2011)
15. Lienhart, R.W.: Comparison of automatic shot boundary detection algorithms pp. 290–301 (1998), +http://dx.doi.org/10.1117/12.333848
16. Schwartz, R.: Finding similar images. Linux Magazine pp. 1 – 6 (2003), http://www.linux-mag.com/id/1437/
17. Song, J., Yang, Y., Huang, Z., Shen, H.T., Hong, R.: Multiple feature hashing for real-time large scale near-duplicate video retrieval. In: Proceedings of the 19th ACM international conference on Multimedia. pp. 423–432. MM '11, ACM, New York, NY, USA (2011), http://doi.acm.org/10.1145/2072298.2072354
18. Wichterich, M., Assent, I., Kranen, P., Seidl, T.: Efficient emd-based similarity search in multimedia databases via flexible dimensionality reduction. In: SIGMOD Conference. pp. 199–212 (2008)
19. Wu, X., Hauptmann, A.G., Ngo, C.W.: Practical elimination of near-duplicates from web video search. In: Proceedings of the 15th international conference on Multimedia. pp. 218–227. MULTIMEDIA '07, ACM, New York, NY, USA (2007), http://doi.acm.org/10.1145/1291233.1291280
20. Wu, X., Zhao, W.L., Ngo, C.W.: Near-duplicate keyframe retrieval with visual keywords and semantic context. In: Proceedings of the 6th ACM international conference on Image and video retrieval. pp. 162–169. CIVR '07, ACM, New York, NY, USA (2007), http://doi.acm.org/10.1145/1282280.1282309
21. Zauner, C.: Implementation and Benchmarking of Perceptual Image Hash Functions. Master's thesis, Upper Austria University of Applied Sciences, Hagenberg Campus (Jul 2010)
22. Zhou, X., Schmucker, M., Brown, C.: Perceptual hashing of video content based on differential block similarity. In: Hao, Y., Liu, J., Wang, Y.P., Cheung, Y.m., Yin, H., Jiao, L., Ma, J., Jiao, Y.C. (eds.) Computational Intelligence and Security, Lecture Notes in Computer Science, vol. 3802, pp. 80–85. Springer Berlin Heidelberg (2005), http://dx.doi.org/10.1007/11596981_12
23. Zitová, B., Flusser, J.: Image registration methods: a survey. Image and Vision Computing 21(11), 977–1000 (Oct 2003), http://dx.doi.org/10.1016/s0262-8856(03)00137-9

Selección de características en la predicción a medio plazo del precio del aceite de oliva virgen extra

Antonio J. Rivera[1], Pedro Pérez-Recuerda[1] M. D. Pérez-Godoy[1], M. J. del Jesus[1], M. P. Frías[2], and M. Parras[3]

[1] Dept. of Computer Science, University of Jaén
arivera@ujaen.es, pedro.perez@ujaen.es, lperez@ujaen.es, mjjesus@ujaen.es
[2] Dept. of Statistics and Operation Research, University of Jaén
mpfrias@ujaen.es
[3] Dept. of Marketing, University of Jaén
mparras@ujaen.es

Abstract. Uno de los sectores que más demanda la tarea de predicción de series temporales es el sector económico-empresarial. Un ejemplo lo podemos encontrar en la industria de producción del aceite de oliva, con un peso estratégico dentro de la economía española, y para la que es muy importante poder predecir el precio en origen del producto obtenido. La predicción de este índice se va a ver influenciada por diversas variables exógenas e indicadores técnicos, por lo que se hace necesaria la utilización de algoritmos de selección de características que ayuden a determinar cuales son las variables más influyentes. En este trabajo, se ha realizado la predicción del precio del aceite de oliva virgen extra a seis meses, utilizando CO^{2}RBFN, un algoritmo de diseño de Redes de Funciones de Base Radial y otros métodos *soft-computing*. La predicción se ha realizado utilizando el conjunto de todas las variables exógenas y sobre el conjunto de variables seleccionado mediante algoritmos de selección de características. La experimentación realizada nos ha permitido detectar un conjunto de variables de influencia en el precio del aceite de oliva virgen extra.

Keywords: Predicción, Precio Aceite de Oliva Virgen Extra, Selección de Características, Algoritmos Evolutivos

1 Introducción

La predicción de series temporales es un problema común a muchos campos tales como economía, hidrología, electricidad, etc. El interés en esta clase de problemas está motivado por diferentes razones entre otras, la necesidad de controlar procesos o de calcular los beneficios económicos obtenidos.

Actualmente, el aceite de oliva es un sector empresarial con un mercado en expansión. Según el Consejo Oleícola Internacional,[4] España es el mayor

[4] http://www.internationaloliveoil.org

productor y exportador de aceite de oliva. En España, el Mercado de Futuros del Aceite de Oliva (MFAO) [5] es una sociedad cuyo objetivo es negociar precios futuros del aceite de oliva y que evidentemente está muy interasada en predecir el precio de este producto.

El problema de predicción de series temporales se ha venido abordando mediante métodos estadísticos y econométricos [1]. Sin embargo, en los últimos años se ha conseguido mejorar los resultados utilizando métodos *soft-computing* [2] [3] [4].

Los agentes en el sector del aceite de oliva, especialmente el Mercado de Futuros del Aceite de Oliva, consideran que es importante una predicción a medio plazo. El objetivo de este trabajo es predecir el precio del aceite de oliva virgen extra a seis meses. Para ello, además del precio se van a considerar 9 variables exógenas y un conjunto de indicadores técnicos. La combinación de ambos provoca que se tenga que trabajar con muchas variables, por lo que se hace necesaria la utilización de algoritmos de selección de características que ayuden a determinar qué variables son la que más influencia tienen sobre el futuro precio del aceite de oliva.

Para la tarea de predicción en sí se ha utilizado CO^{2}RBFN, un algoritmo desarrollado por los autores para el diseño cooperativo-competitivo de Redes de Funciones de Base Radial (RBFN) y que se ha aplicado con éxito a la predicción a corto plazo del precio de aceite pero sin tener en cuenta variables externas al precio del aceite [5]. Tanto CO^{2}RBFN como otros métodos *soft-computing*, típicos en tareas de prediccin de series temporales, se han aplicado antes y después de realizar la selección de características.

El resto del trabajo se organiza de la siguiente forma: en la sección 2 se expondrán generalidades sobre la predicción de series temporales y los métodos de pre-procesamiento. En la sección 3 se describe brevemente CO^{2}RBFN, el método diseado por los autores y aplicado en este caso, junto a otros métodos *soft-computing* a la predicción a medio plazo del aceite de oliva virgen extra. La experimentación realizada y los resultados obtenidos se explican en la secciones 4 y 5 respectivamente. Finalmente en la sección 6 se esbozan las conclusiones del trabajo.

2 Preliminares

2.1 Predicción de series temporales

Las razones que motivan la investigación en series temporales son muy diversas: la necesidad de predecir el comportamiento de una variable en el futuro, el control de un proceso, aumentar los beneficios de la empresa, anticipar caídas o subidas en el mercado, la simulación de fenómenos que no pueden llevarse a la práctica. En general, el objetivo final es siempre aumentar nuestro conocimiento sobre un fenómeno o aspecto de nuestro entorno partiendo de datos pasados y presentes. Por tanto, se pretende extraer las regularidades que se observan

[5] htpp://www.mfao.es

en el comportamiento pasado de la variable, es decir, obtener el mecanismo que la genera, para tener un mejor conocimiento de la misma en el tiempo. Además, bajo el supuesto de que las condiciones estructurales que conforman la serie objeto de estudio permanecen constantes, también se trata de predecir el comportamiento futuro.

Las series pueden tener una periodicidad anual, semestral, trimestral, mensual, etc., según los periodos de tiempo en los que están recogidos los datos que la componen. El tipo de análisis, así como los modelos en los que basemos el estudio, dependerán en gran medida del tipo de preguntas que queramos responder. Cuando las observaciones corresponden a una única variable, el análisis de series temporales suele tener como objetivo construir un modelo para explicar la estructura (descripción) y prever la evolución (predicción) de la variable de interés. Por tanto, en el modo descriptivo podemos utilizar los datos para encontrar patrones de comportamiento de los datos, reglas de hechos y asociaciones entre sus ocurrencias. También se buscan anomalías de comportamiento y en general se emplea para caracterizar los datos. En el modo predictivo, los datos se analizan para descubrir un modelo de comportamiento futuro de los mismos para poder estimar con antelación, posibles valores y tendencias.

Dentro del campo de la minería de datos son muchos los métodos que se utilizan para predicción de series temporales. Así por ejemplo, es bien conocida la capacidad de predicción de métodos tales como Redes Neuronales o Sistemas Basados en Reglas Difusas [6], [7], [8].

2.2 Pre-procesamiento de datos

Los algoritmos de minería de datos pueden encontrar muchos problemas cuando se aplican a bases de datos con un gran número de atributos [9]. Este problema es conocido como "curse of dimensionality" [10] y se puede solventar mediante algoritmos de selección de características que reduzcan la dimensionalidad [11].

El uso de algoritmos de selección de características puede reportar varios beneficios. Por un lado, pueden eliminar las características irrelevantes y su ruido asociado. Por otro lado, al reducir la dimensionalidad se consiguen modelos más sencillos. Finalmente, se reduce el tiempo y la memoria requeridos por los algoritmos de minería de datos.

Dentro del campo de la reducción de dimensionalidad [11], una técnica de selección de características debería seleccionar el mejor subconjunto de características de entrada, eliminando los atributos irrelevantes o redundantes.

Existen dos aproximaciones estándar para la selección de características: métodos *filter* y *wrapper*. En los primeros, se realiza la selección de características y después se aplica el método de minería de datos de forma independiente. En los segundos, el objetivo del algoritmo de minería de datos consiste en obtener una medida de evaluación de un subconjunto de atributos dado.

En este trabajo se va a utilizar la aproximación *filter* dado que es independiente del algoritmo de aprendizaje. Otra razón más, es que esta aproximación es más rápida y se puede aplicar a conjuntos de datos con muchas características.

3 Descripción de CO²RBFN

CO²RBFN [12] es un método evolutivo cooperativo-competitivo para el diseño de RBFNs. En esta propuesta cada individuo de la población representa una función base y la población entera es la responsable de la solución final. Se presenta un entorno en el que los individuos cooperan para alcanzar la solución definitiva, no obstante, también compiten por la supervivencia, dado que si el trabajo de un individuo no es bueno dicho individuo será eliminado. En este entorno de cooperación-competición en el que la solución depende del comportamiento de muchos componentes, el fitness de cada individuo es conocido como asignación de crédito. Para medir la asignación de crédito de un individuo, se proponen tres factores que evalúan el papel de cada RBF en la red. Dichos factores tienen en cuenta la aportación de la RBF a la red, el error cometido por la RBF y la posible interacción de una RBF con otras. El algoritmo usa un Sistema Basado en Reglas Difusas (SBRD) para generar la probabilidad de aplicación de los operadores a las RBFs. Se utiliza un esquema de codificación real. Cada cromosoma representa el centro (con un número de coordenadas igual al número de características de los ejemplos) y el radio de una RBF. Las etapas más importantes del algoritmo se muestran el la Figura (1):

```
Inicialización de la RBFN
Mientras (No Fin) Hacer
  Entrenamiento de la RBFN
  Evaluación de las RBFs
  Aplicación de los operadores
    a las RBFs
  Introducción de nuevas RBFs
Fin_Mientras
```

Fig. 1. Principales pasos de CO²RBFN

4 Experimentación

La Figura 2 muestra la serie temporal del precio en origen del aceite de oliva virgen extra en España, calculado de forma mensual y por toneladas. Dicha serie se ha obtenido en colaboración con Poolred [6], una iniciativa de la Fundación para la Promoción y Desarrollo del Olivar y el Aceite de Oliva, situada en Jaén (España). Concretamente, la serie temporal contiene los datos obtenidos desde el primer mes de 2002 hasta el último mes de 2009.

[6] http://www.oliva.net/poolred/

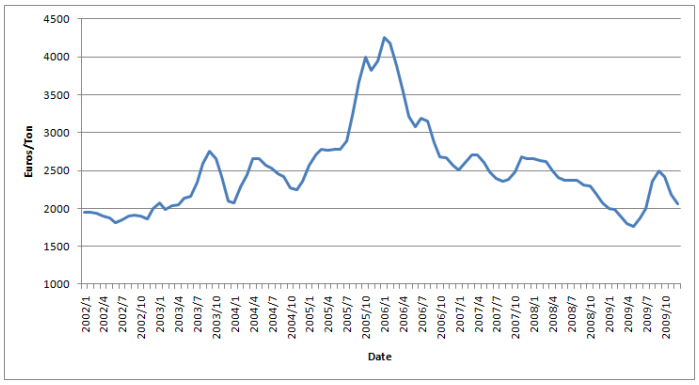


Fig. 2. Serie temporal del precio del aceite de oliva virgen extra

4.1 Variables exógenas e indicadores técnicos

En la Tabla 1 se muestran las variables exógenas elegidas dada su contribución al precio del aceite de oliva virgen extra. Las fuentes de dichas variables son: el citado Poolred, la Agencia para el Aceite de Oliva (AAO) [7] en España, el Instituto Nacional de Estadística de España (INE) [8] y el Ministerio de Industria, Turismo y Comercio (MITC) [9]. Todas las variables son mensuales.

Se han usado un conjunto de indicadores técnicos muy referenciados en la bibliografía especializada [3], de forma que ayuden a extraer información adicional de los datos anteriores.

Dichos indicadores se pueden observar en la Tabla 1, donde i_t es el valor del índice en el momento t, H_{t-k} y L_{t-k} representan el valor más alto y más bajo respectivamente, durante un periodo de tiempo k; H_n y L_n son el valor más alto y más bajo respectivamente desde el inicio de la serie temporal.

Las primeras ejecuciones realizadas tienen en cuenta todas las posibles combinaciones, un total de setenta variables de entrada, considerando las nueve variables exógenas, el precio del aceite de oliva virgen extra y los seis indicadores técnicos además del valor absoluto o bruto de cada variable. Todas las variables y los indicadores técnicos utilizados se pueden observar en la Tabla 4. Todos los datos están normalizados en el intervalo [0, 1].

4.2 Algoritmos utilizados

Para realizar la selección de características [11] se ha utilizado el software de minería de datos, Weka [13]. Como ya se ha mencionado anteriormente se han elegido métodos *filter*.

[7] http://aao.mapa.es
[8] http://www.ine.es
[9] http://www.mityc.es

Table 1. A la izquierda las variables exógenas, a la derecha los indicadores técnicos y sus fórmulas

Variable	Descripción
TgtPrice	Precio en destino del aceite de oliva virgen extra (Poolred)
OpStock	Existencias iniciales de aceite de oliva (AAO)
ClStock	Existencias finales de aceite de oliva (AAO)
InMarket	Mercado interior del aceite de oliva (AAO)
Imports	Importaciones de aceite de oliva (AAO)
Exports	Exportaciones de aceite de oliva (AAO)
ConMK	Consumo de aceite de oliva en millones de kilos (MITC)
GenCPI	Índice de Precios de Consumo (INE)
FoodCPI	Índice de Precios de Consumo de alimentación (INE)

Nombre del indicador	Descripción	Fórmula
Momentum 1	Medida de la variación de un índice en 1 mes	$i_t - i_{t-1}$
Momentum 3	Medida de la variación de un índice en 3 meses	$i_t - i_{t-3}$
Momentum 6	Medida de la variación de un índice en 6 meses	$i_t - i_{t-6}$
Stochastic %k	Medida del último valor de un índice con respecto a su precio durante un periodo dado, $k = 6$ en nuestro caso	$\frac{i_t - L_{t-k}}{H_{t-k} - L_{t-k}} \times 100$
Williams %R	Indicador que mide niveles de sobrecompra/sobreventa	$\frac{H_n - i_t}{H_n - L_n} \times 100$
Disparity6	Disparidad de 6 días. Mide la diferencia entre el precio actual y la media móvil de 6 meses	$\frac{i_t}{MA_6} \times 100$

En Weka para realizar tareas de selección de características se pueden elegir de forma independiente el método de búsqueda, que define el conjunto de atributos y la función evaluadora de las características. En este tabajo se ha elegido CfsSubsetEval [14] como función evaluadora del conjunto de características. Este método basa su evaluación en cálculos de correlaciones entre característica y clase y entre característica y característica.

Con el objetivo de determinar el mejor subconjunto de atributos, se han seleccionado los siguientes métodos: BestFirst [15], GeneticSearch [16], GreedyStepwise [15], LinearForwardSelection [15], ScatterSearch [17], SubsetSizeForwardSelection [15].

CfsSubEval se ha utilizado como método de evaluación para todos los algoritmos de búsqueda, de forma que se han obtenido seis métodos de selección de características.

Para la realizar la predicción se han elegido, además del método CO^{2}RBFN, los siguientes métodos *soft-computing*: FuzzyGAP [18], MLPConjGrad [19], NU-SVR [20] y RBFNLMS [21], .

La implementación de estos métodos se ha obtenido desde KEEL [22] y se han ejecutado con los parámetros recomendados por los autores. Para ejecutar CO^{2}RBFN se han considerado 10 RBFs y 200 generaciones del ciclo principal.

5 Resultados

Los datos de los que se dispone se han particionado tal y como se muestra en la Tabla 2.

Table 2. Conjunto de datos

Datos	Años de entrenamiento	Años de test
Test2006	2002 2003 2004 2005	2006
Test2007	2002 2003 2004 2005 2006	2007
Test2008	2002 2003 2004 2005 2006 2007	2008
Test2009	2002 2003 2004 2005 2006 2007 2008	2009

Para conseguir nuestro objetivo de predecir el precio en origen del aceite de oliva virgen extra (SrcPrice) a seis meses, se han elegido el valor absoluto de SrcPrice, nueve variables exógenas (Tabla 1) y se han obtenido variables de entrada extras procesando cada variable exógena con seis indicadores técnicos (Tabla 1). Los algoritmos *soft-computing* se aplican a este conjunto inicial de datos con setenta variables de entrada.

En la Tabla 3 se pueden observar los resultados obtenidos en cuanto a la media y desviación típica del error MAPE para diez repeticiones. CO^{2}RBFN comete el error más bajo, seguido de RBFNLMS (el otro método de diseño de RBFNs). CO^{2}RBFN también tiene el error con la menor desviación típica, con lo que se puede afirmar que es robusto.

Table 3. Error Mape en test con todas las variables de entrada

Year	CO^{2}RBFN	FuzzyGap	MLPConjGrad	NUSVR	RBFNLMS
Test2006	0.2366 ± 0.0418	0.2085 ± 0.1411	0.4209 ± 0.3329	0.3603 ± 0.3105	0.1953 ± 0.1090
Test2007	0.0630 ± 0.0209	0.2179 ± 0.0990	0.3901 ± 0.2338	0.2294 ± 0.2193	0.1284 ± 0.0949
Test2008	0.0999 ± 0.0194	0.1752 ± 0.0932	0.1524 ± 0.1391	0.1006 ± 0.1115	0.1156 ± 0.0944
Test2009	0.1998 ± 0.0255	0.2245 ± 0.1470	0.2747 ± 0.2254	0.1863 ± 0.0977	0.2539 ± 0.1689
Mean	**0.1498 ± 0.0269**	0.2065 ± 0.1201	0.3095 ± 0.2328	0.2191 ± 0.1847	0.1733 ± 0.1168

Para decrementar el número de variables de entrada y así incrementar la interpretabilidad del problema, se han aplicado algoritmos de selección de características. De esta forma se construye un nuevo conjunto de datos con las variables seleccionadas que se puede observar en la Tabla 4. En dicha tabla, la primera fila muestra los diferentes índices, la primera columna contiene los indicadores técnicos (se usa Absoluto cuando no se ha aplicado un indicador a una variable) y cada celda representa el número de veces que una variable de entrada (definida mediante la combinación de fila-columna) es elegida por algún algoritmo en algún conjunto (año). Por ejemplo, la celda 2,2 (fila 1, columna 2) muestra que la variable Absolute/SrcPrice se elige 12 veces mediante diferentes algoritmos y conjuntos de datos, pero ningún algoritmo de selección elige la variable Momentum1/SrcPrice para ningún año.

De esta forma, se puede observar que existen variables exógenas importantes que son seleccionadas independientemente del indicador técnico. Se puede decir

Table 4. Resultados de aplicar los métodos de selección de características

	SrcPrice	TgetPrice	OpStock	ClStock	InMarket	ConMK	Imports	Exports	GenCPI	FoodCPI	Total
Absolute	12	6	0	0	0	0	0	2	0	0	20
Momentum1	0	0	0	0	0	0	0	0	5	4	9
Momemtum3	11	4	0	0	0	5	0	0	0	0	20
Momemtum6	23	4	0	0	0	12	0	0	12	12	63
Stochastic %k	0	12	5	1	0	0	0	0	0	1	19
Williams %R	14	6	0	0	0	4	0	2	18	10	54
Disparity6	1	0	0	0	0	0	2	0	0	13	16
Total	61	32	5	1	0	21	2	4	35	40	-

que en la predicción a seis meses del precio del aceite de oliva virgen extra, las variables más influyentes son (ordenadas por el número de veces que han sido seleccionadas): SrcPrice, FoodCPI, GenCPI, TgetPrice y ConMK. Hay un segundo grupo, que han sido seleccionadas un similar número de veces: FoodCPI, GenCPI y TgetCPI. La última variable resaltable es ConMK que se ha seleccionado de forma moderada. El resto de variables sólo se han seleccionado de forma puntual.

Para construir un nuevo conjunto de datos a partir de los datos ofrecidos por los algoritmos de selección, se eligen como variables de entrada aquellas que ha sido elegidas al menos una vez en cualquier año por cualquiera de los métodos.

Los resultados de aplicar los métodos *soft-computing* a este nuevo conjunto de datos variables seleccionadas se muestra en la Tabla 5.

De nuevo el método CO^{2}RBFN consigue el mejor resultado en test (tanto en media como en desviación típica) de entre todos los algoritmos utilizados en el estudio.

Table 5. Error Mape en test con el conjunto de variables seleccionadas

Year	CO^{2}RBFN	FuzzyGap	MLPConjGrad	NUSVR	RBFNLMS
Test2006	0.2454 ± 0.0483	0.2419 ± 0.1579	0.4893 ± 0.3916	0.3853 ± 0.2499	0.1434 ± 0.1101
Test2007	0.0711 ± 0.0312	0.2466 ± 0.1317	1.0056 ± 0.4911	0.4040 ± 0.1359	0.1875 ± 0.0453
Test2008	0.0963 ± 0.0138	0.2381 ± 0.1061	0.5144 ± 0.3724	0.2273 ± 0.1618	0.1319 ± 0.1053
Test2009	0.2090 ± 0.0287	0.1422 ± 0.0976	0.3309 ± 0.2046	0.1902 ± 0.1578	0.2326 ± 0.1604
Mean	**0.1555 ± 0.0305**	0.2172 ± 0.1233	0.5850 ± 0.3649	0.3017 ± 0.1764	0.1739 ± 0.1053

Si se comparan los resultados obtenidos por los métodos usando el conjunto de variables seleccionas frente a los conseguidos sobre el conjunto total de variables, se puede observar que los resultados son similares, con lo que se consigue el objetivo de simplificar el problema habiendo identificado las variables que influyen en el precio futuro del aceite de oliva virgen.

6 Conclusiones

La predicción de series temporales constituye un área de interés en sectores como ciencia, ingeniería, economía, etc, o concretamente para la industria productora de aceite de oliva donde España el primer país productor y exportador.

Los agentes implicados en este sector del olivar están muy interesados en el uso de métodos de predicción enfocados a diferentes objetivos, entre los que destaca la predicción del precio del aceite de oliva virgen extra. Ésto es especialmente importante para el Mercado de Futuros del Aceite de Oliva (MFAO).

En esta línea, y en colaborción con la Fundación para la Promoción y Desarrollo del Olivar y Aceite de Oliva (Poolred), se ha identificado que la predicción a seis meses del precio del aceite de oliva virgen extra, constituye una tarea de interés.

Los métodos *soft-computing* y particularmente los de diseño de RBFNs han demostrado su eficiencia en la resolución de problemas de predicción. Por esta razón los autores proponen la aplicación de CO^{2}RBFN, un algoritmo evolutivo con enfoque cooperativo-competitivo de diseo de RBFN, para resolver el problema de predicción a medio plazo del precio del aceite de oliva.

Son muchas las variables exógenas e indicadores técnicos que se tienen que manejar para realizar la predicción con lo que los modelos obtenidos son complicados. Para simplificarlos se han aplicado métodos de selección de características que tratan de encontrar cúales son los aspectos que realmente afectan a la predicción, en este caso del precio del aceite de oliva.

Tanto CO^{2}RBFN como el resto de los métodos *soft-computing* se han aplicado sobre el conjunto con el total de características y sobre el conjunto con las características seleccionadas. Los resultados obtenidos muestran que CO^{2}RBFN obtiene los mejores resultados en media para los dos conjuntos de datos y que es un método robusto.

Desde el punto de vista de la selección de características, se puede concluir que variables como precio en origen, precio en destino, IPC general, IPC en alimentación y consumo influyen en el precio futuro del aceite de oliva virgen extra.

Agradecimientos. Subvencionado por el proyecto español del Ministerio de Ciencia y Tecnología TIN 2012-33856 y por el Plan de Investigación Andaluz TIC-3928.

References

1. Box, G., Jenkins, G., Reinsel, G.: Time series analysis: forecasting and control,4th Edition. Wiley (2008)
2. Mochón, A., Quintana, D., Sáez, Y., Isasi, P.: Soft computing techniques applied to finance. Applied Intelligence **29**(2) (2008) 111–115
3. Atsalakis, G., Valavanis, K.: Surveying stock market forecasting techniques - part ii: Soft computing methods. Expert Systems with Applications **36**(3, Part 2) (2009) 5932 – 5941

4. Vanstone, B., Hahn, T.: Designing Stock Market Trading Systems: with and without Soft Computing. Harriman House (2010)
5. Pérez-Godoy, M., Pérez, P., Rivera, A., del Jesus, M., Carmona, C., Frías, M., Parras, M.: co^2rbfn for short-term forecasting of the extra virgin olive oil price in the spanish market. International Journal of Hybrid Intelligent Systems **7**(1) (2010) 75–87
6. Co, H., Boosarawongse, R.: Forecasting thailand's rice export: Statistical techniques vs. artificial neural networks. Computers and Industrial Engineering **53**(4) (2007) 610–627
7. Khashei, M., Reza Hejazi, S., Bijari, M.: A new hybrid artificial neural networks and fuzzy regression model for time series forecasting. Fuzzy Sets and Systems **159**(7) (2008) 769–786
8. Pino, R., Parreno, J., Gomez, A., Priore, P.: Forecasting next-day price of electricity in the spanish energy market using artificial neural networks. Engineering Applications of Artificial Intelligence **21**(1) (2008) 53–62
9. Maimon, O., Last, M.: Knowledge discovery and data mining - The info-fuzzy network (ifn) methodology. Kluwer (2000)
10. Elder, J., Pregibon, D.: A statistical perspective on knowledge discovery in databases. In: Advances in knowledge discovery and data mining, AAAI/MIT Press (1996)
11. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. The Journal of Machine Learning Research **3** (2003) 1157–1182
12. Pérez-Godoy, M., Rivera, A., del Jesus, M., Berlanga, F.: co^2rbfn: An evolutionary cooperative-competitive RBFN design algorithm for classification problems. Soft Computing **14**(9) (2010) 953–971
13. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.: The weka data mining software: An update. SIGKDD Explorations **11**(1) (2009)
14. Hall, M., Smith, L.: Feature selection for machine learning: Comparing a correlation-based filter approach to the wrapper. In: Twelfth International FLAIRS Conference, AAAI/MIT Press (1999)
15. Gütlein, M., Frank, E., Hall, M., Karwath, A.: Large scale attribute selection using wrappers. In: Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining. (1999)
16. Goldberg, D.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading, MA (1989)
17. García, F., García, M., Melián, B., Moreno, J., Marcos, J.: Solving feature subset selection problem by a parallel scatter search. European Journal of Operational Research **169** (2006) 477–489
18. Sánchez, L., Couso, I.: Fuzzy random variables-based modeling with ga-p algorithms. In: B. Bouchon, R.R. Yager, L. Zadeh (Eds.) Information, Uncertainty and Fusion (2000) 245–256
19. Moller, F.: A scaled conjugate gradient algorithm for fast supervised learning. Neural Networks **6** (1990) 525–533
20. Fan, R., Chen, P., Lin, C.: Working set selection using the second order information for training svm. Journal of Machine Learning Research **6** (2005) 1889–1918
21. Widrow, B., Lehr, M.: 30 years of adaptive neural networks: perceptron, madaline and backpropagation. Proceedings of the IEEE **78**(9) (1990) 1415–1442
22. Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F.: Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. J. of Mult.-Valued Logic & Soft Computing **17** (2011) 255–287

Improving Ensemble Methods for Imbalanced Datasets with Disturbing Neighbors

Juan J. Rodríguez, José F. Díez-Pastor, Jesús Maudes, and César García-Osorio

University of Burgos
jjrodriguez,jfdpastor,jmaudes,cgosorio@ubu.es

Abstract. One of the successful approaches for dealing with imbalanced data are methods that combine classical, not specific for imbalance, ensemble methods (e.g., Bagging, Boosting), with pre-processing approaches designed for imbalance (e.g., random undersampling, SMOTE). Disturbing Neighbors ($\mathcal{DN}$) is a method for improving the diversity of the base classifiers generated by an ensemble method, this usually improves the performance of the ensemble. This work considers the combination of three ensemble methods (i.e., SMOTEBoost, SMOTEBagging and RUSBoost) with $\mathcal{DN}$. The results show that $\mathcal{DN}$ improves significantly these methods.

1 Introduction

In classification problems, the data is imbalanced when the proportion of the classes is very different. There are several specific techniques for this kind of data, because they can give better results than methods that do not take into account this situation. Not only the classifier construction methods have to be specific, also there are specific performance measures because with imbalanced data, conventional classification measures (e.g., accuracy) are not useful. Two measure used for imbalance are the AUROC and the AUPRC. Both are areas under one curve, the first one under the ROC (Receiver Operation Characteristic) [1], the second one under the Precision-Recall curve [2].

There are several approaches for dealing with imbalanced data [3]. One way is to change the class distributions, with some pre-processing. For instance, in SMOTE [4] artificial instances of the minority class are generated. Ensemble methods, as for other classification problems, are one of the successful approaches. Several ensemble methods for imbalanced datasets are obtained including a pre-processing step in a conventional ensemble method.

SMOTEBoost [5] combines Boosting with SMOTE. In each iteration of Boosting, the data is augmented with minority instances generated with SMOTE. SMOTEBagging [6] combines Bagging with SMOTE. The base classifiers are constructed with perfectly balanced data sets, the classes have the same number of instances. Some instances are obtained resampling the original training data while others are obtained from SMOTE. The percentage of instances from

resampling and SMOTE varies in order to increase the diversity. RUSBoost [7] is similar to SMOTEBoost, the difference is that instead of augmenting the minority class with SMOTE, the majority class is reduced with random undersampling.

This paper studies the performance of these ensemble methods when used with Disturbing Neighbors ($\mathcal{DN}$) [8].

The rest of the paper is organized as follows: Sect. 2 describes the $\mathcal{DN}$ method. The experiments and results are presented in Sect. 3. Finally, Sect. 4 shows the conclusions.

2 Disturbing Neigbors

$\mathcal{DN}$ is a method for generating diversity in the base classifiers of an ensemble. Diversity is a required ingredient of successful ensembles, the other is the accuracy of the base classifiers. $\mathcal{DN}$ can be used by itself for generating ensembles, but it can also be combined with any other ensemble method. These combinations usually improve the two combined methods.

In $\mathcal{DN}$, the training data for each base classifier is augmented with a few features. First, a random subset of the training instances is selected. The instances in this subset are the $\mathcal{DN}$. A nominal feature is added to the data set, its values identify the nearest $\mathcal{DN}$. For instance, if the number of $\mathcal{DN}$ is 10, this feature has 10 possible values. Then, some boolean features are also added. There are as many features as $\mathcal{DN}$. Each boolean feature is true if the closest $\mathcal{DN}$ for that instance is the corresponding $\mathcal{DN}$.

The distances to the $\mathcal{DN}$ are calculated in a random subspace, with the purpose of augmenting the diversity. For each base classifier, a random subset of the features is selected, distances to the $\mathcal{DN}$ will be calculated in that subspace. Otherwise, two close instances would usually have the same values for all the additional features.

As different base classifiers will have different values for these features and they can seem useful for classification, the base classifiers can be more diverse without necessarily being less accurate.

In [9] the $\mathcal{DN}$ method was used with imbalanced data, but this method was only combined with Bagging and the results were not compared with other ensemble methods for imbalanced datasets.

3 Experiments

3.1 Data sets

Two collections of datasets were used. The HDDT collection[1] contains the binary imbalanced datasets used in [10]. The KEEL collection[2] contains the binary imbalanced datasets from the repository of KEEL [11].

[1] Available at `http://www.nd.edu/~dial/hddt/`.
[2] Available at `http://sci2s.ugr.es/keel/imbalanced.php`.

Table 1 shows the characteristics of the 20 data sets in the HDDT collection and Table 2 the 66 data sets in the KEEL collection. Many data sets in these two collections are available or are modifications of data sets in the UCI Repository [12].

Table 1. Characteristics of the data sets from the HDDT collection.(#E: number of examples, #A: number of attributes (numeric/nominal), IR: imbalance ratio)

Data set	Examples	Attributes		Imbalance
		Numeric	Nominal	Ratio
boundary	3505	0	175	27.50
breast-y	286	0	9	2.36
cam	18916	0	132	19.08
compustat	13657	20	0	25.26
covtype	38500	10	0	13.02
credit-g	1000	7	13	2.33
estate	5322	12	0	7.37
german-numer	1000	24	0	2.33
heart-v	200	5	8	2.92
hypo	3163	7	18	19.95
ism	11180	6	0	42.00
letter	20000	16	0	24.35
oil	937	49	0	21.85
optdigits	5620	64	0	9.14
page	5473	10	0	8.77
pendigits	10992	16	0	8.63
phoneme	5404	5	0	2.41
PhosS	11411	480	0	17.62
satimage	6430	36	0	9.29
segment	2310	19	0	6.00

3.2 Settings

Weka [13] was used for the experiments. Unless explicitly specified, the parameters for the different methods take the default values given by Weka.

The decision tree method used for constructing the base classifiers was J48 (Weka's re-implementation of C4.5 [14]). As recommended for imbalanced data [10], it was used without pruning and collapsing but with Laplace smoothing at the leaves. C4.5 with these options is called C4.4 [15].

Several ensemble methods were considered. The first ensemble method is Bagging [16]. It was the best method for imbalanced data in [10] and [17]. In [9] it was used with $\mathcal{DN}$, improving the results of Bagging without $\mathcal{DN}$.

As ensemble methods for imbalance the following were used: SMOTEBoost, SMOTEBagging and RUSBoost. SMOTEBoost has a parameter, the number of artificial instances to generate. Three values were considered: 100%, 200% and 500% of the number of instances in the minority class.

For each considered ensemble method there are two versions: one without $\mathcal{DN}$ and the other with $\mathcal{DN}$. The parameter with the number of neighbors in $\mathcal{DN}$ was 10. For all the ensembles the number of classifiers was 100.

Table 2. Characteristics of the data sets from the KEEL collection. #E: number of examples, #N: number of numeric attributes, #D: number of discrete attributes, IR: imbalance ratio

data set	#E	#N	#D	IR	data set	#E	#N	#D	IR
abalone19	4174	7	1	129.44	glass5	214	9	0	22.78
abalone9-18	731	7	1	16.40	glass6	214	9	0	6.38
cleveland-0_vs_4	177	13	0	12.62	haberman	306	3	0	2.78
ecoli-0-1-3-7_vs_2-6	281	7	0	39.14	iris0	150	4	0	2.00
ecoli-0-1-4-6_vs_5	280	6	0	13.00	led7digit-0-2-4-5-6-7-8-9_vs_1	443	7	0	10.97
ecoli-0-1-4-7_vs_2-3-5-6	336	7	0	10.59	new-thyroid1	215	5	0	5.14
ecoli-0-1-4-7_vs_5-6	332	6	0	12.28	new-thyroid2	215	5	0	5.14
ecoli-0-1_vs_2-3-5	244	7	0	9.17	page-blocks-1-3_vs_4	472	10	0	15.86
ecoli-0-1_vs_5	240	6	0	11.00	page-blocks0	5472	10	0	8.79
ecoli-0-2-3-4_vs_5	202	7	0	9.10	pima	768	8	0	1.87
ecoli-0-2-6-7_vs_3-5	224	7	0	9.18	segment0	2308	19	0	6.02
ecoli-0-3-4-6_vs_5	205	7	0	9.25	shuttle-c0-vs-c4	1829	9	0	13.87
ecoli-0-3-4-7_vs_5-6	257	7	0	9.28	shuttle-c2-vs-c4	129	9	0	20.50
ecoli-0-3-4_vs_5	200	7	0	9.00	vehicle0	846	18	0	3.25
ecoli-0-4-6_vs_5	203	6	0	9.15	vehicle1	846	18	0	2.90
ecoli-0-6-7_vs_3-5	222	7	0	9.09	vehicle2	846	18	0	2.88
ecoli-0-6-7_vs_5	220	6	0	10.00	vehicle3	846	18	0	2.99
ecoli-0_vs_1	220	7	0	1.86	vowel0	988	13	0	9.98
ecoli1	336	7	0	3.36	wisconsin	683	9	0	1.86
ecoli2	336	7	0	5.46	yeast-0-2-5-6_vs_3-7-8-9	1004	8	0	9.14
ecoli3	336	7	0	8.60	yeast-0-2-5-7-9_vs_3-6-8	1004	8	0	9.14
ecoli4	336	7	0	15.80	yeast-0-3-5-9_vs_7-8	506	8	0	9.12
glass-0-1-2-3_vs_4-5-6	214	9	0	3.20	yeast-0-5-6-7-9_vs_4	528	8	0	9.35
glass-0-1-4-6_vs_2	205	9	0	11.06	yeast-1-2-8-9_vs_7	947	8	0	30.57
glass-0-1-5_vs_2	172	9	0	9.12	yeast-1-4-5-8_vs_7	693	8	0	22.10
glass-0-1-6_vs_2	192	9	0	10.29	yeast-1_vs_7	459	7	0	14.30
glass-0-1-6_vs_5	184	9	0	19.44	yeast-2_vs_4	514	8	0	9.08
glass-0-4_vs_5	92	9	0	9.22	yeast-2_vs_8	482	8	0	23.10
glass-0-6_vs_5	108	9	0	11.00	yeast1	1484	8	0	2.46
glass0	214	9	0	2.06	yeast3	1484	8	0	8.10
glass1	214	9	0	1.82	yeast4	1484	8	0	28.10
glass2	214	9	0	11.59	yeast5	1484	8	0	32.73
glass4	214	9	0	15.46	yeast6	1484	8	0	41.40

The results were obtained with 5×2-fold cross validation [18]. Average ranks [19,20] were used for comparing several methods across several data sets.

3.3 Results

Table 3 shows the comparison of methods with and without $\mathcal{DN}$, according to the AUROC. Each entry in the table shows the number of wins (W), ties (T) and losses (L) when comparing the method with $\mathcal{DN}$ with the original method. For 20 data sets, according to a two-tailed sign test at $\alpha = 0.05$ a classifier is significantly better than another if it performs better on at least 15 data sets [19]. For the HDDT collection in all the considered pairs the number of wins of $\mathcal{DN}$ is at least 15. For 66 data sets, the required number of wins is 41. For the KEEL collection the number of wins is at least 51.

Table 3. Comparison of methods with and without $\mathcal{DN}$, according to the AUROC.

Method	HDDT W/T/L	KEEL W/T/L
Bagging	20/0/0	59/1/6
SMOTEBagging	19/0/1	59/2/5
SMOTEBoost-100%	16/0/4	54/2/10
SMOTEBoost-200%	15/0/5	56/2/8
SMOTEBoost-500%	17/0/3	51/2/13
RUSBoost	15/0/5	51/2/13

Fig. 1 shows for the considered methods the histograms of the differences between the AUROC for the method with $\mathcal{DN}$ and the method without $\mathcal{DN}$. Fig. 2 shows these histograms for the KEEL collection. These histograms show for how many data sets the version with $\mathcal{DN}$ is better or worse than the version without $\mathcal{DN}$, but they also show the magnitudes of the differences.

Table 4 shows the comparison of methods with and without $\mathcal{DN}$, according to the AUPRC. For all the considered methods, with only one exception (SMOTEBoost-200% for HDDT), the number of wins is greater or equal than the minimum number required according to the sign test: 15 for the HDDT collection and 41 for the KEEL collection.

Fig. 3 shows for the considered methods the histograms of the differences between the AUPRC for the method with $\mathcal{DN}$ and the method without $\mathcal{DN}$. Fig. 4 shows these histograms for the KEEL collection.

Until now the results in this section show the comparison of a method and the corresponding method augmented with $\mathcal{DN}$. They show cleary the advantage of $\mathcal{DN}$, but another interesting issue is which are the best methods among all the considered. Average ranks [19,20] are used to compare all these methods. For each data set all the methods are sorted, from best to worst, according to the considered performance measure. The best method has rank 1, the second rank

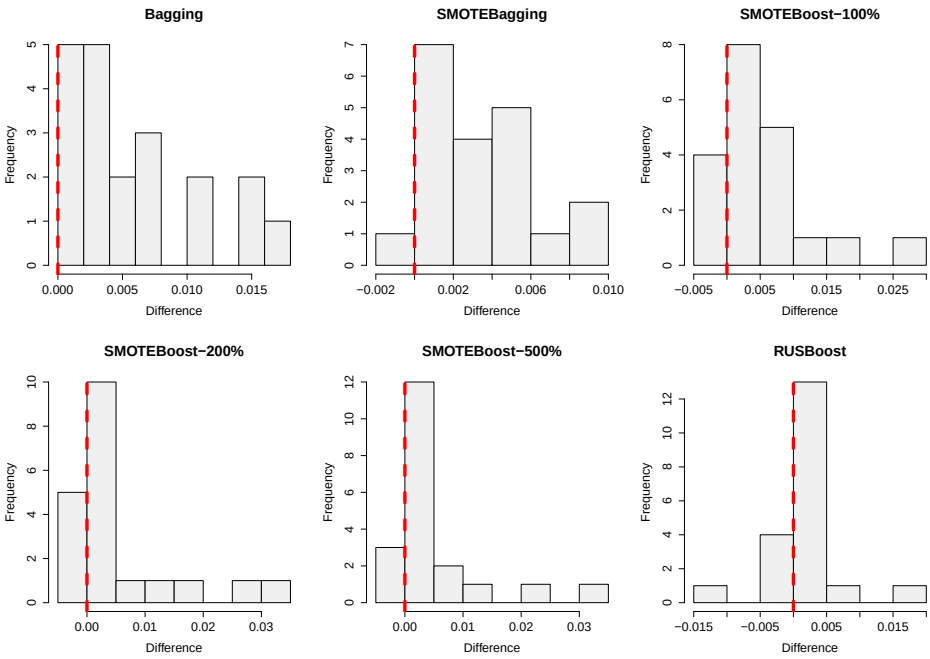


Fig. 1. Histograms of the differences between the AUROC when using the method with and without $\mathcal{DN}$, for the HDDT collection.

Table 4. Comparison of methods with and without $\mathcal{DN}$, according to the AUPRC.

Method	HDDT W/T/L	KEEL W/T/L
Bagging	18/0/2	60/1/5
SMOTEBagging	17/0/3	58/2/6
SMOTEBoost-100%	18/0/2	58/2/6
SMOTEBoost-200%	14/0/6	56/2/8
SMOTEBoost-500%	18/0/2	55/2/9
RUSBoost	16/0/4	51/2/13

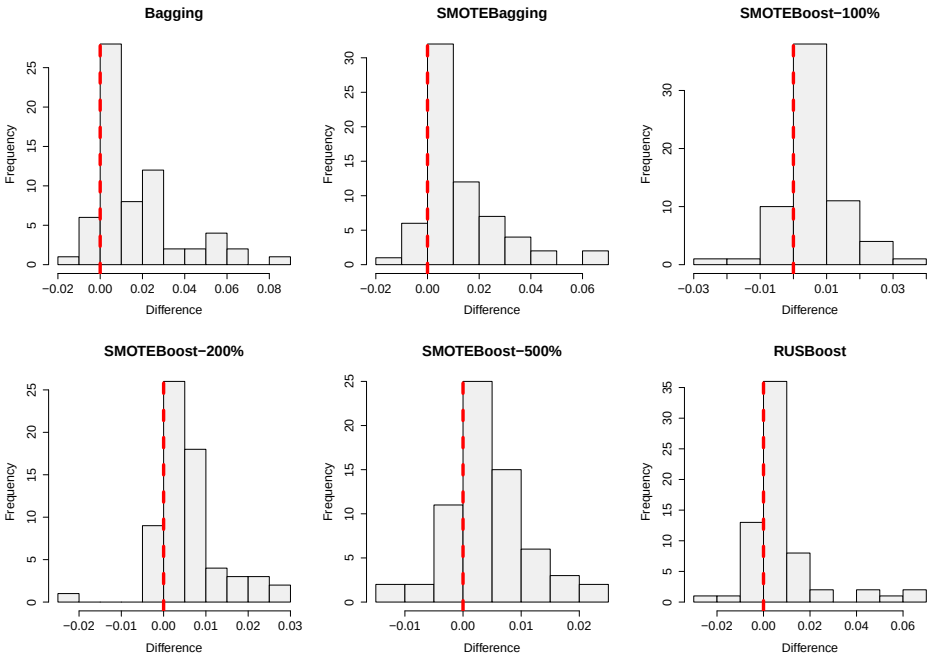


Fig. 2. Histograms of the differences between the AUROC when using the method with and without $\mathcal{DN}$, for the KEEL collection.

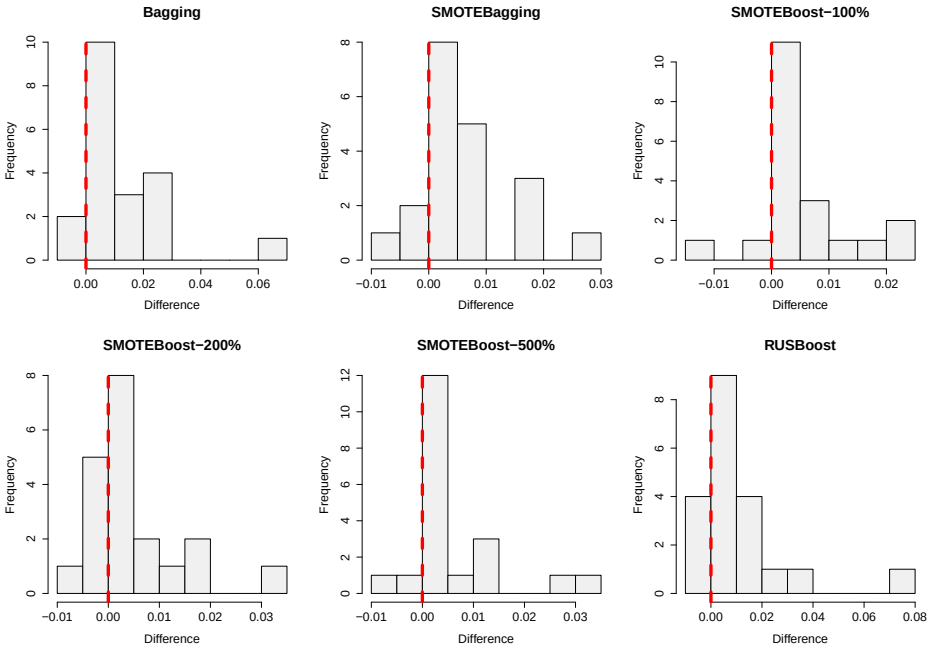


Fig. 3. Histograms of the differences between the AUPRC when using the method with and without $\mathcal{DN}$, for the HDDT collection.

two and so on. If several methods have the same performance they are assigned an average rank (e.g., if two methods have the best result, both have a rank of 1.5). The average rank of a method is calculated as the mean across all the data sets. The Iman and Davenport Test is used to determine the presence of differences among all the compared methods. As a post-hoc procedure, using the best method as the control, the Hochberg procedure is used [20].

Table 5 shows the average ranks according to the AUROC, while Table 6 shows them according to the AUPRC. For the two measures and the two collections of data sets the top positions in the ranks are for methods with $\mathcal{DN}$. For the KEEL collection all the methods with $\mathcal{DN}$ have better average ranks than all the methods without $\mathcal{DN}$. This does not happen for the HDDT collection, but because there is only one exception: $\mathcal{DN}$-SMOTEBagging. These tables also show the adjusted p-value according to the Hochberg procedure.

4 Conclusions

The performance of Disturbing Neighbors have been studied when included in ensemble methods for imbalanced data. Three ensemble methods have been considered: SMOTEBagging, SMOTEBoost and RUSBoost. According to the AUROC and the AUPRC in two sets of data sets (HDDT and KEEL) including $\mathcal{DN}$ gives significant improvements.

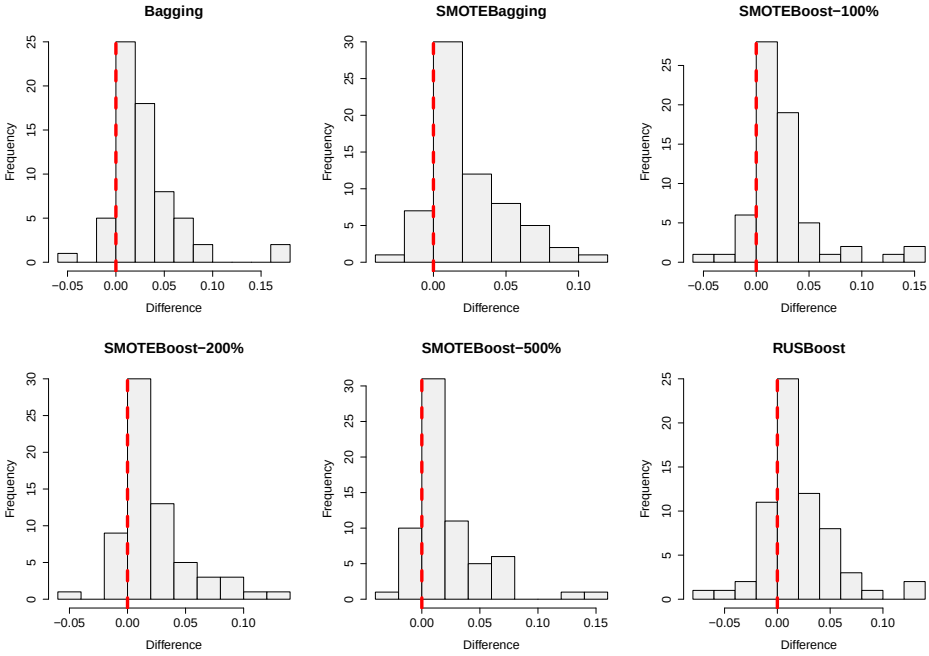


Fig. 4. Histograms of the differences between the AUPRC when using the method with and without $\mathcal{DN}$, for the KEEL collection.

Table 5. Average ranks according to the AUROC

(a) HDDT

Method	Ranking	$p_{Hochberg}$
$\mathcal{DN}$-SMOTEBoost 100%	4.25	
$\mathcal{DN}$-SMOTEBoost 200%	4.60	0.758866
$\mathcal{DN}$-SMOTEBoost 500%	5.55	0.508426
$\mathcal{DN}$-RUSBoost	5.75	0.508426
$\mathcal{DN}$-Bagging	6.25	0.317643
SMOTEBoost 100%	6.40	0.296694
$\mathcal{DN}$-SMOTEBagging	6.50	0.290713
SMOTEBoost 200%	6.75	0.198331
RUSBoost	6.85	0.180695
SMOTEBoost 500%	7.35	0.058953
SMOTEBagging	8.50	0.001934
Bagging	9.25	0.000127

(b) KEEL

Method	Ranking	$p_{Hochberg}$
$\mathcal{DN}$-RUSBoost	4.0682	
$\mathcal{DN}$-SMOTEBoost 200%	4.7803	0.256547
$\mathcal{DN}$-Bagging	5.0758	0.216845
$\mathcal{DN}$-SMOTEBoost 500%	5.0985	0.216845
$\mathcal{DN}$-SMOTEBoost 100%	5.1818	0.216845
$\mathcal{DN}$-SMOTEBagging	5.8333	0.024592
RUSBoost	6.2500	0.003052
SMOTEBoost 500%	7.5682	0
SMOTEBoost 200%	7.8258	0
SMOTEBoost 100%	8.1136	0
SMOTEBagging	9.0833	0
Bagging	9.1212	0

Table 6. Average ranks according to the AUPRC

(a) HDDT

Method	Ranking	$p_{Hochberg}$
$\mathcal{DN}$-SMOTEBoost 100%	3.95	
$\mathcal{DN}$-SMOTEBoost 200%	4.55	0.598725
$\mathcal{DN}$-Bagging	5.75	0.228808
$\mathcal{DN}$-SMOTEBoost 500%	6.00	0.216546
$\mathcal{DN}$-RUSBoost	6.05	0.216546
SMOTEBoost 200%	6.15	0.216546
SMOTEBoost 100%	6.30	0.216546
$\mathcal{DN}$-SMOTEBagging	6.90	0.067708
RUSBoost	7.95	0.003609
SMOTEBoost 500%	8.00	0.003440
SMOTEBagging	8.15	0.002299
Bagging	8.25	0.001786

(b) KEEL

Method	Ranking	$p_{Hochberg}$
$\mathcal{DN}$-RUSBoost	3.9470	
$\mathcal{DN}$-Bagging	4.7273	0.213786
$\mathcal{DN}$-SMOTEBoost 100%	4.8333	0.213786
$\mathcal{DN}$-SMOTEBoost 200%	4.8485	0.213786
$\mathcal{DN}$-SMOTEBoost 500%	5.2652	0.142845
$\mathcal{DN}$-SMOTEBagging	5.7197	0.023684
RUSBoost	6.6288	0.000116
SMOTEBoost 200%	8.3258	0
SMOTEBoost 100%	8.3712	0
Bagging	8.3939	0
SMOTEBoost 500%	8.4318	0
SMOTEBagging	8.5076	0

As base classifiers, only decision trees have been used. Other base classifiers could give better results or affect in different ways the behaviour of the ensemble methods. The performance of the different ensemble methods with other base classifiers can be studied in future work.

Acknowledgments. This work was supported by the Projects TIN2011-24046 and IPT-2011-1265-020000 of the Spanish Ministry of Economy and Competitiveness.

References

1. Flach, P., Hernandez-Orallo, J., Ferri, C.: A coherent interpretation of AUC as a measure of aggregated classification performance. In: 28th International Conference on Machine Learning (ICML-11), ACM (June 2011) 657–664
2. Davis, J., Goadrich, M.: The relationship between Precision-Recall and ROC curves. In: Proceedings of the 23rd international conference on Machine learning. ICML '06, New York, NY, USA, ACM (2006) 233–240
3. Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., Herrera, F.: A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on **42**(4) (july 2012) 463 –484
4. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research **16** (2002) 321–357
5. Chawla, N.V., Lazarevic, A., Hall, L.O., Bowyer, K.W.: SMOTEBoost: Improving prediction of the minority class in boosting. In: Knowledge Discovery in Databases: PKDD 2003. (2003) 107–119
6. Wang, S., Yao, X.: Diversity analysis on imbalanced data sets by using ensemble models. In: IEEE Symposium Series on Computational Intelligence and Data Mining(IEEE CIDM 2009). (2009) 324–331
7. Seiffert, C., Khoshgoftaar, T., Van Hulse, J., Napolitano, A.: Rusboost: A hybrid approach to alleviating class imbalance. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on **40**(1) (2010) 185–197

8. Maudes, J., Rodríguez, J.J., García-Osorio, C.: Disturbing neighbors diversity for decision forests. In Okun, O., Valentini, G., eds.: Applications of Supervised and Unsupervised Ensemble Methods. Volume 245 of Studies in Computational Intelligence. Springer (2009) 113–133
9. Rodríguez, J.J., Díez-Pastor, J.F., Maudes, J., García-Osorio, C.: Disturbing neighbors ensembles of trees for imbalanced data. In Wani, M.A., Khoshgoftaar, T., Zhu, X.H., Seliya, N., eds.: 11th International Conference on Machine Learning and Applications, ICMLA 2012. Volume 2., IEEE (2012) 83–88
10. Cieslak, D., Hoens, T., Chawla, N., Kegelmeyer, W.: Hellinger distance decision trees are robust and skew-insensitive. Data Mining and Knowledge Discovery **24**(1) (2012) 136–158
11. Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S.: KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. Multiple-Valued Logic and Soft Computing **17**(2-3) (2011) 255–287
12. Frank, A., Asuncion, A.: UCI machine learning repository (2010) `http://archive.ics.uci.edu/ml`.
13. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: An update. SIGKDD Explorations **11**(1) (2009)
14. Quinlan, J.R.: C4.5: Programs for Machine Learning. Machine Learning. Morgan Kaufmann, San Mateo, California (1993)
15. Provost, F., Domingos, P.: Tree induction for Probability-Based ranking. Machine Learning **52**(3) (September 2003) 199–215
16. Breiman, L.: Bagging predictors. Machine Learning **24**(2) (1996) 123–140
17. Rodríguez, J.J., Díez-Pastor, J.F., García-Osorio, C., Santos, P.: Using model trees and their ensembles for imbalanced data. In: Advances in Artificial Intelligence: 14th Conference of the Spanish Association for Artificial Intelligence, Springer (2011) 94–103
18. Dietterich, T.G.: Approximate statistical test for comparing supervised classification learning algorithms. Neural Computation **10**(7) (1998) 1895–1923
19. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research **7** (2006) 1–30
20. García, S., Fernández, A., Luengo, J., Herrera, F.: Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Information Sciences **180**(10) (May 2010) 2044–2064

Meta-aprendizaje para la recomendación de algoritmos de clasificación con conjuntos de datos educacionales

Cristóbal Romero, Juan Luis Olmo, and Sebastián Ventura

Departamento de Informática y Análisis Numérico,
Universidad de Córdoba, Campus de Rabanales, Edificio Albert Einstein,
14071 Cordoba, España. Tel.:+34957212218. Fax:+34957218630
{cromero,jlolmo,sventura}@uco.es

Resumen En este artículo se propone utilizar una estrategia de meta-aprendizaje con el fin de recomendar el mejor subconjunto de algoritmos de clasificación de caja blanca para utilizarlos con conjuntos de datos educativos. Se ha analizado un caso de estudio con 32 conjuntos de datos Moodle, considerando no sólo las características estadísticas tradicionales, sino incorporando características de complejidad y específicas del dominio educativo. Para evaluar la productividad de los algoritmos en rangos se utilizaron diferentes medidas de rendimiento de clasificación, así como test estadísticos no paramétricos. Por otro lado, un enfoque de vecinos cercanos fue empleado para recomendar el subconjunto de algoritmos óptimo dado un nuevo conjunto de datos. Los experimentos demuestran que los mejores resultados de recomendación se obtienen cuando se utilizan simultáneamente los tres tipos de características de los conjuntos de datos.

Keywords: Meta-aprendizaje, clasificación, predicción del rendimiento de estudiantes

1. Introducción

La predicción del rendimiento de los estudiantes como una tarea de clasificación es uno de los problemas originales y más estudiados dentro de la minería de datos en educación (EDM, *Educational Data Mining*) [11]. Así, se han aplicado numerosos algoritmos y técnicas para predecir el éxito y los resultados académicos. Sin embargo, la selección e identificación del algoritmo más adecuado para clasificar un nuevo conjunto de datos es una tarea difícil, si tenemos en cuenta que no existe un único clasificador que se comporte, en general, de forma superior a todos los demás para cada conjunto de datos, tal como enuncia el teorema del *No Free Lunch* [13,6]. Así, la elección del algoritmo de clasificación apropiado para un determinado conjunto de datos es de gran dificultad e importancia en la práctica.

Por otro lado, el meta-aprendizaje consiste en el estudio de los métodos principales que explotan metaconocimiento para obtener modelos eficientes y soluciones mediante la adaptación del proceso de minería de datos (DM, *Data Mining*) [2]. La tarea de recomendación se puede realizar de diversas formas, como por ejemplo recomendar el mejor algoritmo de un conjunto, un subconjunto de algoritmos, un ranking de algoritmos o el rendimiento estimado de los mismos. Nuestra propuesta para el caso de conjuntos de

datos educativos se centra en utilizar diversas medidas de evaluación de rendimiento en clasificación y test estadísticos para crear un ranking de algoritmos, utilizando a continuación un enfoque de vecinos más cercanos [3] para recomendar el subconjunto de mejores algoritmos para clasificar un nuevo conjunto de datos de este tipo.

El meta-aprendizaje se ha empleado principalmente sobre dominios generales y utilizando conjuntos de datos estándar, disponibles públicamente, como los de la UCI (Universidad de California en Irvine) [5]. Sin embargo, no tenemos constancia de que existan trabajos que apliquen el concepto de meta-aprendizaje para abordar la selección de algoritmos en el dominio específico de EDM. De hecho, existe un único trabajo que utiliza meta-aprendizaje para apoyar la selección de los valores adecuados para los parámetros del algoritmo J48 utilizando diversos conjuntos de datos educacionales [8]. En este dominio, la comprensibilidad del conocimiento descubierto y de los modelos inferidos en la tarea de clasificación tiene una gran importancia, ya que tienen que poder ser interpretados por usuarios que no son expertos en DM, tales como instructores, autores de cursos y otros miembros de la comunidad educativa, de manera que puedan incorporar dicho conocimiento en el proceso de toma de decisiones. De hecho, los modelos de DM de caja blanca basados en reglas se prefieren ante los modelos de caja negra, como las redes bayesianas o las redes neuronales artificiales, que a pesar de que normalmente obtienen una mayor exactitud, son mucho menos comprensibles [10]. Por otro lado, para caracterizar conjuntos de datos en meta-aprendizaje se suelen emplear medidas estadÌsticas y de teoría de la información, aunque recientemente se han empezado a utilizar también medidas relacionadas con la complejidad de los datos [7]. Sin embargo, en nuestra propuesta incorporamos, además, medidas específicas del dominio para caracterizar los conjuntos de datos.

El resto del trabajo se organiza como sigue. En la próxima sección se introduce la metodologÌa seguida en este trabajo. En la Sección 3 se describen los conjuntos de datos Moodle que se usan en la experimentación. La Sección 4 describe los experimentos, resultados y el modelo obtenidos. Finalmente, las conclusiones y posibles líneas de trabajo futuro se enuncian en la Sección 5.

2. Metodología

La metodología de meta-aprendizaje que proponemos en este artículo se compone de dos pasos o fases, tal como recoge la Figura 1:

- Una fase fuera de línea o de entrenamiento, para crear una meta base de datos a partir de los conjuntos de datos educacionales y los algoritmos de clasificación. Por un lado, identificamos características y propiedades relevantes de los conjuntos de datos (estadísticas, de complejidad y propias del dominio) y generamos metacaracterísticas. Por otro lado, utilizamos algoritmos de clasificación de caja blanca (árboles de decisión y clasificadores basados en reglas) para evaluar su rendimiento sobre todos los conjuntos de datos disponibles. Por cada conjunto de datos, calculamos diversas medidas de rendimiento obtenidas por cada algoritmo, con lo que dispondremos de tantas instancias como medidas se consideren. A continuación,

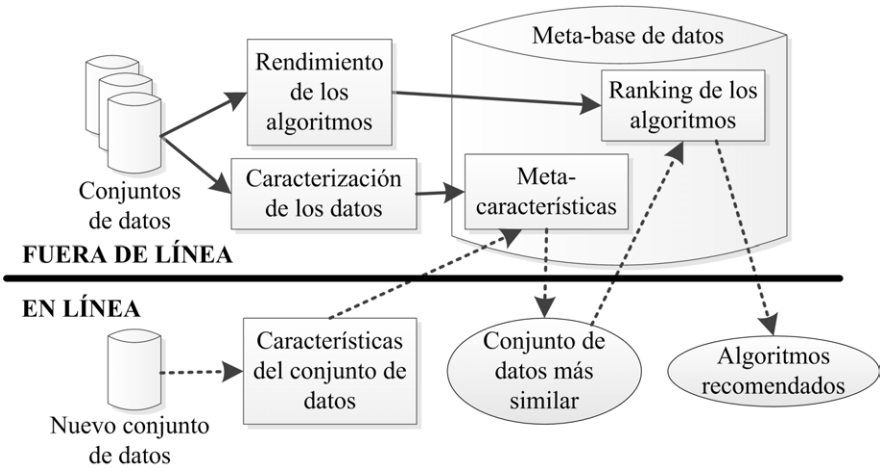


Figura 1. Metodología de meta-aprendizaje

empleando test estadísticos no paramétricos [4], se calcula el rango de los algoritmos sobre el conjunto de medidas consideradas, así como el conjunto de algoritmos, partiendo del que mejor rendimiento presenta, que no se comportan estadísticamente peor que éste. De este modo, se habrá identificado por cada conjunto de datos educacional el subconjunto de algoritmos que mejor abordan dicho problema de clasificación.

- Una fase en línea o de predicción, donde se recomienda el mejor subconjunto de algoritmos de clasificación para un nuevo conjunto de datos utilizando un enfoque basado en vecinos cercanos. Para ello, en primer lugar se comparan las características del nuevo conjunto de datos con todas las metacaracterísticas de modo que se encuentre el conjunto de datos más afín. Posteriormente, el subconjunto de algoritmos que se recomiendan para el nuevo conjunto de datos corresponderá con el de su vecino más cercano.

3. Conjuntos de datos

Para este trabajo se han empleado 32 conjuntos de datos educacionales de clasificación, cuyo atributo clase refleja el rendimiento final del alumno, y que han sido obtenidos a partir de datos de uso de Moodle [11]. Como atributos predictivos, estos conjuntos de datos utilizan información variada acerca de la interacción de los alumnos con el sistema Moodle, y la clase predicha se corresponderá directamente con la calificación final obtenida por los estudiantes en el curso. Todos los datos utilizados en este trabajo se corresponden con asignaturas de Grado e Ingeniería Informática en la universidad de Córdoba, recogidos entre los años 2007 y 2012.

Entonces, para cada conjunto de datos se identificaron 16 características que se pueden agrupar bajo uno de los tres tipos siguientes, y que se muestran en el Cuadro 1:

Cuadro 1. Atributos estadísticos y del dominio de los conjuntos de datos

	Estadísticas					Complejidad			Dominio
Dataset	Ni	Nna	Nca	Nc	IR	F1	...	T2	Domain
dataset1	98	4	0	2	1,08	0,046	···	48,5	Report
dataset2	194	0	4	2	1,39	1,452	...	48,25	Report
dataset3	786	6	0	3	9,8	1,499	...	297,66	Quiz
dataset4	658	0	6	3	9,1	0,005	...	298	Quiz
dataset5	67	40	0	2	1,23	0,101	...	1,675	Quiz
dataset6	922	6	0	3	19,27	2,745	...	153,66	Quiz
dataset7	910	0	6	3	19,24	0,018	...	153,16	Quiz
dataset8	114	0	11	2	1,19	1,198	...	10,364	Forum
dataset9	42	0	11	2	6	1,029	...	3,818	Forum
dataset10	103	0	11	2	1,53	0,648	...	10,364	Forum
dataset11	114	11	0	2	1,43	0,079	...	10,364	Forum
dataset12	98	0	6	2	1,91	0,872	...	13,5	Forum
dataset13	81	6	0	2	1,19	0,094	...	13,5	Forum
dataset14	33	0	12	2	32	26,43	...	2,75	Forum
dataset15	82	0	12	2	3,1	0,11	...	6,833	Forum
dataset16	113	40	0	4	23,5	0,037	...	2,825	Quiz
dataset17	105	41	0	3	1,06	0,055	...	2,488	Quiz
dataset18	123	0	10	4	3,89	2,482	...	10,3	Quiz
dataset19	102	10	0	3	1,06	0,168	...	10,2	Quiz
dataset20	75	0	8	2	2,12	0,517	...	9,375	Report
dataset21	52	0	4	2	1,89	2,274	...	13	Report
dataset22	208	10	0	2	3,25	0,023	...	20,7	Report
dataset23	438	0	10	4	15,41	0,398	...	43,8	Report
dataset24	421	10	0	4	14,2	0,065	...	43,8	Report
dataset25	84	6	0	4	5,43	0,042	...	14	Report
dataset26	168	6	0	4	11,25	0,003	...	28	Report
dataset27	136	6	0	4	11,5	-1	...	22,667	Report
dataset28	283	0	10	2	1,67	0,842	...	28,2	Report
dataset29	155	0	10	2	1,21	0,18	...	15,5	Report
dataset30	72	6	0	4	11	-1	...	12	Report
dataset31	40	0	10	2	1,2	0,476	...	2	Quiz
dataset32	48	10	0	2	1,8	2,327	...	2	Quiz

1. Estadísticas: número de instancias o estudiantes (Ni), número de atributos numéricos (Nna), número de atributos categóricos (Nca), número de clases o etiquetas de la nota como Aprobado/Suspenso, Alta/Media/Baja, etc. (Nc), y ratio de desbalanceo (IR), que corresponde al ratio entre el número de instancias de la clase mayoritaria y la minoritaria.
2. De complejidad. Caracterizan la complejidad aparente de los conjuntos de datos para aprendizaje supervisado, como la razón discriminante de Fischer (F1), el volumen de la región de solapamiento (F2), la máxima eficiencia individual de los atributos (F3), la minimización de la suma de error en la distancia por Programación Lineal (L1), el error del clasificador lineal por Programación Lineal (L2), la

no linealidad del clasificador lineal por programación lineal (L3), la fracción de puntos en los bordes de las clases (N1), la media de la distancia de vecinos más cercanos intra/inter-clases (N2), el error del clasificador 1-NN (N3), la no linealidad del clasificador 1-NN (N4), la fracción de puntos con subconjuntos adheridos (T1) y la media de puntos por dimensión (T2). Para obtener los valores de complejidad de las medidas anteriores en nuestros conjuntos de datos hemos hecho uso de la librería DCoL (Data Complexity Library) [9].

3. Del dominio. Se ha identificado un único atributo correspondiente a este grupo, el cual indica la fuente concreta de cada conjunto de datos, que bien puede ser un informe de Moodle (*report*), un cuestionario (*quiz*) o un foro (*forum*). Un informe es un resumen general sobre la interacción de cada estudiante en Moodle, con información acerca del tiempo de uso de la herramienta, número de accesos/sesiones, número de recursos vistos, número de tareas realizadas, puntuación media en las tareas, tiempo total para efectuar las tareas, número de actividades realizadas, tiempo total para hacer las actividades, etc. Por su parte, los cuestionarios son resúmenes específicos acerca de la interacción de cada estudiante con los cuestionarios, y recogen información como el número de cuestionarios respondidos, tiempo total en hacer los cuestionarios, número de cuestionarios superados, preguntas respondidas correcta e incorrectamente, conocimiento de cada concepto evaluado en el cuestionario, etc. Por último, los conjuntos de datos de foro contienen información relativa a la interacción de cada estudiante en los foros, tal como el tiempo total en foros, número de mensajes enviados, número de mensajes leídos, número de hilos creados, número de respuestas recibidas, número de palabras y frases escritas, etc.

4. Experimentos

En primer lugar se seleccionaron un conjunto de algoritmos de clasificación de caja blanca para emplearlos en la predicción del rendimiento de los estudiantes en los conjuntos de datos de Moodle. Únicamente se han tenido en cuenta clasificadores basados en reglas y árboles de decisión debido al hecho de que proporcionan modelos que pueden ser fácilmente interpretables por los usuarios no expertos en minería de datos y, por ende, incorporados directamente en el proceso de toma de decisiones. Se han empleado los 19 algoritmos siguientes que proporciona la herramienta Weka 3.6 [12]:

- Basados en reglas: ConjunctiveRule, DecisionTable, DTNB, JRip, NNge, OneR, PART, Ridor y ZeroR.
- Basados en árboles: BFTree, DecisionStump, J48, LADTree, LMT, NBTree, RandomForest, RandomTree, REPTree y SimpleCart.

Nótese que cada algoritmo fue ejecutado sobre todos los conjuntos de datos Moodle, siguiendo una estrategia de validación cruzada estratificada en 10 particiones. En total se efectuaron 6080 ejecuciones (19 algoritmos $*$ 32 conjuntos de datos $*$ 10 particiones). Para la configuración de los parámetros de los algoritmos se emplearon sus valores por defecto recomendados. Para comparar el rendimiento de los algoritmos se emplearon las siguientes medidas de evaluación de clasificación: sensibilidad (Sen), precisión (Prec), F-Medida (F-M), Kappa (Kap) y área bajo la curva ROC (AUC). A

modo de ejemplo se recogen en el Cuadro 2 los valores medios de las diez ejecuciones de cada algoritmo en el conjunto de datos dataset1.

Cuadro 2. Atributos estadísticos y del dominio de los conjuntos de datos

Algoritmo	Sen	Prec	F-M	Kap	AUC
RConjunctiveRule	0,845	0,869	0,846	0,694	0,852
DecisionTable	0,840	0,866	0,841	0,684	0,840
DTNB	0,851	0,863	0,852	0,701	0,889
JRip	0,840	0,870	0,841	0,685	0,837
NNge	0,742	0,740	0,739	0,461	0,726
OneR	0,845	0,873	0,846	0,695	0,862
PART	0,845	0,869	0,846	0,694	0,843
Ridor	0,851	0,866	0,852	0,702	0,861
ZeroR	0,582	0,339	0,429	0,000	0,485
BFTree	0,835	0,855	0,836	0,672	0,873
DecisionStump	0,856	0,888	0,856	0,716	0,836
J48	0,845	0,869	0,846	0,694	0,847
LADTree	0,830	0,848	0,831	0,662	0,829
LMT	0,840	0,855	0,841	0,681	0,862
NBTree	0,861	0,873	0,862	0,721	0,876
RandomForest	0,840	0,855	0,841	0,681	0,854
RandomTree	0,830	0,848	0,831	0,662	0,838
REPTree	0,861	0,887	0,862	0,725	0,852
SimpleCart	0,840	0,858	0,841	0,682	0,844

En segundo lugar, con el fin de determinar qué algoritmos se comportaban mejor para cada conjunto de datos teniendo en cuenta simultáneamente las 5 medidas consideradas, hicimos uso del test de Iman&Davenport [4], un test estadístico no paramétrico de comparaciones múltiples. Por cada uno de los 32 conjuntos de datos, este test produjo una lista ordenada de algoritmos por rango (rango medio de los algoritmos sobre las cinco medidas de rendimiento), de manera que el algoritmo con mejor rango (valor de rango más bajo) es el que mejor se comporta de manera global para las medidas estudiadas. De acuerdo con el test de Iman&Davenport, si se acepta la hipótesis nula, se afirma que los algoritmos se comportan de un modo equivalente. Por contra, si se rechaza, se afirma que existen diferencias significativas en el comportamiento de los mismos. En todos los casos el test de Iman&Davenport rechazó la hipótesis nula a un nivel de significación $\alpha = 0{,}1$, lo que indica que los clasificadores no se comportan de un modo equivalente.

Para poder discernir entre qué algoritmos se producen esas diferencias significativas de rendimiento es necesario proceder con un test no paramétrico a posteriori. Dado que todos los algoritmos son comparados frente a uno de control (el de mejor rendimiento), es posible efectuar el test de Bonferroni-Dunn, centrándonos en todas las posibles comparaciones por pares. El valor crítico obtenido por este test al mismo nivel de significación $\alpha = 0{,}1$ fue de $9{,}5351$. Así pues, para cada conjunto de datos, si ese valor se

suma al rango del algoritmo de control, los algoritmos cuyo rango sea superior a dicho valor se comportan estadísticamente peor. Por tanto, los algoritmos que se pueden recomendar como adecuados para el conjunto de datos concreto serán aquellos cuyo rango pertenezca al intervalo $[rango_control, rango_control + valor_critico]$.

Por ejemplo, de entre todos los algoritmos ejecutados, el subconjunto de clasificadores que se recomendarían para el conjunto de datos $dataset1$ se recoge en el Cuadro 3, donde el intervalo crítico se corresponde con $[2, 2 + 9{,}5331]$. El resto de algoritmos no incluidos en esta tabla no se recomendarían debido al hecho de que su rango supera el valor superior de dicho intervalo.

Cuadro 3. Subconjunto de algoritmos recomendados para $dataset1$

Algoritmo	Rango
NBTree	2
REPTree	2,667
DecisionStump	6
DTNB	5,25
Ridor	5,667
OneR	6,333
ConjunctiveRule	8,083
J48	8,417
PART	8,833

Todo lo explicado hasta ahora en esta sección se correspondería con la fase que hemos denominado como fuera de línea en la Figura 1. En la fase en línea, dado un nuevo conjunto de datos educacional, el objetivo será recomendar el subconjunto de clasificadores adecuados para el mismo. A tal efecto se ha utilizado un enfoque de vecinos más cercanos (1-NN) [1] para determinar el conjunto de datos que presenta una mayor similitud con el nuevo de entre los empleados en la fase de entrenamiento, utilizando para ello la distancia euclídea normalizada sin pesos. En el caso de los atributos categóricos (el atributo *domain*), se consideró una distancia nula en caso de concordancia, y de 1 en caso contrario. De este modo, una vez se localiza el vecino más cercano, su subconjunto de algoritmos recomendados es el que se devuelve al usuario como recomendación para el nuevo conjunto de datos.

Finalmente, para comprobar la influencia del tipo de características empleadas como metacaracterísticas, llevamos a cabo experimentos utilizando diferentes combinaciones. Así, podemos determinar si los vecinos más cercanos obtenidos para cada conjunto de datos difiere en función del tipo de características tenidas en cuenta. Por ejemplo, para $dataset1$ se obtuvieron los vecinos más cercanos indicados en el Cuadro 4 según diferentes combinaciones de características.

Cuadro 4. Vecino más cercano para $dataset1$ dependido de la combinación de características

Estadísticas	Complejidad	Estadísticas + Complejidad	Estadísticas + Complejidad + Dominio
dataset13	dataset11	dataset11	dataset22

Como se puede observar en el Cuadro 4, *dataset13* es el conjunto de datos más similar a *dataset1* cuando sólo se emplean características estadísticas. Si se emplean características de complejidad únicamente, o en conjunción con las estadísticas, el más similar es *dataset11*. Finalmente, si se emplean todos los grupos de características, incluyendo también el dominio, el más similar es *dataset22*. En este caso resulta interesante observar cómo siendo *dataset1* de tipo *report*, cuando se tienen en cuenta las características estadísticas o de complejidad, los datasets más similares pertenecen al dominio *forum*. En cambio, cuando se incluye la característica dominio el dataset más cercano es *dataset22*, el cual sí que es de tipo *report*, al igual que el original. Esto demuestra cómo la característica dominio influye en gran medida a la hora de calcular la similitud entre dos conjuntos de datos, lo cuál es especialmente significativo en entornos educativos.

Entonces, utilizando el método de *hold-one-out* [3], se llevaron a cabo cuatro test separados para determinar qué combinación de metacaracterísticas permitía obtener una mejor recomendación de clasificadores en conjunto. Siguiendo este procedimiento *hold-one-out*, primero calculamos la similitud entre cada conjunto de datos y los 31 restantes para seleccionar el conjunto de datos más similar. A continuación, el subconjunto de algoritmos recomendados por cada conjunto de datos se consideró como salida real, mientras que el del vecino más cercano se consideró como salida predicha. De este modo, aplicamos medidas de evaluación utilizadas ampliamente en sistemas de recuperación de información, tales como motores de búsqueda y sistemas de recomendación, para evaluar la calidad de las recomendaciones. Las métricas empleadas fueron la *precision* y el *recall*, que se definen en términos del conjunto de documentos recuperados en dominios de recuperación de información, pero que en nuestro trabajo redefinimos en función de los algoritmos recomendados:

$$precision = \frac{|\{algoritmos_reales\} \cap \{algoritmos_predichos\}|}{|algoritmos_predichos|} \tag{1}$$

$$recall = \frac{|\{algoritmos_reales\} \cap \{algoritmos_predichos\}|}{|algoritmos_reales|} \tag{2}$$

Es importante tener en cuenta que existe una relación inversa entre las medidas de *precision* y *recall*, de tal manera que la obtención de valores superiores en una implica obtener valores más bajos para la otra. No obstante, existe una medida, denominada *F-measure*, que combina *precision* y *recall*, calculando la media armónica de ambas:

$$recall = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{3}$$

Los resultados de *F-Measure* obtenidos por las cuatro combinaciones de metacaracterísticas empleadas se muestran en los diagramas de cajas de la Figura 2, que recogen gráficamente el mínimo valor observado, el cuartil inferior, la media, el cuartil superior y el valor máximo observado. Como se puede apreciar en la figura, se obtienen mejores resultados de recomendación cuando se tienen en consideración de manera conjunta las características estadísticas y de complejidad, en lugar de por separado. Es más, los mejores resultados se obtienen cuando, además, se considera la medida del dominio.

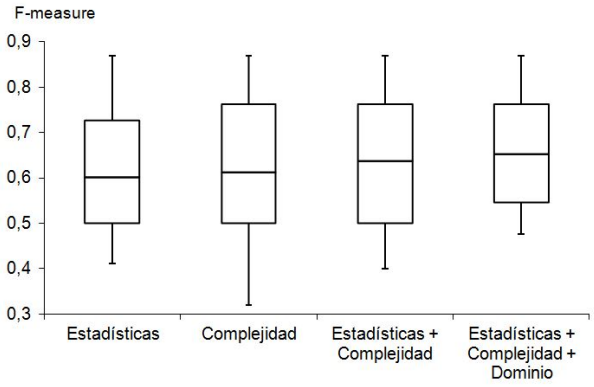


Figura 2. Diagramas de caja para la *F-measure*

5. Conclusiones

En este trabajo se ha propuesto un enfoque de meta-aprendizaje para abordar el problema de recomendación del mejor subconjunto de clasificadores de caja blanca a utilizar con conjuntos de datos educativos, obtenidos del sistema Moodle. Se han utilizado diversas medidas de rendimiento en clasificación junto con test estadísticos no paramétricos de comparaciones múltiples para asignar un rango a los clasificadores en base a su rendimiento, seleccionando el subconjunto de clasificadores que se comportan estadísticamente mejor. Los resultados muestran que la utilización de atributos de complejidad y del dominio para caracterizar conjuntos de datos permite mejorar la calidad de la recomendación.

Agradecimientos

Este trabajo ha sido financiado conjuntamente por los proyectos P08-TIC-3720 de la Junta de Andalucía, TIN-2011-22408 del Ministerio de Economía y Competitividad, y por la fundación FEDER.

Referencias

1. Aha, D.W., Kibler, D.F., Albert, M.K.: Instance-based learning algorithms. Machine Learning 6, 37–66 (1991)
2. Brazdil, P., Giraud-Carrier, C., Soares, C., Vilalta, R. (eds.): Metalearning: Applications to Data Mining. Cognitive Technologies, Springer (2009)
3. Cios, K., Pedrycz, W., Swiniarski, R., Kurgan, L.: Data Mining: A Knowledge Discovery Approach. Springer (2010)
4. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30 (2006)
5. Frank, A., Asuncion, A.: UCI machine learning repository (2010), `http://archive.ics.uci.edu/ml`
6. Hämälainen, W., Vinni, M.: Handbook of educational data mining, chap. Classifiers for educational data mining. Chapman & Hall / CRC (2011)
7. Ho, T.K., Basu, M.: Complexity measures of supervised classification problems. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(3), 289–300 (March 2002)
8. Molina, M.D.M., Romero, C., Ventura, S., Luna, J.M.: Meta-learning approach for automatic parameter tuning: A case of study with educational datasets. In: 5th International Conference on Educational Data Mining (EDM). pp. 180–183 (2012)
9. Orriols-Puig, A., Macià, N., Ho, T.: Documentation for the data complexity library in c++. Tech. rep., La Salle - Universitat Ramon-Llull (2010)
10. Romero, C., Espejo, P.G., Zafra, A., Romero, J.R., Ventura, S.: Web usage mining for predicting final marks of students that use moodle courses. Computer Applications in Engineering Education 21(1), 135–146 (2013)
11. Romero, C., Ventura, S.: Data mining in education. Wiley Interdisciplinary Rewievs: Data Mining and Knowledge Discovery 3(1), 12–27 (2013)
12. Witten, I.H., Frank, E., Hall, M.A.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, 3rd edn. (2011)
13. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. Evolutionary Computation, IEEE Transactions on 1(1), 67–82 (1997)

Clustering and Interpretation on Real Nutritional Data

Beatriz Sevilla Villanueva[1], Karina Gibert[1,2], and Miquel Sànchez-Marrè[1]
bsevilla@lsi.upc.edu, karina.gibert@upc.edu, miquel@lsi.upc.edu

[1] Knowledge Engineering and Machine Learning Group (KEMLg), Universitat Politècnica de Catalunya - BarcelonaTech, Barcelona, Spain
[2] Department of Statistics and Operations Research, Universitat Politècnica de Catalunya - BarcelonaTech, Barcelona, Spain

Abstract. Nutritional Genomics studies diet-gene-disease interactions and aims to promote health and disease prevention. It is based on the idea that everything ingested into a person's body affects the genome of the individual and, therefore, both genes and nutrients modify the same metabolic processes. This paper presents an application of clustering and interpretation over real heterogeneous data coming from a nutritional study. The individuals are clustered by their diet and physical activity habits and the resulting clustering is interpreted. This work is part of a methodology to deal with data from dietary intervention studies.

1 Introduction

Nutrition has an active role in the people's health. In last decade the efforts of nutrition are oriented to the personalized diet. They wonder *Why apparently similar persons have different responses to the same nutrient?* For answering this question several clinical studies have been performed. In addition with the enormous advances in genomics, those studies can include genetic information to obtain a new view of why nutrients affect persons in a different way [31].

This field is emerging as important new research area. The reason is that it is becoming increasingly evident that damage to genome is the most fundamental disease. A risk for developing a disease increases with DNA damage, which is dependent on nutritional status and an optimal dietary intake. These genes are involved directly or indirectly in the uptake and metabolism of nutrients [9].

In this work we will analyze part of the data coming from a nutritional genomic trial. This trial is a randomized dietary intervention study for relating the gene expression with the Mediterranean diet. This study collects different information such as biometric, sociodemographic characteristics, as well as clinical test. Also, it includes a survey about their habits both diet and physical activity. The aim of this work is to cluster the individuals depending on these habits. This is a step of a methodology which is being defined for finding the diet effect, but locally to each one of the profiles of persons, in such a way that the complex interactions can be decomposed.

The data set of diet and physical activity habits is heterogeneous. Therefore, the clustering has to be adapted to this type of data. In this work, we use a hierarchical clustering using the Gower's metric to handle both numerical and categorical attributes.

A good characterization and interpretation of the resulting clusters is very important in this kind of applications. In this paper, the interpretation of these clusters is done by means of a graphical representation of the attributes against the clusters.

2 Related Work

Advances in Nutritional Genomics are being enormous in last years. Specially since the gene expression can be translated into microarrays formats which can be afterwards analyzed. Thus, this genomic data can be involved in studies such as the dietary intervention pre-post studies. Notwithstanding, there is still a long way before a highly personalized diet can be reached [31].

The publications in nutritional genomics are multiple but not all reliable as it is explained in [27]. In this review of the gene-environmental factor interactions, it is reported that in many studies there is a lack of statistical significance, many published results are contradictory, and in addition, many studies are not reproducible.

In [14] is pointed out a need for new methodologies to analyze the efficacy of interventions when contextual factors that can interact with the intervention itself are not all well-known. These conditions make difficult to design a correct pre-post studies as they were initially conceived and decrease the reliability of the provided results. The traditional techniques used in these studies not always extract all relevant information from data.

In the particular field of nutrition, several nutritional intervention trials can be found. As an example, the one conducted in Linxian, China: Dysplasia and General Population Trial [43]. They tested the effect of multiple vitamins and minerals in the prevention of esophageal cancer. Basic statistical pairwise test are mainly used to conclude in these kind of studies.

Few works are found in this field using clustering. Next works are about environmental factors including diet but do not include an intervention. In [24], diet and other lifestyle factors are analyzed using k-means. In [37], individuals are clustered depending on their diet, physical activity and anthropometric characteristics. In other medical fields, some references are found about epidemiological pre-post studies which introduce some AI techniques for the analysis. In [5], the hypertension is studied in Korea with Insurance data by comparing logistic regression, decision trees and association rules. In [38], statistical models and association rules were used to study the comorbidity of attention deficit-hyperactivity disorder in Taiwan.

Within the specific field of nutritional genomics, clustering using data generated by gene expression profiling can be used to identify sub-populations of subjects that respond differently to a given diet intervention. Nevertheless, few

works are found using clustering. Working with animals, in [28] clustering was used to typify the different kind of diets of a set of cows with more o less fats.

An interesting antecedent is [40] focused on clustering a human subcutaneous adipose tissue gene expression data obtained during low-calorie diet intervention to aid in the prediction of 6-month weight loss maintenance. This work timidly points to a need of *local studies over the different groups of intervention.*

2.1 Cluster Interpretation

In many data mining processes there is still a gap between raw data mining results and effective decision-making. Postprocessing data mining results to approach them to the decision makers is crucial for an impact of the analysis on reality. Some works points to this issue [6, 17] and for the particular case of clustering interpretation and conceptualization of clusters is a key issue to this purpose. Some ideas have been developed in previous works [2, 15, 11, 12, 36]. The main assumption is that the resulting clusters are different, at least, in base to the data used to create them. Thus, one could expect that not every attribute will be of equal importance when describing the different groups (clusters). Therefore, a comparative analysis between the attributes that have similar values together in the different clusters can be conducted, which provides new insights into the complexity of the cluster description. In [4], they rank the attributes involved into the clustering with *logWorth*. This index is based on p-values which are assessed with the χ^2-Independence Test.

Efforts in cluster interpretation also aims at visualizing the attributes through the resulting clusters. In [21] the means of the standardized attributes are displayed for every cluster. Besides, *Class Panels Graphs* are introduced in [15] as a graphical representation in form of panel, containing attributes in the columns and clusters in the rows. Conditional distributions of the attributes against the clusters are shown per column. This visualization is interesting to understand better the meaning of the clusters [13] and it is used in this work.

3 Methodology proposal

Nowadays, results obtained from classical pre-post studies are based on traditional and often basic, statistical techniques. Till now, these techniques have not been expressive enough to allow the extraction of complex relationships. The level and degree of interactions between different subset of attributes is too complex, in this context, to be captured by simple data analysis or pre-post statistical testing.

This work focuses on finding the profiles of the individuals before they started the dietary intervention. This purpose belongs to the proposal of a general methodology [34] which will help to cluster the individuals properly for subsequently analyzing the pre-post effects. This classification allows to decompose the posterior intervention analysis in such a way that the effect can be studied locally to these profiles.

Most of the clinical studies refer to numerical data. However, categorical attributes may be also of importance for the analysis. Upon [1], three main strategies may be followed in front of mixed data: *i) Partitioning:* analyzing the dominant type; *ii) Converting:* all attributes to a unique type, conserving as much original information as possible; *iii) Compatibility measures:* allowing clustering on heterogeneous data matrices without transforming the attributes themselves. Main advantages of the later approach are no loss of information, no need to take previous arbitrary decisions which can bias results, allowing studies with all types of attributes together, and the analysis of interactions between attributes of different types. In the literature, several proposals are found [16]. In this work Gower [19] is used for those algorithms where it is available and Heterogeneous Euclidean-Overlap Metric (HEOM)[42] for the other ones.

For this first step, hierarchical clustering based on Ward's method is used [41]. The Ward's method aims at finding compact and distinguishable clusters. The Ward's method is widely used because it is a criterion related to the quantity of information of the clusters, and the resulting clusters can be often interpreted easily. Hierarchical clustering does not need number of clusters as input.

Criteria to cut the dendrogram is based on optimal Calisnki-Harabasz Index [3].Calinsky-Harabasz is a compromise between both the between-cluster distance and the within-cluster distance.

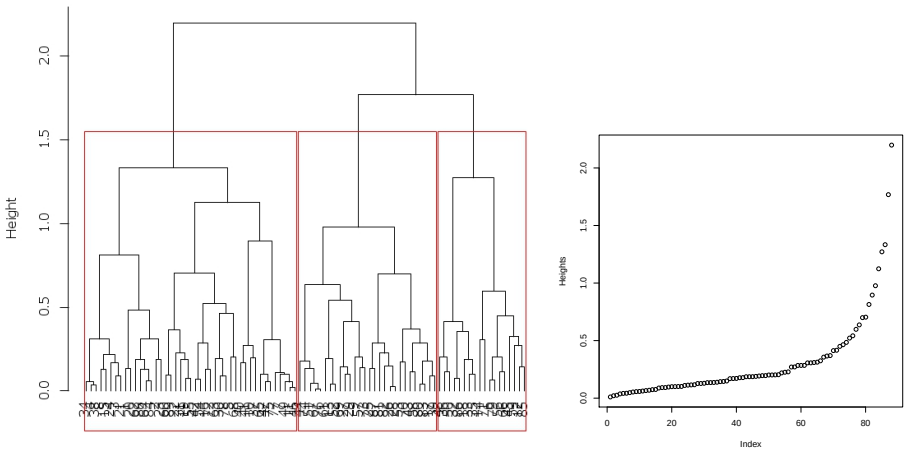
Then, the resulting partition is compared with partitions obtained by using different clustering methods, on the basis of the performance computed by 10 cluster validity indexes available in R [32]: Dunn [20], Pearson version of Hubert's gamma coefficient (Pearson)[20], average of silhouette width [33], Calinsky-Harabasz Index (CH)[3], average distance between clusters, minimum cluster separation, Separation index [23], average distance within clusters, Goodman and Kruskal's G3 index [18] and maximum cluster diameter [23].

The resulting clusters are interpreted using *Class Panel Graphs* [15] as the interpretation support tool.

4 Application & Results

As said before, the data used in this work comes from a real nutritional genomic study. Dietary habits are described through 14 categorical attributes; these attributes follow same questionnaire published in [26]. Physical activity is designed through 10 numerical attributes reporting weekly and daily activity as well as totals. Only the 4 attributes referred to the physical activity per week are used to avoid redundancies.

Therefore, the clustering is applied over these 18 selected attributes. As it was mentioned, a hierarchical clustering based on Ward's method with Gower's metrics is performed. The dendrogram of the resulting clustering is depicted in Fig. 1a. Also, in Fig. 1b, the histogram of level indexes of the dendrogram is shown. From these figures, it is possible to determine a convenient cut in 3 clusters according to max leaps in the graphics: $H = \{H_1, H_2, H_3\}$ (consistent with Calinski-Harabasz recommendation).



(a) Dendrogram cut in 3 clusters (b) Heights of level indexes

Fig. 1: Hierarchical clustering with Ward's Method for Diet and Physical Activity

Partition	Num.Clusters	Dunn	Pearson	Avg. silhouette	CH	Avg. Between	Min Separation	Separation	Avg Within	G3	Max Diameter	Partition Specifics
H	3	0.15	**0.30**	**0.13**	**13.24**	0.33	**0.09**	**0.11**	**0.27**	**0.30**	**0.58**	Ward's Method [41], Gower dist
P_1	3	0.10	**0.30**	**0.14**	**14.72**	0.33	0.06	0.07	**0.27**	**0.29**	**0.63**	PAM [25]. Gower dist, 3 clusters, selection by Calinski-Harabasz
P_2	3	**0.16**	0.27	0.12	7.04	**0.36**	**0.09**	**0.11**	0.29	0.33	**0.58**	Complete Linkage method[35], Gower dist
P_3	3	0.11	0.29	0.08	7.74	**0.34**	0.07	0.09	0.28	0.32	0.67	Model-Based Clustering [10]. 3 mixtures, model "EEE", ellipsoidal distribution, equal shapes and volumes, HOEM dist
P_4	6	**0.20**	0.29	-0.08	3.23	**0.34**	**0.13**	**0.13**	0.28	0.31	0.65	DBSCAN[8]. Epsilon:0.13, min. points:2, HOEM dist
P_5	3	0.04	0.10	0.03	4.25	0.32	0.03	0.05	0.30	0.44	0.71	k-means [29]. Max iterations: 1000, 3 clusters, HOEM dist.

Table 1: Cluster Validation Indexes. The two best values are in bold.

Table 1 describes the obtained partition, together with the results obtained with other 5 methods for comparison. R software [32] has been used for all cases, with packages: *stats, StatMatch [7], cluster[30], mclust[10], fpc[22], MASS[39].* For every partition, the 10 cluster validity indexes introduced in section 3 have been evaluated (Table 1). The first 7 indexes indicates better quality with higher values; the last 3 with lower values.

It is possible to observe from this Table 1 that H and P_1 partitions generally perform better than the rest while P_4 and P_5 have the worst index values. For instance, partition P_4 has one cluster which contains most of the instances, that is why the diameter is so high. Also, its Dunn index (based on separation) is higher because the rest of the clusters are small and isolated. Between H and P_1 small differences are observed but H improves P_1 in 4 of the 10 indexes while P_1 improves H in 3. Thus H is going to be interpreted in next section.

4.1 Cluster Interpretation

Class panel graphs are used to visualize empirical conditional distributions of attributes against clusters as it is shown in Table 2 and Table 3. Table 2 shows the 14 diet attributes and in Table 3, the original 10 physical attributes. Based on these *Class Panel Graphs* the following profiles can be described.

H_1 is a cluster that eats more vegetables but less fresh fruit. Comparing with other clusters, they consume a lower quantity of red meat, less legumes and commercial bakery. In addition, this cluster eats more fish than H_2. Their exercise is low in general, but some does from light to hard physical activity.

H_2 contains some persons that consume less oil per day. Comparing with other clusters, their consumption of red meat is higher and they use slightly more butter and takes more gas drinks. They eat less fish and nuts than other clusters. This cluster is the one which practices less physical activity.

H_3 is a cluster that eats more fresh fruit than other clusters. They eat few portions of red meat and uses less butter. This cluster eats some fish and consumes more commercial bakery and nuts than the rest of clusters. This cluster practices slightly more physical activity with a significant proportion of persons doing moderate physical activity.

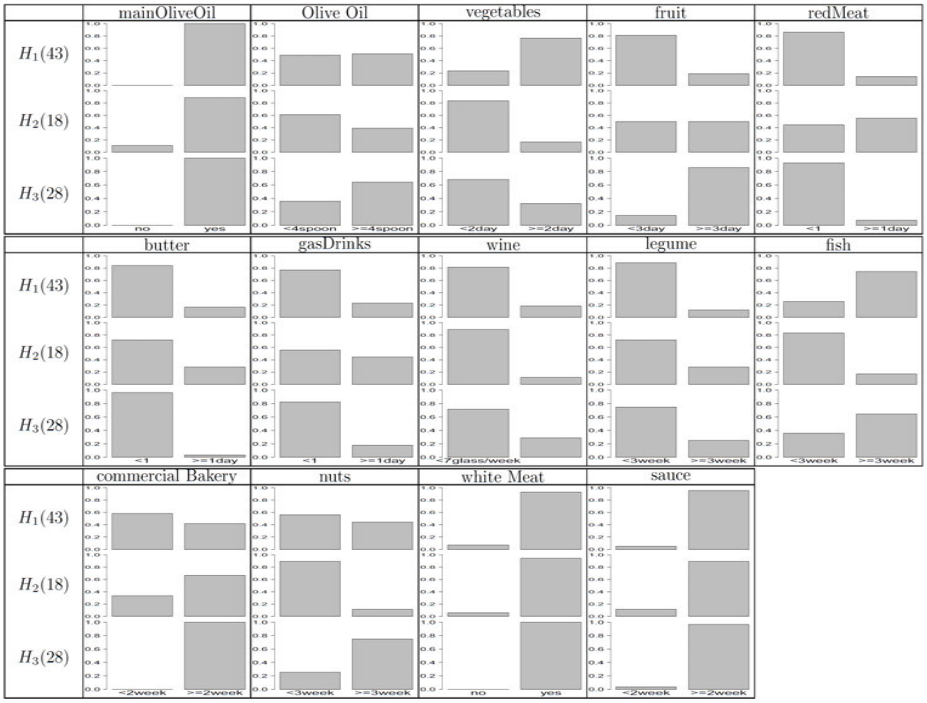


Table 2: Diet characteristics

Table 3: Physical Exercise characteristics

5 Conclusions

In this work we have introduced the drawbacks of the nutritional genomics and how AI techniques can help in extracting knowledge from data. As a first step to solve these difficulties, we have presented the classification of the individuals before the dietary intervention. Although few works are found, we are convinced that using clustering for finding patterns of response to diet may be of help to assist decision making in diet design. Therefore, clustering can bring light to this aspect because it can identify multivariate patterns in a more natural way without requiring a clear previous knowledge about the laws governing the target phenomena. Finding a prior substructure of the problem can make easier to get richer models.

Besides, the evaluation of the resulting clusters based on cluster validity indexes has been successful. Notwithstanding, the main goal achieved is the use of clustering with heterogeneous real data and the visualization to assists the interpretation of the resulting classes. Regarding cluster interpretation, not many research works are found in the literature. Methods or tools for obtaining and helping cluster interpretation seem to be a problem not yet solved.

Experts have found meaningful the composed profiles from a medical point of view. Thus, their validation of the interpretation has been positive.

Next steps will focus on finding the diet intervention effect locally to these obtaining profiles.

6 Acknowledgements

The authors acknowledge the project AGAUR (2009SGR 1365) from the catalan government oriented to support excellent research groups.

References

1. M.R. Anderberg. *Cluster Analysis for aplications.* Academic Press, 1973.
2. K. Barnard, P. Duygulu, et al. Clustering art. In *CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 2, pages II–434–II–441, 2001.
3. T. Calinski and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3:1–27, 1974.
4. W. Cecere and D.A Abreu. A method for improving list building: Cluster profiling. In *Proc. of the Survey Research Methods Section, AMSTAT*, 2010.
5. Y.M Chae, S.H Ho, et al. Data mining approach to policy analysis in a health insurance domain. *IJMI*, 62(2.3):103–111, 2001.
6. P. Cortez and M.J Embrechts. Using sensitivity analysis and visualization techniques to open black box data mining models. *Information Sciences*, 2012.
7. M. D'Orazio. *StatMatch: Statistical Matching*, 2012. R package version 1.0.5.
8. M. Ester, H.P. Kriegel, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of 2nd International Conference on Knowledge Discovery and Data Mining, KDD*, 1996.

9. M. Fenech. The human genome, nutrigenomics and nutrigenetics. *Innovation: Management, Policy & Practice*, 2008.
10. C. Fraley and A.E Raftery. Model-based clustering, discriminant analysis, and density estimation. *ASA*, 97(458):611–631, 2002.
11. K. Gibert. Automatic generation of classes interpretation as a bridge between clustering and decision making. *IJMCDM*, 2013.
12. K. Gibert, D. Conti, et al. Assisting the end-user in the interpretation of profiles for decision support. an application to wastewater treatment plants. *EEMJ*, 11(5):931–944, 2012.
13. K. Gibert, C. García-Alonso, et al. Integrating clinicians, knowledge and data: expert-based cooperative analysis in healthcare decision support. *Health research policy and systems*, 8(1):28, 2010.
14. K. Gibert and A. García. Posibilidades de aplicación de minería de datos para el descubrimiento de conocimiento a partir de la práctica clínica. *Tecnologías Aplicadas al Proceso Neurorrehabilitador*, 2008.
15. K. Gibert, A. García-Rudolph, et al. The role of kdd support-interpretation tools in the conceptualization of medical profiles: An application to neurorehabilitation. *Acta Inform Med*, 16(4):178–182, 2008.
16. K. Gibert, R. Nonell, et al. Knowledge discovery with clustering: impact of metrics and reporting phase by using klass. *Neural Network World*, 4:319–326, 2005.
17. K Gibert, G Rodríguez-Silva, et al. Post-processing: Bridging the gap between modelling and effective decision-support. the profile assessment grid in human behaviour. *Mathematical and Computer Modelling*, 2011.
18. A.D Gordon. *Classification, 2nd Edition (Chapman & Hall/CRC Monographs on Statistics & Applied Probability)*. Chapman and Hall/CRC, 2 edition, June 1999.
19. J.C. Gower. A General coefficient if similarity and some of its properties. *Biometrics*, 27:857–874, 1971.
20. M. Halkidi, Y Batistakis, et al. On clustering validation techniques. *JIIS*, 17:107–145, 2001.
21. D. Haughton, P. Legrand, and S. Woolford. Review of three latent class cluster analysis packages: Latent gold, polca, and mclust. *AMSTAT*, 63(1):81–91, 2009.
22. C. Hennig. *fpc: Flexible procedures for clustering*, 2013. R package version 2.1-5.
23. C. Hennig and T.F Liao. Comparing latent class and dissimilarity based clustering for mixed type variables with application to social stratification. Technical report, Technical report, 2010.
24. K.F Hulshof, M. Wedel, et al. Clustering of dietary variables and other lifestyle factors (dutch nutritional surveillance system). *JECH*, 46(4):417–424, 1992.
25. L. Kaufman and P.J. Rousseeuw. Clustering by means of medoids. *Statistical Data Analysis Based on the L1–Norm and Related Methods*, pages 405–416, 1987.
26. V. Konstantinidou, M.A Covas, et al. In vivo nutrigenomic effects of virgin olive oil polyphenols within the frame of the mediterranean *FASEB J*, 24(7):2546–57, 2010.
27. Y. Lee, C. Lai, J.M. Ordovas, and L. Parnell. A database of gene-environment interactions pertaining to blood lipid traits, cardiovascular disease and type 2 diabetes. *JDMGP*, 2(1):1–8, 2011.
28. J.J Loor, M. Bionaz, et al. Systems biology and animal nutrition: insights from the dairy cow during growth and the lactation cycle. *Systems Biology and Livestock Science*, 2011.
29. J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. of the 5th Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. California, USA, 1967.

30. M. Maechler, P. Rousseeuw, et al. *cluster: Cluster Analysis Basics and Extensions*, 2012. R package version 1.14.2.
31. J.M Ordovas and V. Mooser. Nutrigenetics and nutrigenomics. *Casopis Lekaru Ceskych*, 146(2):837–839, 2007.
32. R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2012.
33. P.J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.
34. B. Sevilla, M. Sànchez-Marrè, and others. Using ai in nutritional genomics. In *Poster in Sematica 2012, Granada*, November 2012.
35. R. Sibson. Slink: an optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1):30–34, 1973.
36. M. Siponen, J. Vesanto, et al. An approach to automated interpretation of som. In *Advances in Self-Organising Maps*, pages 89–94. Springer, 2001.
37. S. Swaminathan, T. Thomas, et al. Clustering of diet, physical activity and overweight in parents and offspring in south india. *EJCN*, 2012.
38. Y.M Tai and H.W Chiu. Comorbidity study of adhd: Applying association rule mining (arm) to national health insurance database of taiwan. *IJMI*, 78(12):e75 – e83, 2009.
39. W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, New York, fourth edition, 2002. ISBN 0-387-95457-0.
40. N. Viguerie, C. Poitou, et al. Transcriptomics applied to obesity and caloric restriction. *Biochimie*, 87(1):117 – 123, 2005.
41. J.H Ward Jr. Hierarchical grouping to optimize an objective function. *ASA*, 58(301):236–244, 1963.
42. D. Randall Wilson and Tony R. Martinez. Improved heterogeneous distance functions. *Journal of Artificial Intelligence Research*, 6:1–34, 1997.
43. X.N Zou, P.R Taylor, et al. Seasonal variation of food consumption and selected nutrient intake in linxian, a high risk area for esophageal cancer in china. *Int J Vitam Nutr Res*, 72(6):375–382, Dec 2002.

17-20 sept 2013
MULTICONFERENCIA
CAEPIA'13
Madrid

I Simposio de Fusión de la Información y Ensembles [FINO 2013]

FINO'13

Editor asociado:
Emilio Corchado

1466

Prefacio

Este volumen contiene los trabajos aceptados para su presentación oral en el primer simposium FINO (Fusión de la información y ensembles), desarrollo bajo el marco de CAEPIA-CEDI 2013, los días 17 al 20 de septiembre en Madrid, España.

El objetivo de FINO es convertirse en un foro de encuentro para investigadores tanto del entorno universitario como empresarial que trabajan en temas asociados a la Fusión de la información y los ensembles.

La selección de los artículos fue extremadamente rigurosa para mantener una alta calidad y por ello queremos agradecer a los miembros del Comité Organizador por el gran trabajo realizado.

Hector Quintián
Emilio Corchado

Organización

Comité Organizador

Emilio Corchado (PC Co-Chair)	Universidad de Salamanca
Álvaro Herrero	Universidad de Burgos
Bruno Baruque	Universidad de Burgos
Héctor Quintián	Universidad de Salamanca
Roberto Vega	Universidad de Salamanca
Jose Luis Calvo	Universidad de Coruña
Ángel Arroyo	Universidad de Burgos
Laura García-Hernández	Universidad de Cordoba
Vicente Vera González	Universidad Complutense de Madrid

Comité de Revisores

Ajith Abraham, Estados Unidos
Jose L. Casteleiro, España
Juan Corchado , España
Laura Garcia-Hernandez, España
Cesar Hervas, España
Jan Platos, República Checa
Raquel Redondo, España
Vaclav Snasel, República Checa
Vicente Vera, España
Michal Wozniak, Polonia
Jose L. Calvo Rolle, España
Emilio Corchado, España
Richard Duro, España
Manuel Grana, España
Pavel Kromer, República Checa
Héctor Quintián, España
Dragan Simic, Serbia
Mª Belen Vaquerizo, España
José Ramón Villar, España

Sponsors

- IEEE Systems, Man and Cybernetics Society

IEEE Systems, Man and Cybernetics Society
Spain

Selección dinámica de clasificadores para la estrategia Uno-contra-Uno: Evitando los clasificadores no competentes

Mikel Galar[1], Alberto Fernández[2], Edurne Barrenechea[1], Humberto Bustince[1], and Francisco Herrera[3]

[1] Departamento de Automática y Computación,
Universidad Pública de Navarra, 31006, Pamplona, España
{mikel.galar,edurne.barrenechea,bustince}@unavarra.es
[2] Departamento de Ciencias de la Computación,
Universidad de Jaén, 23071, Jaén, España
alberto.fernandez@ujaen.es
[3] Departamento de Ciencias de la Computación e Inteligencia Artificial,
Universidad de Granada, 18071, Granada, España
herrera@decsai.ugr.es

Resumen Una de las estrategias de descomposición más utilizada a la hora de afrontar problemas multi-clase es la estrategia Uno-contra-Uno, que divide el problema multi-clase original en tantos nuevos subproblemas como pares de clases puedan considerarse. Posteriormente, cada uno es afrontado por un clasificador binario independiente. Las nuevas instancias se clasifican combinando las salidas obtenidas por cada clasificador base. Sin embargo, esta forma de descomposición trae consigo ciertos problemas a la hora de la combinación, como es el denominado problema de los clasificadores no competentes. Se definen como aquellos cuya salida no es relevante para la clasificación de la instancia, es decir, no fueron aprendidos con instancias de la clase real del ejemplo a clasificar. En este trabajo proponemos un modelo de selección dinámica de clasificadores para la estrategia Uno-contra-Uno con la intención de evitar estos clasificadores no competentes, para lo que consideraremos el vecindario de la instancia a clasificar. Estudiamos la validez del modelo propuesto mediante un estudio experimental exhaustivo con diferentes clasificadores base y basados en un estudio estadístico adecuado.

Palabras clave: multi-clase, uno-contra-uno, descomposición, no competencia, selección de clasificadores

1. Introducción

Los problemas de clasificación se pueden dividir en dos tipos dependiendo del número de clases que consideran. Los problemas binarios tratan únicamente con un par de clases, mientras que los multi-clase consideran más de dos clases y por tanto son más generales. Normalmente, la clasificación con múltiples

clases es más difícil debido a la complejidad en el establecimiento de las fronteras de decisión. Adicionalmente a esta dificultad, inherente en los problemas multi-clase, algunos sistemas de clasificación están intrínsecamente diseñados para trabajar con solo dos clases, y su extensión a problemas con múltiples clases no está todavía establecida; por ejemplo, este es el caso de las Máquinas de Soporte Vectorial (*Support Vector Machines*, SVMs) [21] o el *Positive Definite Fuzzy Classifier* (PDFC) [4]. En estos casos, la forma natural de afrontar los problemas multi-clase es mediante técnicas de descomposición [17], que tratan de dividir el problema multi-clase original en problemas binarios más sencillos de resolver. Debemos destacar que este tipo de técnicas han mostrado su validez incluso en el caso en el que los clasificadores base utilizados puedan afrontar el problema multi-clase de forma directa [9].

En la literatura especializada existen multitud de estrategias de descomposición [17]. Entre ellas, la estrategia Uno-contra-Uno (*One-vs-One*, OVO) [15] es una de las más conocidas y utilizadas. En OVO se obtienen tantos nuevos subproblemas como pares de clases podamos obtener; cada uno de ellos es afrontado por un clasificador base independiente. Finalmente, una nueva instancia se clasifica sometiéndola a todos los clasificadores base y combinando sus salidas. Esta estrategia tan simple ha mostrado su potencial en trabajos previos [9], siendo además utilizada como estándar para afrontar problemas multi-clase con SVMs en los paquetes software más utilizados como WEKA [12], LIBSVM [3] o KEEL [2].

Una vez fijada la estrategia de descomposición, la forma en la que se agregan las salidas de los clasificadores es el siguiente factor clave para la clasificación. En [9] se llevó a cabo un extenso estudio experimental comparando las agregaciones del estado del arte para la estrategia OVO. Las estrategias analizadas iban desde las de estimación de probabilidades [24] hasta los modelos basados en relaciones de preferencia [7], entre otras. Uno de los problemas de OVO que más atención ha recibido por parte de los investigadores ha sido el de la región no clasificable cuando se considera la estrategia del voto simple. Sin embargo, esta atención no se ha visto reflejada en una mejora significativa de los resultados frente a otros modelos más simples como el voto ponderado [14]. Otro de los problemas mencionados en [9] es el de los clasificadores no competentes, cuyo estudio se indicó como una posible línea futura de trabajo prometedora ya que todavía no ha sido tratado de manera directa. La no competencia es inherente al método de descomposición de OVO; cada clasificador es entrenado únicamente con instancias de las dos clases que debe distinguir, ignorando las instancias del resto de clases. Es decir, son clases desconocidas para el clasificador y por tanto las salidas de estos clasificadores son irrelevantes para la clasificación de instancias de dichas clases. Sin embargo, estas salidas son agregadas de la misma forma que lo son las relevantes por lo que pueden llevar a etiquetar incorrectamente la instancia en determinadas situaciones.

Evidentemente, la competencia de los clasificadores no puede ser conocida a priori, ya que en ese caso el problema de clasificación estaría resuelto. Por ello, en este trabajo nuestra intención es presentar una nueva estrategia de agregación

basada en el concepto de Selección Dinámica de Clasificadores (SDC) [11,16] que pueda reducir el número de clasificadores no competentes que se utilizan en la combinación. De esta forma, podrían evitarse algunas clasificaciones erróneas debido a estos clasificadores. Mediante este modelo trataremos de considerar solo aquellos clasificadores que son probablemente competentes. Para ello, analizaremos el vecindario de la instancia a clasificar y a partir de este seleccionaremos el subconjunto de clasificadores para la fase de agregación.

En la literatura los métodos de SDC están principalmente dirigidos a conjuntos de clasificadores en los que todos los clasificadores son capaces de discernir entre todas las clases (cada uno está especializado en un área del espacio de entrada) [23]; por contra, su aplicación en la estrategia OVO todavia no ha sido estudiada, probablemente debido a que su aplicación es más difícil y está más restringida, ya que OVO establece el área de competencia de cada clasificador a priori, y no depende del espacio de entrada sino del de salida.

Para evaluar la validez del modelo propuesto hemos desarrollado un estudio experimental que es el mismo que hemos utilizado en [9], de tal forma que podemos basarnos en las conclusiones obtenidas en dicho trabajo para reducir la comparativa. Consideramos diecinueve conjuntos de datos del repositorio de KEEL [2,1], contrastamos los resultados mediante los test estadísticos adecuados [6,10] y comparamos nuestro modelo utilizando clasificadores base provenientes de diferentes paradigmas.

El resto del trabajo esta organizado de la siguiente forma: en la Sección 2 recordamos la estrategia OVO e introducimos nuestro modelo dinámico. En la Sección 3 mostramos el marco experimental utilizado para la comparativa que llevamos a cabo en la Sección 4. Finalmente, la Sección 5 concluye este trabajo.

2. Selección dinámica de clasificadores para la estrategia Uno-contra-Uno

En esta sección introducimos nuestro modelo para evitar los clasificadores no competentes en OVO. Para ello, comenzamos recordando el funcionamiento de dicha estrategia (Subsección 2.1), posteriormente introducimos el modelo dinámico (Subsección 2.2) y mostramos un ejemplo sencillo de su funcionamiento (Subsección 2.3).

2.1. Estrategia Uno-contra-Uno

En OVO se divide un problema de m clases en $m(m-1)/2$ subproblemas binarios (todos los posibles pares de clases) que son afrontados por clasificadores base independientes. Una nueva instancia se clasifica obteniendo la salida de cada uno de los clasificadores. Cada clasificador que distingue un par de clases $\{C_i, C_j\}$ devuelve un grado de confianza $r_{ij} \in [0,1]$ en favor de la clase C_i ($r_{ji} = 1 - r_{ij}$). Estas salidas pueden almacenarse en lo que se denomina matriz

de votos de la siguiente forma:

$$R = \begin{pmatrix} - & r_{12} & \cdots & r_{1m} \\ r_{21} & - & \cdots & r_{2m} \\ \vdots & & & \vdots \\ r_{m1} & r_{m2} & \cdots & - \end{pmatrix} \tag{1}$$

Para establecer la clase a la que pertenece la instancia puede usarse cualquiera de las agregaciones existentes en la literatura [9] sobre la matriz de votos. Una de las estrategias más utilizadas por su robustez es la estrategia del voto ponderado (*Weighted Voting*, WV) [14]) en donde cada confianza devuelta por los clasificadores base vota por la clase correspondiente y aquella que alcanza la mayor confianza total es predicha: $Class = arg \max_{i=1,\ldots,m} \sum_{1 \leq j \neq i \leq m} r_{ij}$

2.2. OVO Dinámico

Tal y como hemos expuesto anteriormente, nuestro objetivo es tratar de evitar aquellos clasificadores no competentes que pueden perjudicar a la decisión final del sistema cuando alguno de los clasificadores que distinguen la clase correcta han fallado (e incluso cuando no fallan, ver ejemplo en la Subsección 2.3). Para ello, consideramos el uso de modelos SDC; sin embargo, estos modelos de selección dinámica no se pueden aplicar directamente sobre OVO a diferencia de lo que ocurre con los conjuntos de clasificadores clásicos, ya que no podemos establecer las regiones en el espacio de entrada para cada clasificador base, ni podemos estimar sus precisiones locales. Por tanto, debemos adaptar estas técnicas al marco de trabajo de OVO, ya que los clasificadores base con los que trabajamos son únicamente competentes para las dos clases utilizadas en su entrenamiento. Evidentemente, no podemos restringir nuestra búsqueda de los clasificadores competentes a aquellos que han considerado la clase real en el entrenamiento, la cual lógicamente es desconocida; pero podemos intentar buscar un pequeño subconjunto de clases a las cuales la pertenencia de la instancia sea más probable. De este modo, podemos considerar una nueva matriz de votos donde consideramos únicamente los clasificadores entrenados con dichas clases; por tanto, podemos reducir el número de clasificadores en la clasificación eliminando aquellos de los que estamos suficientemente seguros de que no contribuyen a la decisión correcta, o que incluso pueden perjudicarla.

Siguiendo esta filosofía y de manera similar a modelos SDC previos [23], consideramos el uso de los vecinos de la instancia a ser clasificada para decidir si un clasificador es competente o no. Sin embargo, en nuestro caso, en vez de utilizar los vecinos para estimar la precisión local de los clasificadores, los utilizaremos para seleccionar las clases que tomarán parte en la matriz de votos reducida. El proceso de selección dinámica propuesto lo detallamos en los siguientes pasos:

1. Calcular los k vecinos más cercanos (k *Nearest Neighbors*, kNN) de la instancia a clasificar (k es un parámetro del método, actualmente fijado a $3 \cdot m$).

2. Obtener las clases presentes en dicho vecindario (si entre los k vecinos solo hay una clase, se aumenta el número de vecinos hasta un máximo de $2 \cdot k$, en cuyo caso utilizamos el OVO estándar).
3. Considerar una nueva matriz de votos en las que únicamente están presentes los clasificadores entrenados con pares de clases del subconjunto seleccionado.
4. Aplicar cualquiera de las agregaciones existentes para OVO [9].

La simplicidad del método es una importante ventaja, ya que consiste en utilizar kNN dinámicamente para hacer una pre-selección de clases sobre un amplio número de vecinos (de ahí el por qué de los valores de los parámetros). Gracias a ello, es difícil fallar una instancia por la eliminación de la clase correcta de la matriz de votos. Una vez llevada a cabo la pre-selección dinámica, podemos seguir utilizando cualquiera de las agregaciones existentes para OVO [9]. En nuestro caso, proponemos el uso del WV ya que su robustez ha sido probada tanto teórica [14] como empíricamente [9].

2.3. Ejemplo ilustrativo

Para mostrar como funciona el modelo propuesto, a continuación presentamos un ejemplo ilustrativo donde resolvemos el problema de la no competencia. Supongamos que tenemos que clasificar una instancia x, cuya clase real es c_1 y cuya matriz de votos OVO obtenida presentando dicha instancia a los clasificadores binarios es la siguiente:

$$R(x)= \begin{pmatrix} & \mathbf{c_1} & \mathbf{c_2} & \mathbf{c_3} & \mathbf{c_4} & \mathbf{c_5} \\ \mathbf{c_1} & - & 0{,}55 & 0{,}6 & 0{,}75 & 0{,}7 \\ \mathbf{c_2} & 0{,}45 & - & 0{,}4 & 1 & 0{,}8 \\ \mathbf{c_3} & 0{,}4 & 0{,}6 & - & 0{,}5 & 0{,}4 \\ \mathbf{c_4} & 0{,}25 & 0{,}0 & 0{,}5 & - & 0{,}1 \\ \mathbf{c_5} & 0{,}3 & 0{,}2 & 0{,}6 & 0{,}9 & - \end{pmatrix} \tag{2}$$

Si aplicamos la estrategia WV para obtener la clase final sobre $R(x)$, obtendríamos la clase c_2 como salida (Ecuación 3). Sin embargo, ninguno de los clasificadores que considera la c_1 en el entrenamiento ha fallado (primera fila); pero el resto de clasificadores que distinguen a la clase c_2, a pesar de su no competencia (la cual es la fuente del fallo), han votado fuertemente por ella, y por eso en este caso sería la predicha.

$$R(x)= \left(\begin{array}{cccccc|c} & \mathbf{c_1} & \mathbf{c_2} & \mathbf{c_3} & \mathbf{c_4} & \mathbf{c_5} & \text{WV} \\ \mathbf{c_1} & - & 0{,}55 & 0{,}6 & 0{,}75 & 0{,}7 & 2{,}6 \\ \mathbf{c_2} & 0{,}45 & - & 0{,}4 & 1 & 0{,}8 & \mathbf{2.65} \\ \mathbf{c_3} & 0{,}4 & 0{,}6 & - & 0{,}5 & 0{,}4 & 1{,}9 \\ \mathbf{c_4} & 0{,}25 & 0{,}0 & 0{,}5 & - & 0{,}1 & 0{,}85 \\ \mathbf{c_5} & 0{,}3 & 0{,}2 & 0{,}6 & 0{,}9 & - & 2{,}1 \end{array}\right) \tag{3}$$

En el caso de aplicar el modelo propuesto en vez del WV sobre la misma matriz de votos, debemos calcular primero los k vecinos más cernos de x (donde $k = 3 \cdot 5$ clases $= 15$ vecinos). Supongamos que el subconjuto de clases en este vecindario es el formado por las clases $\{c_1, c_4, c_5\}$. Por consiguiente, eliminamos de la matriz de votos aquellos clasificadores que no consideran pares de clases del subconjunto (Ecuación 4), es decir, cualquier clasificador que haya sido entrenado

con instancias de las clases $\{c_2, c_3\}$. Finalmente, aplicando el WV sobre la nueva matriz de votos. En este caso, la clase predicha (c_1) es la correcta.

$$R_{dyn(x)}=\left(\begin{array}{c|cccccc} & \mathbf{c_1} & \mathbf{\sout{c_2}} & \mathbf{\sout{c_3}} & \mathbf{c_4} & \mathbf{c_5} & \text{WV} \\ \mathbf{c_1} & - & \sout{0.55} & \sout{0.6} & 0{,}75 & 0{,}7 & \mathbf{1.45} \\ \mathbf{\sout{c_2}} & \sout{0.45} & - & \sout{0.4} & \sout{1} & \sout{0.8} & - \\ \mathbf{\sout{c_3}} & \sout{0.4} & \sout{0.6} & - & \sout{0.5} & \sout{0.4} & - \\ \mathbf{c_4} & 0{,}25 & \sout{0.0} & \sout{0.5} & - & 0{,}1 & 0{,}35 \\ \mathbf{c_5} & 0{,}3 & \sout{0.2} & \sout{0.6} & 0{,}9 & - & 1{,}2 \end{array}\right) \tag{4}$$

3. Marco Experimental

En esta sección presentamos el marco experimental utilizado para el desarrollo de la comparativa experimental de la Sección 4. Debemos destacar que todo el marco experimental utilizado coincide con el de [9] y por tanto, partimos de las conclusiones obtenidas en dicho estudio para llevar a cabo un estudio experimental directo, en el que comparamos nuestro modelo contra los mejores encontrados en dicho trabajo. Primero describimos los clasificadores considerados y sus parámetros en la Subsección 3.1. Después, en la Subsección 3.2 recordamos las mejores estrategias de agregación para cada clasificador base en [9]. Posteriormente, detallamos los conjuntos de datos y los tests estadísticos considerados Subsección 3.3.

3.1. Clasificadores base y parámetros

Hemos considerado diferentes clasificadores base de diferentes paradigmas para comparar nuestro modelo en situaciones diferentes. Los algoritmos de aprendizaje utilizados son los siguientes: Las máquinas de soporte vectorial **SVM** [21], el árbol de decisión **C4.5** [20], los k vecinos más cercanos k**NN** [18], el aprendizaje de reglas **Ripper** [5], y el clasificador difuso **PDFC** [4].

La mayoría de los métodos de combinación para OVO llevan a cabo su predicción en base a la confianza dada por los clasificadores base para la decisión. En este trabajo obtenemos las confianzas para cada uno de los anteriores de la siguiente forma:

- **SVM:** Modelo logístico para la estimación de probabilidades [19].
- **C4.5:** Pureza de la hoja que lleva a cabo la predicción.
- k**NN:** Confianza $= \frac{\sum_{l=1}^{k} \frac{e_l}{d_l}}{\sum_{l=1}^{k} \frac{1}{d_l}}$ donde d_l es la distancia entre la instancia y el vecino l-ésimo y $e_l = 1$ si el vecino l es de la clase y 0 en otro caso.
- **Ripper:** Pureza de la regla que realiza la predicción.
- **PDFC:** Confianza total para la clase predicha.

En caso de que ocurran empates, la clase mayoritaria es la predicha, si el empate continúa, la clase se selecciona de manera aleatoria.

Los parámetros utilizados en cada clasificador base pueden verse en la Tabla 1. Estos parámetros son comunes para todos los problemas y han sido seleccionados de acuerdo a las recomendaciones de los autores. Hacer notar que usamos

dos configuraciones de SVM para comprobar la robustez de la propuesta. Finalmente, para el modelo dinámico propuesto utilizamos la distancia Euclidea para la búsqueda de los vecinos más cercanos (excepto cuando el conjunto de datos contiene valores nominales que utilizamos la distancia HVDM).

Tabla 1. Especificación de parámetros.

Algoritmo	*Parámetros*
SVM_{Poly}	C = 1.0, Tolerancia = 0.001, Epsilon = 1.0E-12, Kernel Type = Polinomial, Grado polinomio = 1
SVM_{Puk}	C = 100.0, Tolerancia = 0.001, Epsilon = 1.0E-12 Kernel Type = Puk, PukKernel ω = 1.0, PukKernel σ = 1.0
C4.5	Poda = True, Nivel confianza = 0.25, Mínimo número de item-sets por hoja = 2
3NN	k = 3, Distancia = Heterogeneous Value Difference Metric (HVDM)
Ripper	Tamaño del subconjunto = 66 %, Repeticiones de la optimización = 2
PDFC	C = 100.0, Tolerancia = 0.001, Epsilon = 1.0E-12, Kernel Type = Polinomial Grado Polinomio = 1, Tipo PDRF = Gaussiano

3.2. Agregaciones consideradas

Utilizamos como agregaciones representativas para cada clasificador las mismas seleccionadas en [9]. La única excepción es con las SVMs, ya que no se encontraron diferencias significativas entre los métodos y el modelo más utilizado es el de Wu *et al.* [24], que además nos permite utilizar las mismas matrices OVO para todos los métodos. Por tanto, las agregaciones para cada clasificador son las siguientes: **SVM** – PE (Wu *et al. Probability Estimates by Pairwise Coupling* [24]). **C4.5** – WV (*Weighted Voting strategy*). k**NN** – ND (*Non-Dominance criterion* [7]). **Ripper** – WV (*Weighted Voting strategy*). **PDFC** – PC (*Probability Estimates by Pairwise Coupling* [13]).

Una descripción completa y detallada de los métodos puede consultarse en [8]. Debemos recalcar que todos los métodos utilizan exactamente las mismas matrices de votos y que por tanto todas las diferencias mostradas en el estudio experimental se deben únicamente a la estrategia de agregación.

3.3. Conjuntos de datos y evaluación

Consideramos los mismos diecinueve conjuntos de datos del repositorio KEEL [1] utilizados en [9], cuyas propiedades se muestran en la Tabla 2. Utilizamos un modelo de validación cruzada de 5 particiones para obtener los resultados de precisión de cada método. Para llevar a cabo una evaluación honesta de los resultados, utilizamos test estadísticos no paramétricos [6,10]. Dado que realizamos comparativas por pares utilizaremos el test de Wilcoxon [22].

4. Estudio experimental

En esta sección vamos a estudiar la utilidad del modelo propuesto. Para ello, comparamos nuestro modelo dinámico frente a las mejores estrategias del estado del arte para cada clasificador base. Nuestro objetivo es investigar si el hecho de evitar los clasificadores no competentes se traduce en una mejora de los

Tabla 2. Descripción de los conjuntos de datos.

C. datos	#Ej.	#Atr.	#Num.	#Nom.	#Cl.
Car	1728	6	0	6	4
Lymphography	148	18	3	15	4
Vehicle	846	18	18	0	4
Cleveland	297	13	13	0	5
Nursery	1296	8	0	8	5
Page-blocks	548	10	10	0	5
Autos	159	25	15	10	6
Dermatology	358	34	1	33	6
Flare	1066	11	0	11	6
Glass	214	9	9	0	7
Satimage	643	36	36	0	7
Segment	2310	19	19	0	7
Shuttle	2175	9	9	0	7
Zoo	101	16	0	16	7
Ecoli	336	7	7	0	8
Led7digit	500	7	0	7	10
Penbased	1100	16	16	0	10
Yeast	1484	8	8	0	10
Vowel	990	13	13	0	11

resultados; o por contra, si al tratar de evitarlos mediante el mecanismo dinámico estamos también eliminando las clases que deben ser predichas del proceso de decisión.

La Tabla 3 muestra los resultados de precisión en test de los diferentes métodos. Mostramos para cada clasificador base el resultado de la mejor agregación anterior y el de la estrategia dinámica (Dyn) con el WV. Podemos comprobar como la propuesta funciona aparentemente bien en todos los clasificadores base, aumentando la precisión media en todos ellos y destacando en un gran número de conjuntos de datos. Sin embargo, para obtener conclusiones relevantes debemos llevar a cabo el estudio estadístico adecuado, en este caso, mediante el test de Wilcoxon, cuyos resultados se muestran en la Tabla 4.

Tabla 3. Resultados de precisión media en test para las agregaciones representativas y la estrategia dinámica (con WV) para cada clasificador base.

	3NN		C4.5		Ripper		SVM_{Poly}		SVM_{Puk}		PDFC	
C. datos	ND	Dyn^{WV}	WV	Dyn^{WV}	WV	Dyn^{WV}	PE	Dyn^{WV}	PE	Dyn^{WV}	PC	Dyn^{WV}
autos	76.75	**77.38**	**81.17**	79.92	78.65	**79.27**	74.80	**75.42**	**68.53**	67.88	76.71	**78.61**
car	92.82	**93.06**	93.00	**93.29**	91.32	**92.36**	**92.71**	92.65	63.60	**84.72**	99.88	**99.94**
cleveland	56.58	**56.58**	51.53	**52.53**	48.45	**49.81**	**58.25**	57.59	45.09	**45.42**	52.83	**53.85**
derma.	89.96	**93.58**	96.37	**98.31**	93.02	**93.30**	94.13	**94.69**	**96.09**	**96.09**	84.36	**92.18**
ecoli	**80.96**	80.66	79.47	**80.07**	**75.89**	75.60	77.69	**77.99**	75.31	**75.90**	**82.75**	**82.75**
flare	**72.14**	71.95	74.20	**74.29**	**72.61**	72.61	74.67	**75.33**	69.42	**72.89**	**72.98**	**72.98**
glass	**73.38**	**73.38**	70.53	**71.47**	**74.30**	**74.30**	61.26	**62.66**	70.60	**71.09**	68.25	**70.11**
led7digit	72.60	**72.80**	72.20	**72.60**	70.00	**71.60**	73.00	**74.00**	70.20	**71.40**	71.80	**72.60**
lymph.	**83.72**	**83.72**	73.63	**76.30**	76.28	**77.63**	**81.68**	**81.68**	**80.34**	**80.34**	**78.94**	**78.94**
nursery	**92.52**	**92.52**	**89.04**	88.89	**89.43**	89.28	**91.90**	**91.90**	81.33	**89.12**	**96.84**	**96.84**
pageblocks	94.70	**94.88**	95.61	**95.79**	**95.25**	95.25	**94.70**	94.52	94.16	**94.34**	**95.43**	**95.43**
penbased	96.18	**96.27**	90.64	**90.73**	89.91	**90.82**	95.27	**95.64**	97.82	**97.91**	**98.27**	**98.27**
satimage	**86.47**	86.32	81.65	**82.74**	79.78	**80.40**	**84.14**	83.67	84.92	**85.08**	**87.41**	87.10
segment	96.71	**96.97**	**97.06**	96.97	95.84	**96.41**	92.55	**92.60**	**97.10**	97.06	**96.62**	96.54
shuttle	99.54	**99.63**	**99.72**	99.68	99.49	**99.68**	96.37	**97.52**	**99.72**	99.68	97.47	**98.21**
vehicle	**71.40**	**71.40**	71.39	**72.81**	71.04	**71.75**	72.46	**73.29**	80.49	**80.61**	83.57	**83.69**
vowel	**95.86**	**95.86**	80.00	**81.21**	**78.99**	78.48	69.90	**70.30**	**99.39**	**99.39**	**97.98**	**97.98**
yeast	**54.72**	54.52	**59.91**	59.57	56.00	**56.07**	**59.10**	58.96	**56.54**	56.27	**59.10**	59.03
zoo	92.10	**94.10**	**93.10**	**93.10**	94.10	**95.10**	95.05	**96.05**	84.19	**85.19**	**97.05**	**97.05**
Mean	83.11	**83.45**	81.59	**82.12**	80.54	**81.04**	81.03	**81.39**	79.73	**81.60**	84.12	**84.85**

Los resultados de los test estadísticos son claros, siendo la estrategia dinámica significativamente mejor que las representativas para cada clasificador base en cinco de los seis casos estudiados (todos menos 3NN), con p-valores muy bajos. En el caso de 3NN, aunque el test no se rechaza, los rangos están a favor de la estrategia dinámica con un p-valor bajo. Por tanto, estos resultados muestran la superioridad de la estrategia dinámica frente a los métodos clásicos. Además

Tabla 4. Comparativa de agregaciones mediante el test de Wilcoxon. R^+ corresponde a la suma de rangos para la agregación clásica y R^- para la dinámica.

Clasificador	Comparativa	R^+	R^-	Hypótesis	p-valor
3NN	Dyn^{WV} vs. ND	133.5	56.5	No Rechazada	0.140146
C4.5	Dyn^{WV} vs. WV	153.5	36.5	**Rechazada para Dyn^{WV} at 5 %**	0.017621
Ripper	Dyn^{WV} vs. WV	162.5	27.5	**Rechazada para Dyn^{WV} at 5 %**	0.005684
SVM_{Poly}	Dyn^{WV} vs. PE	149.5	40.5	**Rechazada para Dyn^{WV} at 5 %**	0.024520
SVM_{Puk}	Dyn^{WV} vs. PE	154.0	36.0	**Rechazada para Dyn^{WV} at 5 %**	0.015086
PDFC	Dyn^{WV} vs. PC	138.0	52.0	**Rechazada para Dyn^{WV} at 5 %**	0.040860

queda claro que el evitar algunos de los clasificadores no competentes puede llevar a mejoras significativas de los resultados, lo que a su vez pone de manifiesto este problema en la estrategia OVO.

5. Conclusiones

En este trabajo hemos presentado una nueva estrategia para agregar los resultados de los clasificadores base en la estrategia OVO mediante un modelo SDC. De esta forma, logramos evitar que las decisiones de alguno de los clasificadores no competentes perjudique la clasificación. Este modelo utiliza el vecindario de la instancia a clasificar para establecer la competencia de cada clasificador, pasando a utilizar únicamente aquellos que, con nuestro método, son competentes.

La nueva estrategia ha mostrado su utilidad a pesar de su simplicidad en el estudio experimental llevado a cabo. A su vez, este hecho ha puesto de manifiesto la importancia del problema de los clasificadores no competentes en OVO, cuyo tratamiento puede llevar a mejoras significativas en los resultados.

Agradecimientos

Este trabajo ha sido parcialmente subvencionado por el Ministerio de Educación y Ciencia bajo los proyectos TIN2010-15055 y TIN2011-28488 y el plan Andaluz de investigación P10-TIC-6858.

Referencias

1. Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F.: KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. J. Mult.-Valued Logic Soft Comput. 17, 255 – 287 (2011)
2. Alcalá-Fdez, J., Sánchez, L., García, S., del Jesus, M.J., Ventura, S., Garrell, J.M., Otero, J., Romero, C., Bacardit, J., Rivas, V.M., Fernández, J., Herrera, F.: KEEL: a software tool to assess evolutionary algorithms for data mining problems. Soft Comput. 13(3), 307–318 (2009)
3. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Trans. Intell. Syst. Technol. 2, 27:1–27:27 (2011)
4. Chen, Y., Wang, J.Z.: Support vector learning for fuzzy rule-based classification systems. IEEE Transactions Fuzzy Systems 11(6), 716–728 (2003)

5. Cohen, W.W.: Fast effective rule induction. In: ICML'95: Proceedings of the Twelfth International Conference on Machine Learning. pp. 1–10 (1995)
6. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. 7, 1–30 (2006)
7. Fernández, A., Calderón, M., Barrenechea, E., Bustince, H., Herrera, F.: Solving mult-class problems with linguistic fuzzy rule based classification systems based on pairwise learning and preference relations. Fuzzy Sets Syst. 161(23), 3064 – 3080 (2010)
8. Galar, M., Fernández, A., Barrenechea, E., Bustince, H., Herrera, F.: Aggregation schemes for binarization techniques. methods' description. Tech. rep. (2011), http://sci2s.ugr.es/ovo-ova/AggregationMethodsDescription.pdf
9. Galar, M., Fernández, A., Barrenechea, E., Bustince, H., Herrera, F.: An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes. Pattern Recogn. 44(8), 1761 – 1776 (2011)
10. García, S., Herrera, F.: An extension on "statistical comparisons of classifiers over multiple data sets"for all pairwise comparisons. J. Mach. Learn. Res. 9, 2677–2694 (2008)
11. Gunes, V., Ménard, M., Loonis, P., Petit-Renaud, S.: Combination, cooperation and selection of classifiers: A state of the art. Int. J. Pattern Recogn. 17(8), 1303 – 1324 (2003)
12. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. SIGKDD Explor. Newsl. 11, 10 – 18 (2009)
13. Hastie, T., Tibshirani, R.: Classification by pairwise coupling. Ann. Statist. 26(2), 451–471 (1998)
14. Hüllermeier, E., Vanderlooy, S.: Combining predictions in pairwise classification: An optimal adaptive voting strategy and its relation to weighted voting. Pattern Recogn. 43(1), 128–142 (2010)
15. Knerr, S., Personnaz, L., Dreyfus, G.: Single-layer learning revisited: A stepwise procedure for building and training a neural network. In: Fogelman Soulié, F., Hérault, J. (eds.) Neurocomputing: Algorithms, Architectures and Applications, NATO ASI Series, vol. F68, pp. 41–50. Springer-Verlag (1990)
16. Kuncheva, L.I.: Combining Pattern Classifiers: Methods and Algorithms. Wiley-Interscience (2004)
17. Lorena, A.C., Carvalho, A.C., Gama, J.M.: A review on the combination of binary classifiers in multiclass problems. Artif. Intell. Rev. 30(1-4), 19–37 (2008)
18. McLachlan, G.J.: Discriminant Analysis and Statistical Pattern Recognition. John Wiley and Sons (2004)
19. Platt, J.C.: Fast training of support vector machines using sequential minimal optimization. MIT Press, Cambridge, MA, USA (1999)
20. Quinlan, J.R.: C4.5: Programs for Machine Learning. San Mateo-California: Morgan Kaufmann Publishers, 1st edn. (1993)
21. Vapnik, V.: Statistical Learning Theory. New York: Wiley (1998)
22. Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics Bull. 1(6), 80–83 (1945)
23. Woods, K., Philip Kegelmeyer, W., Bowyer, K.: Combination of multiple classifiers using local accuracy estimates. IEEE Trans. Pattern Anal. Mach. Intell. 19(4), 405 – 410 (1997)
24. Wu, T.F., Lin, C.J., Weng, R.C.: Probability estimates for multi-class classification by pairwise coupling. J. Mach. Learn. Res. 5, 975–1005 (2004)

Analyzing the structure of ensembles based-on evolving classifiers

José Antonio Iglesias, Agapito Ledezma, and Araceli Sanchis

Carlos III University of Madrid,
Avda. de la Universidad, 30, 28911 Leganés (Madrid), Spain
{jiglesia, ledezma, masm}@inf.uc3m.es

Abstract. Evolving classifiers was inspired by the idea of evolve their structure in order to adapt to the changes of the environment. The base of these classifiers are fuzzy rules that are evolved from streaming data. Two main characteristics of these classifiers are their open structure of the rule-base and the online learning mechanism which takes into account this flexible rule-base structure. Also, it is important to clarify that these classifiers are gradually developing but they are not genetic or evolutionary. On the other hand, ensembles of classifiers are well established as a method for obtaining highly accurate classifiers by combining less accurate ones.
In this paper, we present an analysis of the results of an approach (*eEnsemble*) for constructing an ensemble system based on Evolving classifiers. The main characteristic of this approach is that all the individual classifiers are self-developing and their structure could be totally different according to their input data samples stream.

1 Introduction

An ensemble of classifiers is defined by Ditterich [1] as a set of classifiers whose individual decisions are combined in some way (typically by weighted or unweighted voting) to classify new examples. The strategy in ensemble systems is therefore to create a collection of individual accurate classifiers, and combine their outputs such that the combination improves upon the performance of a single classifier. This requires, however, that individual classifiers make errors on different instances, that is, that ensemble members are diverse.These systems are also known as multiple classifier systems, committee of classifiers or mixture of experts [2].

Many different researches about constructing good ensembles of classifiers have been proposed. The researches in this field come to the conclusion that ensembles are often much more accurate than the individual classifiers that make them up. According to Ditterich [1], ensembles can improve performance because uncorrelated errors made by individual classifiers can be removed by voting. However, there are still many questions about the best way to construct ensembles as well as issues about how best understand the decisions made by ensembles.

In [3], a proof-of-concept ensemble method in which all the individual classifiers are Evolving Fuzzy-rule-based Classifier (*eClass*) was proposed. Based on that research, this paper analyzes the structure of this kind of ensembles. Evolving Classifiers have been applied to a wide range of problems; in this case, we analyze in detail its behavior as part of an ensemble.

An Evolving Classifier (*eClass*) is a fuzzy rule-based classifier with rules that are evolved from streaming data. Thus, the use of an *eClass* does not need to know beforehand in how many classes the data will be classified as new classes can be introduced during the learning process [4]. An *eClass* learns new rules from new data gradually preserving/inheriting the rules learned already. Thus, *eClass* can be defined as a self-developing classifier which has both their parameters but also (more importantly) their structure self-adapting on-line. In addition, an *eClass* can start learning "*from scratch*". It is important to clarify that evolving classifiers are gradually developing but they are not genetic or evolutionary.

2 Background and Related Work

In this section, we will define briefly the concept of ensemble of classifier. In addition, the evolving fuzzy classifiers will be presented more in detail. However, first, it is interesting to remark the theoretical reasons why an ensemble system is preferred. According to Polikar [2], the main reasons are: statistical reasons, large volumes of data, too little data, divide and conquer, data fusion. In this research, as we are using evolving classifiers, they can cope with huge amounts of data since they can be generated online from streaming data. Thus, the use of large volumes of data or little data is not a reason for using ensembles in our case. However, the other reasons (especially, statistical and *"divide an conquer"* reasons) are quite interesting so it will be analyzed if these reason are also true using evolving classifiers.

2.1 Ensemble of Classifiers

In 1979, one of the first works on ensemble systems [5] proposed the partition of the feature space using two or more classifiers. Other research in 1990 [6] concluded that the generalization performance of a neural network can be improved using an ensemble of neural networks. Bagging [7] and Boosting [8] are two of the most well-known ensemble learning methods due to their theoretical performance guarantees and strong experimental results.

Many different approaches propose different methods for generating individual classifiers, and strategies for combining the outputs of these classifiers. Some of these approaches were defined as combination of multiple classifiers [9], dynamic classifier selection [10], classifier fusion [11], classifier ensembles [12] and many others.

2.2 Evolving Fuzzy Classifiers: eClass

eClass (evolving Classifier) family [4] is a set of evolving neuro-fuzzy classifiers which take its roots in evolving Takagi-Sugeno (eTS). The first evolving fuzzy rule-based systems were introduced in [13] and further developed in [14].

During training in these classifiers, a set of fuzzy rules that describes the most important features for the classification of each class is formed. These rules are constantly adjusted to the available training data. One of the advantages of *eClass* is that it does not require parameter optimization as its only parameter 'scale' can be directly inferred from the training data. This technique [15] is based on partitioning the data space into overlapping local regions through *Recursive Density Estimation* (RDE) and associating clusters (respectively fuzzy sets) to them.

As it is explained in [4], the main differences between *eClass* family and a conventional Fuzzy Rule-Based (FRB) classifier are: 1) The open structure of the rule-base: *eClass* self-develops on-line 'starting from scratch' while in a conventional FRB classifier it is determined offline and then fixed. 2) The online learning mechanism which takes into account this flexible rule-base structure.

eClass family has been applied in many different areas with a great success. An important area of application of these evolving classifiers is robotics and autonomous systems. Some examples of application include: autonomous landmark recognition [16], self-localization and mapping [17], object detection and tracking [18][19], collision avoidance [20], activity recognition from sensor streams [21][22] and user modeling [23][24].

eClass family includes two different architectures and on-line learning methods: 1) *eClass0* with the classifier consequent representing class label. 2) *eClass1* for regression over the features using first order eTS fuzzy classifier. In this research, it will be analyzed the use of individual evolving classifiers (*eClass0*) in an ensemble.

3 Ensemble Method Based on Evolving Classifiers

As it was proposed in [3], the ensemble method based on evolving classifiers (which will be named as *eEnsemble*) can be explained taking into account two concepts: 1) The evolving classifier used for the individual classifiers: *eClass0*. 2) the architecture of the proposed ensemble method.

3.1 Evolving Classifier: eClassO

eClass0 possesses a zero-order Takagi-Sugeno consequent, so a fuzzy rule in the *eClass0* model has the following structure:

$$Rule_i = IF(X_1 \; is \; P_1) \; AND \ldots AND \; (X_n \; is \; P_n) \\ THEN \; Class \; = \; Class_i \tag{1}$$

where i represents the number of rule; n is the number of input variables; the vector X stores the input variables and the vector P stores the values of the features of one of the prototypes (cluster centre) of the corresponding class. $Class \in \{\text{set of different classes}\}$.

The *eClass0* model is composed of several fuzzy rules per class (the number of rules depends on the heterogeneity of the input data of the same class). During the training process, a set of rules is formed from scratch using an evolving clustering approach to decide when to create new rules. The inference in *eClass0* is produced using the "winner takes all" rule and the membership functions that describe the degree of association with a specific prototype are of Gaussian form. The *potential* (Cauchy function of the sum of distances between a certain data sample and *all* other data samples in the feature space) is used in the partitioning algorithm. However, it is very important to remark that in these classifiers, the potential is calculated recursively, which makes the algorithm faster and more efficient. This recursive concept is essential in these classifiers since for this reason, they do not need to store all the samples.

All details about the *eClass0* model and the learning algorithm can be found in [15].

3.2 Architecture of the proposed method: *eEnsemble*

The ensemble proposed in [3] is composed of several simple classifiers (individual or base-classifiers). The number of these individual classifiers (k) must be defined at the beginning by hand. All of these k classifiers have the same *eClass0* structure, but they will be trained with different (and disjoint) sets of data.

Unlike most of the ensembles of classifiers, the proposed method has the following characteristic: 1) The classification is online, 2) The inputs of the method are streaming data. These aspects are the most important novelty of this approach.

The architecture of the proposed method is shown in Figure 1 and it can be divided in the following three steps: Receiving data streams, creating the individual classifiers and combining individual classifiers.

Receiving data streams: As streaming data arrive continuously, there is not an initial data set. In this case, when a new sample arrives, it is sent randomly to one of the k individual classifiers. This on-line partitioning is done without taking into account any variable or technique; it is done entirely randomly. Thus, the examples are randomly distributed across the set of individual classifiers.

Creating the individual classifiers: Once an example is received by an individual classifier, it is analyzed using *eClass0*. As we already explained before, and this is one of main novelties of this method, this classifier is designed with an evolving (self-developing) FRB structure. Also, this classifier can start "from scratch" and it is no need to define any variable or parameter for the different classifiers.

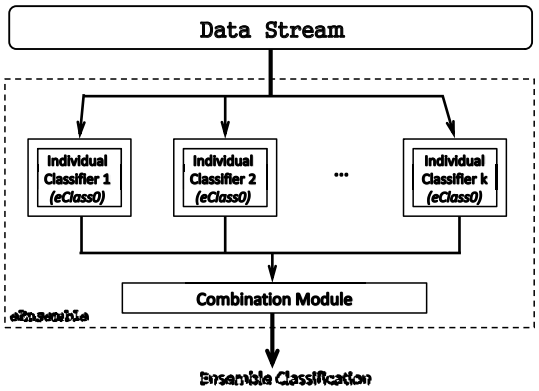


Fig. 1. *eEnsemble* Architecture.

It is important to highlight that the (flexible) rule-base structure of the individual classifiers will be different since the data streams that they receive are also different. Thus, the number of rules and the focal points of the fuzzy rules will be different in the individual classifiers. This is the main characteristic of the proposed method.

Combining individual classifiers: Once the individual classifiers have been trained with example data streams, we can use the proposed *eEnsemble* in order to classify a new example/instance. In this case, the new example will be tested by all the classifiers and their output will be combined in order to obtain a final classification. This task can be achieved by using different methods: Linear functions like average function [25]; non-linear combination methods, like majority voting (Bagging, Boosting) or meta-learning methods [26] [27].

The ensemble that we consider in this paper will use the simplest approach: unweighted vote. The outputs of the individual classifiers will be combined by taking a majority vote of their classification. For any given instance, the class chosen by most individual classifiers is the ensemble classification. This method is also used in Bagging [7].

Although this is a very simple method, the experience in the forecasting literature has been that simple, unweighted voting is robust [28]. However, this module could be changed without changing the base idea of the *eEnsemble*.

Finally, it is important to remark before the experimental section that a variation of *eEnsemble* (called *eEnsembleR*) was proposed in [3]. In that method, when an instance arrives, it would go (randomly) to one of the base-classifiers or to all of them. However, although the number of samples of each classifier increases, the results of this variation were not significantly better than the obtained by *eEnsemble*. Thus, in this paper, the results of *eEnsembleR* are not analyzed.

4 Experimental Setup and Results

In [3], *eEnsemble* was tested on 7 benchmark problems from the UCI machine learning repository [29]. In this paper, we will analyze in detail 3 different datasets with a number of instances very different. This analysis will let us to know who the *eEnsembles* work and how the number of instances, attributes and classes can vary their results. These datasets are detailed as follows:

- Wine Data Set (178 instances, 13 attributes and 3 Classes): These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivators. The analysis determined the quantities of 13 constituents found in each of the 3 types of wines.
- Page Blocks Data Set (5473 instances, 10 attributes and 6 Classes): This is a set of 5473 examples which come from 54 distinct documents. Each observation concerns one block.
- Statlog (Shuttle) Data Set (58000 instances, 9 attributes and 8 Classes): This data set contains 58000 instances and a high number of them belongs to one class.

In order to apply *eEnsemble* to these data sets, they are considered as pseudo-online streams. Thus, although the datasets were used in an offline mode, the proposed method was designed to be used with data streams.

As we already mentioned, before applying *eEnsemble*, it is necessary to parametrize the number of individual classifiers (represented as k) that will make up the ensemble. As the result of the *eEnsemble* varies according to this parameter, we have used there different values for k: 3, 5 and 10.

The performance of the ensemble that we propose depends on how the data stream is distributed among the individual classifiers. As this distribution is randomly done, the result of the *eEnsemble* can change because of this aspect. For this reason, the results shown in this section are the average of 100 different executions of *eEnsemble* (the standard deviation is also represented).

Finally, in order to evaluate the performance of *eEnsemble*, it is compared with the *eClass0*. Also, to measure the performance of the different methods, for each data set, 80 percent of the data are kept as training instances while the rest are used as the pool of training examples.

Table 1 shows the rate for *eClass0* and the average rate for *eEnsemble* in the 3 different datasets. Major rows correspond to the 3 different datasets and columns show the results of the approach taking into account the number of individual classifiers (k).

According to these results, we can conclude that, in general, the results of the proposed ensemble are comparable with those obtained from eClass0. However, these results change considerably according to the number of individual classifiers. In the next section we will analyze in detail these results and the structure of the proposed ensemble taking into account a fixed number of individual classifiers (k=3).

Table 1. Experimental results: *eClass0* vs. eEnsemble

		Our approaches - rate % (average)		
Dataset	**eClass0**	**eEnsemble**		
	(rate %)	k=3	k=5	k=10
DataWine	94,4	**96,72 ± 2,1**	$95,78 \pm 3,3$	$91,11 \pm 5,8$
PageBlocks	69,59	$76,97 \pm 3,3$	$80,99 \pm 4,2$	**89,95 ± 2,0**
Shuttle	**92,98**	**92,89 ± 2,0**	$95,36 \pm 1,6$	$88,82 \pm 4,6$

4.1 Analyzing the structure of eEnsemble

In order to simplify the analysis of the 3 different structures proposed (k=3, k=5 and k=10), we will analyze in detail only the structure of the eEnsamble with k=3. In this analysis, we will take into account:

1. The percentage of samples correctly classified by each individual classifier.
2. The number of fuzzy rules which form each individual evolving classifiers. We should take into account these fuzzy rules describe the most important features for the classification of each class. Thus, an specific class can be defined by one or several rules. For example, if we are analyzing the dataset *Shuttle* which consists of 9 different attributes (and 8 classes), the structure of the rule (in this case, $Rule_1$) could be similar to:

$$Rule_1 = IF(X_1 \; is \; 37) \; AND \ldots AND \; (X_9 \; is \; 4) \\ THEN \; Class \; = \; Class_1 \tag{2}$$

In addition, as these values could change according to the distribution of the samples, the results in this case are obtained as the average of 100 different executions of *eEnsemble*.

Figure 2 shows the result of this analysis. In the first dataset (Data Wine), if only one (evolving) classifier is considered (right part of the Figure), 17 fuzzy rules are created. On the other hand, if we observe the method proposed with k=3 (left part of the Figure), we can observe that the number of rules of each individual (evolving) classifier of *eEnsemble* is around 11. These values are not an integer number because the result result is the average of 100 different executions. However, the sum of the rules of the different classifiers is around 33, which is a much higher value than the rules of a unique classifier (17).

In addition, it is interesting how the total rate of samples correctly classified for each individual classifier is around 93%, but if we combine the result of the three classifiers, this value increases to 96,72%. As we observe, this rate is higher than the obtained for only one classifier (94,49%).

The conclusions obtained by observing the results of the other two datasets are similar. We observe that the number of (fuzzy) rules that need to be created increases using *eEnsembles* but the results are, in general, better.

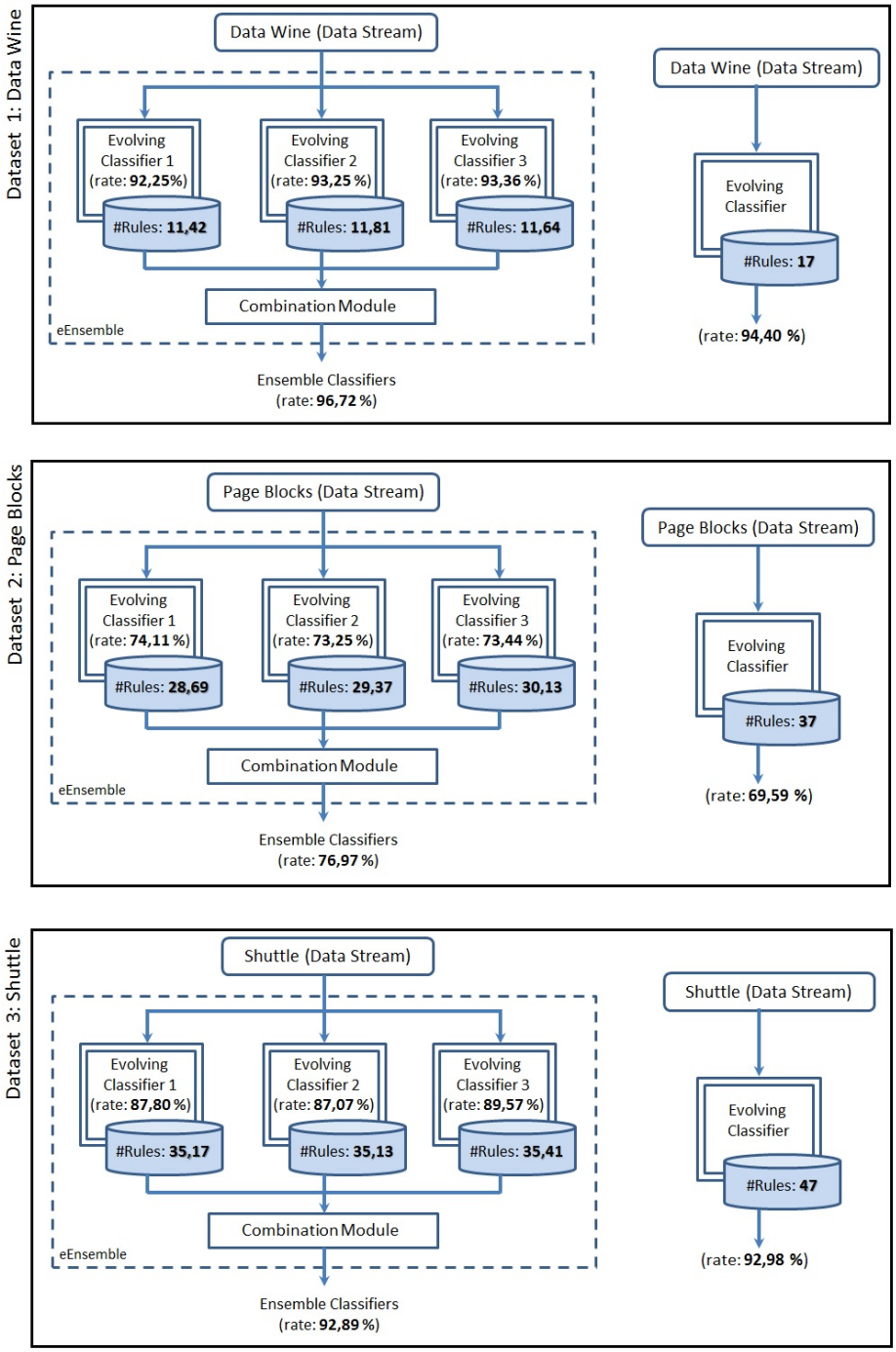


Fig. 2. eEnsembles: Analysis Result using 3 different datasets

5 Conclusions and Future Work

In this paper, we have presented an analysis of the results of an approach (*eEnsemble*) for constructing an ensemble system based on Evolving classifiers. The main characteristic of this approach is that all the individual classifiers are self-developing and their structure could be totally different according to their input data samples stream. Thus, although the kind of classifier is the same (*eClass0*), their structure (number of fuzzy rules) changes according to the received data. It is important to remark that these individual classifiers do not need any initialization value or threshold since they can start from scratch and their changes are done on-line and recursively.

According to the experimental results, although the number of (fuzzy) rules that need to be created increases using *eEnsembles*, the obtained results are comparable and, in general, better. Also, taking into account the results, the number of instances of the data set is not crucial for the result of the ensemble.

Finally, it would be interesting as future work that the value k (number of individual classifiers of the ensemble) could be calculated automatically in order to find the best configuration of the ensemble. Thus, as it happens with the evolving classifiers that their structure changes according to the (input) data, this value could also change according to the data received.

Acknowledgment

This work has been supported by the Spanish Government under i-Support (Intelligent Agent Based Driver Decision Support) Project (TRA2011-29454-C03-03).

References

1. Dietterich, T.G.: Machine-learning research – four current directions. AI MAGAZINE **18** (1997) 97–136
2. Polikar, R.: Ensemble Based Systems in Decision Making. IEEE Circuits and Systems Magazine **6**(3) (2006) 21–45
3. Iglesias, J., Ledezma, A., Sanchis, A.: Ensemble method based on individual evolving classifiers. In: Evolving and Adaptive Intelligent Systems (EAIS), 2013 IEEE Symposium on. (apr 2013) 56–61
4. Angelov, P.P., Zhou, X.: Evolving fuzzy-rule-based classifiers from data streams. IEEE T. Fuzzy Systems **16**(6) (2008) 1462–1475
5. Dasarathy, B.V., Sheela, B.V.: Composite classifier system design: concepts and methodology. In: Proceedings of the IEEE. Volume 67. (1979) 708–713
6. Hansen, L.K., Salamon, P.: Neural network ensembles. IEEE Trans. Pattern Anal. Mach. Intell. **12**(10) (October 1990) 993–1001
7. Breiman, L.: Bagging predictors. Mach. Learn. **24**(2) (August 1996) 123–140
8. Freund, Y., Schapire, R.E.: Experiments with a New Boosting Algorithm. In: International Conference on Machine Learning. (1996) 148–156

9. Rogova, G.: Combining the results of several neural network classifiers. Neural Networks **7**(5) (May 1994) 777–781
10. Woods, K.S., Kegelmeyer, W.P., Bowyer, K.W.: Combination of multiple classifiers using local accuracy estimates. IEEE Transactions on Pattern Analysis and Machine Intelligence **19**(4) (1997) 405–410
11. Kuncheva, L.I., Bezdek, J.C., Duin, R.P.W.: Decision templates for multiple classifier fusion: an experimental comparison. Pattern Recognition **34** (2001) 299–314
12. Kuncheva, L.I.: Classifier ensembles for changing environments. In: In Multiple Classifier Systems, Springer (2004) 1–15
13. Angelov, P.P.: Evolving rule-based models: a tool for design of flexible adaptive systems. Springer-Verlag, London, UK (2002)
14. Angelov, P., Zhou, X.W.: Evolving fuzzy systems from data streams in real-time. In: Proceedings of the Internat. Symp. on Evolving Fuzzy Systems. (2006) 29–35
15. Angelov, P., Filev, D.: An approach to online identification of takagi-sugeno fuzzy models. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on **34**(1) (feb. 2004) 484 – 498
16. Zhou, X., Angel, P.: Real-time joint landmark recognition and classifier generation by an evolving fuzzy system. In: Fuzzy Systems, 2006 IEEE International Conference on. (2006) 1205 –1212
17. Angelov, P.P., Zhou, X.: Evolving fuzzy classifier for novelty detection and landmark recognition by mobile robots. In: Mobile Robots: The Evolutionary Approach. Volume 50 of Studies in C.I. Springer (2007) 89–118
18. Angelov, P., Ramezani, R., Zhou, X.: Autonomous novelty detection and object tracking in video streams using evolving clustering and takagi-sugeno type neuro-fuzzy system. In: Neural Networks, 2008. IJCNN 2008. (june 2008) 1456 –1463
19. Liu, Y., Meng, H., Wang, D., Wang, X.: Adaptive staggering time estimation for target tracking in periodic nonuniform sampling system. Electronics Letters **43**(24) (2007) 1385–1387
20. Angelov, P., Bocaniala, C.D., Xideas, C., Patchett, C., Ansell, D., Everett, M., Leng, G.: A passive approach to autonomous collision detection and avoidance. Computer Modeling and Simulation, International Conference on **0** (2008) 64–69
21. Iglesias, J.A., Ordonez, J., Ledezma, A., de Toledo, P., Sanchis, A.: Evolving activity recognition from sensor streams. In: Proceedings of the 2012 IEEE Evolving and Adaptive Intelligent Systems (EAIS-2012). (May 2012) 96–101
22. Ordonez, J., Iglesias, J.A., Ledezma, A., de Toledo, P., Sanchis, A.: Online activity recognition using evolving classifiers. Expert Systems with Applications (2012)
23. Iglesias, J.A., Angelov, P.P., Ledezma, A., de Miguel, A.S.: Creating evolving user behavior profiles automatically. IEEE Transactions on Knowledge and Data Engineering **24**(5) (5 2012) 854–867
24. Garca-Cuesta, E., Iglesias, J.A.: User modeling in changeable environments. In: Proceedings of the 2012 IEEE Evolving and Adaptive Intelligent Systems (EAIS-2012). (May 2012) 182–185
25. Sesmero, M., Alonso-Weber, J., Gutirrez, G., Ledezma, A., Sanchis, A.: A new artificial neural network ensemble based on feature selection and class recoding. Neural Computing and Applications **21** (2012) 771–783
26. Ledezma, A., Aler, R., Sanchis, A., Borrajo, D.: Ga-stacking: Evolutionary stacked generalization. Intelligent Data Analysis **14**(1) (January 2010) 89–119
27. Wolpert, D.H.: Stacked generalization. Neural Networks **5** (1992) 241–259
28. Clemen, R.T.: Combining forecasts: A review and annotated bibliography. International Journal of Forecasting **5**(4) (1989) 559–583
29. Frank, A., Asuncion, A.: UCI machine learning repository (2010)

Dental milling process optimization by the use of ensemble classification

Santiago Porras[1], Hector Quintian[2], Bruno Baruque[1], Beatriz Hernando[3], Ignacio Aliaga[3], Vicente Vera[3], and Emilio Corchado[2]

[1] Dept. of Civil Engineering, University of Burgos (Spain)
{sporras,bbaruque}@ubu.es.
[2] Dept. de Informática y Automática, University of Salamanca (Spain)
{hector.quintian,escorchado}@usal.es.
[3] Facultad de Odontología, University Complutense of Madrid (Spain)
{Ialia01@odon.ucm.es,viventevera}@odon.ucm.es.

Abstract. A novel soft computing method for predicting the behavior of a dental milling machine is presented in this study. This Artificial Intelligent system makes possible to optimize the implementation conditions in the manufacturing process of high precision parts, including finishing precision, while saving time, financial costs and/or energy. The proposed system is modeled as an ensemble of classifiers using two different stages. First of all, an unsupervised model known as Self-Organizing Map is used to segment the complete hypothesis space of the function parameters of the machine, obtaining an initial reference of the milling conditions. This step has the additional advantage that is easy to supervise by a human operator. Then a more fine-grained classification of the data is performed using only conditions present in the corresponding cluster of the data space, constructing an ensemble of weak classifiers. The weak classifiers used are well-known simple classification algorithms such as Naive Bayes and Multilayer Perceptron. The experiments and results shown prove that the system is able to correctly identify the functioning characteristics of the milling machine in order to predict the interval of several of the final conditions of the process.

Keywords: soft computing; ensemble Learning; automated classification; industrial application; dental milling

1 Introduction

There is an increasing interest in the use of computing techniques as tools to assist in many complex tasks that have traditionally been performed entirely by human experts in knowledge fields such as Engineering[1] or Medicine [2]. Computers can perform mathematically complex operations and repetitive tasks with a high level of accuracy. That is how new areas of expertise, such as Computer Aided Design (CAD) or Computer Aided Manufacture (CAM) have emerged. Traditionally, there are more subtle human tasks that are much more difficult for computers to solve (especially those that involve uncertainty or problems

that are not very well defined). In these cases, the specific computing field of Artificial Intelligence (or Soft Computing) provides interesting tools that allow computers cope with these kinds of complex problems.

Machine parameter optimisation in the fabrication process includes the development of models to assess the behaviour of the variables in the process and to find the best functioning parameters that can be optimised. The machine parameter optimisation should help the experts in better understanding the production process itself in order to manufacture products using new materials in a short period of time. The application of the optimisation process in the field of Medical Therapeutics (Odonto-Stomatology), a booming industry, is both novel and economically advantageous [3,4,5]. The optimisation process of machine parameters, for example the time parameter, permits significant economic savings due to the high number of dental pieces produced daily by the same high-precision dental milling machine centre. This could significantly help to increase a company's efficiency, and substantially contribute to cost reductions in the preparation and setting of the machines processes. Another example is the marginal adjustment of a dental prosthesis to the remaining tooth structure (dentine and enamel), thus avoiding tissue invasion and/or unprepared anatomical areas. This is the goal of any dental treatment, since the success now lies in the adjustment of approximately 15-25 micrometers between the prosthesis and the remaining structures. This would enhance and prevent the filtering of germs and oral fluids that within the short or medium term will lead to treatment failure.

The optimal outcome would be the so-called passive adjustment, i.e. adjusted between 15-25 micrometers which would lead to success in dental restorative treatment. As can be expected, these are tasks where an extremely high accuracy is needed, so it is currently of great interest to optimise processes [3,6] related to the preparation of dental prostheses (dental structures of materials such as cobalt chromium, titanium, ceramics and/or resin) with the aid of Soft Computing in order to reduce the processing errors.

In the case of these study, data obtained from the real processing of several prosthetic pieces, is processed by an automatic classification algorithm that, based on the initial processing conditions, is able to determine the time that the process will take to finish (within a small range). This process is able to determine the difference with the theoretical estimated processing time, with a lower percentage of error.

2 Dental Milling Process Description

This multidisciplinary study uses a 5-axis milling (latest generation) device. In addition to controlling movement between the tool and the piece in three axes, this 5-axis machine can also control both the rotation of the piece in two axes, one perpendicular to the axis of the tool and the other parallel to it, and the rotation of the piece on a horizontal axis with the inclination of the tool around an axis perpendicular to the former.

The current research describes the way in which a soft computing system can be applied to optimise the data gathered by means of a Machining Milling Center of HERMLE type-C 20 U (iTNC 530), with swiveling rotary (280 mm), with a control system using high precision drills and bits, by optimising the time error detection for manufacturing dental metal.

The case study is described by an initial data set of 218 samples obtained by the dental scanner in the manufacturing of dental pieces with different tool types (plane, toric, spherical and drill) and characterized by 8 input variables (Tool, Radius, Revolutions, Feed rate X, Y and Z, Initial Temperature) and 1 output variable (Time Error for manufacturing) as shown in Table 1. Time error for manufacturing is the difference between the estimated time by the machine itself and real production time (negative values indicate that real time exceeds estimated time).

Variable (Units)	Range of values
Type of tool	plane, toric, spherical, drill
Radius (mm.)	0.25 to 1.5
Revolutions per minute (RPM)	7.5 to 38.0
Feed rate X (mm. by minute)	0.0 to 3.0
Feed rate Y (mm. by minute)	0.0 to 3.0
Feed rate Z (mm. by minute)	2.0 to 50.0
Temperature (°C)	24.1 to 31
Estimated time	6 to 6318
Time errors for manufacturing (s)	-28 to -255

Table 1: Values of each variable used in the process

3 Topology Preserving Maps for Data Clustering

3.1 The SOM Algorithm

Topology preserving mapping comprises a family of techniques with a common target: to produce a low-dimensional representation of the training samples that preserves the topological properties of the input space. From among the various techniques, the best known is the Self-Organizing Map (SOM) algorithm [7,8]. SOM aims to provide a low-dimensional representation of multi-dimensional data sets while preserving the topological properties of the input space. The SOM algorithm is based on competitive unsupervised learning; an adaptive process in which the neurons in a neural network gradually become sensitive to different input categories, which are sets of samples in a specific domain of the input space [9].

The update of neighbourhood neurons in SOM is expressed as:

$$w_k(t+1) = w_k(t) + \alpha(t) \cdot \eta(v,k;t) \cdot (x(t) - w_k(t)) \tag{1}$$

where, x denotes the network input, w_k the characteristics vector of each neuron; α, is the learning rate of the algorithm; and $\eta(v,k;t)$ is the neighbourhood function, in which v represents the position of the winning neuron (BMU) and k the positions of the neurons in its neighbourhood in the lattice.

3.2 Use of the SOM for Data Clustering

The clustering capabilities of the SOM algorithm have been extensively used in many works [10,11]. Mainly intended as a visual aid for data clusters exploration, several calculations over the final map have been proposed. After the calculation is performed, it can be represented over the final resulting map as a colour scheme. In the case of this work the three measures presented in [12,13] are used:

U-Matrix. This matrix represents for each unit of the map, the sum of the euclidean distances between its corresponding weights vector and all the data samples that are recognized by it. Its mathematical definition is expressed in 2

$$uh(n_i) = \sum_{n_j \in U_i} d(n_i, n_j) \tag{2}$$

It permits to get a visual idea of how concentrated or disperse is the data set in the manifold represented by the map. Obviously, it is a valuable tool to detect and determine clusters and cluster borders in the data set. n refers to a determined unit or neuron of the map, while d represents the euclidean distance between two units.

P-Matrix. In this case, the values represented in the matrix are calculated as the density measured in the data space at a specific point, where that point is the weight vector associated with each unit of the SOM. Its mathematical definition is expressed in Eq. 3.

$$ph(n_i) = p(w(n), X) \tag{3}$$

where $p(x, X)$ is an empirical density estimation at point x in the data space X. As with previous matrix, the measure of the concentration of data serves as a very good measure for finding clusters in data.

U*-Matrix. Finally, matrix combines the distance based U-Matrix and the density based P-Matrix. It consists in using the U-Matrix as a basis for the final matrix, and the P-Matrix. This is mathematically defined as shown in 4 and 5.

$$u^*h(n_i) = uh(n_i) \cdot Scale\ Factor\ (n_i) \tag{4}$$

$$Scale\ Factor\ (n_i) = \frac{ph(n_i) - mean(ph)}{mean(ph) - min(ph)} \tag{5}$$

Again, this matrix is used to obtain data clusters, especially in thin populated regions of the data space; where distances are more important than density to determine the similarity of data. As the results obtained by all these three matrix calculations consist basically in numerical values for each of the units composing the maps, they can be used as an automatic way of determining the number of clusters in a data set but more importantly in this case, can inform about things like the boundary of those clusters or the density of data included in them. This information is used in the presented model to decide the structure that the ensemble of base classifiers will have.

3.3 Other Clustering Options

Several ways of generating clusters have been taken from this subject's literature in order to add more variability to the way the data input space can be segmented [14]. In the case of this study, five different approaches have been adapted to be used over the trained SOM, in order to obtain an initial clustering of the data space.

When deciding the overall way of constructing the clusters, usually two main strategies are considered: *Agglomerative*, in which each unit of the SOM is considered as a cluster and more units are added to that cluster, depending whether or not are within a certain distance threshold. And *Divisive*, in which the whole data space is considered as a cluster and it is divided into pairs of clusters, adding each unit to one or the other depending on their distances. This is repeated until there are not divisible clusters left.

When considering how the distances between two clusters can be calculated in order to decide if they need to be merged/divided, there are also several options to regard: *Simple*, where only the distance between the two closest units belonging to each of the cluster is considered; *Complete*, where only the distance between the two furthest units belonging to each of the cluster is considered; and *Average*, where the average distance of all pairs of units belonging to each of the clusters is considered.

All this five different approaches have been included in the tests.

4 Segmentation and classification algorithm

As explained in many previous publications in literature[15,12], one of the best ways of enhancing the capabilities of a classifier without introducing a too higher complexity - and avoiding an overfitting to the dataset - is the use of ensembles. It consists in training several different but overlapping simple classifiers, in an attempt to obtain a better grasp over the complete dataset. In the case of this study, this is achieved by the segmentation of the hypothesis space of the problem by the use of the SOM algorithm, to later try to classify independently samples assigned to a certain cluster in space.

The processing devised can be therefore summarized as in Algorithm 1.

Summarizing the algorithm, its main steps are: firstly a SOM is trained using a subset of the data, then it calculates the data clusters by using one of the models of Section 3. Data contained in each of the clusters is used to train a base classifier, hopefully making it the best expert for that region of the data space. To perform the classification of new instances, they are firstly presented to the SOM. The cluster to which each sample belongs to, is determined by finding the BMU. It is then presented to the classifier(s) assigned to that cluster. The final output will be that of the assigned classifier.

Algorithm 1 SOM Clustering and Selecting Ensemble

Input: A training data set $D \epsilon \mathbb{R}^n$*, a clustering threshold* θ_c*, an inclusion threshold* θ_m
Output: A Classification Ensemble $C_1 ... C_n$

1: **procedure** CONSTRUCT ENSEMBLE($C_1 ... C_n$)
2: Train a Self-Organizing Map over the input data set.
3: Label the units of the map with the entries for which it has been considered as the BMU.
4: Calculate the requested matrix values for each map unit (U-Matrix, P-Matrix, U^*-Matrix).
5: Perform a clustering of units depending on the difference of the values calculated in step 3 for each unit and its neighbours, using θ_c.
6: Include each data sample in each of the clusters found in step 4, according to the cluster its corresponding BMU belongs to.
7: Eliminate the clusters with a number of samples lower than θ_m.
8: Train a base classifier with the data entries that form part of each of the clusters.
9: **end procedure**

10: **procedure** CLASSIFY SAMPLE(s)
11: Present sample to the Self-Organizing Map.
12: Determine the corresponding cluster (C) by finding the BMU for that sample (BMU_s).
13: Present sample (s) to the classifier(s) corresponding to the cluster (C).
14: Output the majority vote of classifications obtained in the previous step (in the case that more than one classifier were assigned to the cluster)
15: **end procedure**

5 Experiments and Results

5.1 Experimental set up

In this study the experiments set try to determine whether an automatic classification algorithm would be able to correctly predict the behaviour of the milling machine, given its initial functioning conditions. That way, they could be automatically corrected by the machine, avoiding the waste of time and materials of a potential undesirable final result. In order to test this hypothesis, a dataset of milling work examples completed by the machine discussed measured directly on its functioning parameters and results has been compiled.

The dataset contains 218 instances or examples and 8 features (see Table 1). In order to do the experiments the data has been preprocessed. First, very few missing values (due to minor error in the measures) have been replaced by the average of the all values of the feature, then a data normalization has been calculated.

Two different set of tests have been performed with different objectives. In a first set, data has been used in its original form, using a regression algorithm to test if the algorithm is able to predict the difference between the estimated and real operation time with a high degree of accuracy. In a second set a discretization

of the output variable has been performed, in order to obtain a classification of samples with a clear explanation. This discretization has been made by bins: the dataset has been divided in 7 bins of equal size, obtaining 6 classes (one of the bins did not contain any samples). In both cases, the class information is not needed for the training of the SOM maps for the initial data space segmentation (unsupervised phase), but only for the training of the final classifiers (supervised phase).

The parameters used in the SOM training are included in Table 2. The SOM and later clustering parameters have been obtained experimentally in a process of manual fine-tuning; by selecting the parameters of the best performing ones. Future improvements of the algorithm will try to include a heuristic method for obtaining the parameters.

The classifiers are defined using the same parameters in all tests: the Naive Bayes parameters include a kernel distribution and a uniform probability distribution and the Multilayer Perceptron is configured with one hidden layer and 9 neurons in the first layer and 4 in the output one.

size	8x8
times	1700
learning rate	0.1
σ (neighb. funct.)	4

(a) Parameters used for the training of the SOM

	Lower limit	Divisive Limit
U-Matrix	0	0.25
P-Matrix	0	3
U*- Matrix	5	0.3

(b) Parameters used for the calculation of the clusters

Table 2: Parameters used in the experiments

5.2 Results

Results compare the use of simple classification algorithms with the use of the cluster and classification counterparts in order to compare their performance in classifying data from the experimental results obtained with the milling machine. All results are the average of two different 5-fold cross validation experiments.

The values shown in Table 3 have been calculated as the regular Mean Absolute Error, which is calculated as the accumulated difference of values between the predicted value and the measured one. A single MLP trained over the complete dataset obtained a MAE of 1.5. The presented algorithm improves this result in all its variants, by obtaining lower error levels.

The values shown in Table 4 are a measure of the percentage of the error of each algorithm when presented the test data, so lower values are more desirable. The simple Naive Bayes (without including it in any ensemble) achieves an average error of 13.3%, while the simple Multilayer Perceptron, an error of 11%. Results obtained by the ensembles of these models, including all combinations of clustering methods described in Section 3.3 are shown in Table 4a for the Naive Bayes and in Table 4b for the Multilayer Perceptron Results.

		U-Matrix	P-Matrix	U*-Matrix
Agglomerative	Simple	0.34	0.42	0.46
	Complete	0.51	0.45	0.49
	Average	0.43	0.37	0.47
Divisive	Simple	0.41	0.36	0.5
	Complete	0.45	0.45	0.41
	Average	0.41	0.43	0.42

Table 3: Results for the regression task using a Multilayer Perceptron, presenting the Mean Absolute Error.

		U-Mat	P-Mat	U*-Mat
Agglom.	Simple	11.5%	13.8%	14.1%
	Complete	13.7%	14.1%	12.7%
	Average	14.5%	13.0%	12.6%
Div.	Simple	13.5%	13.0%	13.6%
	Complete	13.0%	13.9%	11.7%
	Average	14.1%	12.9%	11.3%

(a) Results for the experiments using a Naive Bayes classifier as the base classifier.

		U-Mat	P-Mat	U*-Mat
Agglom.	Simple	9.4%	11.0%	9.7%
	Complete	10.8%	11.7%	9.8%
	Average	10.3%	11.8%	10.5%
Div.	Simple	10.1%	10.2%	11.3%
	Complete	9.9%	10.6%	10.6%
	Average	11.3%	10.4%	10.1%

(b) Results for the experiments using a Multilayer Perceptron as the base classifier.

Table 4: Result for the classification task (quantized labels), presenting average error percentages.

As can be seen in Table 4a, when using the SOM segmentation cluster, results improving the simple classifier (those lower than 13.3%) are achieved mainly when using the P-Matrix and U*-Matrix for clustering calculations. This means that the dataset is better clustered and therefore, classified when the data concentration is taken into account, over data distances. It is interesting to note, nevertheless, that the U-Matrix with the simple calculation for cluster distances obtains a very low error in this experiment. Analysing results when using a MLP as classifier, in Table 4b, we can observe again the same idea behaviour of the classification as with the Bayesian classifier: when taking into account the data density, the ensemble of classifiers obtains better results than the simple classifier.

As pointed out in other previous approach to this problem [5], the dataset is distributed along the whole data space, but at the same time, it is a sparse distribution, where the data is grouped in very compact clusters. It is therefore a convenient strategy to look for dense clusters of data in order to organize the

data space for further analysis. It is also important to note that, although the ensemble algorithm improves the results obtained by the single classifier, in the case of this not very complex dataset, the difference in the results is in the range of 2% or 3%.

Although in the initial state of this study the number of samples is quite reduced, it is natural to assume that the number of samples to analyze will increase highly in the foreseeable future. When the drilling machine operates continuously, even an on-line functionality is desirable. In that sense, it is interesting to note that the structure proposed is able to improve results for a single model with test samples and is expected to escalate well for this particular problem, minimizing the problems of classifiers overfitting and bottle neck procedures, as it is highly parallelizable.

6 Conclusions

The novel soft computing optimisation process described in this study can be used to optimise machine parameters for industrial processes, based on the obtained results. This method increases the companies' efficiency and substantially reduces the cost of preparing and setting machine processes. It also helps in the production process using new materials. We have used this method for optimisation and adjustments during the manufacturing process of dental pieces such as implants according to medical specifications for precise moldings.

In the present study, automated classification models, implemented as a cluster and classification ensemble have been used to automatically determine the time that will be necessary to finish a drilling work. The results have been compared with simpler classification models, proving that the model is able to improve those results. This results are promising, as they are an evidence of the feasibility of a software system that could anticipate the conditions of the process is being executed in a determined moment in order to correct/modify the functioning parameters of the machine in case it would be needed. This is an obvious advantage as it can easily result in higher accuracy finishes of the pieces, saving of time and material in its processing and a less demanding task for the human operator.

Acknowledgement This research is partially supported by the Spanish Ministry of Economy and Competitiveness under project TIN2010-21272- C02-01 (funded by the European Regional Development Fund), SA405A12-2 from Junta de Castilla y León. This work was also supported by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070).

References

1. Calvo-Rolle, J., Corchado, E., Quintián, H., Ferreiro, R., Román, J., Hernandez, P.: A novel hybrid intelligent classifier to obtain the controller tuning parameters for

temperature control. In: 7th International Conference Hybrid Artificial Intelligent Systems, HAIS 2011. Volume vol. 1. (2012) 677–689
2. Lasso, M.C., Gutiérrez, C.D., Curiel, L., Baruque, B., Ibañez, A.F., Tárrago, C.P., Corchado, E., Rodríguez, M.A.M., Abad, S.M., Tupon, V.P., Morentin, A.B.M., Gonzalez, J.F.L., Garcia, A.B., Gómez, N.: Detection of bacterial endocarditis complications with machine learning programs. European Journal of Internal Medicine **22, Supplement 1**(0) (2011) S23 – S24
3. Al-Fadda, S., Zarb, G., Finer, Y.: A comparison of the accuracy of fit of 2 methods for fabricating implantprosthodontic frameworks. Int J Prosthodont **20** (2007) 125–131
4. Veran, V., Sedano, J., Corchado, E., Redondo, R., Hernando, B., Camara, M., Laham, A., Garcia, A.: hybrid system for dental milling parameters optimisation. In: 6th International Conference Hybrid Artificial Intelligent Systems, HAIS 2011. Volume vol. 2. (2011) 437–446
5. Vera, V., Corchado, E., Redondo, R., Sedano, J., Garcia, A.E.: Applying soft computing techniques to optimise a dental milling process. Neurocomputing **109** (2013) 94–104
6. Beuer, F., Schweiger, J., Edelhoff, D.: Digital dentistry: an overview of recent developments for CAD/CAM generated restorations. Br Dent J **204**(9) (2008) 497–502
7. Kohonen, T.: Self-Organizing Maps. Volume 30. Springer, Berlin, Germany (1995)
8. Kohonen, T.: Data management by self-organizing maps. In: WCCI. (2008) 309–332
9. Kohonen, T., Lehtio, P., Rovamo, J., Hyvarinen, J., Bry, K., Vainio, L.: A principle of neural associative memory. Neuroscience **2**(6) (1977) 1065–1076
10. Lampinen, J., Oja, E.: Clustering properties of hierarchical self-organizing maps. Journal of Mathematical Imaging and Vision **2** (1992) 261–272
11. Dara, R., Kremer, S.C., Stacey, D.A.: Clustering unlabelled data with SOMs improves classification of labelled real-world data. In: Proc. IEEE World Congress on Computational Intelligence. (May 2002) 2237–2242
12. Kuncheva, L.I.: Clustering-and-selection model for classifier combination. In: KES. (2000) 185–188
13. Ultsch, A.: U*-matrix: A tool to visualize clusters in high dimensional data. Technical report, Department of Computer Science, University of Marburg (2003)
14. Xu, R., Wunsch, D.: Survey of clustering algorithms. IEEE Transactions on Neural Networks **16**(3) (May 2005) 645–678
15. Akhand, M., Islam, M.M., Murase, K.: Comparative study of data sampling techniques for constructing neural network ensembles. International Journal of Neural Systems **19** (2009) 67–89

17-20 sept 2013

MULTICONFERENCIA

CAEPIA'13

Madrid

I Simposio de Agentes y Sistemas Multi-Agente: de la teoría a la práctica [ASMas 2013]

Editor asociado:

Juan M. Corchado

Simposio organizado por

VNiVERSiDAD D SALAMANCA

UNIVERSIDAD POLITÉCNICA MADRID

RESEARCH GROUP BISITE .usal.es

1500

Prefacio

La investigación en Agentes y Sistemas multi-agente ha madurado durante la última década, obteniéndose muchas aplicaciones basadas en esta tecnología. El congreso en Agentes y Sistemas multi-agente ASMas, en el marco de CAEPIA 2013 proporciona una gran oportunidad para aunar expertos de diferentes áreas, y para intercambiar sus experiencias en todos los ámbitos de los agentes y sistemas multi-agente, especialmente en aquellos que afectan a su aplicación práctica, metodologías, técnicas y herramientas.

ASMas intenta reunir a investigadores y desarrolladores de los ámbitos académico e industrial, para mostrar los últimos avances, científicos y técnicos en agentes y sistemas multi-agente, discutir y debatir los principales aspectos a tener en cuenta en la evolución de este área de investigación, y mostrar los últimos avances y tecnologías obtenidos. Se trata de un área multidisciplinar que puede atraer a científicos y profesionales de CAEPIA y CEDI. De esta forma, se propone un foro multidisciplinar para discutir cómo la tecnología de agentes puede ayudar al diseño y desarrollo de sistemas reales, con la mejora añadida que ello supone. Los participantes podrán conocer de primera mano el potencial y las oportunidades que proporcionan los agentes y los sistemas multi-agente.

Queremos agradecer a los autores, los miembros del comité científico y organizador el gran trabajo realizado que ha contribuido a llevar a cabo ASMas.

Juan Manuel Corchado
Javier Bajo

Organización

Comité Organizador

Juan M. Corchado	Universidad de Salamanca
Javier Bajo	Universidad Politécnica de Madrid
Angélica de Antonio	Universidad Politécnica de Madrid
Jacinto González-Pachón	Universidad Politécnica de Madrid
Josefa Z. Hernández	Universidad Politécnica de Madrid
Ricardo Imbert	Universidad Politécnica de Madrid
Antonio Jiménez	Universidad Politécnica de Madrid
Juan F. de Paz	Universidad de Salamanca
Fernando de la Prieta	Universidad de Salamanca
Sara Rodríguez	Universidad de Salamanca
Alejando Sánchez	Universidad de Salamanca
Gabriel Villarrubia	Universidad de Salamanca
Carolina Zato	Universidad de Salamanca

Comité Científico

Carlos Alonso González, U. Valladolid
Antonio Berlanga, U. Carlos III
Vicente Botti, U. P. Valencia
Carlos Carrascosa, U. P. Valencia
Fernando Díaz, U. Valladolid
Alberto Fernández, U. Rey Juan Carlos
Rubén Fuentes, U. Complutense
Evelio J. González, U. La Laguna
Daniel Glez-Peña, U. Vigo
Vicente Julian, U. P. Valencia
Faraón Llorens, U. Alicante
José M. Molina, U. Carlos III
Antonio Moreno, U. Rovira y Virgili
Sascha Ossowski, U. Rey Juan Carlos
Juan Pavón, U. Complutense
Juan L. Posadas, U. P. Valencia
Maria del Mar Pujol, U. Alicante
Ramón Rizo, U. Alicante
Mireia L. Sempere, U. Alicante
Fernando Tadeo, U. Valladolid
María B. Vaquerizo, U. Burgos
Ana de Luis, U. Salamanca
Bruno Baruque, U. Burgos
Lourdes Borrajo, U. Vigo
Javier Carbó, U. Carlos III
Luis F. Castillo, U. Caldas
Florentino Fdez-Riverola, U. Vigo
Antonio Fernández-Skarmeta, U. Murcia
Jorge J. Gómez-Sanz, U. Complutense
Carina González, U. La Laguna
Stella M. Heras, U. P. Valencia
Rosalia Laza, U. Vigo
José R. Méndez, U. Vigo
María N. Moreno, U. Salamanca
Martí Navarro, U. P. Valencia
Xavier G. Pañeda, U. Oviedo
Marta Posada, U. Valladolid
José L. Poza-Luján, U. P. Valencia
Manuel Resinas, U. Sevilla
Elena Sánchez-Nielsen, U. La Laguna
José Soler, U. P. Valencia
Aida Valls, U. Rovira y Virgili
José R. Villar, U. Oviedo

Service-oriented Multi-Agent Systems for Ubiquitous Domains

Jorge Agüero, Miguel Rebollo, Carlos Carrascosa,Vicente Julián

Departamento de Sistemas Informáticos y Computación
Universitat Politècnica de València
Camino de Vera S/N 46022 Valencia (Spain)
jaguero, mrebollo, carrasco, vinglada@dsic.upv.es

Abstract. This work proposes to use a MDD (Model-Driven Development) approach to facilitate the development process of Agent-Based Pervasive Systems, providing the user with a set of abstractions that ease the implementation of Pervasive Systems and the deployment of a platform for their execution. The proposer architecture allows to design pervasive applications using high-level abstractions, avoiding the low-level implementation details and, after that, the Pervasive System deployment (with embedded agents and devices) is generated by using automatic transformations. In this way, a non-expert programmer will be able to develop Agent-Based Pervasive Systems, reducing the gap between the design and the implementation phases.

1 Introduction

The development of *Ubiquitous computation* or *Pervasive Systems* is a complex task, with multiple actors, devices and different hardware environments; where it is difficult to find a compact view of all components. The requirements of this kind of systems are very different [1, 2]. However, two of them are basic: (i) integration of external devices and software systems, the services that are provided by *Pervasive System* can be supplied by physical devices and also by existing software systems, and it is essential that the system supports these issues; (ii) the isolation of the technology and the manufacturer-dependent devices, in order to facilitate the development of this kind of systems, the manufacturer dependent devices must be well encapsulated in independent and generic functionalities. So, it is easy to think that this paradigm implementation requires, the development of applications in different software and hardware platforms depending on the diversity of the objects in the environment. This raises big challenges. In this way, one way to implement *Ubiquitous computation* is with embedded intelligent agents. The embedded hardware containing such agent is usually called embedded agent [3]. Each embedded agent is an autonomous entity allowing to communicate and cooperate with other agents, as part of a MAS.

MDD approach facilitates and simplifies the design process and improves the software quality for *Pervasive Systems*. It allows to re-use software and automatic transformation between models [4]. This methodology can be applied in

the development of embedded agents for ubiquitous computation, where different technologies and developing platforms coexist. This work is centered in the deployment phase of a MDD approach for the development of embedded agents for ubiquitous computation. In this sense, a MDD approach to develop agent-based software for Pervasive Environments is presented. This MDD approach simplifies the design and implementations of application prototypes. Our approach provides a method for the specification of *Pervasive Systems*, which allows to face the development of such systems from a higher abstraction level. The deployment over different execution platforms is achieved by means of automatic transformations among models that described entities and the environment (UML-like). The result is a simplified and homogeneous deployment process for *Agent-Based Pervasive Systems*. Finally, in our approach the implementation of *Pervasive System* over a service-based framework (OSGi-based) and it allows agents to manage the services, and adapt the environment, giving the possibility to create more advanced and powerful Pervasive Environments.

2 πVOM Approach

One fundamental challenge when defining a meta-model is to select which components will be included in order to model the system. To do this, common components in existing MAS and *Pervasive Systems* methodologies have been identified and incorporated to the Computation Independent Model (CIM) level. These models can be adjusted as MDD models that specify the concepts of the system, as roles, behaviors, tasks, environment, or devices. The models can be used to describe an *Agent-Based Pervasive System* without focus on platform-specific details, as a Platform Independent Model (PIM). After that, it is possible to transform PIM models into Platform Specific Models (PSM).

The proposed set of meta-models integrate different MAS modeling approaches, and it mainly focuses on the integration of Services and MAS techniques for supporting dynamical and open MAS societies [5]. This set of meta-models is called πVOM (*Platform-Independent Virtual Organization Model*). πVOM is structured in five meta-models or views: *Structure*, *Functionality*, *Normative*, *Agent*, and its *Environment*. Therefore, to model the characteristics of these components in our approach, five key concepts are used: *Organizational Unit*, *Service*, *Environment*, *Norm*, and *Agent* [5,6]. These five elements describe those members (entities) that form the organization: the topology of the organization; the services and features that the organization offer; the evolution of the organization over time; the environment where the organization is situated; and the rules about the behavior of members respectively.

Development process: The design process starts trying to model the agents and the environmental devices using the abstract components of the proposed meta-models. The *Pervasive System* design process is formed by a set transformations that finally will obtain the OSGi-Java code.

Step 1: At the beginning, the developer must create the different diagrams which model the different devices, resources or services of the agents. To perform this step, an Eclipse IDE with a set of *plugins* is employed.
Step 2: Once the model has been developed, it is necessary to select in which platforms the user wants to execute the agents. This phase corresponds to the PSM model definition of each agent. PIM-to-PSM model transformations are applied using the ATL *plug-in* and incorporating the appropriated set of transformation rules (explained in detail in [6, 7]). It is important to remark that the same agent model can be transformed into different specific agent platforms.
Step 3: The developer must apply a transformation to convert the models into the MAS code (OSGi-based). To do this, we must use a PSM-to-code transformation. In this case, we use *MOFScript* which is an Eclipse plug-in that uses templates to do the translation. These templates have been developed for two MAS platforms: JADE-Leap and ANDROMEDA [6, 10].
Step 4: Finally, the process finishes by adding the necessary drivers for the different environment devices needed in the *Pervasive System*. All the functionality of physical devices are encapsulated as OSGi services, which allow agents to use them without worrying about low-level features. However, it is necessary to provide the driver/firmware/protocol of the new devices that are not in the OSGi *bundle* library. This application has a *bundle* library, which store the *EnvironmentService* (service devices) for frequent use or re-use.

3 Deployment Process

this work proposes to use a homogeneous and unified model for implementing *Agent-Based Pervasive Systems* permitting its translation into different execution MAS platforms through MDD. The user can get the agent code automatically using MDD with minimal user intervention. The drivers or firmware must be added to support environmental devices. These drivers will be encapsulated within a service (an OSGi service), to export their functionality as a high-level abstraction (which will be managed by agents). Finally, this code should be compiled for execution over an OSGi framework, as it is showed in Figure 1.

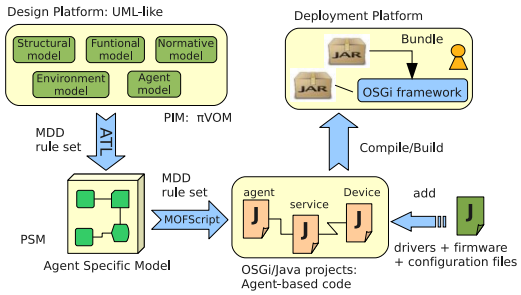


Fig. 1. Implementing Agent-based Pervasive System using MDD transformations

3.1 Deployment Platform

In order to do the before commented, it is necessary to employ an appropriated deployment platform, which includes all the computational resources needed to give support to systems designed according to the proposed development process. Many research efforts are currently being developed on prototype implementations [1]. However, some *Pervasive System* prototypes share a common architectural style, which correctly fits the requirements of these systems.

The requirements of *Pervasive Systems* are basically two: (i) is essential that the *Pervasive System* must support the integration of services provided by external devices and software systems (as another services); (ii) the isolation of the low-level abstraction of the devices (manufacturer dependent), the environmental devices must be well encapsulated in independent and generic functionalities. Therefore, an architecture style that meets these requirements is a layered architecture [1]. By means of this architecture, the system elements are organized in different levels with well-defined responsibilities. Figure 2 shows our deployment platform, which is a framework based on OSGi technology. The main layers of this deployment platform are [8]:

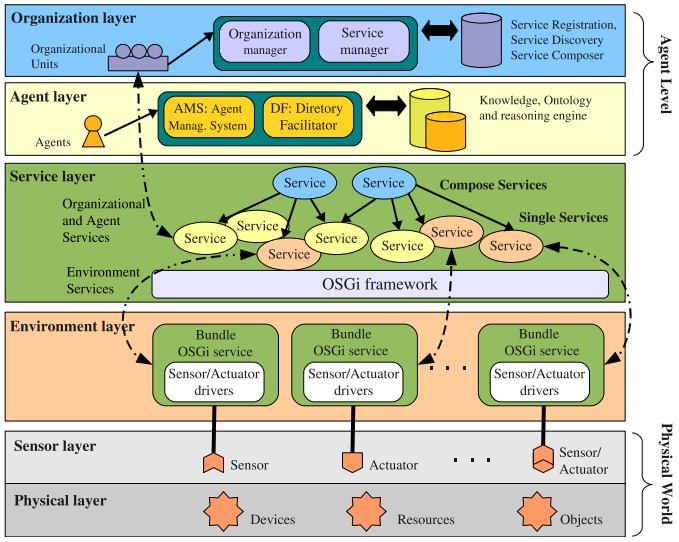


Fig. 2. Deployment architecture of Pervasive System

Physical layer, has the resources/devices that are perceived in the environment. Fully represents the real world, where the *Pervasive System* is located.

Sensor layer, has the responsibility of accessing to physical devices, through actuators and sensors, which allow to change or read the state of the devices.

Environment layer, has the responsibility of encapsulating the manufacturer dependent technology of the environment devices. The drivers that conform

this level directly export their functionalities through a *bundle*. The *bundles*, which manage similar devices of software systems from different technology or vendors, are implemented as a common interface, in order to provide a uniform way of communicating within the environment devices.

Service layer, provides the system functionality, offering services that the *Pervasive System* must supply. The services are provided by the devices located in the physical world, and by the MAS entities (agent or organizational unit). Also, at this level the services can be *single* or *composite services*, which are formed by the composition of other services.

Agent layer, supports the mechanisms in order to register, de-register and discovery of agents. In this layer the agents work together through different interactions to support complex tasks in a collaborative and dynamic way. This layer also supports the information management (*knowledge*) and the needed *knowledge models*, including the *reasoning engine* and *ontologies* needed by the agents. Furthermore, this layer provides the necessary mechanisms to support the communications and needed languages used by agents.

Organizational layer, is used as a regulatory framework for the coordination, communication, and interaction among different computational entities. This layer supports high level interoperability to integrate diverse information systems in order to share knowledge and facilitate collaboration among entities. This layer is an open system formed by the grouping and collaboration of heterogeneous entities. From a technical view, these functionalities are obtained using the THOMAS platform [9], which consists basically of a set of modular services that enable the development of agent-based organizations in open environments.

4 Implementing an application example

In order to illustrate this approach, a case study for an **Intelligent Mall** is briefly described. The intelligent Mall allows mobile users to know the product deals for a Mall based on the user profile, ie, the user receives recommendations for possible purchase and use of Mall's Services.

The system proposed provide wireless data services, which allows mobile devices (mobile phones, tablets, netbooks) to communicate to each other and to a number of servers. The network architecture provides access to wireless Services for users equipped with mobile wireless devices, via a set of access point deployed in key points around of the Mall(usually shops which offer services and products). This architecture is built upon a number of wireless communication standards (WiFi or Bluetooth) which are utilized to deliver these services to registered users. This network architecture is shown in Figure 3(a).

The *third layer* is composed of mobile devices equipped with intelligent agents that act as Personal Assistants (PA) for the users. The *second layer* consist of the shop agents, which act as access point or gateway for the supply of services and products. These shop agents are located in various points of the Mall and facilitate access to mobile users. Finally, the *first layer* is composed for the Mall which act as a "container" of different organizations Mall (shops), allowing the

creation and registration of shops and their services, and ,in addition, to control and synchronize the whole information system.

4.1 Intelligent Mall through pervasive system

The physical environment is composed of a vast and rich infrastructure of hidden electronic devices and components which may differ significantly. This approach uses different technologies adopting open technologies (services and standards, by example, standard communication protocols) reducing the seriousness of any interoperability issues that may arise at some future point. Therefore, the Mall environment will support a number of services, and one function that it could fulfill concerns the composition of elementary services into sophisticated services that would ultimately enhance the quality of the clients experience. Figure 3(b), shows how the Mall components (and technologies) are integrated and how they interact with the environment.

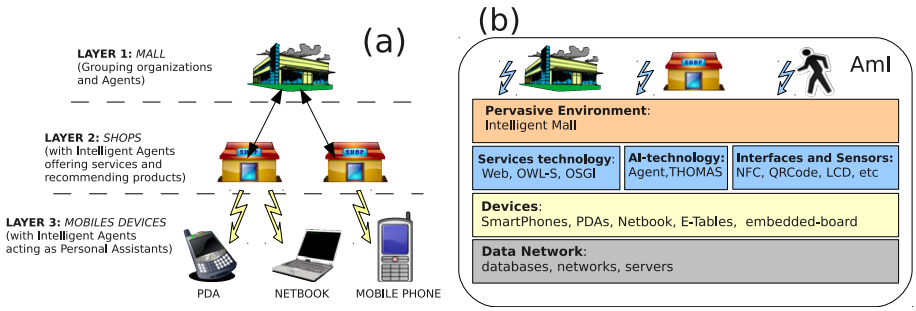


Fig. 3. The Mall: (a) network architecture, (b) as a Pervasive System

4.2 Deployment Platform for the Mall

Some agent platforms (and others technologies) are needed to implement the Mall example. The Mall is designed as a virtual organizations supporting an open agents society. As agents can enter and leave the organization at any time, the THOMAS platform is used. Agents using the *services and products* of the Mall have been designed as embedded agents on mobile phone by means of ANDROMEDA and JADE-Leap. These technologies are described below briefly. Figure 4 shows as the Intelligent Mall actors (and technologies) are coupled on the deployment platform.

A. Organizational Layer: THOMAS framework.

THOMAS [9] (MeTHods, Techniques and Tools for Open Multi-Agent Systems), is an open Multi-Agent System architecture consisting of a related set of modules that are suitable for the development of systems applied in environments that working as a "society". The use of THOMAS will give us the

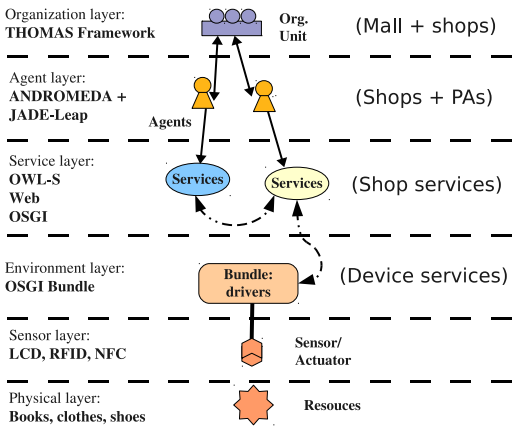


Fig. 4. Intelligent Mall over deployment platform

necessary tools for the deployment of shop and personal assistants agents into a virtual organization infrastructure. This infrastructure allows agents to dynamically enter or exit in the organization, to assign roles to each agent and to include rules or norms that agents must fulfill.

B. Agent Layer: Embedded Agent Platforms. With respect the agent deployment, we will use two agent platforms:

Andromeda[1] (ANDROid eMbeddED Agent platform) [10] is an agent platform specifically oriented to embedded agents over the *Android*[2] operating system. *Android* can be seen as a software system specifically designed for mobile devices which includes an operating system, a middleware and key applications. ANDROMEDA platform includes all the abstract concepts of the πVOM agent meta-model. The agents run as any application of *Android*, because ANDROMEDA platform can be interpreted as a new layer that is inserted into the *Android* architecture, as shown in [10].

JADE[3] one of the most popular or relevant platforms that support the agents execution, widely used because it provides programming concepts that simplify the MAS implementation. JADE is FIPA compliant in the communication infrastructure between agents. Moreover, **JADE-Leap** (JADE Light Extensible Agent Platform) is a JADE version used to implement embedded agents, which running over J2ME (Java Micro Edition).

C. Service Layer. This layer can provide various types of services to facilitate and improve the client experience on the Mall. The types of services offered by the Mall can be grouped into three categories (depending on the type of service, the role of client and provider may change): (i) Recommendation Service; (ii) Instant Information Service; (iii) Adaptive Service. The implementation of

[1] http://www.gti-ia.upv.es/sma/tools/Andromeda/
[2] Android System, http://code.google.com/android/
[3] http://jade.tilab.com/

these services is based on the use of OWL-S technology, which enables clients (PAs) and Shops (Shop agents) to automatically discover, invoke, compose, and monitor Web resources offering services, under specified constraints.

For example, Figure 5(a) shows the protocol service of a clothing store. The customer comes to the store, then the system registers it, then the client receives the services from store. The client requests the daily specials, and the store responds with special offers products that fit the customer's style.

D. Environment Layer. This layer has the responsibility of encapsulating the devices drivers (dependent manufacturer) in a single and generic service. These encapsulation is performed by OSGi technology (using a *Bundle*). The *Bundle* implemented a common interface, in order to provide a uniform way of communicating within the devices driver. The *Bundles* runs over embedded card (the BeagleBoard[4] in this approach). The BeagleBoard is an embedded low-power open-source hardware single-board computer, based on TI's ARM.

E. Sensor and Physical Layers. These layers contain the sensors and actuators of used on the Mall, such as: LCD screens, RFID cards, NFC cards, QR codes, automatic locks, parking barriers, etc. Each shop has one or more embedded card that can run an embedded agent (with ANDROMEDA over BeagleBoard), whose responsibility is the sensors management.

4.3 Implementing the embedded agents

The agent implementation starts by defining the different components used in the agent, these components are part of πVOM agent meta-model. In this case, we only describe agents running on mobile phones using ANDROMEDA or JADE-Leap platforms, due to space limitations of article. These agents (clients and consumers) receive service information of the Mall's shops (which depends of the user profile). The Shops responds with different options for products and services to the *client* agent, then the user must select his best option.

The *client* agent model represents a consumer. This agent has various behaviors that allow managing services offered by the shops. The two most important behaviors allows the agent to acquire the role of client and to discover the shops and send profiles. In Figure 5(b) these two behaviors can be observed. *Behaviour* `ShopDiscover` allows the agent to perform the activities necessary to find a shop and send the user profiles, while the other *Behaviour* (`ShopClient`) does the necessary tasks to use service and view the products offered.

These behaviors are composed of two individual capabilities (see Figure 5). As an example the *Behaviour* `ShopDiscover` includes the *Capabilities* `NetManager` and `Broadcast`. These *Capabilities* realize the actions of looking for a shop and send the user profiles respectively. The *Behaviour* `ShopClient` can be designed in similar way, as can be seen in Figure 5(b).

Once we know the different components of the system, the design continues with the creation of embedded agents and the Mall system, which is modeled as an organization that will be executed over the THOMAS platform [9]. This step

[4] http://beagleboard.org

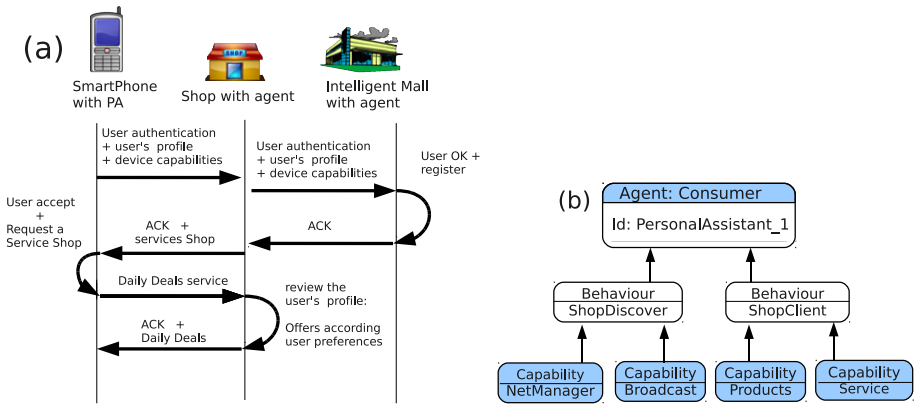


Fig. 5. (a) Daily specials service example. (b) Embedded agent model

is accomplished using the EMF Toolkit that allows users to design all the agent components according to the πVOM agent meta-model. Once the agent model is designed, EMF Toolkit allows verifying the designed system (in a semantic and syntactic way). Figure 6(a) shows a snapshot of the components of the client agent model and his correspondence with the πVOM agent meta-model.

After this, the next step is to generate agent code using this toolkit. The designer must decide in which mobile platform will be executed each agent. In this case, the *client* agent is selected to be executed over an agent platform: Andromeda or JADE-Leap. The agent in Andromeda is implemented using the same concepts employed in the abstract model. This design greatly simplifies the automatic transformation between PIM and PSM, because the Andromeda PSM is very similar to the πVOM agent model. Thus, the second step of the MDD process is not necessary. In this case, the needed transformation rules are mainly model-to-text, generating directly Andromeda code which can be combined with additional user code. The code template generated by the transformations is shown in Figure 6(b).

5 Conclusions

This work presents the application of the ideas proposed by the MDD for the design of Agent-Based Pervasive Systems. Although the use of MDD refers primarily to methodologies of object-oriented software, it was verified that the approach can be adopted in the development of Agent-Based Pervasive Systems. Concretely, a layered deployment architecture which is part of a Model Driven Development specifically designed for this kind of systems has been presented in this work. This development process allows to design pervasive applications using high-level abstractions, avoiding the low-level implementation details and, after that, the Pervasive System deployment (with embedded agents and devices) is generated by using automatic transformations. Moreover, a case study

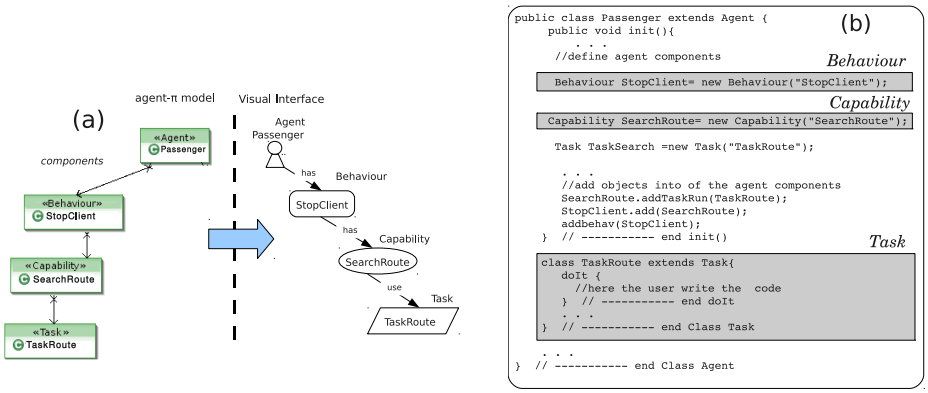


Fig. 6. (a) Designing the client agent. (b) Code template of ANDROMEDA Agent

has been presented in order to illustrate the advantages of this proposal.

Acknowledgments: This work was partially supported by TIN2012-36586-C03-01 and CONSOLIDER-INGENIO 2010 under grant CSD2007-00022 of the Spanish government.

References

1. Endres, C., Butz, A., MacWilliams, A.: A survey of software infrastructures and frameworks for ubiquitous computing. Mobile Inf. Sys. **1**(1) (2005) 41–80
2. Serrano, E., Botia, J.: Validating ambient intelligence based ubiquitous computing systems by means of artificial societies. Information Sciences **222** (2013) 3–24
3. Pavón, J., Gómez-Sanz, J., Fernández-Caballero, A., Valencia-Jiménez, J.J.: Development of intelligent multisensor surveillance systems with agents. Robotics and Autonomous Systems **55**(12) (2007) 892–903
4. Selic, B.: The pragmatics of model-driven development. IEEE software **20**(5) (2003) 19–25
5. Argente, E., Julian, V., Botti, V.: MAS Modelling based on Organizations. In: 9th Int. Workshop on Agent Oriented Software Engineering (AOSE08). (2008) 1–12
6. Agüero, J., Rebollo, M., Carrascosa, C., Julián, V.: Towards the development of agent-based organizations through MDD. International Journal on Artificial Intelligence Tools (IJAIT) **2**(2) (2013) 1350002:1–1350002:35
7. Agüero, J., Rebollo, M., Carrascosa, C., Julián, V.: Agent design using Model Driven Development. In: Int. Conference PAAMS2009. Vol. 55, (2009) 60–69
8. Agüero, J., Rebollo, M., Carrascosa, C., Julián, V.: Developing Pervasive Systems as Service-oriented Multi-Agent Systems. In: Mobiquitous2010, (2012) 78-89
9. Argente, E., Botti, V., Carrascosa, C., Giret, A., Julian, V., Rebollo, M.: An abstract architecture for virtual organizations: The THOMAS approach. Knowledge and Information Systems **29**(2) (2011) 379–403
10. Agüero, J., Rebollo, M., Carrascosa, C., Julián, V.: Does Android Dream with Intelligent Agents?. In: Int. Symposium DCAI 2008. Vol. 50, (2008) 194–204

A multi-agent architecture to combine heterogeneous inputs in multimodal interaction systems

David Griol, Jesús García-Herrero, José Manuel Molina

Applied Artificial Intelligence Group (GIAA),
Computer Science Department,
Carlos III University of Madrid,
Spain
{david.griol,jesus.garcia,josemanuel.molina}@uc3m.es

Abstract. In this paper we present a multi-agent architecture for the integration of visual sensor networks and speech-based interfaces. The proposed architecture combines different techniques related to Artificial Intelligence, Natural Language Processing and User Modeling to provide an enhanced interaction with their users. Firstly, the architecture integrates a Cooperative Surveillance Multi-Agent System (CS-MAS), which includes several types of autonomous agents working in a coalition to track and make inferences on the positions of the targets. Secondly, the proposed architecture incorporates enhanced conversational agents to facilitate human-computer interaction by means of speech interaction. Thirdly, a statistical methodology allows to model the user conversational behavior, which is learned from an initial corpus and posteriorly improved with the knowledge acquired from the successive interactions. A technique is proposed to facilitate the multimodal fusion of these information sources and consider the result for the decision of the next system action.

Keywords: Software agents, Multimodal fusion, Visual sensor networks, Surveillance applications, Spoken interaction, Conversational Agents, User Modeling, Dialog Management.

1 Introduction

Research on multimodal interaction has grown considerably during the last decade as a consequence of the advent of innovative input interfaces, as well as the development of research fields such as speech interaction and natural language processing [1–3]. Speech and natural language technologies allow users to communicate in a flexible and efficient manner, making possible to access applications in which traditional input interfaces cannot be used (e.g. in-car applications, access for disabled persons, etc). Also speech-based interfaces work seamlessly with small devices and allow users to easily invoke local applications or access remote information. For this reason, multimodal conversational agents

are becoming a strong alternative to traditional graphical interfaces which might not be appropriate for all users and/or applications [4, 5].

In human conversation, speakers adapt their message and the way they convey it to their interlocutors and to the context in which the dialog takes place. The performance of a multimodal conversational agent also depends highly on its ability to adapt to the environmental conditions, such as other people speaking near the system or noise generated by other devices. This way, information related to the environment and users presence and location is essential to achieve this adaptation [6, 7].

Adaptation can play a much more relevant role in speech-based applications [8]. For example, users have diverse ways of communication. Novice users and experienced users may want the interface to behave completely differently, such as maintaining more guided versus more flexible dialogs. In these cases, processing context is not only useful to adapt the systems' behavior, but also to cope with the ambiguities derived from the use of natural language [9, 10]. For instance, contextual information can be used to resolve anaphoric references depending on the context of the dialog or the user location.

In order to acquire this information, visual sensor networks (VSN) present a number of benefits. Firstly, the use of these networks is growing rapidly as powerful public safety and security tools (for instance, in airports [11], sea environments [12], railways or undergrounds [13], and other critical environments). Secondly, the use of agents to develop VSNs provides important advantages, like "reactivity" (agents can perceive and respond to a changing environment), "social ability" (by means of which agents interact with other agents), and "proactivity" (through which agents behave in a goal-directed way). In addition, VSNs allow to know users current position (also considering users specific speeds, directions or even specific behaviors or physical features), but also to estimate users intentions and future actions (e.g., by detecting one or more users getting closer or moving away, looking at specific places, etc.).

In this work we present a novel architecture for the integration of visual sensor networks and speech-based interfaces. Our proposal is based on the multiagent framework for deliberative camera-agents forming visual sensor networks described in [14]. In this framework, each camera is represented and managed by an individual software agent, called a surveillance-sensor agent [15]. In addition, a visual fusion agent guarantees that objects of interest are successfully tracked across the whole area, assuring continuity and seamless transitions.

As far as we are concerned, there are not previous works proposing the integration of the information provided by visual sensor networks to improve human-machine interaction by means of conversational agents. To integrate speech interaction and visual sensor networks, we propose the incorporation of enhanced conversational agents [5, 4]. This kind of agents can be defined as computer programs that accept natural language as input and produces natural language as output, engaging in a conversation with the user. To successfully manage the interaction with users, conversational agents usually carry out five main tasks: automatic speech recognition (ASR), natural language understanding (NLU), di-

alog management (DM), natural language generation (NLG), and text-to-speech synthesis (TTS). These tasks are usually implemented in different modules.

In this paper we propose to incorporate two additional modules to generate enhanced conversational agents acting in conjunction with visual sensor networks. The first module, that we have called User Modeling Module, generates a prediction of the next user action by taking into account the previous interactions with the conversational agent. User profiles are considered in this module for a better prediction. The second module, that we have called Multimodal Fusion Module, generates the next input for the dialog manager by considering the spoken interaction and the information provided by the VSN.

The main contributions of this work are: (i) To provide a detailed architecture that considers heterogeneous information generated by cooperative surveillance multi-agent systems (CS-MAS) and conversational agents; (ii) To describe a multimodal fusion methodology that takes these information sources into account to generate and encode the input of the dialog manager in the conversational agent; (iii) To propose a statistical user modeling methodology to predict the current task of the dialog and the next user action; (iv) To provide a statistical methodology for dialog management that considers the data generated by the multimodal fusion and user modeling methodologies for the selection of the next system action.

2 Proposed architecture

As described in the previous section, the proposed architecture to integrate visual sensor networks and speech interaction is based on [14]. As Figure 1 shows, different types of autonomous agents interact to fulfill this integration. The *Surveillance-Sensor Agent* tracks all the targets moving within its local field of view (FoV) and sends data to the *Visual-Fusion Agent*. It also sends information to the *Context Agent*. This agent is coordinated with other agents in order to improve surveillance quality. It can play different roles (individualized agent, object recognition agent, face recognition agent), each with different specific capabilities, but only one role at a time.

The *Visual-Fusion Agent* integrates the information sent from the associated surveillance-sensor agents. It analyzes the situation in order to manage the resources and coordinate the surveillance-sensor agents. This agent has the global view of the environment being monitored by all the surveillance-sensor agents. It is in charge of creating the dynamic coalitions of surveillance-sensor agents using contextual information and the prediction of certain situations requiring a cooperative fusion process. This agent also integrates the information from the different cameras and assures continuity and seamless transitions.

The *Recorder Agent* belongs to a specific camera with recording features only [14]. The *Planning Agent* has a general vision of the whole scene. It makes inferences on the targets and the situation. The *Context Agent* provides monitored context-dependent information. This agent indicates the semantic distance between different surveillance-sensor agents. The context agent stores information

about static objects that could provoke partial conclusions of the tracked targets but it also stores dynamic information about the scene [16]. The *Interface Agent* provides a graphical user interface that shows the evolution of the targets that are being tracked.

As described in [14], the coordination among Surveillance-Sensor Agents makes possible to jointly achieve a surveillance task. This way, the proposed CS-MAS architecture improves trajectory tracking by fusing data from several neighboring surveillance-sensor agents (camera agents in a visual sensor network), which are in a coalition.

In this paper, we propose the use of the information provided by the visual sensor network to facilitate the interaction with users by means of enhanced *Conversational Agents*. As Figure 1 shows, two main modules has been incorporated to enrich the general architecture of a conversational agent previously described. As stated in the previous section, the User Modeling module considers the previous dialog interactions and specific users features (defined by means of user profiles) to calculate a prediction of the next user action. The Multimodal Fusion module takes as input this prediction, the current user utterance, and the information provided by the surveillance sub-system. Using this information this module generates the input of the dialog manager, which selects the next system action. The following subsections describe the statistical methodologies proposed for the development of these modules.

2.1 The User Modeling module

Research in techniques for user modeling has a long history within the fields of language processing and speech technologies [17]. The main purpose of a user intention model in this field is to improve the usability of a conversational agent through the generation of corpora of interactions between the system and the user model [18].

Our proposed technique for user modeling simulates the user intention level by means of providing the next user dialog act in the same representation defined for the natural language understanding module. The lexical, syntactic and semantic information (e.g., words, part of speech tags, predicate-arguments structures, and name entities) associated to speaker u's ith clause is denoted as c_i^u.

Our model is based on the proposed in [19]. In this model, each user clause is modeled as a realization of a user action defined by a subtask to which the clause contributes, the dialog act of the clause, and the named entities of the clause. For speaker u, DA_i^u denotes the dialog label of the ith clause, and ST_i^u denotes the subtask label to which the ith clause contributes. The dialog act of the clause is determined from the information about the clause and the previous dialog context (i.e., k previous utterances) as shown in Equation 1.

$$DA_i^u = \underset{d^u \in \mathcal{D}}{\operatorname{argmax}} P(d^u | c_i^u, ST_{i-1}^{i-k}, DA_{i-1}^{i-k}, c_{i-1}^{i-k}) \tag{1}$$

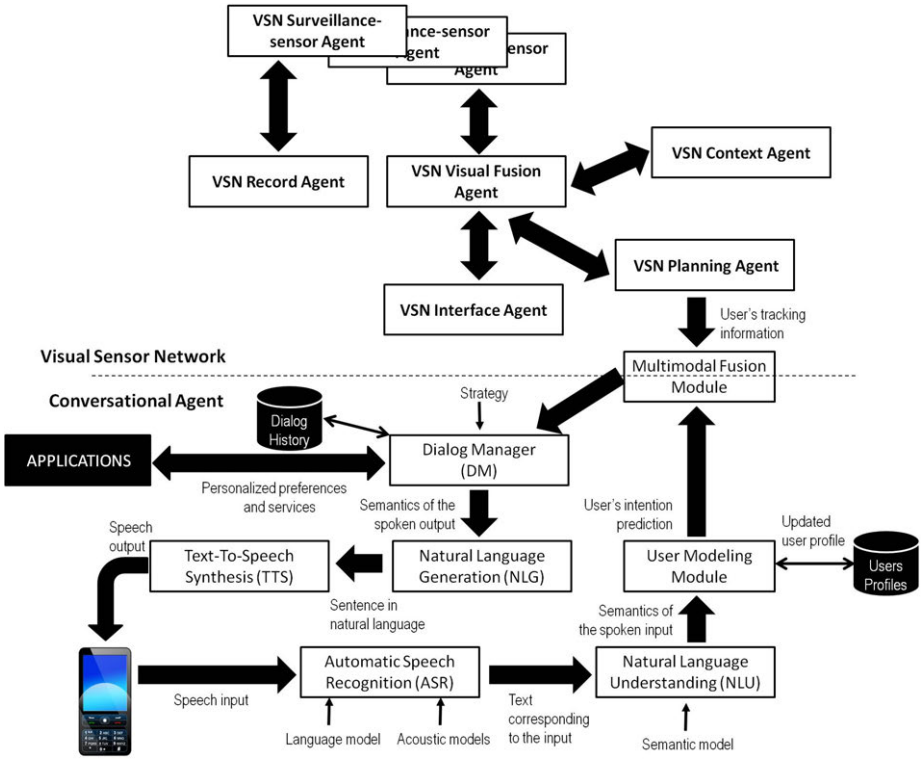


Fig. 1. Proposed multi-agent architecture to combine visual sensor networks and spoken interaction

In a second stage, the subtask of the clause is determined from the lexical information about the clause, the dialog act assigned to the clause according to Equation 1, and the dialog context, as shown in Equation 2.

$$ST_i^u = \underset{s^u \in \mathcal{S}}{\operatorname{argmax}} P(s^u | DA_i^u, c_i^u, ST_{i-1}^{i-k}, DA_{i-1}^{i-k}, c_{i-1}^{i-k}) \tag{2}$$

In our proposal, we consider both static and dynamic features to estimate the conditional distributions shown in Equations 1 and 2. Dynamic features include the dialog act of each utterance and the task/subtask of each utterance. Static features include the words in each utterance (unigrams, bigrams, and trigrams), the part of speech tags in each utterance (unigrams, bigrams, and trigrams), supertags in each utterance (unigrams, bigrams, and trigrams), and a set of features that has been included in a user profile. This profile is comprised of user's:

- Id, which he can use to log in to the system;
- Gender;

- Experience, which can be either 0 for novel users (first time the user calls the system) or the number of times the user has interacted with the system;
- Skill level, estimated taking into account the level of expertise, the duration of their previous dialogs and the time that was necessary to access a specific content and the date of the last interaction with the system. A low, medium, high or expert level is assigned using these measures;
- Most frequent objective of the user;
- Reference to the location of the previous interactions and the corresponding objective and subjective parameters for the user.

2.2 Multimodal Fusion and Dialog Management

When dealing with multiple input sources, fusion of these input sources is a necessary feature of multimodal interaction creation tools. In fact, fusion of input data can be considered as one of the distinguishing features of multimodal interaction. Typical algorithms for decision-level fusion are frame-based fusion, unification-based fusion, and hybrid symbolic/statistical fusion [20]. Symbolic/statistical fusion [21] is an evolution of standard symbolic unification-based approaches, which adds statistical processing techniques to the fusion techniques previously described. These kinds of "hybrid" fusion techniques have been demonstrated to achieve robust and reliable results.

The methodology that we propose to develop the multimodal fusion module considers the set of information sources (spoken interaction, user modeling, and video tracking) by using different machine-learning techniques. The main objective of this module is to successfully associate the visual situation detected by the VSN and the user interaction with the conversational agent.

As described in [19], the conditional distributions shown in Equations 1 and 2 can be estimated by means of the general technique of choosing MaxEnt distribution that properly estimates the average of each feature in the training data [22]. This can be written as a Gibbs distribution parameterized with weights λ as Equation 3 shows, where V is the size of the label set, X denotes the distribution of dialog acts or subtasks (DA_i^u or ST_i^u) and Φ denotes the vector of described features for user modeling.

$$P(X = st_i|\phi) = \frac{e^{\lambda_{st_i}\cdot\phi}}{\sum_{st=1}^{V} e^{\lambda_{st_i}\cdot\phi}} \tag{3}$$

Each of the classes can be encoded encoded as a bit vector such that, in the vector for class, the ith bit is one and all other bits are zero. Then, one-versus-other binary classifiers are used as Equation 4 shows.

$$P(y|\phi) = 1 - P(\overline{y}|\phi) = \frac{e^{\lambda_y\cdot\phi}}{e^{\lambda_y\cdot\phi} + e^{\lambda_{\overline{y}}\cdot\phi}} = \frac{1}{1 + e^{-\lambda'_{\overline{y}}\cdot\phi}} \tag{4}$$

where $\lambda_{\overline{y}}$ is the parameter vector for the anti-label $\overline{y}$ and $\lambda'_{\overline{y}} = \lambda_y - \lambda_{\overline{y}}$.

Once the users action prediction has been calculated, a prediction of the system action can also been generated using a similar process. Each system

action is also defined in terms of the subtask to which it contributes and the dialog act to be performed. The determination of the system action, therefore, also proceeds in two stages: prediction of the system subtask (Equation 5), and prediction of the dialog act (Equation 6).

$$ST_i^a = \underset{s^a \in \mathcal{S}}{\operatorname{argmax}} P(s^u | ST_{i-1}^{i-k}, DA_{i-1}^{i-k}, c_{i-1}^{i-k}) \tag{5}$$

$$DA_i^a = \underset{d^a \in \mathcal{D}}{\operatorname{argmax}} P(d^a | ST_i^a, ST_{i-1}^{i-k}, DA_{i-1}^{i-k}, c_{i-1}^{i-k}) \tag{6}$$

The dialog manager decides the next action of the conversational agent. In addition, it updates the dialog history, provides a context for interpreting the sentences, and coordinates the other modules of the multimodal system. Thus, the dialog manager has to deal with different sources of information such as the semantic interpretations of the users utterances, database queries results, application domain knowledge, knowledge about the users and the dialog history.

A conventional dialog manager maintains a state n such as a form or frame and relies on two functions for control, G and F. For a given dialog state n, $G(n) = a$ decides which system action to output, and then after observation o has been received, $F(n, o) = n_0$ decides how to update the dialog state n to yield n_0. This process repeats until the dialog ends.

In a statistical approach, the conventional dialog manager is extended in three respects: firstly, its action selection function $G(n) = a$ is changed to output a set of one or more (M) allowable actions given a dialog state n, $G(n) = \{a_1, a_2, \cdots, a_M\}$. Next, its transition function $F(n, o) = n_0$ is extended to allow for different transitions depending on which of these actions was taken, $F(n, a, o) = n_0$.

In order to control the interactions with the user, our proposed statistical dialog management technique represents dialogs as a sequence of pairs (A_i, U_i), where A_i is the output of the dialog system (the system answer) at time i, and U_i is the semantic representation of the user turn (the result of the understanding process of the user input) at time i; both expressed in terms of dialog acts [23]. This way, each dialog is represented by:

$$(A_1, U_1), \cdots, (A_i, U_i), \cdots, (A_n, U_n)$$

where A_1 is the greeting turn of the system, and U_n is the last user turn. We refer to a pair (A_i, U_i) as S_i, the state of the dialog sequence at time i.

In this framework, we consider that, at time i, the objective of the dialog manager is to find the best system answer A_i. This selection is a local process for each time i and takes into account the previous history of the dialog, that is to say, the sequence of states of the dialog preceding time i:

$$\hat{A}_i = \underset{A_i \in \mathcal{A}}{\operatorname{argmax}} P(A_i | S_1, \cdots, S_{i-1}) \tag{7}$$

where set $\mathcal{A}$ contains all the possible system answers.

Following Equation 7, the dialog manager selects the following system prompt by taking into account the sequence of previous pairs (A_i, U_i). The main problem to resolve this equation is regarding the number of possible sequences of states, which is usually very large. To solve the problem, we define a data structure in order to establish a partition in this space, i.e., in the history of the dialog preceding time i). This data structure, which we call *Interaction Register* (IR), contains the following information:

- sequence of user dialog acts provided by the user throughout the previous history of the dialog (i.e., the output of the NLU module);
- predicted user dialog act (generated by means of Equation 1);
- predicted user subtask (generated by means of Equation 2);
- predicted user position (provided by the agents in the virtual sensor network as explained in [14]);
- predicted system dialog act (generated by means of Equation 5);
- predicted system subtask (generated by means of Equation 6);

After applying these considerations and establishing the equivalence relation in the histories of dialogs, the selection of the best A_i is given by Equation 8.

$$\hat{A}_i = \underset{A_i \in \mathcal{A}}{\operatorname{argmax}} P(A_i | IR_{i-1}, S_{i-1}) \tag{8}$$

We propose the use of a classification process to decide the next system action following the previous equation. Specifically, we propose a multilayer perceptron (MLP) for the classification, where the input layer receives the current state of the dialog, which is represented by the term (IR_{i-1},A_i). The values of the output layer can be viewed as the a posteriori probability of selecting the different user intention given the current situation of the dialog. Figure 2 summarizes the operation of the proposed multimodal fusion and dialog management methodologies. As it can be observed, the user modeling module provides predictions of the next user dialog act and the current subtask of the dialog. Then, the system prediction module considers this information to generate the corresponding estimations for the system. The complete set of predicted values and the user position prediction provided by the planning agent are inputs of the fusion module to generate the interaction register. The dialog manager considers this register and the current user turn for the selection of the next system action.

3 Conclusions

In this paper we have described an architecture to develop multi-agent systems that considers the information generated by cooperative surveillance systems to provide user-adapted spoken interaction. To do this, we propose the integration of enhanced conversational agents in the CS-MAS architecture described in [14]. Two main modules have been incorporated in the classical architecture of a conversational agent to achieve the integration between visual sensor networks and conversational agents. These modules respectively allow to predict the next

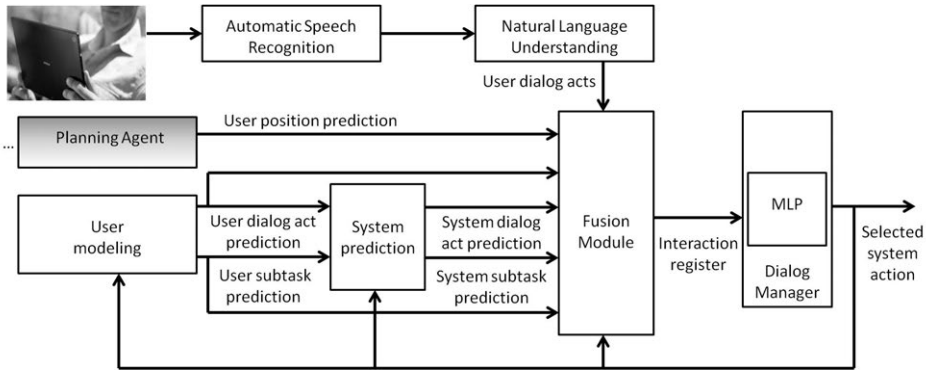


Fig. 2. Proposed multimodal fusion and dialog management methodologies for the development of conversational agents

user response for the conversational agent and carry out the fusion of visual and spoken information. The proposed multimodal fusion and dialog management techniques allow considering these heterogeneous information sources to select the next system action according to the current dialog and visual situations. Although the different methodologies proposed to develop the described modules have been evaluated in previous works [14, 24, 19], as a future work we propose the application of the described architecture to develop and evaluate a practical system in a real environment.

Acknowledgements

This work was supported in part by Projects MINECO TEC2012-37832-C02-01, CICYT TEC2011-28626-C02-02, CAM CONTEXTS (S2009/TIC-1485).

References

1. D. Gibbon, I. Mertins, and R. K. Moore(Eds.), *Handbook of Multimodal and Spoken Dialogue Systems: Resources, Terminology and Product Evaluation.* Kluwer Academic Publishers, 2000.
2. T. Heinroth and W. Minker, *Introducing Spoken Dialogue Systems into Intelligent Environments.* Springer, 2012.
3. O. Lemon and O. Pietquin(Eds.), *Data-Driven Methods for Adaptive Spoken Dialogue Systems. Computational Learning for Conversational Interfaces.* Springer, 2012.
4. R. López-Cózar and M. Araki, *Spoken, Multilingual and Multimodal Dialogue Systems.* John Wiley & Sons Publishers, 2005.
5. R. Pieraccini, *The Voice in the Machine: Building Computers that Understand Speech.* The MIT Press, 2012.

6. P. Osland, B. Viken, F. Solsvik, G. Nygreen, J. Wedvik, and S. Myklbust, "Enabling Context-Aware Applications," in *Proc. of ICIN'06*, 2006, pp. 1–6.
7. T. Lech and L. W. M. Wienhofen, "AmbieAgents: A Scalable Infrastructure for Mobile and Context-Aware Information Services," in *Proc. of AAMAS'05*, 2005, pp. 625–631.
8. P. Strauss and W. Minker, *Proactive Spoken Dialogue Interaction in Multi-Party Environments*. Springer, 2010.
9. S. Seneff, M. Adler, J. Glass, B. Sherry, T. Hazen, C. Wang, and T. Wu, "Exploiting Context Information in Spoken Dialogue Interaction with Mobile Devices," in *Proc. of IMUx'07*, 2007, pp. 1–11.
10. J. McCarthy, "Generality in Artificial Intelligence," *Communications of the ACM*, vol. 30, no. 12, pp. 1030–1035, 1987.
11. M. E. Weber and M. L. Stone, "Low altitude wind shear detection using airport surveillance radars," in *Record of IEEE Radar Conference*, 1994, pp. 52–57.
12. P. Avis, "Surveillance and canadian maritime domestic security," *Canadian Military Journal*, vol. 1, no. 4, pp. 9–15, 2003.
13. B. P. L. Lo, J. Sun, and S. A. Velastin, "Fusing visual and audio information in a distributed intelligent surveillance system for public transport systems," *Acta Automatica Sinica*, vol. 29, no. 3, pp. 393–407, 2003.
14. F. Castanedo, J. García, M. A. Patricio, and J. M. Molina, "Data fusion to improve trajectory tracking in a Cooperative Surveillance Multi-Agent Architecture," *Information Fusion*, vol. 11, pp. 243–255, 2010.
15. M. Wooldridge and N. R. Jennings, "Surveillance and Canadian maritime domestic security," *The Knowledge Engineering Review*, vol. 10, no. 2, pp. 115–152, 1995.
16. A. M. Sánchez, M. Patricio, J. García, and J. M. Molina, "Video tracking improvement using context-based information," in *Proc. of 10th Int. Conference on Information Fusion*, 2007, pp. 1–7.
17. J. Schatzmann, K. Weilhammer, M. Stuttle, and S. Young, "A Survey of Statistical User Simulation Techniques for Reinforcement-Learning of Dialogue Management Strategies," in *Knowledge Engineering Review*, vol. 21(2), 2006, pp. 97–126.
18. D. Griol, J. Carbó, and J. M. Molina, "Agent Simulation to Develop Interactive and User-Centered Conversational Agents," *Advances in Intelligent and Soft Computing*, vol. 91, pp. 69–76, 2011.
19. S. Bangalore, G. D. Fabbrizio, and A. Stent, "Learning the Structure of Task-Driven HumanHuman Dialogs," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 7, pp. 1249–1259, 2008.
20. D. Lalanne, L. Nigay, P. Palanque, P. Robinson, J. Vanderdonckt, and J.Ladry, "Fusion engines for multimodal input: a survey," in *Proc. of ICMI-MLMI'09*, 2009, pp. 153–160.
21. L. Wu, S. L. Oviatt, and P. R. Cohen, "From members to teams to committee-a robust approach to gestural and multimodal recognition," *IEEE Transactions on Neural Networks*, vol. 13, no. 4, pp. 972–982, 2002.
22. A. Berger, S. Pietra, and V. Pietra, "A maximum entropy approach to natural language processing," *Comput. Linguist*, vol. 22, no. 1, pp. 39–71, 1996.
23. D. Griol, L. F. Hurtado, E. Segarra, and E. Sanchis, "A statistical Approach to Spoken Dialog Systems Design and Evaluation," *Speech Communication*, vol. 50, no. 8-9, pp. 666–682, 2008.
24. D. Griol, J. Molina, and Z. Callejas, "Bringing together commercial and academic perspectives for the development of intelligent AmI interfaces," *Journal of Ambient Intelligence and Smart Environments*, vol. 4, no. 3, pp. 83–207, 2012.

Asignación de roles automática en organizaciones virtuales de agentes

Gabriel Villarrubia[1], Javier Bajo[2], Juan F. De Paz[1], Carolina Zato[1], Juan M. Corchado[1]

[1]Departamento de Informática y Automática, Universidad de Salamanca
Plaza de la Merced, s/n, 37008, Salamanca, España
{gvg, fcofds,carol_zato, corchado}@usal.es

[2]Departamento de Inteligencia Artificial, Universidad Politécnica de Madrid
Campus Montegancedo, Boadilla del Monte, Madrid
ffds

Resumen. La asignación de roles en organización de agentes es un proceso que requiere del reparto de tareas en los agentes que hay disponibles en las organizaciones. En determinadas ocasiones el número de roles es muy elevado por lo que resulta complejo realizar la asignación aplicando tecnologías de acuerdo y más aún en situaciones iniciales en los que no hay una distribución de agentes. En este trabajo se propone un sistema que permite realizar un reparto de roles entre agentes de una organización virtual aplicando programación lineal entera de modo que se cumplan determinadas restricciones en la organización y se maximicen determinados objetivos. El sistema se ha incluido en la plataforma de agentes PANGEA para realizar el reparto automático de roles entre los agentes.

Palabras clave: organizaciones virtuales, programación lineal, distribución de roles.

1 Introducción

Actualmente, la inclusión de aspectos organizativos está ganando terreno dentro de la investigación en los sistemas multi-agentes. La evolución de los sistemas multi-agentes a las organizaciones virtuales de agentes pretende introducir los conceptos derivados de la organización social, como son, roles, grupos y normas. Uno de los primeros intentos para incluir organizaciones en sistemas multi-agenets (MAS) se puede ver en este trabajo [5]. En el trabajo MACE (Multi-Agent Computing Environment) se introduce el concepto de rol en los MAS (Multi-Agent Systems) [6]. Más trabajos sobre el uso del concepto de organización se pueden ver en [7] y [8] pero en estos trabajos no se gestiona como diseñar las organizaciones. Una excepción se puede encontrar en [9] que describe los MAS en términos de organizaciones.

Una vez asentado el concepto de organización y rol, el siguiente reto es la organización en sí de dichas estructuras y una posibilidad es la utilización de sistemas mul-

ti-agente. El concepto de agente ha evolucionado desde una entidad autónoma a una entidad colaborativa que forma parte de una sociedad, grupo u organización. De esta manera, la interacción, coordinación y adaptación son puntos clave de una sociedad de agentes y estrechamente relacionados con la organización y reorganización que puede llegar a determinar comportamientos y por supuesto, canales de comunicación.

Las organizaciones de agentes se pueden realizar agrupando agentes que tienen intereses comunes para la realización de una determinado tarea. Al igual que en los MAS los agentes colaboran para la realización un determinado objetivo global [5] desempeñando una serie de roles. Los roles se deben de asignar en función de las capacidades de los agentes debido a que estos sistemas deben de ser capaces de adaptarse a entornos reales altamente dinámicos. Los agentes pueden cambiar de organización y en función de la organización en la que se encuentran se les asignan unos determinados roles [10]. El cambio de organización de los agente conlleva una reorganización y/o reestructuración y puesto que el paradigma de las organizaciones virtuales se basa en las sociedades de humanos, es lógico pensar que éstas también deben llevar a cabo reorganizaciones de forma eficiente. El problema reside en el cuándo y el cómo. Una alta dinamicidad conlleva una alta necesidad reorganización y encontrar una manera efectiva y formal de ordenar las sociedades de forma. En trabajos anteriores se han realizado estudios sobre diferentes alternativas para la distribución de roles [19] en agentes. No obstante, en estos trabajos la problemática no era abordable para la obtención de soluciones óptimas y eficientes en entornos altamente dinámicos.

En este trabajo se propone un modelo de distribución de roles entre los agentes de una organización para minimizar un determinado parámetro. El modelo se basa en programación lineal para búsqueda de la solución óptima de modo que se garantiza que la solución dada es la mejor en caso de que exista. El sistema se ha integrado en la arquitectura PANGEA [17] del grupo de investigación BISITE permitiendo hacer una distribución de roles entre los agentes disponibles según un modelo de organización, restricciones, capacidades y costes de los agentes. El sistema se ha validado en un caso de estudio para comprobar el correcto funcionamiento de la propuesta.

Este artículo se divide como sigue: la sección 2 describe el estado del arte, la sección 3 presenta el modelo propuesto, y la sección 4 describe los resultados y las conclusiones obtenidos.

2 Revisión del estado del arte

Los mecanismos más comunes aplicados en los sistemas adaptativos se basan en el balanceo de carga [13] [14] o asignación dinámica de tareas [11] [12]. En el caso de la asignación dinámica de tareas, se modifica la estructura global del sistema aunque se mantienen los mismos agentes a lo largo del tiempo [1]. Una aproximación diferente es la participación dinámica cuando las interacciones entre los agentes son modeladas como roles [2] [3] [4]. Además de controlar el cambio de roles y la asignación de roles de los agentes, también es necesario crear los procesos necesarios para controlar las interacciones entre los agentes que poseen los diferentes roles.

Otro aspecto de interés es el punto de partida en la generación de una organización. Para crear una organización es necesario crear un reparto de roles entre los agentes disponibles. Existen diferentes topologías de organización, que determinan cómo se ordenarán y comunicarán los agentes, entre ellas [16]:

- Jerarquías: los agentes se ordenan en una estructura de tipo árbol en la que los niveles inferiores tienen la funcionalidad básica y los niveles superiores la toma de decisiones y el control.
- Holarquías: son estructuras anidadas y jerárquicas de holones [15]. Un holón es una parte de entidad mayor resultado de la agrupación de entidades subordinadas. Este tipo de topología suele aplicarse en dominios donde los objetivos se descompongan de forma recursiva en subtareas.
- Coaliciones: son agrupaciones temporales de agentes para la consecución de un objetivo concreto, que suele obtener ciertos beneficios y reducir costes. Las coaliciones se disuelven al alcanzar el objetivo, al no existir ya la necesidad de agrupación o cuando una cantidad crítica de agentes abandona la agrupación. Internamente suele representarse como una estructura plana o con líder (representante grupo) y externamente como una entidad única y atómica.
- Grupos: son agrupaciones de agentes cooperativos que trabajan juntos en la consecución de un objetivo común. De esta manera, maximizan la utilidad del equipo. La representación de objetivos, creencias y planes se lleva a nivel de equipo. Los grupos suelen aplicarse cuando resolver problemas planteados puede conseguirse de una mejor manera de forma conjunta. Los grupos implican una mayor redundancia y flexibilidad para entornos inciertos, aunque también un aumento en las comunicaciones (por la coordinación).
- Congregaciones: son agrupaciones de agentes con características similares o complementarias. En este caso, no conllevan consecución de un objetivo específico pero sí facilitan la búsqueda de colaboradores adecuados para lograr alcanzar ese objetivo. Por este motivo, este tipo de topología suele ser pensada para objetivos a largo plazo.
- Federaciones: son agrupaciones de agentes con un representante. Los miembros del resto de la organización interaccionan solo con el representante, ceden parte de su autonomía. Este agente "representante" actúa también de intermediario entre el grupo y el mundo exterior, haciendo funciones de: (i) Broker: distribuye tareas entre miembros del grupo; (ii) Mediador: facilita interacciones entre diferentes agentes (establece contactos); (iii) Monitor: controla estados de los agentes, informa sobre eventos; (iv) Embassy: controla la comunicación de agentes externos con los de la federación (traductor de ontologías).
- Organizaciones matriciales: en este tipo de topología de organización, un agente puede ser controlado por más de un agente supervisor. Por este motivo, es necesario la utilización de mecanismos de evaluación de compromisos y resolución de conflictos locales. Es como una estructura de tipo grid en la que los agentes Managers se sitúan alrededor de agentes.

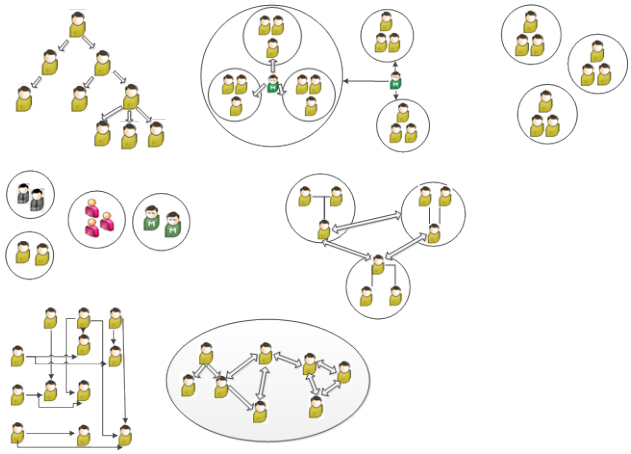


Fig. 1. De izquierda a derecha, jerarquía, holigarquía, coalición, congregación, federación, organización matricial y grupo.

3 Sistema de razonamiento propuesto

El mecanismo de razonamiento desarrollado se ha integrado en la arquitectura PANGEA [17]. Se ha incluido como un comportamiento dentro del agente que posee el rol OrganizationAgent. PANGEA es una plataforma de agentes para el desarrollo de sistemas multi-agente abiertos, especialmente aquellos que incluyen aspectos organizativos. La plataforma permite la integración de las organizaciones y ofrece herramientas para los usuarios finales. PANGEA incluye un protocolo de comunicación basado en IRC (Internet Relay Chat), el cual facilita la estabilidad del sistema.

PANGEA proporciona diferentes roles que son asignados a los agentes para asegurar el correcto funcionamiento. A continuación se indican sólo los roles asociados con el mecanismo de organización de los agentes, el resto de los roles se puede ver en el trabajo [17].

- OrganizationManager: es el responsable del manejo de las organizaciones y las suborganizaciones. Es el responsable de controlar la entrada y salida de agentes y la asignación de roles en colaboración con el OrganizationAgent que es una especialización de este agente.

3.1 OrganizationAgent

El proceso de creación de una organización virtual y la distribución de roles entre los agentes es un proceso habitualmente complicado que se puede afrontar de modo centralizado o bien distribuido. El proceso es más complicado aun cuando no se parte de una organización ya creada y se quiere hacer una distribución inicial de los diferentes roles entre los agentes.

Para llevar a cabo el proceso de distribución de roles entre los agentes se puede recurrir a teoría de acuerdo y dejar que los agentes dialoguen entre ellos para realizar un reparto de roles. Este proceso es factible cuando la organización creada requiere

de modificaciones pero para establecer una distribución inicial o en el caso de una reorganización profunda requeriría del intercambio de un elevado número de mensajes por lo que en este trabajo se recurre a programación lineal entera (PLE) para crear el reparto inicial de roles o en los casos en los que haya una variación importante en el número de agentes o roles.

Para cada una de los tipos de organizaciones se tiene un modelo de representar la información diferente, en la figura 2 se tiene un ejemplo de la representación jerárquica.

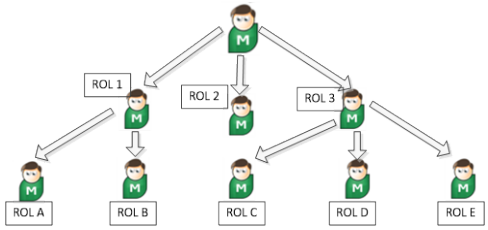


Fig. 2. Representación gráfica de la organización

Además de la información asociada a la estructura jerárquica buscada se establece información asociada a los diferentes agentes en función de las habilidades de cada uno de ellos para el desempeño de un determinado rol. En función de estas características y una serie de restricciones empezaría el funcionamiento de la programación lineal para ir asociando agentes con roles. La información almacenada por cada agente varía según el caso de estudio que se esté analizando, un ejemplo de restricciones asociada a la figura 2 mostrada anteriormente se puede ver en la tupla mostrada en (1) . Por cada rol r_i, se incluye la información de la carga l_i en la jornada que provoca la ejecución por el agente. Además, se incluye información sobre el coste de la jornada laboral c, L la carga máxima en la jornada laboral y finalmente el número máximo de roles que puede realizar R.

$$a = < (r_1, l_1), ..., (r_n, l_n), c, L, R > \quad (1)$$

Una vez se disponga de la información de la estructura y la información sobre las restricciones de los roles ya se puede plantear el problema a maximizar o minimizar. En este caso se va a crear un problema de minimización en función de los costes de cada agente. En la siguiente sección se formula el problema de programación lineal entera que permite resolver estos tipos de problemas de asignación.

3.1.1 Problema de optimización

Para explicar de modo más sencillo el problema de asignación es necesario crear una red que recoja la información sobre el problema de asignación con la información de las habilidades y cargas máximas de los agentes. A partir de la información representada en el grafo se puede plantear de modo más sencillo el problema de minimización. En la figura 3, se tiene el grafo que recoge la información asociada a unas restricciones según lo indicado en (1) . Los roles vienen definidos por los nodos del grafo con el nombre r_i donde i representa el número del rol. Los agentes vienen

definidos por a_i, i representa el identificador del agente. Los nodos 0 y f se han introducido para establecer un origen y un fin. Los valores v_{if} existente en los arcos que une al agente i con el nodo final f y representa el coste por jornada del agente. El valor c_{ij} representa la carga que ejerce el rol i sobre el agente j. El valor (l_i) indica el límite de carga del agente i.

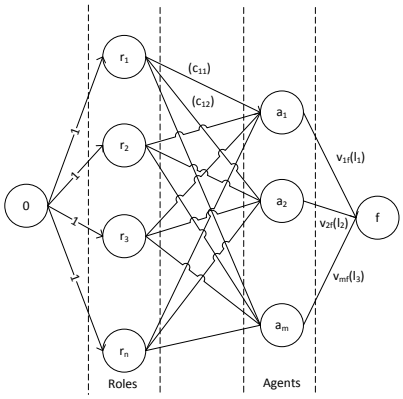


Fig. 3. Representación de las capacidades y restricciones de la información en el proceso de asignación de roles teniendo en cuenta un coste por bloques.

Para resolver el problema de PLE se aplica el algoritmo de Branch & Bound que usa el simplex [18] de modo iterativo para resolver el problema.

4 Resultado y conclusiones

El sistema se ha aplicado en un caso de estudio previo para realizar un reparto de tareas entre personal de una residencia geriátrica. El personal necesario para una jornada laboral está formado por las siguientes figuras laborales (roles): 3 celadores, 15 enfermeros, 3 responsables de planta, 15 cuidadores y 1 gerente. Se tienen en total 50 personas (agentes) disponibles para la realización de la jornada. Cada uno de los miembros del personal puede realizar determinadas tareas en base a los roles indicados. En la tabla 1 se recoge la información que relaciona los roles y los agentes y que permite determinar realizar la asignación de roles. Por simplificar la tabla sólo se ha tomado uno de los roles de cada uno de los tipos y sólo se han representado 5 agentes aunque debería de incluirse todo el personal disponible. La información viene representada según (1) . El número máximo de roles no se ha limitado para ninguno de los agentes ya que se ha limitado de manera indirecta por la carga máxima. La información de la tabla 1 se representa de la siguiente manera: coste de la jornada laboral del agente| tiempo necesario que el agente emplea para realizar el rol correspondiente | tiempo de la jornada laboral | número máximo de roles a asignar al agente. El coste es un coste por jornada laboral lo que supone que si dentro de una jornada laboral el agente puede realizar dos roles el coste será el mismo que si sólo realiza uno. Según se puede ver en la tabla 1, a cada uno de los agentes, un rol le puede suponer una

carga de trabajo diferente aunque suponga las mismas tareas. Este tiempo se estima en función de la experiencia del personal en la realización de dichas tareas.

Tabla 1. Tabla de índices de los diferentes agentes para cada uno de los roles. La información de cada celda viene dada de la siguiente manera coste jornada (v_{ij}) | hora de carga del rol (c_{ij}) | carga máxima del agente l_j | número máximo de roles.

	Agente1	Agente2	Agente3	Agente4	Agente5	…
Celador1	80|8|8|∞					…
Enfermero1		120|6|8|∞	110|7|8|∞			…
Resposable1		120|2|8|∞	110|3|8|∞		150|2|8|∞	…
Cuidadora1		120|6|8|∞	110|6|8|∞	100|6|8|∞		…
Gerente1					150|2|8|∞	…
…	…	…	…	…	…	…

A partir de la información existente en la tabla anterior el problema de optimización quedaría formulado de la siguiente manera:

$$Min\ 80x_{1f}+120x_{2f}+110x_{3f}+100x_{4f}+150x_{5f}$$

st

$$8x_{11} \leq 8$$
$$x_{1f}-x_{11} \geq 0$$
$$6x_{22}+2x_{32}+6x_{42} \leq 8$$
$$x_{2f}-x_{22} \geq 0$$
$$x_{2f}-x_{32} \geq 0$$
$$x_{2f}-x_{42} \geq 0$$
$$7x_{23}+3x_{33}+6x_{43} \leq 8$$
$$x_{3f}-x_{23} \geq 0$$
$$x_{3f}-x_{33} \geq 0$$
$$x_{3f}-x_{43} \geq 0$$
$$6x_{44} \leq 8$$
$$x_{4f}-x_{44} \geq 0$$
$$2x_{35}+2x_{55} \leq 8$$
$$x_{5f}-x_{35} \geq 0$$
$$x_{5f}-x_{55} \geq 0$$
$$x_{11} \geq 1$$
$$x_{22}+x_{23} \geq 1$$
$$x_{32}+x_{33}+x_{35} \geq 1$$
$$x_{42}+x_{43}+x_{44} \geq 1$$
$$x_{55} \geq 1$$
$$x_{11}+x_{22}+x_{32}+x_{42}+x_{23}+x_{33}+x_{43}+x_{44}+x_{35}+x_{55}=5$$
$$x_{ij}\ integer\ number$$

La variable x_{ij} cuando i y j son enteros representa un conexión entre el rol i y el agente j. x_{if} representa una conexión entre el agente i y el nodo final de la red. La variable x_{ij} según las restricciones impuestas va a valer 1 ó 0, la i representa rol y la j

agente. En los casos que aparece x_{if}, f se refiere al nodo final de la red y la i en este caso sería agente.

De manera genérica el problema de asignación se define de la siguiente manera. La primera de las restricciones implica que no se supere la carga máxima de cada uno de los agentes, la segunda es sólo una restricción para que no se asignen tareas a agentes que no van a sumar en la ecuación de optimización, por ejemplo x_{2f}-$x_{32} \geq 0$. La tercera implica que todo rol lo realice al menos un agente y la última que el rol es asignado al menos a un agente, por ejemplo si el role 3 se asigna al agente 2 entonces $x_{32}=1$ y entonces $x_{2f}=1$.

$$Min \sum_i v_{if} x_{if}$$

$$st$$

$$\forall j \quad \sum_i c_{ij} x_{ij} \leq l_j$$

$$\forall i, j \quad x_{jf} - x_{ij} \geq 0$$

$$\forall i \quad \sum_j b_{ij} x_{ij} \geq 1$$

$$\sum_{i,j} b_{ij} x_{ij} = n$$

$$0 \leq x_{ij} \leq 1$$

Para los valores que no exista arco desde el rol i al agente j $c_{ij}=0$ y $b_{ij}=0$, para los que exista arco el valor de c es la carga y el de $b_{ij}=1$, n es el número de roles.

La asignación de los roles tareas quedaría definida por los valores de las variables x_{ij}, si dicho valor es 1 indica que el agente tiene asignado el rol y si es 0 indica que el rol no está asignado al agente. Para el ejemplo mostrado en la tabla 1, el resultado sería el siguiente: el agente 1 realiza el rol celador, el agente 2 no tiene roles asignados, agente 3 el rol enfermero, agente 4 rol cuidador, agente 5 roles responsable y gerente, el coste sería de 440€.

Para realizar una comparativa del funcionamiento del sistema se procedió a comparar el funcionamiento del sistema realizando una distribución manual, una optimización basada en algoritmos genéticos y aplicando el algoritmo propuesto. Se realizó una comparativa del coste total y del personal requerido según cada una de las opciones en función de los datos de los 50 trabajadores y 37 roles del caso de estudio. Se realizó la misma prueba manteniendo las figuras laborales indicadas al inicio de la sección pero modificando las restricciones de los agentes para disponer de varias iteraciones. Los resultados obtenidos para cada iteración se pueden ver en la figura 4. Tal y como se puede ver, la reducción de costes aplicando el sistema es significativa y en todo caso el coste es menor o igual que aplicando el resto de los sistemas.

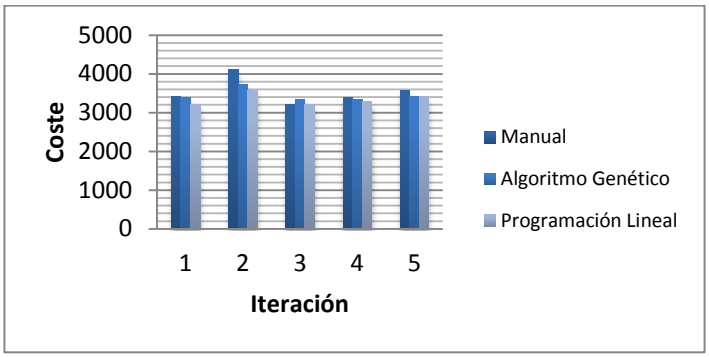


Fig. 4. Coste obtenido

El sistema de organización planteado en el agente OrganizationAgent de la arquitectura PANGEA permite reducir los costes y simplificar el proceso de distribución de roles de una organización de modo eficiente. En caso de que no fueran lineales las restricciones habría que recurrir a alguna heurística para resolver el problema. Como trabajos fututos se analizarán nuevas modelos de organización y restricciones para incorporar al agente OrganizationAgent de la arquitectura PANGEA.

Agradecimientos. Este trabajo ha sido soportado por el ministerio de economía y competitividad (INNPACTO) IPT-2011-0726-430000.

5 Referencias

1. Y, So., Durfee E.: An organizational self-design model for organizational change (1993),
2. Dignum, V.:A Model for Organizational Interaction: based on Agents, founded in Logic, SIKS Dissertation Series, Utrecht University PhD Thesis (2004).
3. Glasser, N., Morignot P.: The reorganization of societies of autonomous agents, in MAAMAW, 98–111 (1997).
4. Tambe M.: Towards flexible teamwork, Journal of Artificial Intelligence Research, (7) 83–124 (1997).
5. Glaser, N. Morignot, P.: The reorganization of societies of autonomous agents. Multi-Agent Rationality, Lecture Notes in Computer Science Volume (1237) 98-111 (1997).
6. Gasser, L., Braganza, C., Herman N.: Mace a flexible testbed for distributed AI research. In Huhns, M. N., editor, Distributed Artificial Intelligence, 119–152 (1987).
7. Fox, M.: An organizational view of distributed systems. IEEE Trans. on Man, Systems and Cybernetics, 11(1) 70–80 (1981).
8. Prasad M. V. N., Decker, K., Garvey, A., Lesser, V.: Exploring organizational designs with TAEMS: A case study of distributed data processing. In M. Tokoro, editor, Proc. of Second International Conference on Multi-Agent Sytems, 283–290 (1996).
9. Collinot, A., Drogoul, A.: Agent oriented design for a soccer robot team. In M. T. Kyoto, editor, Proc. of Second International Conference on Multi-Agent Systems, AAAI Press, 41–47 (1996).
10. Dignum, V., Dignum, F. and Sonenberg, L.: Towards dynamic reorganization of agent societies. In Proceedings of Workshop on Coordination in Emergent Agent Societies, 22-27 (2004).
11. Kivelevitch, E., Cohen, K., Kumar, M.: Market-Based Solution to the Allocation of Tasks

to Agents, Procedia Computer Science, (6) 28-33 (2011).
12. Palmer, D., Kirschenbaum, M., Murton, J., Zajac, K., Kovacina, M., Vaidyanathan, R.: "Decentralized cooperative auction for multiple agent task allocation using synchronized random number generators," Intelligent Robots and Systems, (IROS 2003). Proceedings. 2003 IEEE/RSJ International *C (*2003).
13. Montresor, A., Meling, H., Babaoğlu, Ö.: Messor Load-Balancing through a Swarm of Autonomous Agents. Agents and Peer-to-Peer Computing. Lecture Notes in Computer Science Volume (2530) 125-137 (2003).
14. Young J.L., Geon Y.P., Ho K.S., Hee Y.Y.: "A Load Balancing Scheme for Distributed Simulation Based on Multi-agent System," *Computer Software and Applications Conference Workshops (COMPSACW), 2012 IEEE 36th Annual* , 613-618 (2012).
15. Mazigh, B., Hilaire, V., Koukam, A.: Formal Specification of Holonic Multi-agent Systems: Application to Distributed Maintenance Company. In: Proceeding of 9th International Conference on Practical Applications of Agents and Multiagent Systems (2011).
16. Rodríguez, S.: Modelo adaptativo para organizaciones virtuales de agentes. University of Salamanca, Department of Computer Science and Automation, PhD Thesis (2010).
17. Zato, C., Villarrubia, G., Sánchez, A., Bajo, J., Rodríguez, S., Corchado, J.M.: PANGEA – Platform for Automatic coNstruction of orGanizations of intElligent Agents. Distributed Computing and Artificial Intelligence. Advances in Intelligent and Soft Computing (151), 229-239 (2012).
18. Zheng, Y., Lin, M.C., Manocha, D.: Efficient simplex computation for fixture layout design. Computer-Aided Design. 43 (10) 1307–1318 (2011).
19. Zato, C., De Paz, J.F., de Luis, A., Bajo, J., Corchado, J.M.: Model for assigning roles automatically in egovernment virtual organizations, Expert Systems with Applications. 39 (12) 10389-10401 (2012).

An adaptive multi-agent architecture for workplace integration

Carolina Zato[1], Fernando de la Prieta[1], Sara Rodríguez[1], Yves Demazeau[2], and Juan M. Corchado[1]

[1] Department of Science Computing and Automation,
University of Salamanca, Plaza de la Merced, S/N,
37008, Salamanca, Spain.
asanchezyu@usal.es; gvg@usal.es; carol_zato@usal.es; srg@usal.es; chamoso@usal.es

[2] Laboratoire d'Informatique de Grenoble,
Domaine Universitaire de Saint-Martin-dH'eres, Maison Jean Kuntzmann, 110 Avenue de la Chimie ,
Grenoble , France.
Yves.Demazeau@imag.fr

Abstract. This paper presents a prototype for adapting workplaces for disabled people. The effective integration of people with disabilities in the workplace is a huge challenge to society, and it presents an opportunity to make use of new technologies. The project, called AZTECA, aims to develop new tools that contribute to the employment of groups of people with visual, hearing or motor disabilities in office environments. These different tools for the disabled people have been modelled with intelligent agents that use Web services. These agents are implemented and deployed within the PANGEA platform so that PANGEA conforms the skeleton of the system and allows developing an integrated system. The main target of the prototype presented in this paper is the people detection using ZigBee technology and the personalization of the workplace according to the users disability.
Keywords: personalization workplace; disabled people; open multi-agent system; agent platform; ZigBee; proximity detection; localization.

1 Introduction

In recent years, due to technological advances, intelligent systems have become an important part of our lives. These systems can be found in multiple places, and providing a huge range of facilities. One of these facilities is personalization, as in the case of the publicity. In addition, disabled people face increasing difficulties in their workplaces. Most of the workplaces are not adapted to their disabilities, so they find barriers which cannot overcome alone. Thanks to these new intelligent systems, disabled people are able to overcome difficulties they find when performing their work successfully. Workplace customization will facilitate labour inclusion of people with disabilities. In this paper we present a workplace customization prototype. The rest of the paper is structured as follows: the next

section discusses the technologies for localization systems and introduces open multi-agent system. Section 3 details the proximity prototype system, as well as the agent platform used for its design and its integration. Section 4 explains our case study involving disabled people and finally, in section 5 some conclusions are presented.

2 Background

In this section, we discuss the state of art of the different multi-agent systems that could be used for our prototype. In addition, technologies that could be used for indoor localization of disabled people are presented. They will serve as a background for the architecture we will propose.

2.1 Open multi-agent system

Nowadays, there are many multi-agent frameworks, which help and facilitate to work with agents [1] [10] [7] [5]. The main drawback of these systems is that they are for general purpose. General purpose was consider as a major issue twenty years ago, but it is much less the case now, at a time where personal computers, devices, mobile phones and alike, have exponentially grown. In addition, the needed architecture must be able to assume the tasks for the integration of disabled people in their workplaces. And difficulties from one disabled people to another are very different. Some most known European multi-agent systems projects oriented in our research di-rection are:

- CommonWell [2] proposes architecture to support European citizens with limited mobility, or hearing or visual impairment. However, it focuses on the elderly and does not incorporate either advanced adaptive interfaces or identification and localization elements.
- DTV4A11 [3] proposes the use of digital television to integrate persons with disabilities, but it relies on the television as the only mechanisms to provide services.
- MonAMI [17] proposes a global framework to offer services to the elderly and handicapped people, but it focuses on providing these individuals only with a more independent lifestyle.

At a Spanish national level, we can also find the following relevant frameworks:

- DISCATEL [4] aims to incorporate persons with disabilities to Contact Centers or to allow them to telecommute from their home or residence.
- INREDIS (INterfaces for the RElationship between people with DISabilities) [13] is a CENIT project headed by Technosite, which investigates the concept of using personal devices with interoperability and ubiquitous characteristics to strengthen accessibility of disabled people.

– The eVia platform involves the INCLUTEC [12] study group, and is oriented toward analysis. It promotes the use and development of mobility mechanisms such as assisted wheelchairs and specialized vehicle, alternative and enhanced communication, manipulation, and cognition.

None of these multi-agent platforms can really adapt to our requirements because most of them focus on the elderly themselves or on the social integration of disabled people, but none of them addresses the inclusion of the disabled people in their workplaces. This is the reason why we have developed the PANGEA platform. The novelty of PANGEA is its dynamic and adaptable nature; it integrates new services for including peolple with visual, hearing or mobile impairments into the workforce.

2.2 Technologies for localization systems

Localization systems enable to localize and identify people or other elements whatever is their environment. They are composed of two elements: sensors and tags. Tags are found in the elements to be localized, while sensors are usually placed in fixed locations, thus generating a sensor network that can localize different devices. There are currently different localization systems, depending of the type of technology that is used. Among the different available alternatives, one can find:

– GPS: a GPS system is based on the localization through a group of satellites, which are continuously sending information. Mobile devices collect the information. These receivers calculate the positions based on the satellite coordinates and triangulation. The more satellite references there are available, the greater the precision in calculating locations.
– GSM/GPRS: Mobile phone operators also offer localization services. They function by using the antenna network of the telephone service. The localization is made through the mobile device or the service provider since both the antennas and the devices are both transmitters and receivers. The localization is calculated by using parameters such as signal time reception, angles of incidence, and triangulation of signals or relevant cells.
– RFID: Radio Frequency Identification Technology (RFID) is an alternative used to develop real time localization systems. It functions through a network of readers and RFID tags [19]. An RFID system is essentially composed of four elements: tags, readers, antennas and radios, and processing hardware [8].
– Wi-Fi: Localization systems based on Wi-Fi [16] use devices from wireless networks to calculate position. A mesh of nodes (transmitters and fixed receivers) that function as a reference for mobile nodes. The system calculates the position of the mobile nodes using the signals received by the fixed nodes.
– Bluetooth: This technology works, as Wi-Fi does, in the 2.4 GHZ ISM (Industrial, Scientific and Medical) band. It can be used to build RTLSs (Real-time Locating Systems), based on RSSI (Signal Strength Measurement) measurements. Like Wi-Fi, it also uses localization techniques such as signpost,

fingerprinting or trilateration. The main drawback that invalidates this technology for our concern is the difficulty in building WSNs made up of more than 8 devices [11].
- ZigBee: it is a low-cost, low-power consuming, two-way wireless communication standard developed by the ZigBee Alliance [20]. It allows operating in the ISM band and addresses 2.4GHz almost all over the world. The underlying IEEE 802.15.4 standard is designed to work with low-power limited resources nodes [9]. ZigBee adds network and application layers over IEEE 802.15.4 and allows more than 65,000 nodes to be connected in a mesh topology WSN.

For our workplace inclusion purpose, among this list of possible technologies for our prototype, we remove the GPS technology, since its usage is limited to outdoor places, making its indoor used impossible. We also remove GSM technology because its lack of precision. Viable technologies options are then in fact restricted to Wi-Fi and ZigBee. On the one hand, Wi-Fi technology has a low cost of infrastructure, but its precision is lower than a ZigBee system; since we need a high precision level, we choose ZigBee as the technology for the realization of our prototype. As an additional argument, ZigBee is designed to be embedded in consumer electronics, home and building automation, industrial controls, PC peripherals, medical sensor applications, toys and games, and it is intended for home, building and industrial automation purposes, which perfectly addresses the need of monitoring, control and sensory network applications [20].

3 System Overview

The proximity detection system is based on the detection of presence using ZigBee. Every computer in the room must have a ZigBee router assigned, and the system has to know their exact positions at every moment. Furthermore, all the users have to carry a ZigBee tag that is responsible for identifying each of them. Once the ZigBee tag carried by the person has been detected and identified, its location is delimited within the proximity of the sensor that identified it. The RSSI (Received Signal Strength Indicator) parameter is the responsible of measuring the receiving signal strength. The value fluctuates from an initial 0 to negative values. If the value is close to 0, the user tag is near a computer. If the user moves away from the workplace, the value starts to be negative. To handle this behaviour, we have developed an algorithm based on the RSSI parameter. We have identified 5 levels in the scale. If RSSI is greater or equal than -50, the algorithm increases one level. If the value is lower than -50, the algorithm decreases one level. When the maximum number of levels has been activated, the system understands that the user is within the proximity distance and that the user wants to use her computer. As a consequence, the profile associated to the tag which is specific to the users disability is activated and the computer is remotely switched on. Reaching the initial level of 0 means that the user has moved a significant distance away from his workplace, and

that the computer should be turned off. For switching on the computers, since the system uses a LAN infrastructure, we use the wake-on-LAN protocol. The wake-on-LAN/WAN technology allows a computer to be remotely turned on by a software call. It can be implemented in both Local Area Networks (LAN) and Wide Area Networks (WAN) [15]. It permits several usages, including turning on a Web/FTP server, remotely accessing files stored on a machine, telecommuting, and in this case, remotely turning on a computer when it is turned off [18]. Finally, we will also use an open multi-agent system implemented with PANGEA that will gather all agents and information needed to develop an integral system that will assist disabled people in their workplace. The structure of such open multi-agent system may evolve over time, its components are not known a priori, they may be heterogeneous and they also may evolve differently. The open multi-agent system should allow dealing with heterogeneous agents, and even with agents written in different languages. This makes it difficult to rely on the agents' behaviour, and requires a social control based on societal norms/rules.

3.1 Description of PANGEA

PANGEA is a service-oriented platform that allows the open multi-agent system to take maximum advantage of the distribution of the resources. To this end, all services are implemented as web services. Due to service orientation, different tools modelled with agents that consume web services can be integrated and operated from the platform, regardless of their physical location or implementation. This makes it possible for the platform to include both a service provider agent and a consumer agent, emulating client-server architecture. The ProviderAgent (a basic agent that provides a service) knows how to contact the web service, while the remaining agents know how to contact with the ProviderAgent due to their communication with the ServiceAgent, which contains information about the available services. Once the ClientAgents request has been received, the ProviderAgent extracts the required parameters and establishes the contact. Once the contact is established, the results are sent to the client agent. Using Web Services also allows the platform to introduce the Service-Oriented Architecture (SOA) [14] into multi-agent system systems. SOA is an architectural style for building applications that use services available in a network such as the web. It promotes loose coupling between software components so that they can be reused. Applications in SOA are built based on services. Running PANGEA, the platform automatically launches the following agents:

- OrganizationManager: this agent is responsible for the actual management of organizations and suborganizations. It is responsible for verifying the entry and exit of agents, and for assigning roles. To carry out these tasks, it works with the OrganizationAgent, which is a specialized version of this agent.
- InformationAgent: this agent is responsible for accessing the database containing all pertinent system information.
- ServiceAgent: this agent is responsible for recording and controlling the operation of services offered by the agents.

– NormAgent: this agent ensures compliance with all the refined norms in the organization.
– CommunicationAgent: this agent is responsible for controlling communication among agents, and for recording the interaction between agents and or-ganizations.
– Sniffer: manages the message history and filters information by controlling communication initiated by queries.

3.2 Integration of the agents of the detection prototype within PANGEA

The platform agents are implemented with Java, while the agents of the detection prototype are implemented in .NET and nesC [6]. the following figure presents the system's architecture:

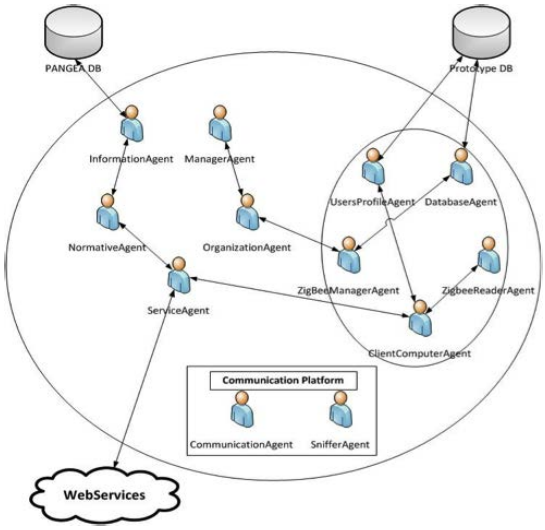


Fig. 1. System architecture

Interactions between the different kinds of agents can be seen in 1:

– ZigBeeManagerAgent: manages communication and events. It is deployed in the server machine.
– UsersProfileAgent: is responsible for managing user profiles. It is also deployed in the server machine.
– ClientComputerAgent: these are user agents located in the client computer. They are responsible for detecting the users presence with ZigBee technology, and for sending the users identification to the ZigBeeManagerAgent. They manage the strength of the signal in order to determine how close or

how far the user is. These agents are responsible for requesting the profile role adapted for the user to the ProfileManagerAgent when the users equipment needs to be turned on and personalized. Each ClientComputerAgent is localized in a piece of office equipment (computer).
- DatabaseAgent: the detection proximity system uses a database that stores data related to the users, sensors, computer equipment and status, and user profiles. It can also communicate with the InformationAgent of PANGEA.
- ZigBeeCoordinatorAgent: is included in the ZigBee device. It is responsible for coordinating the other ZigBee devices in the office.
- ZigBeeReaderAgent: these agents are included in several ZigBee devices that are used to detect the presence of a user. This agent is responsible for managing the personalized information for each user.

In the next section, an example of how these agents interact each other to customize the system is shown. Furthermore, it explains in more detail the parameters and technologies used and the devices involved in the case study.

4 Case Study

This paper presents a proximity detection system used by disabled people to facilitate their integration in workplaces. The main goal of this system is to detect the proximity of a person to a computer using ZigBee technology. For this, the main value used, as explained in previous sections, is the Signal Strength Measurement (RSSI). Thus, once the user is detected, the system can automatically switch on/off the computer, identify the user profile and load it, launch applications, and adapt the workplace to the specific needs of the user. As a result of the ZigBee technology, the prototype is notably superior to existing technologies using Bluetooth, infrareds or radiofrequencies, and is highly efficient with regards to detection and distance. The following table shows a comparison regarding bandwidth, frequency and coverage.

Table 1. Comparison Wireless Techonologies

	Bluetooth	RFID	NFC	ZigBee
Bandwidth	800 Kbps	Varies	106/212/424 Kbps	250 Kbps
Frequency	2.4 GHz	Varies	13.56 MHz	868 MHz, 2.4 GHz, 915 MHz
Coverage	10 meters	3 meters	10 cm	30-100meters

User profiles store data related to applications that are useful to users. These data can be classified according to the application, as shown in the following table.

As it can be seen, the system stores in user profiles the parameters related with the use of the computer such as the cursor speed, the magnification or the colours. All this is configurable in the prototype, which can be adapted to future users having different preferences. Additionally, different types of situations in

Table 2. Users Profiles Stored

Screen		
Parameter	Value	Description
usage	Preferred/unpreferred	Field to describe if this tool is used.
invertColourChoice	1/0	Field to describe if the colors are inverted.
magnification	1-200	Field to describe the level of magnification the user want on the screen.
Narrator		
Parameter	Value	Description
usage	Preferred/unpreferred	Field to describe if this tool is used.
speechRate	1-10	Field to describe the speed of narration.
volume	1-10	Field to describe the volume level of the tool.
Language		
Parameter	Value	Description
language	ISO 3166-1 alfa-3	Field to describe the user's language.
Virtual Keyboard		
Parameter	Value	Description
usage	Preferred/unpreferred	Field to describe if this tool is used.
Head Mouse		
Parameter	Value	Description
usage	Preferred/unpreferred	Field to describe if this tool is used.
cursorAcceleration	1-10	Field to describe the accelerating the cursor.
cursorSpeed	1-10	Field to describe the speed of the cursor.

a work environment have been taken into account, including nearby computers, shared computers, etc. Our case study includes a distribution of computers and laptops in a real office environment, distant by 2 meters. The activation zone is approximately 90cm, a distance considered as close enough to be able to initiate the activation process. It should be noted that there is a sensitive area in which it is unknown exactly which computer should be switched on: two computers being in close proximity may impede the systems efficiency from switching on the desired computer. The experiments have shown that the optimal distance separating two computers should be at least 40cm.

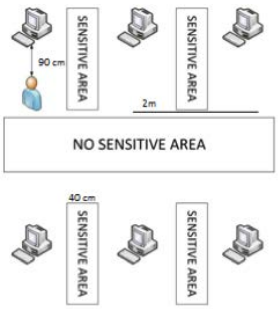


Fig. 2. Distribution of computers in the Case Study

Figure 2 shows a diagram in the case of our prototype. The prototype will be implemented in an office at our institute. The system will be installed in every computer in the workplace; so, users could use the one they want. Every user

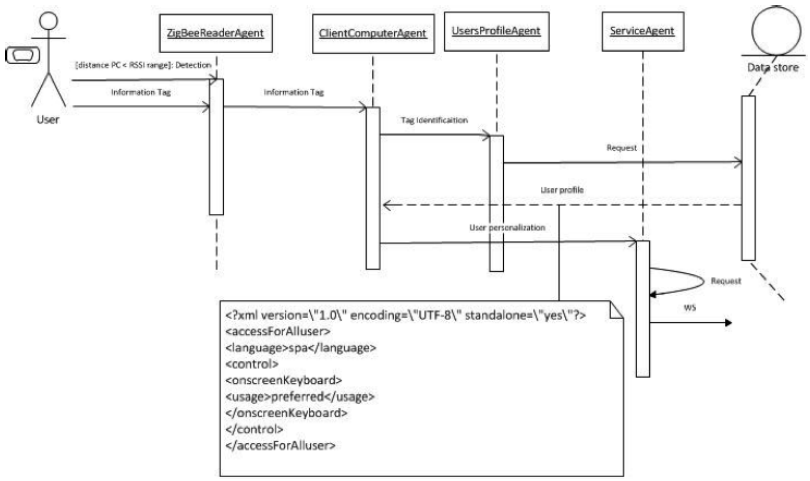


Fig. 3. Request sequence diagram

in the proposed system carries a ZigBee tag, detected by a ZigBeeReader-Agent located in each system terminal and is continuously in communication with the ClientComputerAgent. Thus, when a user tag is sufficiently close to a specific terminal (within a range defined according to the strength of the signal), the ZigBeeReaderAgent can detect the user tag and immediately send a message to the ClientComputerAgent. Next, this agent communicates the tag identification to the UsersProfileAgent, which consults the database to create the XML file that is returned to the ClientComputerAgent. The ClientComputerAgent finally interacts with the ServiceAgent to invoke the Web Services needed to personalize the computer according to the users profile. The figure 3 presents a sequence diagram that shows the interaction between the agents involved in a case of user detection by the prototype. The note of the sequence diagram shows an example of XML used to identify the user profile.

5 Conclusions

The prototype that has been described provides a multi-agent system that is able to communicate with a proximity detection system and that personalizes the workplace improving the companys workflow. The individual adaptation allows that, whatever is the disability of the person, the workplace will be adapted to him automatically. This will increase his productivity by removing the current barriers he usually has to face. One example of turning on the computer using a proximity detection system has been shown. Thanks to the SOA architecture of PANGEA, the system is high scalable. Some of future services that will be embedded include pointer services, predictive writing mechanisms, adaptation for alternative peripheral, virtual interpretations in language of signs, identification of objects by means of RFID, etc.

Acknowledgements

This project is supported by the Spanish project: Sociedades Humano-Agente: Inmersion, Adaptacion y Simulacion. TIN2012-36586-C03-03. Ministerio de Economa y Competitividad. Feder Funds.

References

1. Agent Oriented Software Pty Ltd. (2005) JACK Intelligent Agents Teams Manual, Agent Oriented Software Pty Ltd
2. CommonWell Project. (2010). http://commonwell.eu/index.php 1999.
3. Digital Television for All project. (2010). http://www.psp-dtv4all.org
4. DISCATEL. (2010). http://www.imsersounifor.org/proyectodiscatel
5. Galland S. (2010) JANUS: Another Yet General-Purpose Multiagent Platform. Seventh AOSE Technical Forum, Paris 2010.
6. Gay D., Levis P., von Behren R., Welsh M., Brewer E. and Culler D. (2003) The nesC Language: A Holistic Approach to Networked Embedded Systems. In Proceedings of Programming Language Design and Implementation (PLDI).
7. Giret, A. (2009) An open architecture for Service-Oriented Virtual Organizations. Programming Multi-Agent Systems: 7th International Workshop, ProMAS.
8. Giunchiglia F., Mylopoulos J. and Perini, A. (2002). The tropos software development methodology: Processes, models and diagrams. In: AAMAS02 Workshop on Agent Oriented Software Engineering (AOSE-2002) pp. 6374.
9. Huang Y. and Pang A. (2007) A Comprehensive Study of Low-power Operation in IEEE 802.15.4, Proceeding of the 10th ACM Symposium on Modeling, analysis and simulation of wireless and mobile systems. Chaina, Crete Island, Greece.
10. Hbner, J.F. (2007) J -Moise+ Programming organisational agents with Moise+ and Jason., Technical Fora Group at EUMAS'07.
11. Ilyas M. and Dorf R.C. (2003) The handbook of ad hoc wireless networks. CRC Press Inc., Boca Raton.
12. INCLUTEC. (2011). http://www.idi.aetic.es/evia/es/inicio/contenidos/documentacion/ documentacion_grupos_de_trabajo/contenido.aspx
13. INREDIS. (2011). http://www.inredis.es
14. Josuttis, Nicolai M. (2007) SOA in Practice., OReilly Media, Inc
15. Lieberman, P. (2011) Wake on LAN Technology, White paper. http://www.liebsoft.com/pdfs/Wake_On_LAN.pdf.
16. Martin, D. et. al. (2004) OWL-S: Semantic Markup for Web Services, W3C Member Submission, http://www.w3.org/Submission/OWL-S
17. Monami project. (2010). http://www.monami.info
18. Nedevschi S., Chandrashekar J., Liu J., Nordman B., Ratnasamy S. and Taft N. (2009) Skilled in the art of being idle: reducing energy waste in networked systems, Proceedings of the 6th USENIX symposium on Networked systems design and implementation, p.381-394, Boston, Massachusetts
19. Razavi R., Perrot J. and Guelfi, N. (2005). Adaptive modeling: an approach and a method for implementing adaptive agents. Lecture Notes in Artificial Intelligence , 3446, 136-148.
20. ZigBee Standards Organization (2006): ZigBee Specification Document 053474r13. ZigBee Alliance

17-20 sept 2013
MULTICONFERENCIA
CAEPIA'13
Madrid

Key Works

Editores asociados:

Francisco Herrera
José C. Riquelme

Entidades Colaboradoras

1544

Prefacio

Presentamos en las siguientes páginas un resumen de las catorce propuestas presentadas a la sección de trabajos relevantes publicados (Key works) dentro de la multiconferencia CAEPIA en el seno del CEDI 2013. El objetivo con este track transversal a los congresos participantes en la CAEPIA es difundir, en el marco donde se exponen los avances de la informática española, la investigación más sobresaliente que se está realizado en Inteligencia Artificial.

A pesar de ser la primera vez que se organiza, queremos agradecer a los autores el importante envío de trabajos recibidos, todos ellos de una extraordinaria calidad y que son un buen ejemplo de por qué la Inteligencia Artificial española está a la cabeza de la producción científica en el área de Computer Science. Por ejemplo, España ocupaba el 9º puesto en 2010 con el 4,71 % de las publicaciones del WoS, mientras que Computer Science-Inteligencia Artificial representa el 5,63 %, al que podemos añadir una buena parte de otras categorías como Aplicaciones Interdisciplinares (4,91 %), Cibernética (3,91 %), Sistemas de Información (3,61 %) o Teoría y Métodos (5,13 %) que con gran probabilidad son aportaciones de la IA. Por tanto, no será exagerado suponer que la IA española duplica el porcentaje de producción científica con respecto al total del resto de disciplinas.

Los trabajos presentados cubren casi todas las líneas de trabajo presentes en la CAEPIA: optimización, aprendizaje, aprendizaje automático, planificación, toma de decisiones, etc. Las revistas en las que se han publicado ocupan las primeras posiciones de Jounal Citation Report en sus respectivas líneas: IEEE T Fuzzy Systems, IEEE T Evolutionary Computation, Patter Recognition, JAIR, Information Sciences, etc. Las sesiones para exponer estos trabajos se han integrado como sesiones especiales dentro de los distintos congresos federados en la CAEPIA 2013, intentando que dentro de las limitaciones en el número de sesiones que el CEDI impone, tengan el máximo de visibilidad dentro de la comunidad.

Esperamos que esta iniciativa tenga arraigo durante las próximas ediciones de la CAEPIA y que nos permita una mejor divulgación del importante trabajo que se realiza en nuestros grupos de investigación para que la producción científica de nuestro país esté entre las diez primeras del mundo.

Francisco Herrera
José C. Riquelme

Organización

Comité Ejecutivo

Antonio Bahamonde	Universidad de Oviedo
Francisco Herrera	Universidad de Granada
José C. Riquelme	Universidad de Sevilla

Sponsors

- Asociación Española para la Inteligencia Artificial (AEPIA)

La búsqueda de la distribución de clases óptima: una aproximación para mejorar la efectividad del aprendizaje mediante métodos de remuestreo para bases de datos desbalanceadas

Iñaki Albisua, Olatz Arbelaitz, Ibai Gurrutxaga, Aritz Lasarguren, Javier Muguerza, Jesús María Pérez

Computer Science Faculty, University of the Basque Country (UPV/EHU),
Manuel Lardizabal 1, 20018 Donostia, Spain
{inaki.albisua, olatz.arbelaitz, i.gurrutxaga,
aritz.lasarguren, j.muguerza, txus.perez}@ehu.es
http://www.sc.ehu.es/aldapa

Resumen. Este artículo es un resumen del trabajo publicado en la revista *Progress in Artificial Intelligence* [1] en el que se presenta un análisis de la influencia de la distribución de clases y la utilización de técnicas de remuestreo en el comportamiento de los clasificadores.

Keywords: Distribución de clases óptima, Problemas de clases desbalanceadas, Métodos de remuestreo, SMOTE

1 Introducción

Las diferentes soluciones que se han propuesto para afrontar el problema del desbalanceo de clases se pueden clasificar en dos grandes grupos: las que introducen modificaciones en los algoritmos, y las que, normalmente mediante técnicas de remuestreo (submuestreo o sobremuestreo), equilibran las clases antes de construir el clasificador. Existe una tercera opción menos explorada que combina ambas aproximaciones, sin embargo, la basada en técnicas de remuestreo es la más numerosa debido a que es más versátil por no depender del algoritmo que se vaya a utilizar.

Este trabajo describe una metodología que permite obtener mejores resultados independientemente del método de remuestreo utilizado. Se pretende responder a las siguientes cuestiones: ¿Es el 50% siempre la mejor distribución? Weiss y Provost [2] mostraron que no era así para el submuestreo aleatorio, pero, ¿qué sucede al aplicar otras técnicas de remuestreo inteligentes como SMOTE? ¿Es la distribución óptima independiente de la técnica de remuestreo y del algoritmo de clasificación utilizados? ¿Puede ser adecuado remuestrear conjuntos de datos balanceados?

Aunque la opción de igualar la distribución de clases (50% para el caso biclásico) no es una mala opción y está muy extendida en los estudios analizados, hay ciertos autores que apuntan a que una distribución diferente puede obtener mejores resultados ya que se puede adaptar mejor a la especificidad del problema concreto. Weiss y Provost concluyeron que existe una distribución de clases diferente a la que aparece en la base de datos inicial con la que se pueden obtener mejores resultados. En un trabajo previo nuestro pudimos confirmar que el cambio en la distribución de clases en el conjunto de entrenamiento mejoraba el rendimiento de los clasificadores, pero a diferencia de las conclusiones de Weiss y Provost pudimos observar que la distribución de clases óptima dependía del algoritmo utilizado e incluso de los valores de los parámetros del mismo. Por ello, en este trabajo proponemos una

aproximación para mejorar la efectividad del proceso de aprendizaje que combina el uso de métodos de remuestreo con la distribución de clases óptima (en lugar de la balanceada). Este método se puede aplicar a cualquier conjunto de datos, esté balanceado o no.

Para corroborar la efectividad de este método hemos utilizado 29 bases de datos reales (balanceadas y no) extraídas del repositorio UCI y hemos aplicado 8 métodos diferentes de remuestreo: dos versiones de submuestreo aleatorio ($RANSUB_{ocd}$, RANSUB), sobremuestreo aleatorio (RANOVER), SMOTE, dos variantes *bordeline* de SMOTE (B_SMOTEn), y dos métodos que combinan, en diferente orden, SMOTE y la regla de edición de Wilson (ENN) propuesta para el k-*Nearest Neighbour*. Para construir los clasificadores hemos utilizado los algoritmos C4.5 y PART, y hemos estimado el AUC mediante un 5 x 10-*fold*-CV. Finalmente, hemos utilizado los test estadísticos propuestos por Demsar y García et al. para evaluar la significancia estadística de los resultados obtenidos.

En una primera fase, la metodología propone, siguiendo el trabajo de Weiss y Provost, hacer un barrido de 14 distribuciones diferentes mediante submuestreo aleatorio (por ser el método más simple y computacionalmente más eficiente), incluyendo la distribución original y la del 50%. Para ello se han utilizado 70.000 muestras de cada una de las 29 bases de datos. En una segunda fase, partiendo de la distribución óptima encontrada en la fase anterior para cada base de datos (*ocd*), exploraremos mediante una validación cruzada interna, para cada método de remuestreo y en cada base de datos, esa distribución y otras dos cercanas (±10%) seleccionando la mejor de las tres (*orm*). Comparamos los resultados de ambas distribuciones con los obtenidos mediante la balanceada (*bal*) y la base de datos original, es decir, sin utilizar técnicas de remuestreo (ORIGINAL).

2 Conclusiones

Analizando los resultados obtenidos aplicando la metodología propuesta, en la mayoría de casos se mejora respecto al caso ORIGINAL y a la opción balanceada. Tras la segunda fase, todos los métodos de remuestreo excepto SMOTE-ENN mejoran. SMOTE es el método que independientemente del algoritmo, obtiene diferencias estadísticamente significativas al 95% (Holm). RANSUB, B_SMOTE1 y B_SMOTE2 también obtienen diferencias estadísticamente significativas para diversos casos de la comparativa.

La experimentación realizada confirma que, independientemente de la distribución original, es importante encontrar la distribución de clases óptima para cada problema –que puede no coincidir con la del 50%–. Además, esta distribución óptima es dependiente, no solo de las características de los datos, sino también de la técnica de remuestreo y del algoritmo utilizado.

Bibliografía

1. I. Albisua, O. Arbelaitz, I. Gurrutxaga, A. Lasarguren, J. Muguerza, J.M. Pérez. The quest for the optimal class distribution: an approach for enhancing the effectiveness of learning via resampling methods for imbalanced data sets. *Progress in Artificial Intelligence*, Vol. 1, Issue 2, 45-63, 2013.
2. G.M Weiss, F. Provost. Learning when training are costly: the effect of class distribution on tree induction. *J. Artif. Intell. Res.*, Vol. 19, Issue 1, 315-254, 2003.

Un amplio estudio comparativo de índices de validación de agrupamiento

Olatz Arbelaitz, Ibai Gurrutxaga, Javier Muguerza, Jesús María Pérez, Iñigo Perona

Computer Science Faculty, University of the Basque Country (UPV/EHU),
Manuel Lardizabal 1, 20018 Donostia, Spain
{olatz.arbelaitz, i.gurrutxaga, j.muguerza, txus.perez,
inigo.perona}@ehu.es
http://www.sc.ehu.es/aldapa

Resumen. Este artículo es un resumen del trabajo publicado en la revista *Pattern Recognition* en el que se presenta un extenso análisis del comportamiento de 30 índices de validación del agrupamiento sobre datos con diferentes características.

Keywords: Agrupamiento, Índices de Validación del Agrupamiento, Análisis comparativo.

1 Introducción

En todo proceso de agrupamiento o *clustering*, la validación de la partición obtenida por los algoritmos utilizados es fundamental. Dado que no existe un algoritmo óptimo y que, generalmente, no se conoce el número de grupos que mejor se adapta a la estructura subyacente en los datos de entrada, la validación es primordial. La mayoría de los métodos basan su validación en el cálculo de unos valores numéricos denominados índices de validación del agrupamiento que se pueden clasificar, principalmente, en índices de validación externos e internos. En este estudio se analizan índices internos (*Cluster Validity Index*, CVI), los cuales estiman la bondad de la partición analizando la compactación y separación de los grupos generados.

En la bibliografía se han propuesto una gran cantidad de índices internos de validación diferentes, sin embargo, ninguno de los índices propuestos hasta el momento supera en efectividad al resto de los índices. La revisión más reciente y citada es de hace 25 años [2], por ello, este trabajo presenta una más amplia experimentación sistemática realizada con 30 índices actualizados en diversos contextos con diferentes características (dimensiones, número de *clusters*, grado de solapamiento...). Además, los resultados de la experimentación se han dejado públicamente accesibles en formato electrónico [1].

El estudio comparativo se ha realizado siguiendo una metodología, propuesta por nuestro grupo en un trabajo previo [3], en la que proponíamos evaluar la bondad de las particiones en base a su similitud respecto a la partición óptima, y no sólo en base al número de *clusters* de la partición óptima. Esta propuesta surge del error que se cometía en los análisis tradicionales dando por buenas particiones propuestas con el número de *clusters* óptimo aunque fueran particiones notoriamente incorrectas. Para

aplicar esta metodología es necesario utilizar alguna medida de similitud, y para evitar sesgos hemos replicado la experimentación realizada con tres medidas de similitud diferentes.

Se han aplicado tres algoritmos de *clustering*. Se han creado 720 bases de datos sintéticas para cubrir diferentes posibles combinaciones de cinco factores: nº de *clusters* y de dimensiones, nivel de ruido, solapamiento y densidad de los *clusters*. Se ha definido un hipercubo de tamaño fijo para realizar la experimentación en el que se generan aleatoriamente los *clusters* siguiendo distribuciones normales multivariante y con las restricciones mencionadas anteriormente. Además, se han utilizado 20 bases de datos reales obtenidas del repositorio UCI. El trabajo presenta el comportamiento de los CVIs para cada uno de los factores analizados. También se han aplicado tests estadísticos para determinar si hay o no diferencias significativas en el comportamiento de los índices.

2 Conclusiones

La principal conclusión es que indudablemente existen diferencias estadísticamente significativas en los comportamientos de los índices. A grandes rasgos podemos establecer tres grupos entre los índices: 10 índices en el mejor grupo, 14 en el grupo medio y 6 en el grupo con peores resultados.

Por otro lado, se puede afirmar que no hay diferencias significativas entre índices del mismo grupo, que todos los índices del primer grupo tienen diferencias significativas con respecto a los del tercer grupo, y, además, que el índice con el mejor comportamiento tiene diferencias significativas respecto a 13 del segundo grupo. Por último, sólo 2 índices del segundo grupo tienen diferencias significativas con el tercer grupo. Por tanto, aunque no existe una evidencia contundente de que un subconjunto de índices sea superior al resto, hay un subconjunto de unos 10 índices que parece comportarse mejor.

Los resultados obtenidos pueden servir de guía para seleccionar el índice más apropiado para cada tipo de problema y, además, proveen un análisis en profundidad sobre las diferencias en rendimiento entre los índices propuestos hasta ahora.

Agradecimientos. Este trabajo ha sido financiado por la subvención general a grupos de investigación de la Universidad del País Vasco, UPV/EHU (Aldapa, GIU10/02), por el Departamento de Educación y Ciencia del Gobierno Español (proyecto ModelAccess, TIN2010-15549), por el programa SAIOTEK del Gobierno Vasco (proyectos DATACC y DATACC2, S-PE11UN097 y S-PE12UN064) y por la Diputación Foral de Gipuzkoa (proyecto Zer4you, DG 10/5).

Bibliografía

1. O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J.M. Pérez, I. Perona. An extensive comparative study of cluster validity indices. *Pattern Recognition*, Vol. 46, Issue 1, 243-256, 2013. URL: http://www.sc.ehu.es/aldapa/cvi
2. G.W. Milligan, M.C. Cooper. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50, 159-179, 1985.
3. I. Gurrutxaga, J. Muguerza, O. Arbelaitz, J.M. Pérez, J.I. Martín. Towards a Standard Methodology to Evaluate Internal Cluster Validity Indices. *Pattern Recognition Letters*, 32(3), 505-515, 2011.

On the study of nearest neighbor algorithms for prevalence estimation in binary problems

Jose Barranquero, Pablo González, Jorge Díez, Juan José del Coz
{barranquero,pgonzalez,jdiez,juanjo}@aic.uniovi.es

Artificial Intelligence Center, University of Oviedo at Gijón
Campus de Viesques, 33204 Gijón, Spain

Abstract. This article is a summary of our paper of the same title [1], published in *Pattern Recognition*, to be considered as a part of the *Multi-Conference CAEPIA'13 KeyWorks*. The motivation, the main contributions and some conclusions are briefly summarized.

Keywords: Quantification, prevalence estimation, nearest neighbor, experimental methodology

1 Introduction

Recently, there is a growing interest within the machine learning community regarding the accurate estimation of the distribution of classes from a sample. This relatively new task, termed *quantification* by Forman [2], deals with the prediction of the prevalence of the positive class over a specific dataset. In practical terms, the key objective is to estimate the class distribution of a test set, provided that we have a training set in which this distribution may be noticeably different. Intuitively, this task is directly related to tracking of trends over time, such as early detection of epidemics, endangered species, market and ecosystem evolution, and other kinds of distribution changes in general.

In [1] we presented an extensive study in which the aim was to explore the applicability of *nearest neighbor* (NN) algorithms for binary quantification tasks. The intuition behind this idea was that the inherent behavior of NN algorithms should yield appropriate quantification results based on the assumption that they may be able to *remember* details of the topology of the data, independently of the presence of distribution changes between training and test. Moreover, bearing in mind that once the distance matrix has been constructed we are able to compute many different estimations in a straightforward way, thus these methods offer a cost-effective alternative for this learning problem. At the very least, our experimental study showed that they reveal themselves to be robust competitive baseline approaches, providing performance results that challenge more complex methods proposed before.

Previous quantification methods, mostly based on SVM classifiers [2], exhibited promising empirical results due to theoretical developments aimed at correcting the aggregation of individual classifier outputs. Thus, our main hypothesis was whether we could apply the aforementioned theoretical foundations

with simpler classifiers, such as NN algorithms [3], in order to stress the relevance of corrections of this kind over the use of any specific family of classifiers as base learners. The second objective was to develop a new experimental methodology for quantification based on the widespread 10-fold cross-validation procedure and Friedman-Nemenyi statistical test. This methodology was adapted to the requirements of quantification, which demand evaluating performance i) over whole sets rather than by means of individual classification outputs and ii) over a broad spectrum of test distributions in order for it to be representative.

2 Conclusions

After studying the behavior of several NN-based algorithms in the context of binary quantification, we found that, in general, two proposed weighted variants, identified as PWK and PWK$^{\alpha}$, offered the best performance. The experimental results obtained on datasets from different domains suggested that PWK and PWK$^{\alpha}$ stood out as the best approaches, without statistical differences between both, but offering clear statistical differences with respect to less robust models. Thus, our experiments did not provide any discriminative indicator regarding which of these two algorithms is more recommendable for real-world applications. The final decision should be taken in terms of the specific needs of the problem, the constraints of the environment, or the complexity of the data, among other factors. Notwithstanding, taking into account the observations discussed in the paper, it appears that PWK could be more appropriate when the minority class is much more relevant, while PWK$^{\alpha}$ seems to behave more conservatively with respect to the majority class. Furthermore, PWK is simpler, its weights are more easily interpretable and it only requires calibrating the number of neighbors.

Possible future directions for NN-based quantification could involve the selection of parameters through grid-search procedures, optimizing metrics with respect to equivalent rules, or even using these rules to calibrate the weights of each class during learning. Finally, appropriate collections of data, extracted directly from different snapshots of the same populations and showing natural shifts in their distributions, will be required in order to further analyze the quantification problem from a real-world perspective.

Acknowledgments This work was supported in part by the Spanish Ministerio de Economía y Competitividad, under research project TIN2011-23558. The contribution of Jose Barranquero was also supported by FPI grant BES-2009-027102.

References

1. Jose Barranquero, Pablo González, Jorge Díez, and Juan José del Coz. On the study of nearest neighbour algorithms for prevalence estimation in binary problems. *Pattern Recognition*, 46(2):472—482, 2013.
2. George Forman. Quantifying counts and costs via classification. *Data Mining and Knowledge Discovery*, 17(2):164–206, 2008.
3. Songbo Tan. Neighbor-weighted k-nearest neighbor for unbalanced text corpus. *Expert Systems with Applications*, 28(4):667 – 671, 2005.

A Distance-based Ranking Model Estimation of Distribution Algorithm for the Flowshop Scheduling Problem

Josu Ceberio, Ekhine Irurozki, Alexander Mendiburu, and Jose A. Lozano

Intelligent Systems Group, University of the Basque Country UPV/EHU
{josu.ceberio,ekhine.irurozki,alexander.mendiburu,ja.lozano}@ehu.es

Abstract. This is a summary of our article published in *IEEE Transactions on Evolutionary Computation*, to be considered as a part of the *Multi-Conference CAEPIA'13 KeyWorks*.

Keywords: EDAs, Generalized Mallows, permutations problems, PFSP

1 Introduction

Estimation of distribution algorithms (EDAs) constitute a powerful evolutionary algorithm tool for optimization. Based on machine learning techniques, at each iteration EDAs estimate a probability distribution from a set of candidate solutions in order to capture the most relevant information. By sampling the model, EDAs guide the search towards promising areas of the search space.

EDAs have demonstrated their validity in the solution of problems from diverse fields. However there is a type of problem for which EDAs have not been extensively developed; we refer to the *permutation-based problems*. Due to the *mutual exclusivity constraints* associated with the permutations, adaptations of the standard approaches for the discrete and continuous domains yield poor solutions, and thus, we believe that efficient designs of EDAs for permutations could further improve the current approaches.

2 Proposal and conclusions

Motivated by this idea, in this paper a novel general estimation of distribution algorithm to deal with permutation-based optimization problems is presented. Our proposal, the Generalized Mallows EDA (GM-EDA), matches the classical EDA scheme using as a probabilistic model the Generalized Mallows model (GM).

The GM model is a generalization of a *distance-based ranking model* called Mallows model. This Mallows model has been considered as analogous over permutations to the Gaussian distribution: they are both exponential unimodal models defined by two parameters, the mean and the variance. Under this model, the probability of every permutation $\sigma \in S_n$ depends on two parameters: a spread parameter θ and the distance to a central permutation σ_0, which is calculated by the *Kendall's*$-\tau$ metric $d_\tau(\sigma, \sigma_0)$. Formally, the Mallows model can be expressed as

$$P(\sigma) = \psi(\theta)^{-1} \exp(-\theta d_\tau(\sigma, \sigma_0))$$

where $\psi(\theta)$ stands for the partition function. As an extension to the Mallows model, the GM model defines an $n-1$ parametric model where each parameter describes the spread of a specific term within the solutions, increasing in this way, the expressiveness of the model. Formally,

$$P(\sigma) = \psi(\boldsymbol{\theta})^{-1} \exp(\sum_{j=1}^{n-1} -\theta_j V_j(\sigma, \sigma_0))$$

In order to prove the potential of GM-EDA, the permutation flowshop scheduling problem (PFSP) was used as test problem. The PFSP consists of scheduling n jobs on m machines in such a way that a criterion is minimized. A job consists of m operations and the j-th operation of each job must be processed on machine j for a specific given time without interruption. At a given time, a job can start on the j-th machine when its $(j-1)$th operation has finished on machine $(j-1)$, and machine j is free. The criterion employed in this case is to minimize the total flow time (TFT) which describes the sum of the completion time of the jobs. A solution for the PFSP is naturally expressed as a permutation σ of jobs, where $\sigma(i)$ denotes the rank assigned to the job i.

In a first set of experiments, GM-EDA was compared with NHBSA and EHBSA, the best performing EDAs for PFSP. The stopping criterion was set to a maximum number of evaluations for all the compared algorithms. With regard to the experimentation benchmark, 120 instances of the Taillard's benchmark of the PFSP were evaluated. Each $algorithm - instance$ pair was run 20 times. The results stated that GM-EDA is the most competitive algorithm among the evaluated EDAs.

In order to compare our approach with state-of-the-art algorithms, which are the Asynchronous Genetic Algorithm (AGA) and the Variable Neighborhood Search 4 (VNS_4), a second set of experiments was proposed, where a hybrid approach consisting of the GM-EDA with a variable neighborhood search (VNS) is introduced. The new algorithm, called HGM-EDA, is proposed as a two-stage algorithm, which in the first stage executes the previously introduced GM-EDA, and in the second stage applies a VNS to the solution provided by GM-EDA. The VNS approach that we propose in the paper explores solutions combining the most commonly used neighbourhoods in the literature for PFSP: the interchange neighborhood (N_S) and the insert neighborhood (N_I). For the second set of experiments, besides the 120 instances of the Taillard's benchmark, an additional set of 100 randomly generated instances was evaluated.

The conducted experiments showed that the hybrid version of the GM-EDA (HGM-EDA) is the best algorithm for solving the PFSP with the TFT. According to the results, HGM-EDA obtained the best TFT solution for 152 out of 220, AGA 103 and VNS_4 31.

The discussion addressed with respect to the experimental results pointed to the ability of GM-EDA as the main reason for the success of HGM-EDA. Moreover, this work confirms the good performance that probability models on rankings, such as the Generalized Mallows model, may have in EDAs for optimizing permutation-based problems.

A Methodology to Find the Elementary Landscape Decomposition of Combinatorial Optimization Problems

Francisco Chicano[1], L. Darrell Whitley[2], and Enrique Alba[1]

[1] Universidad de Málaga, Spain
{chicano,eat}@lcc.uma.es
[2] Colorado State University, Fort Collins, CO, USA
whitley@cs.colostate.edu

Resumen Presentamos un resumen de nuestro artículo publicado en 2011 en el número 19(4) de *Evolutionary Computation Journal* [1]. Describimos brevemente la motivación, la contribución y las implicaciones del trabajo.

Keywords: *Landscape* elemental, *fitness landscape*, optimización combinatoria, metodología de descomposición

1. Introducción

La teoría de *landscapes* [2] ha resultado ser útil para comprender ciertos aspectos de los problemas de optimización combinatoria. Esta teoría ofrece un marco formal centrado en el concepto de *landscape*, una tupla (X, N, f) donde X es el conjunto de soluciones tentativas de un problema de optimización; N representa el operador de vecindario, que determina las soluciones cercanas desde el punto de vista del algoritmo empleado; y f es la función objetivo (a optimizar). Dentro de esta teoría se dice que un *landscape* es *elemental* si la función objetivo es un autovector de la matriz Laplaciana asociado al espacio de configuración (X, N). En la mayoría de los casos de interés práctico esto significa que en los *landscapes* elementales se cumple una sencilla igualdad entre el valor promedio de la función objetivo en el vecindario de una solución y el valor de la función objetivo en dicha solución. La igualdad, conocida como *ecuación de onda de Grover* es la siguiente: $\text{avg}\{f(y)\}_{y\in N(x)} = f(x) + \frac{k}{d}\left(\bar{f} - f(x)\right)$, donde k es una constante característica del problema, d es el tamaño del vecindario de las soluciones y $\bar{f}$ es el valor medio de la función objetivo en el espacio de búsqueda completo.

Existe un buen número de problemas de optimización NP-difíciles que son elementales bajo ciertos vecindarios comunes. Un ejemplo es el problema del viajante de comercio (TSP). En los problemas que no son elementales, como MAXSAT o QAP, la función objetivo f puede escribirse como suma de otras funciones objetivo que sí son elementales, obteniendo una expresión similar a la

ecuación de Grover para el valor medio de la función objetivo en el vecindario de una solución. El conjunto formado por la nuevas funciones objetivo elementales se denomina *descomposición en landscapes elementales* del *landscape* original.

2. Conclusiones

Nuestro trabajo propone una metodología basada en conceptos de álgebra lineal para descomponer un *landscape* cualquiera en sus componentes elementales. La metodología distingue cinco pasos que el investigador debe dar para realizar la descomposición. Algunos de estos pasos son completamente mecánicos y no requieren intervención humana, por lo que podrían implementarse (y se ha hecho parcialmente) en una herramienta software. A continuación enumeramos estos cinco pasos: 1) reescribir la función objetivo en términos de una suma ponderada de *funciones básicas* independientes de los datos del problema, 2) obtener de forma explícita la matriz Laplaciana y las funciones básicas (vectores) para instancias de pequeño tamaño, 3) calcular la descomposición en *landscapes* elementales de las funciones básicas en las instancias de pequeño tamaño, 4) analizar la descomposición obtenida para las funciones básicas en las instancias pequeñas para tratar de encontrar una generalización de las mismas en instancias de mayor tamaño y 5) comprobar la propuesta de descomposición obtenida en el paso anterior para el caso general.

Una importante ventaja de esta metodología frente a otros métodos para obtener la descomposición es que se puede aplicar a cualquier problema de optimización combinatorio independientemente de la representación o el vecindario utilizado, siempre y cuando sea posible formalizar la función objetivo. Además, existe una herramienta software de apoyo para ayudar a los investigadores a realizar la descomposición que automatiza algunos pasos de la metodología. El trabajo ilustra los pasos de la metodología con un ejemplo sencillo, la suma de subconjunto, y desarrolla un ejemplo más complejo, la asignación cuadrática (QAP), tras la presentación detallada de la metodología.

Entre las aplicaciones recientes que hacen uso de la descomposición encontramos nuevos operadores de búsqueda, mecanismos para escapar de plateaus, busquedas locales muy eficientes, determinación de probabilidades óptimas de mutación o el cálculo exacto del *fitness-distance correlation* para problemas binarios [3]. Gracias a la metodología propuesta es posible extender estos resultados a cualquier problema de optimización combinatorio.

Acknowledgements Este trabajo ha sido parcialmente financiado por el Ministerio de Ciencia e Innovación y FEDER (proyecto TIN2008-06491-C04-01), por el gobierno de la Junta de Andalucía (proyecto P07-TIC- 03044) y por las fuerzas aéreas de EEUU (proyecto FA9550-08-1-0422).

Referencias

1. Chicano, F., Whitley, L.D., Alba, E.: A methodology to find the elementary landscape decomposition of combinatorial optimization problems. Evolutionary Computation **19**(4) (2011) 597–637
2. Reidys, C.M., Stadler, P.F.: Combinatorial landscapes. SIAM Review **44**(1) (2002) 3–54
3. Chicano, F., Alba, E.: Exact computation of the fitness-distance correlation for pseudoboolean functions with one global optimum. In: Proceedings of EvoCOP. Volume 7245 of LNCS. (2012) 111–123

Optimal Cycle Program of Traffic Lights with Particle Swarm Optimization

José García-Nieto, Ana C. Olivera, and Enrique Alba

Dept. de Lenguajes y Ciencias de la Computación,
Universidad de Málaga, E.T.S.I. Informática,
{jnieto,eat}@lcc.uma.es

Abstract. This is a summary of our article published in *IEEE Transactions on Evolutionary Computation* [1], to be considered as a part of the *Multi-Conference CAEPIA'13 KeyWorks*. The motivations, contributions, experiments, conclusions, and future work are summarized.

Keywords: Programming Cycles of Traffic Lights, Particle Swarm Optimization, SUMO Simulator of Urban Mobility, Smart Cities

1 Introduction

Pollution, congestion, security, parking, and many other problems derived from vehicular traffic are present every day in most cities around the world. Since changes in urban area infrastructure are usually not possible, researchers often agree that a correct staging of traffic lights can help to reduce these problems by improving the flow of vehicles through the cities [2, 4]. Nevertheless, as traffic lights are installed in cities and their number grows, their joint programming becomes more complex due to the huge number of combinations that appear.

The concept of automatic intelligent algorithms has proven to be useful for the optimization of cycle programs of traffic lights [4]. However, in general, authors have addressed specific urban areas with few intersections and a small number of traffic lights (from 1 to 4 intersections with around 2 traffic lights controlling each intersection), and most of them consist of ad-hoc methods designed for only one specific instance. Therefore, the optimization of large and heterogeneous real cases is still an open issue. It is a complex problem, since the greater the number of adjacent intersections, the stronger the interaction between the traffic lights (which increases the complexity of the problem). For a high impact work, an ideal research should go for large parts of the city, with hundreds semaphores and vehicles following real driving rules.

All this motivated us to propose our approach based on a Particle Swarm Optimizer (PSO) that, coupled with the well-known microscopic traffic simulator SUMO [3], will be shown to find successful cycle programs of traffic lights. In this new approach, the initialization method, the solution encoding, the fitness function, and the velocity calculation have been adapted to deal with optimal traffic light cycle programs. The task of SUMO is to evaluate cycle programs (coded as vector-solutions provided by our PSO) of the traffic lights that control the scenario instance. All this has been done on real maps, traffic lights locations, varied source-destination trips, and cities in Europe and America.

We have tested our proposal, called PSO-tl, with two large real urban areas with hundreds of traffic lights (up to 184) and vehicles (100, 300, and 500) throughout the cities of Bahia Blanca in Argentina (American style), and Malaga in Spain (European style). The behavior of our approach is analyzed under different conditions of road network dimension, and traffic density. We have also compared our PSO-tl with other techniques: Random Search, SUMO Cycle Program Generator (SCPG), Differential Evolution (DE), and Standard PSO 2011; and with cycle programs close to real ones, in terms of the two main objectives: the number of vehicles that reach their destination and the overall journey time. An analysis of the computational effort have also been carried out.

2 Conclusions

As a main conclusion, we can claim that PSO-tl performs successfully in the optimization of cycle programs for big realistic traffic scenarios where no automatic intelligent technique was taken before. For all the instances, our proposal obtained robust results statistically better than the other compared algorithms: DE, Standard PSO 2011, SCPG, and Random Search (used for sanity check).

In the scope of the studied urban scenarios, our PSO-tl scales adequately in terms of the number of traffic lights. Concerning an increasing number of vehicles, we have characterized how the scenario topology can influence the scalability power of our proposal, showing accurate results especially with regular route designs. The complete optimization process required a computational mean time in the range from 6.33 to 26.5 minutes, which is completely acceptable for use by human experts in civil engineering working for a urban agency. PSO-tl final solutions can improve the number of vehicles that reach their destination (31%) and the mean journey time (40%), for all the instances.

For future work, we will face the optimal cycle program from the focus of environmental factors reduction. We have obtained promising results concerning pollutant emissions (CO2 and NOx) and hydrocarbon savings that motivate us to follow in this line. We are also interested in modeling heterogeneous traffic scenarios with different feature conditions (climatic, driver expertise, recording time, sensors data, etc.) to endow our solutions with a strong validation.

Acknowledgements Authors acknowledge funds from the Spanish Ministry of Economy and Competitiveness (MICINN) and under contract TIN2011-28194 (roadME `http://roadme.lcc.uma.es`). José García-Nieto is supported by grant BES-2009-018767 from the MICINN.

References

1. **J. Garcia-Nieto**, A. Olivera, and E. Alba. Optimal cycle program of traffic lights with particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, In Press, 2013.
2. J. Toutouh, **J. Garcia-Nieto**, and E. Alba. Intelligent olsr routing protocol optimization for vanets. *IEEE Transactions on Vehicular Technology*, 61(4):1884–1894, 2012.
3. D. Krajzewicz, M. Bonert, and P. Wagner. The open source traffic simulation package SUMO. *RoboCup 2006 Infrastructure Simulation Competition*, 2006.
4. J. Sánchez-Medina, M. Galán, and E. Rubio. Applying a traffic lights evolutionary optimization technique to a real case: "Las Ramblas" area in Santa Cruz de Tenerife. *IEEE Transactions on Evolutionary Computation*, 12(1):25–40, feb. 2008.

Avoiding Redundant Messages in BnB-ADOPT

Patricia Gutierrez and Pedro Meseguer

IIIA - CSIC, Campus Universitat Autònoma de Barcelona
08193 Bellaterra, Spain
patricia|pedro@iiia.csic.es

Abstract. This is a summary of our paper [1] published in the JAIR journal in 2012. We briefly describe a number of relatively simple changes made on the optimal distributed contraint optimization algorithm BnB-ADOPT to remove most of its redundant messages, increasing largely its efficiency (especially communication).

Keywords: distributed constraint optimization, BnB-ADOPT, redundant messages.

1 Introduction

In constraint reasoning, most work done assumes centralized solving: all elements (variables, domains, constraints) of a problem instance are available in a single computer, that performs the considered task (solving the network, eliminating a variable, finding all solutions, etc). Recently, constraint reasoning has faced the distributed context, where problem elements are distributed among different agents (computers) and cannot be joined into a single one for a number of reasons. In this case, resolution is distributed among agents, which communicate by message passing.

Distributed CSP (DisCSP) was considered in the pioneer work of Yokoo [4], who proposed the *Asynchronous Backtracking* (ABT) algorithm, assuming that agents are linked by binary constraints, each holding one variable. ABT is correct, complete and terminates (termination is detected by quiescence in the network, requiring specialized algorithms outside ABT). Each agent requires polynomial memory. ABT has been generalized to dynamic variable ordering and non-binary constraints.

Distributed constraint optimization problems (DCOPs) appear when agents have to optimize the global aggregation of local cost functions. The ADOPT algorithm [2] optimally solves this problem. Changing the search strategy from ADOPT best-first to depth-first branch-and-bound, the BnB-ADOPT algorithm appeared [3]. Both algorithms are correct, complete and terminate, but BnB-ADOPT offers better practical performance.

On BnB-ADOPT we did some relatively simple changes [1] which largely improved efficiency (specially communication). Each agent executing BnB-ADOPT has to send at each iteration messages to its neighbors in the DFS pseudotree (an special arrangement of the constraint graph). We proved that some of these

messages were redundant, so they could be eliminated without compromising correctness, completeness and termination. We added some filters to avoid sending redundant messages. However, this did not cause much benefits in practice. We realized that resending messages had a tight relation with performance (more specifically, the issue of threshold management after reinitialization). Looking for efficiency, we allowed a small degree of redundancy (enough to properly handle thresholds), obtaining substantial benefits in computation and specially in communication (in the reported experiments, the number of exchanged messages has been divided by a number ≥ 3). We also corrected an issue when BnB-ADOPT is generalized to deal with cost functions of arity greater than 2 (applying the very same idea used in ADOPT).

2 Conclusions

Our contribution is of technical nature: we have improved the performance of a recently presented algorithm for distributed constraint optimization. Regarding the methodology used, we mention two main points:

1. Careful algorithm analysis. A deep understanding of the BnB-ADOPT algorithm (implemented by the first author in her PhD thesis) allowed us to perform a careful analysis of it, which finally lead to identifying messages who could be removed without hurting optimality and termination properties.
2. Detailed experimentation. After removing redundant messages, first experiments did not show much benefits. More detailed experiments were needed to discover the issue of thresholds in children agents. After solving this issue, obtained benefits surpassed our expectations.

Acknowledgements

This work is partially supported by the projects TIN2009-13591-C02-02 and Generalitat de Catalunya 2009-SGR-1434. At the time this research was done, Patricia Gutierrez had an FPI scholarship BES-2008-006653.

References

1. P. Gutierrez and P. Meseguer. Removing redundant messages in n-ary bnb-adopt. *JAIR*, 45:287–304, 2012.
2. P. J. Modi, W. M. Shen, M. Tambe, and M. Yokoo. Adopt: asynchronous distributed constraint optimization with quality guarantees. *Artificial Intelligence*, 161(1–2):149–180, 2005.
3. W. Yeoh, A. Felner, and S. Koenig. BnB-ADOPT: An asynchronous branch-and-bound DCOP algorithm. *JAIR*, 38:85–133, 2010.
4. M. Yokoo, E. Durfee, T. Ishida, and K. Kuwabara. The distributed constraint satisfaction problem: Formalization and algorithms. *IEEE Tr. Know. Data Engin.*, 10:673–685, 1998.

Learning Bayesian Network Classifiers from Label Proportions

Jerónimo Hernández-González, Iñaki Inza, and Jose A. Lozano

Intelligent Systems Group, University of the Basque Country UPV/EHU
{jeronimo.hernandez,inaki.inza,ja.lozano}@ehu.es

Abstract. The motivation, contributions, experiments, conclusions and future work of our article published in Pattern Recognition [1] are summarized to be considered as a keywork of the Multi-Conference CAEPIA'13.

Keywords: Supervised classification, learning from label proportions, Structural EM algorithm, Bayesian network classifiers

1 Introduction

In a problem of *learning from label proportions*[2] (LLP), the examples are provided unlabeled and grouped in *bags* —or disjoint sets of examples—. In fact, the labels of the instances in a bag are known but, for some reason, the individual pairing relation (instance, label) is lost. The group of labels can be presented as the proportion of instances in the bag that belong to each class label. We were motivated by the embryo selection problem, in *assisted reproductive technology*.

As a classical supervised learning problem, it is described by a set of n predictive variables $(X_1, \ldots, X_n)$ and a class variable C, and the objective remains that of classifying new individual instances. The dataset, in addition to a set of m unlabeled instances (each instance, a n-tuple that assigns a value to each predictive variable), specifies the distribution of instances in b bags ($D = \mathbf{B}_1 \cup \mathbf{B}_2 \cup \cdots \cup \mathbf{B}_b$ where $\mathbf{B}_i \cap \mathbf{B}_j = \emptyset, \forall i \neq j$). Each bag $\mathbf{B}_i$ groups m_i instances, where $\sum_{i=1}^{b} m_i = m$, and p_{ic} denotes the proportion of instances in $\mathbf{B}_i$ which have the label c, with $\sum_{c \in \mathcal{C}} p_{ic} = 1$.

From this description, the difficulty of pairing each instance with its class label (label uncertainty) depends on the label proportions and the bag size: the uncertainty in a bag $\mathbf{B}_i$ increases as the labels tend to be balanced (minimizing the difference among proportions $\{p_{ic}\}_{c \in \mathcal{C}}$), and the proportion of instances with uncertain labels in the whole dataset indirectly depends on the bag size (m_i).

2 Proposal and conclusions

In this paper, we show that the label proportions associated to the bags provide relevant class information that can be used to learn more accurate classifiers. Our proposal is based on the Structural EM (SEM) strategy which, in the presence of missing data, provides a suitable framework to learn Bayesian network classifiers (NB, TAN and KDB are considered) combining the search of the network structure and the estimation of the model parameters. The SEM strategy has been modified in order to consider joint label assignments, i.e. those individual label

assignments which grouply fulfill the label proportions of the bag (*valid completions*). Given a group of m_i instances and a set $\mathcal{C}$ of possible class labels, the number of assignments without further information is $|\mathcal{C}|^{m_i}$. But, if this group of instances forms a bag $\mathbf{B}_i$, with label proportions $\{p_{ic}\}_{c\in\mathcal{C}}$, the number of possible assignments is reduced to $m_i!/\prod_{c\in\mathcal{C}}(m_i \cdot p_{ic})!$. A valid completion is represented as m_i-tuple of labels, $\mathbf{e} = (e_1, e_2, \ldots, e_{m_i})$, with $e_j \in \mathcal{C}$ representing the class label assigned to the j^{th} instance of $\mathbf{B}_i$ and $\frac{1}{m_i}\sum_{j=1}^{m_i} 1_{[e_j=c]} = p_{ic}, \forall c \in \mathcal{C}$. A first version of our proposal estimates the most-probable valid completion of each bag iteratively at the E-step.

The valid completion concept has been extended to the probabilistic framework, where a matrix $\mathbf{E}$ assigns each instance $\mathbf{x}_j \in \mathbf{B}_i$ to class label c with probability E_{jc} ($\sum_{c\in\mathcal{C}} E_{jc} = 1$ with $E_{jc} \geq 0$). A *probabilistic valid completion* is a probabilistic joint assignment that fulfills, on average, the label proportions of $\mathbf{B}_i$ ($\frac{1}{m_i}\sum_{j=1}^{m_i} E_{jc} = p_{ic}, \forall c \in \mathcal{C}$). A second version of the method implements a probabilistic completion by adding up the probability of the valid completions that assign a specific instance to a class label. Finally, two reliable variants of the method have been specifically designed to deal with high uncertain LLP scenarios, using a MCMC strategy to approximate the probabilistic completion when the exhaustive method is unfeasible.

A new framework for testing LLP methods, which simulates the whole spectrum of LLP scenarios in terms of complexity for a dataset, is presented. Based on this, three experimental settings were designed with different objectives: (1) to test the benefit of using the information provided by the label proportions during the learning process, (2) to analyze the precision of the approximations of our MCMC procedure, (3) to evaluate our proposals through different LLP scenarios of increasing class uncertainty. Moreover, our proposal is compared with state-of-the-art approaches on real datasets from public repositories. The results obtained show a competitive behavior for the proposed algorithm.

The final version of our method, PMEM_{LLP}, combines an exact and an approximate procedure, only using approximate reasoning when the exact approach is unfeasible. In this issue, the future work could consist in studying the convenience of a non-constant and individual threshold to decide which reasoning procedure should be used for each bag. Another idea could be the "relaxation" of the notion of label *proportions*, considering them just a probability distribution over the class labels of the instances of the specific bag.

Acknowledgments This work has been partially supported by the Saiotek and IT609-13 programs (Basque Government), TIN2010-14931 (Spanish Ministry of Science and Innovation), network in computational bio-medicine (Carlos III Health Institute), 2011-CIEN-000060-01 (Diputación Foral de Gipuzkoa) and CRC-Biomarkers project (6-12-TK-2011-014, Diputación Foral de Bizkaia). Jerónimo Hernández-González holds a grant (FPU) from the Spanish Ministry of Science and Innovation.

References

1. Hernández-González, J., Inza, I., Lozano, J.A.: Learning Bayesian network classifiers from label proportions. Pattern Recognition (2013), in press
2. Quadrianto, N., Smola, A.J., Caetano, T.S., Le, Q.V.: Estimating labels from label proportions. In: Proceedings of the 25th ICML. pp. 776–783 (2008)

An Evaluation of Methods for Estimating the Number of Local Optima in Combinatorial Optimization Problems

Leticia Hernando, Alexander Mendiburu, and Jose A. Lozano

Intelligent Systems Group,
Department of Computer Science and Artificial Intelligence,
University of the Basque Country UPV/EHU, San Sebastián, Spain
{leticia.hernando,alexander.mendiburu,ja.lozano}@ehu.es

Abstract. This is a summary of the motivations, contributions, experiments and conclusions of the article titled *An Evaluation of Methods for Estimating the Number of Local Optima in Combinatorial Optimization Problems* published in *Evolutionary Computation*.

Keywords: Combinatorial Optimization Problems, Local Search, Number of Local Optima, Species Estimation Problem, Traveling Salesman Problem, Flow-Shop Scheduling Problem.

1 Introduction

Metaheuristic algorithms are efficient methods for solving hard Combinatorial Optimization Problems (COPs). Among them, we can find local search algorithms that rely on a neighborhood structure over the search space. Depending on this neighborhood used, the performance of the algorithms and the characteristics of the search space will change. For example, the number of local optima. In fact, this property is related to the difficulty of finding the global optima.

Unfortunatelly, given an instance of a COP and a neighborhood, we do not know in advance the number of local optima. So, the development of methods that estimate the number of local optima is very valuable. Therefore, in this work we present an evaluation of methods for estimating the number of local optima. Not only methods proposed in the combinatorial optimization field, but also those developed for the species problem in the statistics arena.

The methods chosen for the analysis, are classified bellow according to the sampling model and the field where they were previously proposed in.

- COMBINATORIAL OPTIMIZATION
  - Multinomial Sampling Model. Based on Confidence Intervals
    - **First Repetition Time**
    - **Maximal First Repetition Time**
    - **Schnabel-Census**

  - Multinomial Sampling Model. Bias Correction
    - **Jackknife**
    - **Bootstrap**
- STATISTICS
  - Multinomial Sampling Model
    - **Chao 1984**
    - **Chao & Lee 1**
    - **Chao & Lee 2**
  - Poisson-Gamma Sampling Model
    - **Chao & Bunge**

In the paper, we describe in detail these methods, explaining also the adaptation of the methods chosen from the statistics area to our problem. After the theoretic details, we test the accuracy of the estimators on three different datasets: simulated instances of COPs, instances of the Traveling Salesman Problem (TSP) taking random distances, and instances of the TSP with real distances between different cities, as well as instances of the Flow Shop Scheduling Problem (FSSP). We report a comparison of the methods, giving recommendations of the methods that provide the best estimations.

2 Conclusions

We obtain two main conclusions from the experiments, which are observed for all the methods in the three scenarios. When the attraction basins are similar in size, the methods provide estimations close to the real number of local optima, and the further the sizes of the attraction basins from the uniformity, the worse the estimations. Obviously, the higher the sample, the more precise the estimations.

Other important conclusions are provided in the paper. If we are able to take a sample of large size with respect to the number of local optima, we recommend using Jackknife. Otherwise, if we suspect that our sample size is small, we recommend Chao & Bunge and Chao & Lee 2. Due to the instability observed for Chao & Bunge, both methods should be executed independently. If the results provided are close, Chao & Bunge is usually the choice. Otherwise, select Chao & Lee 2. Finally, we realized that we can find a situation where taking a sample of size M we reach M different local optima. In this case, none of the previous methods can be applied. We propose base our estimator on the proportion of local optima over the sample.

3 Acknowledgements

This work has been partially supported by the Saiotek and Research Groups 2013-2018 (IT-242-07) programs (Basque Government), TIN2010-14931 (Spanish Ministry of Science and Innovation), COMBIOMED network in computational biomedicine (Carlos III Health Institute) and NICaiA Project PIRSES-GA-2009-247619 (European Comision). Leticia Hernando holds a grant from the Basque Government.

Fuzzy numbers from raw discrete data using linear regression

J. Moreno-Garcia, L. Jimenez-Linares, L. Rodriguez-Benitez, and E. del Castillo

Departamento de Tecnología y Sistemas de Información,
Universidad de Castilla-La Mancha, Spain
{juan.moreno,luis.jimenez,luis.rodriguez,ester.castillo}@uclm.es

Resumen This is a summary of our article published in Information Sciences [3], to be considered as a part of the Multi-Conference CAEPIA'13 KeyWorks. The motivations, contributions, experiments, and conclusions are summarized.

Keywords: Fuzzy Numbers, Data Summarisation, Linear Regression, Expected Value.

1. Introduction

In many applications, input data are represented as raw discrete data. Such data are difficult to manage and use due to their discrete nature and, sometimes, the large amount of available information. Representing data that allow for applications with which they can be efficiently managed is thus necessary. Fuzzy logic allows for easily extracting information from data and modelling data uncertainty. Fuzzy numbers with either trapezoidal or triangular membership functions can represent this kind of data: they can be defined easily, and their representation, i.e., a 3- or 4-tuple of real numbers, guarantees low memory consumption and high processing speed.

The original paper proposes an approach to construct Trapezoidal Fuzzy Numbers (TFN) using Raw Discrete Data (RDD). The proposed technique provides an efficient way to obtain trapezoidal numbers using linear regression. The technique is simple, fast ($O(n\ log\ n)$), and effective. Fuzzy numbers are automatically obtained while the application is running, and our approach can be used in any application domain that processes RDD.

The input for the method is a dataset represented as $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ and the output is a $TFN = \{a, b, c, d\}$ compounded by the segments $s_{a,b}$, $s_{b,c}$ and $s_{c,d}$ (Figure 2 in original paper). An iterative process that cuts Y-axis from the maximum to minimum Y-axis values is performed. A new trapezoid is generated in each iteration. To do this, the intersection points for a straight line parallel to X-Axis with the input data line are used to determine the position of the indices b and c (Figure 3 in original paper). Afterward, $s_{a,b}$ and $s_{c,d}$ are calculated using linear regression techniques, while the $s_{b,c}$ position, parallel to

X, must be computed to avoid discontinuities in the obtained trapezoid. Finally, the error of the obtained trapezoid is calculated: the weighted average of the errors of each segment of the trapezoid is used for this purpose, the selected trapezoid is the one with less error. This process finishes when cut reaches the average of the values from y_a to y_d.

The algorithm needs a constant α. Empirical observations show that Y values close to 0 may cause problems in determining the TFN support [a, d]. More precisely, the obtained TFN support is large (Figures 8 and 9 in original paper), and the zones with values close to 0 are covered when the segments are calculated, and therefore, the segments do not adjust correctly to zones with higher and more reliable values (Section 4 in original paper). The support thus contains the continuous values x_i that have y_i values greater than a percentage of the maximum distribution value.

The experiments are performed using different types of input data: a Gaussian function, a Sigmoidal function, three datasets of synthetic discrete data, and an histogram obtained from a colour satellite image. The weighted average of the errors of each segment is used to measure the error of obtained trapezoid, while the expected value is used to verify the representativity of the fuzzy number compared to the input RDD [1,2]. As we say above, our method uses a parameter called α, changes in this parameter can have a large influence on the final results. A set of tests has thus been applied to study the sensitivity of the algorithm to changes in α.

2. Conclusions

This work presents a new method to obtain a trapezoidal fuzzy number from raw discrete data. The proposed method has an order of complexity of $O(n\ log\ n)$ and is unsupervised. The experimental results demonstrate the validity of the proposed technique. This set of tests shows that the shape differences between the TFN and input function are minimal. Furthermore, to measure the representativity of the fuzzy number compared to RDD, we use the Expected Value. In every experiment, both expected values (input data and TFN) are similar. Finally, the operation parameter α has been studied, indicating that the proposed method produces better results using alpha values greater than 0.

Referencias

1. S. Heilpern. The expected value of a fuzzy number. *Fuzzy Sets and Systems*, 47(1): 81–86, 1992.
2. B. Liu, Y. Liu. Expected value of fuzzy variable and fuzzy expected value models. *IEEE Transactions on Fuzzy Systems*, 10(4): 445–450, 2002.
3. J. Moreno-Garcia, L. Jimenez-Linares, L. Rodriguez-Benitez, E. Del Castillo. Fuzzy numbers from raw discrete data using linear regression. *Information Sciences*, 233: 1–14, Jun 2013.

Multilabel Classifiers with a Probabilistic Thresholding Strategy

José Ramón Quevedo, Oscar Luaces, Antonio Bahamonde
{quevedo,oluaces,antonio}@aic.uniovi.es

Artificial Intelligence Center, University of Oviedo at Gijón
Campus de Viesques, 33204 Gijón, Spain

Abstract This article is a summary of our paper of the same title [1] published in *Pattern Recognition*, to be considered as a part of the *Multi-Conference CAEPIA'13 Key works*.

Keywords: Multilabel Classification, Thresholding Strategies, Posterior Probability, Expected Loss

1 Introduction

Let $\boldsymbol{L}$ be a finite and non-empty set of *labels* or *classes*. In a *multiclass* classification task involving $\boldsymbol{L}$, each input instance has only one single label or class belonging to $\boldsymbol{L}$. In multilabel classification, however, a subset of labels or classes is attached to each instance. This kind of classification arise in different fields. In many databases, text documents, videos, music or movies are tagged with several labels. In biology, the description of the functionalities of genes frequently requires more than one label. [2] have made a detailed presentation of multilabel classification and their applications.

Multilabel classification has received increasing attention due to its applications and also because of its own appeal as an intellectual challenge. The goal is to find a way to take into account the dependence between labels. Notice that it is possible to tackle a multilabel classification task by *simultaneously* learning one binary classifier for each label. However, these binary classifications are not entirely independent. The presence or absence of a label in the set of labels assigned to an instance may be conditioned by other labels. The aim of a multilabel approach is to predict a *set of labels* as opposed to a simultaneous bunch of independent binary predictions.

Some genuine multilabel methods produce a score for each label in $\boldsymbol{L}$ (in particular, this score is a posterior probability) which induces a ranking of labels. Once we have the ranking, a set of labels can be obtained using a threshold. Typically, the threshold used in probabilistic methods is 0.5. However, thresholding can be more sophisticated. [3] report a theoretical and empirical comparative study of existing thresholding methods.

Thresholding can be considered as a method to adopt a global approach from the point of view of the set of labels; in fact, the threshold is selected taking into account the scores of all the labels. Once a threshold is fixed for a given input

instance, the prediction is the set of labels with a higher score. Threshold computations must thus consider the consequences of predicting a set of labels rather than just the individual effects of including each one. Thresholding strategies are frequently effective and mostly efficient multilabel learners.

In our paper [1] we introduced a new family of thresholding strategies based on optimizing the expected loss for a given loss function, we call these strategies *Probabilistic Thresholds* (PT). Our approach is able to select a different threshold for each instance in order to minimize the expected loss, provided we have a reliable estimation of the posterior probability for each label.

The original paper includes a detailed description of the PT strategies, as well as an experimental comparison of our contribution against two other thresholding techniques, *OneThreshold* (OT) and *Metalearner* ($Meta$), which were considered the most powerful strategies in the work of [3]. Our results support that the PT family significantly outperforms all alternative thresholding strategies.

An implementation of PT can be found under the name of MLPTO (Multilabel Probabilistic Threshold Optimizer) in the public domain library MULAN (http://mulan.sourceforge.net) .

2 Conclusions

The work published in *Pattern Recognition* [1] presented new family of thresholding strategies using posterior probabilities, *Probabilistic Thresholds (PT)*, which are based on the estimation of the expected loss of given a loss function.

Our conclusion is that thresholding is an efficient way to implement multilabel learners that somehow takes into account the interdependence of labels for each instance. In fact, thresholding employs a global viewpoint to decide on the set of labels to predict instead of the local approach adopted, for instance, by *Binary Relevance* multilabel learners.

However, although useful for certain loss functions like F_1 or the *Accuracy*, the exploitation of label interdependence seems to provide very little benefit to other such functions. This is the case of the *Hamming loss*, as was pointed out by [4]. The experiments reported here corroborate this fact.

Acknowledgments: The research reported here is supported in part under grant TIN2008-06247 from the MICINN (Ministerio de Ciencia e Innovación, Spain).

References

1. J. R. Quevedo, O. Luaces, and A. Bahamonde, "Multilabel classifiers with a probabilistic thresholding strategy," *Pattern Recognition* **45**, pp. 876–883, 2012.
2. G. Tsoumakas and I. Katakis, "Multi label classification: An overview," *International Journal of Data Warehousing and Mining* **3**(3), pp. 1–13, 2007.
3. M. Ioannou, G. Sakkas, G. Tsoumakas, and I. Vlahavas., "Obtaining bipartitions from score vectors for multi-label classification," in *Proceedings of the 22nd International Conference on Tools with Artificial Intelligence*, IEEE, 2010.
4. K. J. Dembczyński, W. Cheng, and E. Hüllermeier, "Bayes optimal multilabel classification via probabilistic classifier chains," *Procs. of the 27th ICML*, 2010.

Hesitant Fuzzy Linguistic Term Sets for Decision Making

Rosa M. Rodríguez

University of Jaén, Computer Science Department
23071, Jaén, Spain

1 Summary

Decision problems are usually defined in contexts where the information is vague and imprecise. Different approaches have been proposed to deal with such an uncertainty. One of them is the fuzzy linguistic approach that has provided successful results on decision making problems defined under uncertainty.

In decision situations, it might happen that an expert hesitates among several values to provide their assessments over an alternative. These hesitant situations may appear in both quantitative and qualitative contexts. In quantitative settings, Torra introduced the concept of Hesitant Fuzzy Sets (HFS), to manage those hesitant situations where several values are possible for the definition of a membership function of a fuzzy set. Similarly to these situations, in qualitative settings, it may occur that experts hesitate among several values to assess a linguistic variable. Therefore, to model multiple linguistic terms keeping the basis of the fuzzy linguistic approach and extending the idea of HFS to linguistic contexts, we propose the concept of Hesitant Fuzzy Linguistic Term Sets (HFLTS).

Definition 1 *Let S be a linguistic term set, $S = \{s_0, \ldots, s_g\}$, an HFLTS, H_S, is an ordered finite subset of consecutive linguistic terms of S.*

$H_S = \{s_i, s_{i+1}, \ldots, s_j\}$, *such that,* $s_k \in S, k \in \{i, \ldots, j\}$

Different basic operations are defined to compute with HFLTS, such as, *upper bound* and *lower bound*, that obtain the minimum and maximum of a HFLTS; *complement*, *intersection* and *union* of two HFLTS. In addition, the properties of commutativity, associative and distributive have been studied for the intersection and union operations.

In the computational processes of the decision making problems are necessary more complex operations. Thus, to facilitate the computations with HFLTS, we define the concept of envelope of a HFLTS.

Definition 2 *The envelope of a HFLTS, $env(H_S)$, is a linguistic interval whose limits are obtained by means of upper bound (*max*) and lower bound (*min*), hence:*

$$env(H_S) = [H_{S^-}, H_{S^+}], \quad H_{S^-} <= H_{S^+}$$

In order to establish an order between two HFLTS, it has been defined the comparison operation of HFLTS which is based on the concept of envelope of a HFLTS. To do so, the acceptability function proposed by Sengupta has been extended to linguistic intervals.

The concept of HFLTS allows modeling multiple linguistic terms, nevertheless, in decision making problems when an expert hesitates among different linguistic terms, he/she does not use multiple linguistic terms, but rather linguistic expressions similar to the expressions used by human beings in decision problems. Therefore, to improve the flexibility of the elicitation of the linguistic information, it is proposed the use of context-free grammars, because they formalize the generation of linguistic expressions. We have defined a basic context-free grammar G_H, that generates simple but rich linguistic expressions that can be easily represented by means of HFLTS.

Definition 3 *Let G_H be a context-free grammar and $S = \{s_0, \ldots, s_g\}$ a linguistic term set. The elements of $G_H = (V_N, V_T, I, P)$ are defined as follows:*

$V_N = \{\langle primary\ term\rangle\ ,\ \langle composite\ term\rangle\ ,\ \langle unary\ relation\rangle\ ,$
$\langle binary\ relation\rangle\ ,\langle conjunction\rangle\},$

$V_T = \{greater\ than,\ lower\ than,\ between,\ and,\ s_0, \ldots, s_g\},$

$I \in V_N.$

$P = \{I ::= \langle primary\ term\rangle | \langle composite\ term\rangle$

$\langle composite\ term\rangle ::= \langle unary\ relation\rangle\langle primary\ term\rangle |\ \langle binary\ relation\rangle \langle primary\ term\rangle\langle conjunction\rangle\langle primary\ term\rangle$

$\langle primary\ term\rangle ::= s_0|s_1|\ldots|s_g$

$\langle unary\ relation\rangle ::= greater\ than|lower\ than$

$\langle binary\ relation\rangle ::= between$

$\langle conjunction\rangle ::= and\}$

To carry out computational processes with the comparative linguistic expressions generated by the context-free grammar G_H, a transformation function E_{G_H} that converts the comparative linguistic expressions into HFLTS is defined.

Definition 4 *Let E_{G_H} be a function that transforms comparative linguistic expressions, ll, obtained by G_H, into HFLTS, H_S, where S is the linguistic term set used by G_H, and S_{ll} is the set of comparative linguistic expressions generated by G_H.*

$$E_{G_H} : S_{ll} \longrightarrow H_S$$

In addition, we propose a multicriteria linguistic decision making model in which experts can express their assessments by means of comparative linguistic expressions close to the human beings cognitive model or by single linguistic terms. This decision model manages such linguistic expressions by its representation using HFLTS. To combine these linguistic expressions, we have defined two symbolic aggregation operators, *min_upper* and *max_lower*, that provide a linguistic interval for each alternative. Afterwards, an exploitation process based on the application of the non-dominance choice degree to obtain the solution set of alternatives is used.

Finally, a decision making problem is solved to show the usefulness and effectiveness of the proposed model.

Data Mining and User Centred Design to support the Recommendations Elicitation Process in Social Online Learning Environments

Olga C. Santos, Jesus G. Boticario

aDeNu Research Group, Computer Science School, UNED,
Calle Juan del Rosal, 16. Madrid, Spain
{ocsantos, jgb@dia.uned.es} - http://adenu.ia.uned.es/

Abstract. This paper summarizes the contents of our publication in *Experts Systems: The Journal of Knowledge Engineering* where the application of the TORMES methodology to elicit recommendations is reported. In particular, it reports our approach on eliciting recommendations for social online learning environments which combine user centered design and data mining techniques.

Keywords: educational recommender systems, user centered design, ISO 9241-210, data mining techniques, TORMES methodology.

1 Introduction

Social online learning environments provide new recommendation opportunities to foster active participation of learners. However, current educational recommender systems (ERS) do not usually take advantage of these opportunities. To progress on this issue, we have proposed a knowledge engineering approach (i.e., TORMES) based on human computer interaction (i.e., user centred design –UCD– as defined by the standard ISO 9241-210) and artificial intelligence techniques (i.e., data mining –DM) that involve educators in the process of eliciting educational oriented recommendations. Unlike most ERS to date, this approach focuses on identifying relevant actions to be recommended on e-learning services from a user centric perspective, thus widening the range of recommendation types.

In [1], we describe where DM techniques (i.e., association rules, decision trees and clustering) have been applied to complement the UCD methods that have been followed (i.e., questionnaires, interviews, scenarios, focus groups and card sorting) to produce 32 social oriented recommendations that encourage active participation and inclusive involvement of learners through different type of actions (e.g., contribute, enable, enter, fill in, give feedback, join, work). These recommendations focus on sharing experiences among peers in terms of the social services provided by the learning environment (e.g., forum, chat). 49 participants (29 educators and 20 learners) were involved in the experience. A pervasive use of standards has provided semantic descriptions of users' features, resources characterization and device capabili-

ties. In particular, standards and specifications from IMS, ISO and W3C standardisation bodies have been applied to ensure interoperability among the components that make up the service oriented architecture that provides the technical infrastructure for the recommendation delivery process.

2 Conclusions

The main advantage of the TORMES iterative approach is to explicitly take into account the educational requirements when identifying the recommendation needs in social online learning scenarios as well as to consider their evolution over time. By involving the educators in the elicitation process, recommendations focus on adequate items or actions for the learners' learning, and not solely on what learners prefer, and aim to provide an inclusive educationally oriented adaptive navigation support.

Since TORMES has been designed with generality in mind, the 32 recommendations obtained make up a collection of best practices that can be reused. These recommendations go beyond the existing support provided by ERS. For instance, besides recommending learners to read materials that can complement the course contents, they foster the active participation of the learners in forum and chat and explicitly consider the accessibility requirements, suggesting the appropriate contents and the usage of Web 2.0 services from those available in the learning environment to a particular learner in a specific learning situation and can strengthen social awareness as they foster the interaction with other learners facing similar situations and in particular with like-minded peers. Moreover, the access by the learners to the recommendations descriptions in terms of an explicit recommendation model allows informing the learner about the recommendation rationale and thus can facilitate their self-control mechanism so they feel engaged and motivated in their learning.

In the same way, TORMES can be used to elicit recommendations that consider affective issues [2], as literature shows that the inclusion of emotional features in the recommendations modelling provides significant improvement over generic features and is expected to increase motivation and benefit the achievement of learning outcomes.

Acknowledgements. Authors acknowledge funds from the Spanish Ministries of Science and Innovation under contract TIN2008-06862-C04-00/TSI (A2UN@) and of Economy and Competitiveness under contract TIN2011-29221-C03-01 (MAMIPEC).

References

1. Santos, O.C., Boticario, J.G. (2013) User Centred Design and Educational Data Mining support during the Recommendations Elicitation Process in Social Online Learning Environments. In: Experts Systems: The Journal of Knowledge Engineering (in press).
2. Santos, O.C., Boticario, J.G. (2012) Affective Issues in Semantic Educational Recommender Systems. 2nd RecSysTEL workshop. CEUR-WS proceedings, vol. 896, p. 71-82.

A service oriented architecture to provide data mining services for non-expert data miners

Marta Zorrilla, Diego García-Saiz

University of Cantabria, Av. De los Castros SN, 39005, Santander, Spain
{zorrillm,garciasad}@unican.es

Abstract. The present paper summarizes the published paper in Decision Support Systems journal named "A service oriented architecture to provide data mining services for non-expert data miners" [2].

Keywords. Analytics service, BI-as-a-Service, Knowledge discovery database, Data mining, Service-oriented architecture, Web Services

1 Introduction

In today's competitive market, companies need to use discovery knowledge techniques) to make better, more informed decisions. But these techniques require an incredible amount of expertise. Thus, developing tools which enable non-experts users to make the mining process is necessary [1]. This work focuses on describing a data mining service which fulfills this goal. It was developed following a Service Oriented Architecture (SOA), in order to be deployed as a Software-as-a-Service in the future. We show its use in the educational arena, more specifically in the analysis of behavior of distance students since educators are a clear example of non-experts in data mining techniques.

Our software service is divided in five layers: a data layer which encapsulates the data sources; a layer for the enterprise components which wrap the data mining algorithms, processes to visualize the data mining results and tools to access to the data in the different repositories by means of Web Services; the service layer which offers the enterprise components as a pack of services, the business process choreography layer responsible for the choreography and orquestation of the services, and finally, the presentation layer which presents a friendly user web interface.

The interaction between the different elements is carried out by means of personalized XML templates. Its main advantage is that by simply indicating where the data is, the service itself is able to perform all the process, that means, chooses the most appropriate algorithm and its parameters according to the type of problem requested by the user and the intrinsic characteristics of the data. The results are displayed textually and graphically to facilitate their interpretation and comprehension.

An interesting feature is of this service is that it can be easily integrated in any platform which collects data, for instance, learning management systems such as Moodle or Blackboard, in such a way that the service can directly read the data necessary for the KDD process from logs of the platform without user intervention.

2 Conclusions

The delivery of data mining as a service is an emergent necessity, above all for small to medium range organizations which are the most constrained by the high cost of data mining software and the availability of expert data miners to use this software. Until now, the tools deployed as BI-a-as-Service in the cloud are conceived more for license cost-saving than as a product which can be used directly by end-users without data mining knowledge. To respond to this necessity, this paper describes the architecture of a data mining service for non-expert data miners which can be delivered as SaaS. Its main characteristic is that it is based on the use of templates that answer certain previously-defined questions. These templates gather the tasks of the KDD process to be carried out on the data set which is sent by the end-user. The templates are defined by the service administrator. This service is offered as a Web Service which makes it easily accessible from any client application. Furthermore, its extension with other data mining algorithms and visualization tools developed ad-hoc or consumed from a provider in the Internet can be effortlessly incorporated since it is designed following a service-oriented architecture.

ElWM, is a web application which uses the data mining service configured for an educational context; in particular, it helps instructors involved in virtual teaching to discover their students' profile and their behavior in the course. The prototype of this web application has been successfully tested in two virtual courses taught in the University of Cantabria. In the instructors' opinions, the tool is very easy to use and the information which it returns is very useful to better understand what is happening in the course and take actions as soon as anomalous behaviors are detected.

Currently it is being updated, adding new templates supported by SNA and classification techniques.

3 References

1. Romero, C., Ventura, S.: Educational data mining: A review of the state of the art. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 40(6), 601‑618 (2010)
2. Zorrilla, E., García-Saiz, D.: A service oriented architecture to provide data mining services for non-expert data miners. Decision Support Systems. 55(1), 399-411 (2013).

17-20 sept 2013

MULTICONFERENCIA

CAEPIA'13

Madrid

Doctoral Consortium

Editores asociados:

Francisco Herrera

José C. Riquelme

Entidades Colaboradoras

ugr Universidad de Granada

GENIL GRANADA EXCELLENCE NETWORK OF INNOVATION LABORATORIES

1576

Prefacio

Después del éxito del Doctoral Consortium organizado en la CAEPIA de 2011, hemos querido continuar con una nueva propuesta dentro de la Multiconferencia CAEPIA del CEDI 2013. El objetivo es difundir en el marco donde se exponen los avances de la informática española, la investigación más actual que se está realizado en Inteligencia Artificial.

Queremos agradecer a los doctorandos y directores la importante acogida que ha tenido esta iniciativa con 29 trabajos presentados, que representan un gran abanico de todas las líneas de trabajo de los grupos de investigación en IA: optimización, minería de datos, bioinformática, agentes, incertidumbre, etc. Cabe destacar la presencia de importantes retos reales y actuales que demuestra el interés de nuestros investigadores por resolver problemas que la sociedad plantea: eficiencia energética, análisis de imágenes, análisis de redes sociales, accidentes de tráfico, etc.

En segundo lugar queremos dar las gracias a los patrocinadores, en especial a la AEPIA y a los gestores del Campus de Excelencia BioTICs de la Universidad de Granada por su aportación para otorgar los premios de esta edición.

También queremos agradecer al comité de selección el trabajo de distinguir las mejores propuestas, será una difícil tarea fundamentalmente por la calidad de los trabajos presentados.

Por último, queremos transmitir a los doctorandos unas palabras de ánimo en estos tiempos difíciles para la investigación. La mejor herramienta para salir de crisis es la formación y el doctorado es la máxima expresión de la formación que se puede alcanzar. Las tesis doctorales propuestas significan un compromiso de múltiples horas durante numerosos días de trabajo, pero al final estamos seguros de que el esfuerzo vale la pena. Más pronto que tarde la sociedad sabrá reconocer vuestro trabajo y el de vuestros directores. Sólo algunas propuestas conseguirán premio de este Doctoral Consortium, pero por supuesto, no lo entendáis como un desmerecimiento de vuestro trabajo. Sólo pequeños detalles diferenciaran trabajos de tanta calidad. Deseamos que la ocasión de asistir a esta CAEPIA aglutinadora haya sido el mejor premio y que el fructífero intercambio con otros investigadores noveles y veteranos os ayude en la terminación de la tesis y en vuestro futuro.

Francisco Herrera
José C. Riquelme

Organización

Comité Ejecutivo

Francisco Herrera	Universidad de Granada
José C. Riquelme	Universidad de Sevilla

Comité de Programa

Enrique Alba, U. Málaga	Amparo Alonso-Betanzos, U. A Coruña
Antonio Bahamonde, U. Oviedo	Emilio Corchado, U. Salamanca
José A. Gámez, U. Castilla-LM	Pedro Larrañaga, U. P. de Madrid
José A. Lozano, U. País Vasco	Luis Martínez, U. Jaén
José L. Verdegay, U. Granada	Amparo Vila, U. Granada

Sponsors

- Campus de Excelencia BioTICs, Universidad de Granada

- Asociación Española para la Inteligencia Artificial (AEPIA)

A Computational Model of Maspin Effects on Cell Migration Using Artificial Neural Network

M A Al-Mamun[1]

Principle Director: Alamgir Hosasin[1]
Other Directors: Rosemary Bass[1, 2] and Charles Fall[1]
[1]Computational Intelligence Group, Faculty of Engineering and Environment, University of Northumbria at Newcastle, UK
[2] Departments of Applied Sciences, Faculty of Health and Life Sciences, University of Northumbria at Newcastle, UK
mohammed.al-mamun@northumbria.ac.uk

1 Introduction

Tumour growth and progression is a complex biological process, normally beginning with genetic mutations in a single cell. Generally, the growth of a tumour in the body can be divided into three stages: the early or avascular phase where growth is limited by nutrient diffusion, the vascular stage where angiogenesis occurs to stimulate blood vessel production by the secretion of tumour angiogenesis factors (TAF) and finally the metastasitic phase where the tumour spreads from the site of origin to distant sites around the body [1]. When considering these events at the cellular level, these processes involve both intracellular and extracellular mechanisms to influence proliferation, cell-cell adhesion, cell migration, cell-extracellular matrix (ECM) interactions and angiogenesis. In this project we consider cell proliferation, cell migration and cell invasion to model their principle roles in tumour malignancy.

There are many processes involved in tumour growth and metastasis [2, 3], including cell migration, adhesion, invasion and the inhibition of apoptosis. Key to these processes is proteolysis, which involves multiple protease systems including the serine protease family [4]. Inhibition of these proteases is carried out by serine protease inhibitors (serpins) such as plasminogen activator inhibitor-1 (PAI-1), which targets the serine protease urokinase-type plasminogen activator (uPA) [4]. Serpins act via their reactive centre loop (RCL) to bind to their protease target [4]. They then translocate the protease through their β-sheet, inhibiting it. However, some serpins have been found to be non-inhibitory. These include Ovalbumin and Maspin (Mammary Serine Protease Inhibitor). Maspin was first identified in 1994 as a tumour suppressor [1]. It has since been found to play a role in cellular processes including migration, apoptosis and angiogenesis [2].

The cellular mechanisms that maspin uses to influence cellular behaviour are not yet clearly defined, but have been reviewed recently [4]. There are reports that maspin works in both inside and outside of the cell [4]. It is possible that extracellular maspin directly affects cell migration, adhesion and angiogenesis, while indirectly affecting tumour cell proliferation and apoptosis. Maspin has been reported to bind to integrin cell adhesion receptors [3] or to the extracellular matrix [4]. Recently, it has been suggested that the urokinase-type plasminogen activator (uPA)/uPA receptor

(uPAR) complex is involved in the localization and adhesive functions of maspin through a maspin-uPA-uPAR-beta1 integrin mega-complex which regulates mammary epithelial cell adhesion [5].To date there have been no reports of mathematical or computational models to support these data. This diversity motivates us to build a computational model of the effects of maspin to show its potential engagement with multiple cellular phenomena using cellular automata modeling techniques.

2 Hypothesis and Research Questions

The main goal of this PhD is to build a computational model of tumour growth with maspin effects and establish this model with the support of *in vitro* and *in vivo* data, as it is important to build the model with proper validation for getting accepted by oncologist. To achieve this goal, we divided the goal into three sub-goals,

I. To explore existing literatures, tools and techniques in order to build an understanding on current state of art of technology. Building the tissue level model to understand the tumour growth effect [7] in the tissue surface
II. Building the tumour growth model in cellular level considering with cell-cell adhesion, cell-migration, cell-ECM interaction, growth and inhibitory factors and specific gene called maspin.
III. Investigating more insight of maspin phenomena in respect of cytoskeletal changes of the cell environment, interaction with β-integrin and fibronectin etc.

3 Objectives

In this research, we have taken the migration raw data from a previous study [6], where it was reported that the G-helix is essential and sufficient for the influence of maspin on cell migration. This model will focus on the effects of exogenous maspin over the surrounding tumour cells.

The main objective of this PhD is to model proliferation and migration to investigate the unknown effects of maspin; this will allow us to verify the in vivo or in vitro experiments.

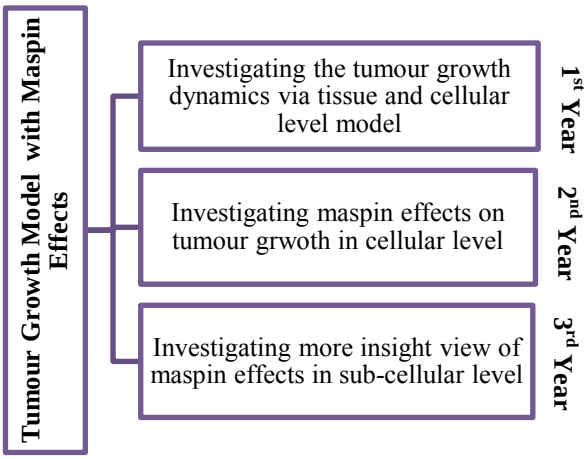


Fig. 1. Objectives of the PhD thesis

4 Methodology

A tumour tissue was developed starting with one cell at the centre of a two dimensional grid. In this cellular automaton model, each grid element was either occupied by an abnormal cell or was empty. The grid elements had local concentrations of oxygen. Each tumour cell was treated as an individual entity and identified through its position coordinates $x(x_x, y_y)$. The cell was strongly influenced by the tumour microenvironment and had its own decision mechanism that determined its behaviour during the growth activity. The decision mechanism was implemented using an artificial neural network. The neural network was told the number of neighbours, the concentration of oxygen and presence/absence of maspin. The response of the network at its output nodes determined the decision or behaviour of the cell. The response was one of three cellular behaviours: proliferation, quiescence or apoptosis.

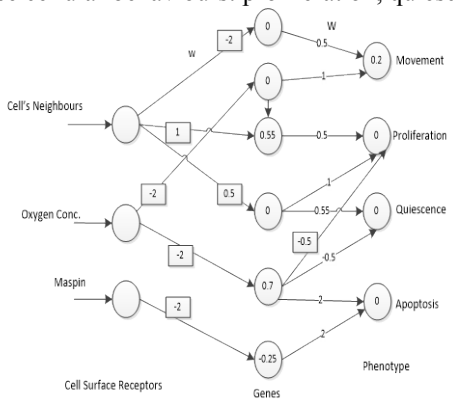


Fig2. The basic structure of the neural network for tumour growth in the presence/absence of maspin

Phase1 (First year): In the early phase, the problem background was analyzed on the basis of existing literatures. The work done in first phase

1. Developed tumour growth model which shows the tumour growth dynamics in tissue level [10]. The first phase also included an intelligent decision support system which helps to analyze the tumour microenvironment constraints like cell-cell adhesion, cell movement, extra-cellular matrix (ECM) and optimal solutions of drug scheduling problem. Hypoxia and drug resistance are also incorporated in the model to achieve the predictive results for every patient as both of them considered as the main reason for chemotherapy and radiotherapy treatment failure.
2. After that we investigated the tumour dynamics in cellular and subcellular level [8]. Recently, [9] established an *in silico* model to calculate tumour mass with consideration of oxygen, glucose, ECM, cell-cell adhesion and cell movement as key micro environmental parameters. The model includes more micro-environment parameters such as protein expression, growth promoters/inhibitors and a bioreductive drug (TPZ) transport model for hypoxic tumour cells [8] on top of the Anderson model.

Publication (Phase1)

1. Kazmi, N., Hossain, M.A., Phillips, R., **Al-Mamun, M.A**. and Bass, R., Avascular tumour growth dynamics and the constraints of protein binding for drug transportation, J Theor Biol. 2012 Nov 21;313:142-52
2. **Al-Mamun**, **M.A.,** Kazmi, N., Hossain, Alamgir, Vickers, Paul and Jiang, Yang (2012), An intelligent decision support system for personalized cancer treatment. In: CIS2012: 11th IEEE Conference on Cybernetic Intelligent Systems, 23-24 August 2012, Limerick, Ireland

Posters

1. An intelligent Optimal Chemotherapy Model for Personalized Cancer Treatment, in Northumbria Research Conference, 2012 (Got best poster awards)
2. An intelligent Optimal Chemotherapy Model for Personalized Cancer Treatment, in London Mathematical Society (LMS) Meeting and workshops on Mathematics of Human Biology - 6 - 8 June, 2012.

Phase2 (Second year):

During the second phase we have built a computational model which mimics the effects of maspin in the tumour growth dynamics. We used our previous tumour growth model [9], but modified the number of neural network inputs. This time we used oxygen, maspin and neighbors as an input. The concentrations of glucose and hydrogen ions did not vary in the *in vitro* experiments, so we have not considered hypoglycemia or acidity in building this model, in the model considering the presence/absence of maspin, oxygen concentration was taken as constant as oxygen and nutrients were essentially constant throughout the *in vitro* experiment [10]. The varying distance of cells from a blood vessel can cause heterogeneity in the tumour microenvironment and the tumour mass (find details in [8]).

Publication (Phase2)

1. M S Alam, M A Hossain, S Algoul, M A A Majumader, **M A Al-Mamun**, R Phillip, (2013) Multi-Objective Multi-drug Scheduling Schemes for Cell Cy-

cle Specific Cancer Treatment in International Journal of Computers & Chemical Engineering (Accepted).
2. **M.A. Al-Mamun,** Hossain, M.A., Alam, M.S., Bass, R., (2013). A Cellular automaton model of the effects of maspin on cell migration. 7th International Conference on Practical Applications of Computational Biology & Bioinformatics, Advances in Intelligent Systems and Computing Volume, 222, pp 53-60. (Got Scientific Excellence Awards)
3. Nadia Alam, Munira Sultana, M. S. Alam, **M. A. Al-Mamun**, M. A. Hossain, (2013). Periodic Chemotherapy Dose Schedule Optimization Using Genetic Algorithm. Distributed Computing and Artificial Intelligence, Advances in Intelligent Systems and Computing Volume 217, 2013, pp 503-511.
4. **M.A. Al-Mamun,** Hossain, M Alamgir, C Fall, R. Bass. (2013) The Effects of Non-Inhibitory Serpin Maspin on Cell Migration Using an Artificial Neural Network in The International Workshop on Computational Systems Biology (WCSB'2013) (Accepted)
5. **M.A. Al-Mamun**., L J Brown, M.A. Hossain,., C. Fall, L Wagstaff., R. Bass, A Hybrid Computational Model for MASPIN Effects on Cancer Cell Dynamics, Journal of Theoretical Biology (In preparation).

Posters

1. A Cellular Automaton Model of the Effects of Maspin on Cell Migration, Northumbria Research Conference, 2013

Phase3 (Third year): Third phase will be considering the mathematical model of core insight of maspin including cytoskeletal changes because of maspin with the help of *in vitro* and *in vitro* data. The upcoming work will be on more biological impact representation in terms of computational model.

5 Relevance of the thesis

First phase of this thesis contributed to the overall modeling of a personalized cancer treatment model [9]. In the second phase, results show that maspin reduces migration by 10-40%, confirmed by published *in vitro* data. The model also shows reduction in cell proliferation by 20-30% in the presence of maspin. From our knowledge, this is the first attempt to model maspins effect in a computational model to verify *in vitro* data [6].

Fig 3. Spatial distribution of the cells at t=20, 60,120. Proliferating cells are shown as light blue, quiescent cells as blue, dead cells as red and empty grid points are white

Although this model does not take into account the whole range of effects of maspin, it can serve as a useful tool as it captures the tumour dynamics at the cellular level. This also presents the knowledge of the subcellular machinery focused on the relationship between environmental inputs and cellular phenotype. The effects of maspin were mimicked in this model using a decision neural network whose parameters served as the genetic material of the cell. The phenotype (outputs) were a function of the genotype (inputs), but also crucially depended on the microenvironment of the cell. Practically, it is still far from being a precise model of the subcellular dynamics, but this attempt brings the complete model a step closer. This model could be used to allow high throughput screening of variable conditions and physiological

considerations as an alternative to biological experiments requiring expense, time and technical expertise. There are many applications of the model presented. Biologists could combine *in vitro* data collected from different experiments using a variety of models in a computer-generated model. This would allow extrapolation of experimentally collected data to model longer-term consequences of cell behaviours than is possible using real cell based models. This will provide a greater consensus on the impact of maspin on cancer.

6 Conclusion

It is worth mentioning that the outcome of the proposed model resemble the in vitro experiments [5]. Our future work will be to establish the model to show the whole complex mechanism of the impact of maspin on cell-cell adhesion, cell-ECM interaction and apoptosis. Also, we will modify the neural network to demonstrate the efficiency with different hidden layers. This will be informed by definition of maspin specific constants which are currently unavailable.

References

1. Zou, Z., Anisowicz, A., Hendrix, M. J., et al. Maspin, a serpin with tumor-suppressing activity in human mammary epithelialcells. Science, 263, 526–529 (1994)
2. Bass, R., Fernandez, A. M. & Ellis, V., 2002. Maspin inhibits cell migration in the absence of protease inhibitory activity. J. Biol. Chem., 277, 46845-8.
3. Bass, R., Wagstaff, L., Ravenhill, L. & Ellis, V., 2009. Binding of extracellular maspin to beta1 integrins inhibits vascular smooth muscle cell migration. J. Biol. Chem., 284, 27712-20.
4. Bodenstine TM, Seftor RE, Khalkhali-Ellis Z, Seftor EA, Pemberton PA, Hendrix MJ. Maspin: molecular mechanisms and therapeutic implications.Cancer Metastasis Rev Dec;31(3-4):529-51 (2012)
5. Endsley, M P., Yanqiu, Hu., Yong Deng et al., 2011. Maspin, the molecular bridge between the plasminogen activator system and *β1* Integrin That Facilitates Cell Adhesion. J. Biol. Chem., vol. 286, NO. 28, pp. 24599–24607.
6. L. Ravenhill , L. Wagstaff , D. R. Edwards, V. Ellis, and R. Bass, G-helix of Maspin Mediates Effects on Cell Migration and Adhesion. The Journal of Biological Chemistry vol. 285, no. 47, pp. 36285–36292, November 19 (2010)
7. Sherratt, J.A. and Chaplain, M.A.J., A new mathematical model for avascular tumour growth, Journal of Mathematical Biology, 43, pp. 291-312 (2001)
8. Kazmi N, Hossain MA, Phillips RM, Al-Mamun MA, Bass R. Avascular tumour growth dynamics and the constraints of protein binding for drug transportation; J Theor Biol. Nov 21;313:142-52 (2012)
9. Al-Mamun M.A., Kazmi, N., Hossain, M. A, Vickers, P and Jiang, Y (2012), An intelligent decision support system for personalized cancer treatment. In: CIS2012: 11th IEEE Conference on Cybernetic Intelligent Systems, 23-24 August 2012, Limerick, Ireland
10. Al-Mamun, M.A., Hossain, M.A., Alam, M.S., Bass, R., 2013. A Cellular automaton model of the effects of maspin on cell migration. Advances in Intelligent Systems and Computing Volume, 222, pp 53-60., 526–529.

Vesta: Análisis y detección de contenido agraviantes en imágenes

Doctorando: Borja Alonso Isla
Director: Pablo García Bringas

DeustoTech Computing –S3lab (Universidad de Deusto)
pablo.garcia.bringas@deusto.es, borjaalonso@deusto.es
http://s3lab.deusto.es/

Keywords: Selección y extracción de características, minería de datos, seguridad y privacidad

1 Introducción

A día de hoy, el crecimiento diario de la cantidad de imágenes que se suben a Internet es tan grande, que ha crecido la necesidad de un detector fiable de contenido. Ya que aparte de los métodos de búsqueda existentes en los buscadores principales, también es importante reconocer contenido no deseado u ofensivo, para así poder filtrarlo. A día de hoy, esta tarea se realiza a mano, causando graves trastornos a los que se encargan de ello[1]. Por ello, es una tarea importante, ser capaz de detectar y censurar de forma automática toda esta clase de contenidos agraviantes para aquellas personas que no deseen verlos.

Dado que la posesión de material de esta naturaleza puede tener consecuencias legales[2], esta investigación se centrará solamente en filtrar contenido adulto. Dicho esto, el primer paso de esta investigación consistiría en la recopilación aleatoria de imágenes en las que existan superficies de piel. A continuación mostramos una lista de la cantidad de factores que hacen variar la coloración de la piel.

- **Sangre:** Frío, calor, rubor, moratón, enfermedades.
- **Melanina:** Asociado a la genética (etnia) y a la exposición reciente al sol.
- **Condiciones de luz:** Luz natural, artificial, a contra luz, flash.
- **Características de la cámara:** Resolución, zoom óptico, obturador, diafragma.
- **Otros factores:** Sudor, gafas, peinado, colores de fondo

Con el afán de eliminar la mayor cantidad de ruido posible, eliminaremos todas aquellas en las que el flash no cree un correcto contraste (p.ej. Que las zonas claras sean completamente blancas), el color de la piel no se corresponda con uno considerado

[1] http://www.buzzfeed.com/reyhan/tech-confessional-the-googler-who-looks-at-the-wo

[2] Artículo 189 - 1.b: http://noticias.juridicas.com/base_datos/Penal/lo10-1995.l2t8.html

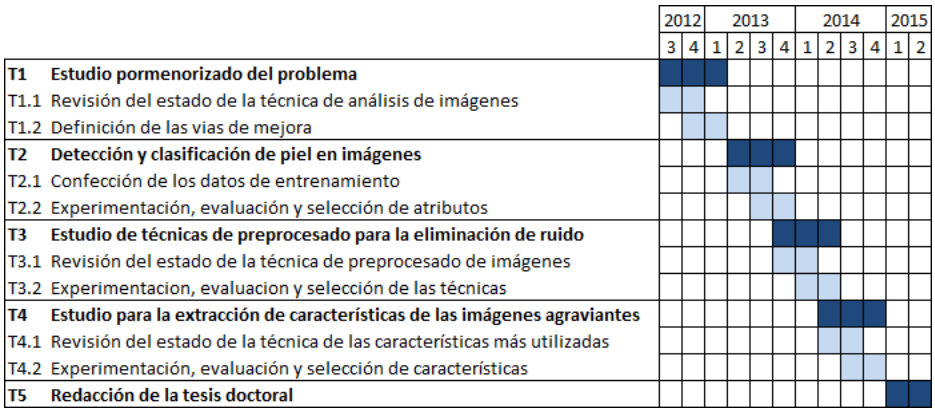
como normal (p.ej.: cuerpo pintado de colores, rule34[3], etc.) o la etnia no se corresponda con la indo-europea (fácilmente extrapolable en un futuro para crear clasificadores de piel por etnia).

Muchas de las investigaciones relacionadas con la detección de personas en imágenes comienzan reconociendo la piel en base a los valores de los componentes de modelos de color [1] para después extraer características [2] que sirvan para la detección de caras en video llamadas, pornografía en internet, etc. [3] [4] [5].

2 Hipótesis

- **H_1** – Crear un clasificador de contenido pornográfico con mayor eficacia y eficiencia que los existentes actualmente.
- **H_a** – Se puede crear un clasificador de contenido pornográfico con mayor eficacia que los existentes actualmente
- **H_0** – No se puede obtener un clasificador de contenido pornográfico con mayor eficacia y eficiencia que los existentes actualmente.

3 Objetivos



La primera tarea que marcará un punto de inflexión en esta investigación será obtener el clasificador de piel más apto en términos de eficacia. Como objetivo secundario se buscaría una mejora en eficiencia, si fuese posible, en todo el proceso de extracción de atributos de los campos de color. Como por ejemplo usando la menor cantidad de atributos posible sin que la eficacia sea vea notablemente afectada.

El segundo hito relevante sería obtener un conjunto de características, de la imagen generada por el detector de piel, en el que la tasa de acierto supere a las actuales técnicas existentes en el estado del arte. Los resultados de las técnicas de preprocesado aplicadas en la tarea 3 no se harán visibles hasta que el modelo de detección de contenido agraviante no esté disponible, así que el modelo de desarrollo sea iterativo entre las tareas 3 y 4, ya que sufren una gran dependencia.

[3] http://rule34.paheal.net/

4 Metodología

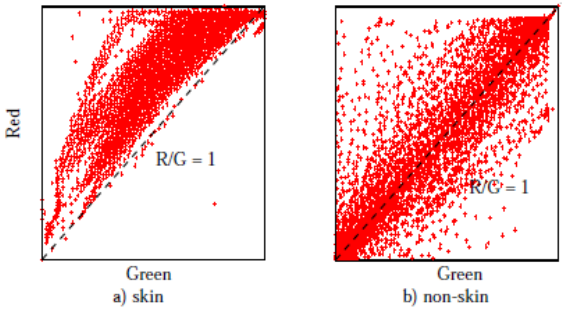
4.1 Clasificador de piel

Una vez hemos recopilado un conjunto relevante de imágenes, hemos de reajustar su tamaño para que a la hora de entrenar los modelos no exista una imagen que aporte más instancias de píxeles que el resto, es decir, para balancear las muestras. Una vez tenemos las imágenes preparadas, es hora de etiquetarlas. Manualmente vamos marcando los pixeles de piel en blanco y los pixeles de no piel en negro. Esta nueva imagen tendrá el mismo tamaño que la original y la denominaremos máscara.

Fig. 1. Máscara de piel y no piel

Una vez tengamos todas las máscaras creadas, procederemos a extraer el valor que poseen los pixeles, que por defecto obtendremos los valores RGB, pero a partir de transformadas podremos obtener HSV, YCbCr, relaciones de R/G, R/B o cualquier otro espacio de color o combinación de atributos que nos pueda interesar para usarlo en el clasificador.

Fig. 2. Ejemplo de transformación de atributos



Se harán los experimentos necesarios para determinar que atributos dan los mejores resultados para el conjunto de datos actual. Desgraciadamente, los resultados de estos

experimentos son muy dependientes del conjunto de datos aún teniendo cuidado a la hora de seleccionarlos [3].

Una vez hayamos determinado el conjunto óptimo de atributos, podremos empezar a centrarnos en la eficiencia. Se podría obtener una relación entre coste de extracción de un atributo y su capacidad para definir la clase piel, y a partir de ahí, ir eliminando los que peor ratio tengan para eliminar atributos importantes que requieren de mucha computación para extraerse, o de atributos que no aportan tanto como otros a la hora de definir la clase y que su omisión no afecta en exceso a la tasa de acierto, pero que en cambio sí que sirve para aligerar el proceso de detección de piel.

4.2 Clasificador de imágenes agraviantes

Una vez se haya realizado el clasificador de piel, podremos empezar el siguiente paso, que sería la extracción de características de imágenes pornográficas. Para iniciar esta tarea, es esencial recopilar imágenes tanto pornográficas (de la naturaleza de las que queremos filtrar) e imágenes no pornográficas. En el estado de la técnica las características extraídas van desde un simple porcentaje de piel en toda la imagen [6], hasta 19 características diferentes [3].

Una vez hayamos finalizado con la recopilación de imágenes, procederemos a usar el clasificador de piel en ellas, ya que hay muchas características relacionadas con la cantidad de piel que hay en la imagen (p.ej. porcentaje de piel en la imagen, cantidad de regiones de piel, porcentaje de la región más grande respecto a la imagen, etc.).

Otras características a tener en cuenta serían la existencia de ciertas formas (p.ej. cara, ojos, boca, senos, genitales, etc.), la aplicación de filtros para eliminar ruidos, y cualquier otra idea que pueda surgir del estado del arte que ayude a definir la clase. La detección de formas se debería aplicar en la imagen original, y a partir de ahí quizás sacar conclusiones (p.ej. si es un primer plano, si esa zona marcada como cara posee ojos, boca y nariz, etc.) que puedan servir como características y añadirlas al conjunto de las extraídas en la imagen binarizada de piel y no piel.

Fig. 3. Ejemplo de extracción de características

También existe la posibilidad de extraer el número de regiones de piel existentes en la imagen, eliminar regiones cuyo tamaño no sea significativo para eliminar ruido, aparte de otras características como densidad o entropía de las regiones de piel.

A partir de aquí, se seguirían los mismos pasos que en la creación del clasificador de piel y se obtendría un subconjunto óptimo en base a los intereses de eficacia y/o eficiencia para crear un clasificador de imágenes pornográficas en consecuencia.

5 Relevancia

A día de hoy se da un rápido crecimiento en la cantidad de imágenes que se suben a internet públicamente, creciendo así la necesidad de un filtrador de contenido fiable. Ya que aparte de los métodos de búsqueda de imágenes, también es importante reconocer el contenido no deseado u ofensivo, para así poder bloquearlo. El mayor grupo de imágenes de Internet que pueda resultar ofensivo para la gente, son, y por mucho, las de contenido adulto, cuya temática forma parte de al menos el 10% de los sitios web[4]. Sobre todo los menores de edad han de ser salvaguardados de ver esta clase de contenido ya que incluso los adultos pueden no desear velo, ni tan siquiera de forma accidental por culpa de los motores de búsqueda.

Por otro lado, cabe destacar que el uso de estas tecnologías comienza cada vez a una edad más temprana. Por lo que es habitual ver a jóvenes en el patio de su colegio compartiendo fotos y videos para sus móviles. Estas personas son altamente problemáticas ya que aún no tienen totalmente definidos sus valores éticos y sociales por lo que pueden caer en ataques a sus compañeros que terminen en una actitud de acoso o de mofa.

Así que dada la gran repercusión que tienen los sucesos relacionados con contenido adulto y en especial la pornografía infantil, es importante investigar por esta línea para proteger la sensibilidad de los más vulnerables. Tanto en centros de enseñanza, como sector público e incluso entornos de trabajo privados.

Por todo lo anterior, es importante investigar en nuevos métodos y técnicas que ayuden a lograr la máxima efectividad posible en temas relacionados con la clasificación de contenido adulto.

6 Referencias

[1] R. Khan, A. Hanbury, J. Stöttinger, F. A. Khan, A. U. Khattak y A. Ali, «Multiple color space channel fusion for skin detection,» *Springer,* 28 Abril 2013.

[2] F. Jiao, W. Gao, L. Duan y G. Cui, «Detecting adult image using multiple features,» de *Conferences on Info-tech and Info-net*, 2001.

[4] Según http://internet-filter-review.toptenreviews.com/internet-pornography-statistics.html en Octubre del 2011

[3] C. X. Ries y R. Lienhart, «A survey on visual adult image recognition,» *Springer*, 31 Mayo 2012.

[4] V. Vezhnevets, V. Sazonov y A. Andreeva, «A Survey on Pixel-Based Skin Color Detection Techniques,» 2007.

[5] D. Chai y K. N. Ngan, «Face Segmentation Using Skin-Color Map in Videophone Applications,» *IEEE Transactions on circuits and systems for video technology*, vol. 9, nº 4, 1999.

[6] J. I. Agbinya, B. Lok, Y. S. Wong y S. Da Silva, «Automatic Online Porn Detection and Tracking,» 2007.

[7] J. Lee, Y. Kuo, P. Chung y E. Chen, «Naked image detection based on adaptative and extensible skin color model.,» de *Pattern Recogn*, vol. 40, 2007, pp. 2261-2270.

[8] Q. Zheng, W. Zeng, G. Wen y W. Wang, «Shape-based adult image detection».

Mejora de la Asistencia en los Accidentes de Tráfico combinando Redes Vehiculares e Inteligencia Artificial

Doctorando: Javier Barrachina
Directores: Piedad Garrido y Francisco J. Martinez

Departamento de Informática e Ingeniería de Sistemas,
Universidad de Zaragoza
{barrachina, piedad, f.martinez}@unizar.es
http://init.unizar.es

Keywords: Estrategias Evolutivas, Redes Vehiculares, Sistemas Inteligentes de Transporte

1 Introducción

A pesar de que las medidas de seguridad en los sistemas de transporte cada vez son mayores, el aumento progresivo del número de vehículos que circulan por las ciudades y carreteras en todo el mundo aumenta, sin duda, la probabilidad de que ocurra un accidente. En este tipo de situaciones, el tiempo de respuesta de los servicios de emergencia es crucial, ya que está demostrado que cuanto menor sea el tiempo transcurrido entre el accidente y la atención hospitalaria de los heridos, mayores son sus probabilidades de supervivencia.

Las redes vehiculares permiten la comunicación entre los vehículos, así como la comunicación entre los vehículos y la infraestructura [4], lo que da lugar a una plétora de nuevas aplicaciones y servicios en el entorno vehicular. Centrándonos en las aplicaciones relacionadas con la seguridad vial, mediante este tipo de comunicaciones, los vehículos podrían informar en caso de accidente al resto de vehículos (evitando así colisiones en cadena) y a los servicios de emergencia (dando información precisa y rápida, lo que sin duda facilitaría las tareas de rescate).

Uno de los aspectos importantes a determinar sería saber qué información se debe enviar, quién será capaz de recibirla, y cómo actuar una vez recibida. Actualmente los vehículos disponen de una serie de sensores que les permiten obtener información sobre ellos mismos (velocidad, posición, estado de los sistemas de seguridad, número de ocupantes del vehículo, etc.), y sobre su entorno (información meteorológica, estado de la calzada, luminosidad, etc.). En caso de accidente, toda esa información podría ser estructurada y enviada a los servicios de emergencia para que pudieran adecuar el rescate a las características específicas y la gravedad del accidente, actuando en consecuencia.

Por otro lado, para que la información enviada por los vehículos accidentados pueda llegar correctamente a los servicios de emergencias, es necesario

disponer de una infraestructura capaz de dar cobertura a todos los vehículos que circulan por una determinada área. Puesto que la instalación y el mantenimiento de dicha infraestructura conlleva un elevado coste, sería conveniente proponer, implementar y evaluar técnicas consistentes en dar cobertura a todos los vehículos, reduciendo el coste total de la infraestructura.

Finalmente, una vez que la información ha sido recibida por las autoridades, es necesario elaborar un plan de actuación eficaz, que permita el rápido rescate de los heridos. Hay que tener en cuenta que, cuando ocurre un accidente de tráfico, el tiempo de personación de los servicios de emergencia en el lugar del accidente puede suponer la diferencia entre que los heridos sobrevivan o fallezcan. Además, es importante conocer si la calle o carretera por la que circulaban los vehículos accidentados ha dejado de ser transitable para el resto de vehículos, y en ese caso, activar los mecanismos necesarios que permitan evitar los atascos asociados. En nuestro trabajo, pretendemos gestionar adecuadamente estas situaciones adversas, distribuyendo el tráfico de manera inteligente para reducir el tiempo de llegada de los servicios de emergencia al lugar del accidente, evitando además posibles atascos.

2 Trabajos Relacionados

Existen numerosas propuestas relacionadas con la aplicación de las redes vehiculares en los sistemas inteligentes de transporte, con el fin de aumentar la seguridad vial. Fogue et al. [3] diseñaron eNOTIFY, una propuesta basada en la combinación de comunicaciones vehículo-a-vehículo y vehículo-a-insfraestructura, diseñado para mejorar la capacidad de respuesta de los servicios de emergencia, optimizando los recursos médicos y de rescate necesarios. Eigner y Lutz [1], propusieron utilizar modelos de contexto ontológico para entornos de seguridad en redes vehiculares adhoc. Su propuesta mostraba la ventaja de aplicar ontologías para la intercomunicación de las distintas partes que se comunican mediante la red, pero no presentaron ninguna ontología específica.

Por otro lado, el uso de Algoritmos Evolutivos para la distribución del tráfico de manera inteligente, también se ha mostrado como una solución potente en ese tipo de entornos [6]. Entre las diferentes propuestas, cabe destacar el trabajo presentado por Wang y Shi [7], en el que se utilizan algoritmos genéticos para solventar problemas de interrupción del tráfico, como los atascos y los accidentes. El inconveniente de esta propuesta es que los autores simplemente la han probado en escenarios teóricos, que no se ajustan a la realidad. Yoshikawa y Terai [8] proponen combinar algoritmos genéticos con el algoritmo Dijkstra para el cálculo de rutas de manera eficiente. La solución presentada es similar al Problema del Viajante [5], muy utilizado como problema base en la enseñanza de los algoritmos evolutivos. En este trabajo, al igual que en el anterior, los autores validan su propuesta en un escenario teórico.

A diferencia de otros trabajos propuestos, nuestro trabajo propone la integración de las Redes Vehiculares (combinando las comunicaciones vehículo-a-vehículo y vehículo-a-insfraestructura) y los Algoritmos Evolutivos para la gestión eficiente

del tráfico y de los servicios de emergencia en caso de accidente. Además, las pruebas realizadas se llevarán a cabo utilizando escenarios reales.

3 Objetivos de la Propuesta

Los objetivos principales son los siguientes:

- Proponer una estructura para la información que queremos enviar cuando ocurra un accidente. Nos centraremos en dos factores: (i) qué información enviar y (ii) cómo estructurarla.
- Proveer de la infraestructura necesaria para dar cobertura de comunicaciones a todos los vehículos con el menor coste posible. Queremos encontrar una solución que permita ahorrar costes, pero sin comprometer las posibilidades de comunicación entre los vehículos y la infraestructura.
- Utilizar algoritmos inteligentes para distribuir el tráfico con el fin de evitar atascos y reducir el tiempo de llegada de los servicios de emergencia al lugar del accidente. Resultados preliminares nos demuestran que el tiempo necesario para que una ambulancia llegue al escenario del accidente se puede reducir hasta en un 50% en un entorno urbano, sin que aumente de forma significativa el tiempo de tránsito del resto de vehículos.

Buscamos que nuestra propuesta sea capaz de reducir el tiempo necesario de rescate de los heridos, lo que salvará vidas, y evitará los atascos, reduciendo por tanto las emisiones de gases nocivos, preservando el medio ambiente y la salud de los habitantes de las ciudades.

4 Metodología y Plan de Trabajo

Como se ha comentado anteriormente, uno de los aspectos importantes a determinar consiste en saber qué información se va a enviar, quién es capaz de recibir esa información y cómo actuar una vez recibida. Una vez propuesta la información que se quiera enviar será necesario darle una estructura con el fin de que haya un entendimiento común entre los vehículos y las autoridades. Para ello, se propone utilizar una ontología, concretamente la VEhicularACident ONtology (VEACON) que ya ha sido propuesta y utilizada de forma satisfactoria en trabajos previos (ver apartado Publicaciones Relacionadas con la Tesis).

Una vez seleccionado el contenido y diseñada la estructura de la información que se desea enviar, es necesario proveer de infraestructura de comunicaciones a los lugares por los que circulan los vehículos. Cada nodo de infraestructura (Road Side Unit, o RSU, en inglés) instalado requiere de una inversión inicial y un mantenimiento, por lo que sería deseable reducir el número de este tipo de nodos, reduciendo de esta forma el coste de implantación. Para ello, se propone el sistema D-RSU (ver apartado Publicaciones Relacionadas con la Tesis), el cual demuestra que teniendo en cuenta la densidad de vehículos que circulan habitualmente por una zona determinada se puede hacer un despliegue

de infraestructura más eficiente. Esto es debido a que en zonas con alta densidad de tráfico será necesario un menor número de RSUs, ya que los propios vehículos retransmitirán los mensajes de alerta de accidente, pudiendo llegar éstos correctamente a su destino.

En el momento en el que la información sobre un accidente llega a los servicios de emergencia, éstos parten hacia el lugar del mismo. Además, en las zonas próximas al accidente pueden surgir atascos y retenciones, debido a que los vehículos involucrados en él pueden obstaculizar la vía. Aunque los vehículos de los servicios de emergencia tienen prioridad de paso, el tener que atravesar un atasco, o conducir por vías en las que hay una elevada congestión de tráfico puede hacer que éstos se demoren más de lo deseable. En el presente trabajo se propondrá el uso de Algoritmos Evolutivos para solucionar este problema, recalculando las rutas de los vehículos para permitir que los servicios de emergencia lleguen los antes posible al lugar del accidente, y además se eviten los posibles atascos que se puedan generar por el mismo. También se tendrá en cuenta la prioridad de las calles en las intersecciones, así como la densidad de vehículos en determinadas zonas para evitar que se produzcan nuevos atascos. El uso de las redes vehiculares permitirá obtener información valiosa para el sistema de gestión del tráfico, y al mismo tiempo permitirá enviar las indicaciones a todos los conductores de forma que éstos puedan desplazarse de forma óptima.

Finalmente, puesto que en esta propuesta es necesario conocer la densidad de vehículos, se considera interesante el desarrollo de un sistema capaz de estimar el número de vehículos que están circulando por una determinada zona.

Todos los experimentos que se lleven a cabo para realizar las distintas tareas y su posterior validación, se realizarán mediante el simulador de movilidad del tráfico SUMO y el simulador de redes ns-2 [2]. Además, para hacer las simulaciones más realistas los escenarios serán obtenidos mediante OpenStreetMaps, un sistema de cartografía abierto que incorpora con gran precisión los mapas de la gran mayoría de ciudades y carreteras de todo el mundo.

5 Relevancia

A nivel científico, el presente trabajo propone varias contribuciones en el ámbito de los Sistemas Inteligentes de Transporte, tanto a nivel de las Redes Vehiculares, como de los Sistemas Evolutivos. Para hacernos una idea del impacto del mismo, hasta la fecha se han publicado un total de 6 artículos en revistas y congresos de primer nivel internacional.

En lo relativo a la transferencia de los resultados de investigación, el grupo al que pertenezco ha colaborado con varias empresas del sector del automóvil, entre las que destaco Applus IDIADA[1], empresa líder a nivel mundial en el sector, y con presencia en países de Europa, Asia y América. A partir de las ideas presentadas, pretendemos implementar un sistema de gestión del tráfico que utilizará los algoritmos propuestos en nuestro trabajo para calcular las mejores

[1] http://www.applusidiada.com/es/

rutas tanto para los servicios de emergencia (ambulancias, bomberos, etc.), como para el resto de vehículos. Además también se implementará un prototipo de Unidad de a bordo (que se instalará en los vehículos) que permita recoger y comunicar a los servicios de emergencia los datos más importantes sobre el accidente, así como facilitar las indicaciones necesarias al resto de los vehículos para que puedan desplazarse sin verse seriamente afectados por la situación de emergencia.

Finalmente, me gustaría destacar que todos los aspectos anteriormente comentados, también favorecen la vertebración del territorio, permitiendo que regiones pequeñas y despobladas como es el caso de Teruel, ciudad en la que trabajo y resido, puedan participar en la Investigación Internacional y crecer económicamente en la medida de lo posible, gracias a los resultados obtenidos.

6 Publicaciones Relacionadas con la Tesis

Como resultado directo e indirecto del trabajo que se está realizando, hasta la fecha se han publicado un total de 6 artículos, a destacar: 1 Revista indexada en el JCR (Q2), 1 Revista indexada en el SJR de SciMago (Q1), 2 Congresos indexados como CORE A, y 1 Congreso CORE B. A continuación se presenta una lista con los mismos:

- Barrachina, J., Garrido, P., Fogue, M., Martinez, F.J., Cano, J.C., Calafate, C.T., Manzoni, P.: Road Side Unit Deployment: A Density-Based Approach. To appear in the IEEE Intelligent Transportation Systems Magazine (Sept. 2013)
- Barrachina, J., Fogue, M., Garrido, P., Martinez, F.J., Cano, J.C., Calafate, C.T., Manzoni, P.: I-VDE: A Novel Approach to Estimate Vehicular Density by Using Vehicular Networks. In: The 12th International Conference on Ad Hoc Networks and Wireless (ADHOC-NOW 2013) (July 2013)
- Barrachina, J., Fogue, M., Garrido, P., Martinez, F.J., Cano, J.C., Calafate, C.T., Manzoni, P.: Assessing Vehicular Density Estimation Using Vehicle-to-Infrastructure Communications. In: The Fourteenth International Symposium on a World of Wireless, Mobile and Multimedia Networks (IEEE WoWMoM) (June 2013)
- Barrachina, J., Garrido, P., Fogue, M., Martinez, F.J., Cano, J.C., Calafate, C.T., Manzoni, P.: VEACON: A Vehicular Accident Ontology designed to improve safety on the roads. Journal of Network and Computer Applications, vol. 35, no. 6, pp. 1891-1900 (2012), http://dx.doi.org/10.1016/j.jnca.2012.07.013
- Barrachina, J., Garrido, P., Fogue, M., Martinez, F.J., Cano, J.C., Calafate, C.T., Manzoni, P.: D-RSU: A Density-Based Approach for Road Side Unit Deployment in Urban Scenarios. In: International Workshop on IPv6-based Vehicular Networks (Vehi6), collocated with the IEEE Intelligent Vehicles Symposium. pp. 1-6 (Jun 2012)
- Barrachina, J., Garrido, P., Fogue, M., Martinez, F.J., Cano, J.C., Calafate, C.T., Manzoni, P.: CAOVA: A Car Accident Ontology for VANETs. In:

IEEE Wireless Communications and Networking Conference (WCNC). pp. 1864-1869 (April 2012), http://dx.doi.org/10.1109/WCNC.2012.6214089

Además, se han enviado los siguientes 3 artículos: 1 Revista indexada en el JCR (Q1), 1 Revista indexada en el JCR (Q2), y 1 Congreso CORE B, que están siendo revisados, y por tanto, pendientes de aceptación. A continuación se presentan:

- Barrachina, J., Garrido, P., Fogue, M., Martinez, F.J., Cano, J.C., Calafate, C.T., Manzoni, P.: Reducing Emergence Services arrival time by using Vehicular Communications and Evolution Strategies. Expert Systems with Applications (2013)
- Barrachina, J., Garrido, P., Fogue, M., Martinez, F.J., Cano, J.C., Calafate, C.T., Manzoni, P.: A V2I-based Real-Time Traffic Density Estimation System. Computer Networks (2013)
- Barrachina, J., Fogue, M., Garrido, P., Martinez, F.J., Cano, J.C., Calafate, C.T., Manzoni, P.: Redistributing traffic routes in case of an accident by combining VNs and AI. In: IEEE International Conference on Tools with Artificial Intelligence (ICTAI) (2013)

References

1. Eigner, R., Lutz, G.: Collision avoidance in VANETs - an application for ontological context models. In: Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom). pp. 412–416 (March 2008)
2. Fogue, M., Garrido, P., Martinez, F.J., Cano, J.C., Calafate, C.T., Manzoni, P.: A Realistic Simulation Framework for Vehicular Networks. In: 5th International ICST Conference on Simulation Tools and Techniques (SIMUTools 2012), Desenzano, Italy. pp. 37–46 (Mar 2012)
3. Fogue, M., Garrido, P., Martinez, F.J., Cano, J.C., Calafate, C.T., Manzoni, P., Sanchez, M.: Prototyping an automatic notification scheme for traffic accidents in vehicular networks. In: IFIP Wireless Days (WD). pp. 1–5 (Oct 2011)
4. Fogue, M., Garrido, P., Martinez, F., Cano, J.C., Calafate, C., Manzoni, P.: Analysis of the most representative factors affecting warning message dissemination in vanets under real roadmaps. In: IEEE 19th International Symposium on Modeling, Analysis Simulation of Computer and Telecommunication Systems (MASCOTS). pp. 197–204 (2011)
5. Johnson, D., McGeoch, L.: The travelling salesman problem: A case study in local optimization. In: Local Search in Combinatorial Optimization. Wiley. pp. 215–310 (1997)
6. Mester, D., Bräysy, O.: Active guided evolution strategies for large-scale vehicle routing problems with time windows. Computers & Operations Research 32(6), 1593–1614 (2005)
7. Wang, Z., Shi, J.: A model for urban distribution system under disruptions of vehicle travel time delay. In: Second International Conference on Intelligent Computation Technology and Automation (ICICTA). vol. 3, pp. 433–436 (Oct 2009)
8. Yoshikawa, M., Terai, H.: Car navigation system based on hybrid genetic algorithm. In: 2009 WRI World Congress on Computer Science and Information Engineering. vol. 5, pp. 62–65 (Apr 2009)

Detección de tráfico de control de *botnets* mediante el modelado del flujo de paquetes

Doctorando: Félix Brezo[2]
Director: Dr. Pablo García Bringas[1]

Institución
pablo.garcia.bringas@deusto.es,felix.brezo@deusto.es
http://www.deustotech.deusto.es/

1. Resumen general de la propuesta de tesis

El origen del término *botnet* surge como fusión de las voces inglesas *robot-networks*. A día de hoy las redes de ordenadores secuestrados, que representan cerca del 25 % de todos los intentos de conexión maliciosos, son una mezcla de amenazas estrechamente relacionadas con otras áreas de *malware*: pueden propagarse como los gusanos, sirven para la ocultación de virus informáticos y permiten además el control remoto de la máquina infectada por parte de terceros. Estas circunstancias, junto con otros pruebas relacionadas con la escritura de código colaborativa a partir de iniciativas *open-source* o de código abierto —como ocurre con muestras de diversas familias como SDBot, Agobot o variantes del propio Zeus, cuyo código está comentado incluso por diferentes autores— permiten la proliferación de una amplia gama de variantes, modificaciones y mutaciones de los *bots* en función de la finalidad específica a la que las quiera dedicar su autor.

Así, hoy en día, los tipos de ataques que pueden ser llevados a cabo empleando este tipo de redes son muy variados: desde los sencillos pero efectivos ataques distribuidos de denegación de servicio —o DDoS, por las siglas en inglés de *Distributed Denial of Service attacks*— [1] o el robo masivo de credenciales o información confidencial [2], hasta ataques que explotan nuevas oportunidades de monetización de la infección como el *click-frauding* [3], el robo de carteras de divisas virtuales como Bitcoin [4] o movimientos aún más sofisticados que se hacen con el control de máquinas industriales con los más diversos fines y de dudosa procedencia como Stuxnet [5].

En este sentido, la utilización de técnicas de análisis de tráfico para la identificación de comportamientos automatizados ofensivos de carácter activo (como puede ser la ejecución de un ataque de denegación de servicio) es una línea de acción bien documentada en el estado del arte de la detección de *botnets* [6,7]. Sin embargo, estas aproximaciones son incapaces de detectar otro tipo de comportamientos maliciosos vinculados a la sustracción de información o a la ejecución de actividades maliciosas en las máquinas infectadas de forma remota, problemática que afecta cada vez con más frecuencia a empresas y corporaciones de todos los sectores económicos, así como a organismos de carácter público y administraciones.

Otros enfoques emplean técnicas más vinculadas al análisis de los contenidos que fluyen por la red. Estas aproximaciones eran capaces de detectar en el pasado conexiones automatizadas en los primitivos canales de control de *botnets* implantados sobre plataformas IRC [8,9], pero la utilización de estas técnicas quedará obsoleta a partir del momento en que los atacantes generalicen la utilización de conexiones cifradas para el envío y recepción de comandos como ya ocurre con la práctica totalidad del *malware* profesional. Sin embargo, el enfoque propuesto pretende dar una respuesta que no esté basada en el contenido de las comunicaciones —dejando de lado el análisis de los datos en sí mismos— y que tampoco se centre en evaluar la procedencia de la conexión si no en otros atributos relativos al direccionamiento de estas y a los tiempos con que los paquetes se envían y reciben.

1.1. Solución propuesta

La solución que se propone en el presente documento consiste en la creación de un sistema de detección basado en red para la identificación de comunicaciones de control de *botnet*. Para ello, se dispondrá de herramientas que permitirán monitorización del tráfico entrante y saliente de la red y se extraerán al vuelo atributos relativos a las comunicaciones que están teniendo lugar. El objetivo es construir las consiguientes representaciones de cada conexión en base a lo observado para conformar lo que en adelante se conocerá como secuencias de n-paquetes. La construcción de cada una de las representaciones se realizará en base a la observación de los atributos de control de cada paquete que fluye por la red (longitudes de las cabeceras y del tipo de datos, *flags*, atributos opcionales, etc.) así como a características espacio-temporales específicas de dicha conexión (e. g., la periodicidad de las sondeos efectuados entre cliente y servidor). En el caso del modelo que emplea secuencias de $n = i$ ($\forall i \in \mathbb{N}$ tal que $1 <= i <= 12$) paquetes se contará con más características: los atributos del primer paquete, los del segundo, etc., y así sucesivamente hasta alcanzar el enésimo paquete que componga dicha representación.

1.2. Hipótesis fundamental

En este contexto, la definición de metodologías capaces de hacer frente de una forma efectiva a la amenaza de las *botnets* se torna fundamental. Así, el autor enuncia a continuación una metodología para el análisis de muestras de tráfico de *botnet* pertenecientes a canales de *Command & Control* de modo que puedan ser utilizadas para la identificación de amenazas relativas a este fenómeno empleando algoritmos de clasificación supervisada, partiendo de la hipótesis en la que se basa esta tesis doctoral y ya introducida en [10]:

> «*El uso de características atómicas para detectar aquellas comunicaciones que compongan el proceso de envío y recepción de comandos maliciosos hacia o desde las víctimas infectadas por malware concebido para el despliegue de una botnet*».

1.3. Objetivos generales y específicos

Con el objetivo último de poder corroborar o rechazar la hipótesis de partida anterior, se han definido los siguientes objetivos generales:

Objetivo principal 1 *Desarrollar una metodología que permita la obtención de muestras de tráfico para su validación.*

Objetivo principal 2 *Obtener un modelo de representación de las conexiones que permita la clasificación de paquetes en base a sus características.*

Objetivo principal 3 *Obtener un modelo de representación de las conexiones que permita la clasificación de secuencias de paquetes de diferentes longitudes procedentes de una misma conexión.*

Para garantizar la consecución de los anteriores objetivos generales, se han definido los siguientes objetivos específicos:

Objetivo específico 1 *Desarrollar una metodología que permita la obtención de muestras de tráfico para su validación.*

Objetivo específico 2 *Agrupar las conexiones y anonimizar las muestras para eliminar la posibilidad de que aparecieran sesgos —en la forma de direcciones IP o puertos— que trivializaran las muestras.*

Objetivo específico 3 *Seleccionar aquellos atributos representativos de las conexiones maliciosas y filtrar del resto de características.*

Objetivo específico 4 *Establecer medidas que permitan obtener patrones relativos a la frecuencia con la que se producen las comunicaciones en base a diferentes valores de tendencia central.*

Objetivo específico 5 *Generar una metodología capaz de representar muestras de tráfico de secuencias de paquetes de diferentes longitudes.*

Objetivo específico 6 *Evaluar la capacidad de que este modelo de representación sea capaz de identificar tráfico perteneciente a canales de Command & Control.*

Objetivo específico 7 *Determinar cuáles son aquellos clasificadores supervisados que mejores resultados generales obtienen tras ser entrenados con muestras procedentes de botnets diferentes.*

2. Metodología y plan de trabajo

La propuesta considera la tarea de detección como una problemática de clasificación supervisada. En este tipo de problemas se dispone de una base de conocimiento con una cantidad de datos (tanto benignos como maliciosos) etiquetados con anterioridad. Generadas las representaciones, se procede a entrenar los diferentes algoritmos de aprendizaje automático sobre los que recaerá la tarea de ejecución de la predicción. Posteriormente, será este modelo el encargado de clasificar cada una de las nuevas muestras de tráfico obtenidas realizando una serie de predicciones con respecto a la naturaleza de la observación. En este sentido, la limitada carga maliciosa presente en cada paquete hace que el proceso sea menos crítico a como ocurre en otras áreas como la detección de *malware*. Esto se debe a que la filtración de un falso negativo (en nuestro caso, que un paquete malicioso, o una reducida serie de ellos, sea erróneamente etiquetado como benigno) puede ser corregida conforme se vayan adquiriendo un número mayor y/o más representativo de muestras pertenecientes a cada conexión sospechosa.

Fases en las que se divide el proyecto

Con tal fin, se ha propuesto la división del proceso de desarrollo de la tesis doctoral en las tareas recogidas a continuación.

- **Tarea 1: revisión del estado del arte e identificación del problema** (T_1). La tarea T_1 consiste en la elaboración de una revisión exhaustiva del estado del arte actual en lo que a las técnicas de análisis de trafico se refiere para la detección de tráfico procedente de canales de *Command & Control*.
- **Tarea 2: modelado de tráfico simple** (T_2). La tarea T_2 consiste en la proposición de un modelado de tráfico simple que emplea los atributos propios de los paquetes como características a utilizar para obtener las representaciones. Para validar la eficacia del modelo se ha considerado el problema como un asunto de clasificación supervisada. La tarea T_2 se puede dividir en las siguientes subtareas: tarea T_{21}, diseño del modelo de representación simple; tarea T_{22}, obtención de muestras benignas; tarea T_{23}, obtención de muestras de tráfico de Flu; tarea T_{24}, ejecución de experimentos de clasificación supervisada con Flu; tarea T_{25}, obtención de muestras de tráfico de Prablinha; tarea T_{26}, ejecución de experimentos de clasificación supervisada con Prablinha; tarea T_{27}, obtención de muestras de tráfico de Warbot; tarea T_{28}, ejecución de experimentos de clasificación supervisada con Warbot y tarea T_{29}, ejecución de experimentos genéricos de clasificación supervisada.
- **Tarea 3: modelado de tráfico por secuencias de n-paquetes** (T_3). En la tarea T_3 se busca ampliar el alcance del modelo de representación anterior ampliando las características a aquellas propias de secuencias de n paquetes mediante la combinación de todas aquellas pertenecientes a los paquetes que las componen. La tarea T_3 se puede dividir en tres subtareas: la tarea T_{31} relacionada con el diseño del modelo de representación de muestras empleando atributos pertenecientes a secuencias de n-paquetes; la tarea T_{32},

vinculada a la construcción de representaciones de n-paquetes y la tarea T_{33}, ejecución de experimentos de clasificación supervisada con Flu, Prablinha, Warbot y tráfico genérico. En este último apartado se ejecutan las pruebas para las $n-1$ representaciones constituidas por los atributos de $n = k \rightarrow \forall k \in \{2,3,4,5,6,7,8,9,10,11,12\}$ paquetes consecutivos procedentes de una misma conexión.

- **Tarea 4: reducción de esfuerzos** (T_4). Vista la cantidad de paquetes necesaria para afrontar el problema como un problema de clasificación supervisada, en la tarea T_4 se propone la utilización de clasificadores colectivos para estudiar cómo afecta la reducción de las instancias etiquetadas utilizadas para entrenar el modelo de aprendizaje.
- **Tarea 5: redacción de la tesis** (T_5). La tarea T_5 está constituida por el proceso de redacción de la tesis doctoral en sí misma. En esta tarea se incluye también la redacción de anexos, diseño de contenidos gráficos, maquetación y revisión, entre otros aspectos.
- **Tarea final: registro y presentación** (T_F). Por último, la tarea T_F hace referencia a la formulación del proceso de registro, depósito, constitución del tribunal y defensa de la presente tesis doctoral.

3. Relevancia del proyecto e impacto potencial

La implantación de los modelos de representación propuestos es versátil y adaptable a los requerimientos concretos de clientes de varios ámbitos por varios motivos. En primer lugar, por la disponibilidad de herramientas libres para llevar a cabo una monitorización avanzada del tráfico de red en prácticamente todos los sistemas operativos. En segundo lugar, el enfoque propuesto deja abierta la posibilidad de ejecutar el proceso de análisis de tráfico directamente desde la red y de una forma completamente transparente para los equipos conectados a ella mediante la utilización de terminales, tecnologías o equipos destinados únicamente a clasificar e identificar el tráfico malicioso. De esta manera, la utilización de librerías de inteligencia artificial aplicadas no tendría implicaciones negativas en el rendimiento de las máquinas convencionales que hacen uso legítimo de la red al trasladar estas tareas a un tercer equipo destinado a tal fin.

Los usos más representativos de la solución propuesta pasan por identificar la existencia de comunicaciones de control maliciosas en el seno de una red aplicando técnicas de análisis de tráfico no intrusivas en lo que al análisis de contenidos se refiere. Este enfoque complementa otros métodos de detección de *malware* que se basen en en el análisis del equipo porque descarga el trabajo hacia máquinas que monitorizan las características del tráfico de red que genera cada nodo. De esta manera, se puede permitir la identificación de mutaciones de *malware* no identificadas por los métodos tradicionales de detección de firmas y que traten de comunicarse con su servidor de órdenes.

Desde el punto de vista empresarial, el impacto de un sistema de detección de canales de control vinculados al *malware* profesional en general y a la construcción de *botnets* en particular aporta una interesante capa de protección adicional

a los sistemas convencionales basados en *host*. A sabiendas de que la problemática del espionaje industrial, que explota las vulnerabilidades telemáticas de las redes y equipos corporativos, es una amenaza cada vez más extendida, la identificación de filtraciones y de comportamientos maliciosos se torna fundamental para la mitigación de los efectos del ciberespionaje.

En cualquier caso, la explotación económica de una solución de estas características puede enfocarse desde dos ámbitos: por un lado, compañías y/o instituciones con la intención de integrar herramientas similares para proteger el acceso a su propia red y, por otro, terceras empresas que ofrecieran el servicio como parte añadida de sus productos destinados a la detección de *malware* y la identificación proactiva de amenazas.

Referencias

1. E. Cooke, F. Jahanian, and D. McPherson, "The zombie roundup: Understanding, detecting, and disrupting botnets," in *Proceedings of the USENIX SRUTI Workshop*, 2005, pp. 39–44.
2. H. Binsalleeh, T. Ormerod, A. Boukhtouta, P. Sinha, A. Youssef, M. Debbabi, and L. Wang, "On the analysis of the zeus botnet crimeware toolkit," in *Privacy Security and Trust (PST), 2010 Eighth Annual International Conference on.* IEEE, 2010, pp. 31–38.
3. N. Ianelli and A. Hackworth, "Botnets as a vehicle for online crime," *CERT Coordination Center*, pp. 1–28, 2005.
4. F. Brezo and P. G Bringas, "Issues and risks associated with cryptocurrencies such as bitcoin," in *SOTICS 2012, The Second International Conference on Social Eco-Informatics*, 2012, pp. 20–26.
5. G. McDonald, L. O. Murchu, S. Doherty, and E. Chien, "Stuxnet 0.5: the missing line," Symantec, Security Response, Tech. Rep., 2013.
6. J. Goebel and T. Holz, "Rishi: Identify bot contaminated hosts by irc nickname evaluation," in *Proceedings of the USENIX Workshop on Hot Topics in Understanding Botnets (HotBots)*, 2007.
7. S. Racine, "Analysis of internet relay chat usage by ddos zombies," *Master's thesis, Swiss Federal Institute of Technology Zurich*, 2004.
8. A. Karasaridis, B. Rexroad, and D. Hoeflin, "Wide-scale botnet detection and characterization," in *Proceedings of the USENIX Workshop on Hot Topics in Understanding Botnets (HotBots)*, 2007.
9. H. Choi, H. Lee, and H. Kim, "Botgad: detecting botnets by capturing group activities in network traffic," in *Proceedings of the Fourth International ICST Conference on COMmunication System softWAre and middlewaRE, COMSWARE*, vol. 9, 2009, p. 2.
10. F. Brezo, J. Gaviria de la Puerta, X. Ugarte-Pedrero, I. Santos, P. G. Bringas, and D. Barroso, "Supervised classification of packets coming from a http botnet," in *Informatica (CLEI), 2012 XXXVIII Conferencia Latinoamericana En.* IEEE, 2012, pp. 1–8.

Parallel Data Mining Algorithms on GPUs

Doctorando: Alberto Cano Rojas
Directores: Sebastián Ventura y Amelia Zafra

Departamento de Informática y Análisis Numérico
Universidad de Córdoba
{acano,sventura,azafra}@uco.es
http://www.uco.es/grupos/kdis

Keywords: Data Mining, Big Data, Parallel Computing, GPUs

1 Introducción

La minería de datos consiste en la aplicación de algoritmos específicos para extraer conocimiento útil, comprensible y no trivial de conjuntos de datos. En los últimos años, la minería de datos se ha transformado para dar respuesta a nuevos problemas y retos cada vez más complejos sobre conjuntos de datos de creciente dimensionalidad, fenómeno conocido como *bigdata*. Las técnicas clásicas de minería de datos ya no son capaces de obtener resultados satisfactorios en los nuevos tipos de problemas que se afrontan en la actualidad. El tamaño y la complejidad de los nuevos conjuntos de datos, los nuevos tipos de problemas que afrontar, junto con la necesidad de dar resultados exactos y en tiempo razonable, exige una nueva generación de algoritmos que sean capaces de dar respuesta a estas demandas tanto en el ámbito de la investigación científica como en el de aplicaciones empresariales y soluciones para la sociedad del siglo XXI.

La revolución tecnológica ofrece en la actualidad multitud de sistemas con alta capacidad de cómputo a bajo coste y al alcance de empresas y usuarios. Los procesadores multi-núcleo y las tarjetas de procesamiento gráfico (GPUs) inundan el mercado y ya están presentes en la mayoría de equipos en hogares, empresas y centros de investigación. Disponemos de toda una infraestructura de computación ubicua e intercomunicada lista para resolver nuestras crecientes necesidades de cómputo. En este proyecto de tesis doctoral investigaremos la aplicación de la computación en GPUs al diseño y aceleración de algoritmos de minería de datos, especialmente sobre grandes conjuntos de datos.

2 Hipótesis de Partida

La paralelización de los algoritmos en procesadores multi-núcleo ha sido una solución clásica al problema del elevado tiempo de cómputo [1]. Sin embargo, estas soluciones se limitan habitualmente a una mera paralelización en pocos hilos, logrando una aceleración limitada. Por lo tanto, siguen sin dar solución a los verdaderos problemas y retos de computación de los algoritmos.

La computación de propósito general usando GPUs ha demostrado desde sus inicios un verdadero cambio de mentalidad en la forma de abordar la computación paralela. Ya no se trata de una mera paralelización del código, sino de un cambio radical en la forma de diseñar los algoritmos para aprovechar las enormes capacidades de cómputo de las GPUs. Problemas muy complejos o de gran tamaño son resolubles en tiempo razonable empleando millones de hilos que cooperan en la solución del problema. La gran mayoría de tipos de problemas en minería de datos son resolubles mediante algoritmos masivamente paralelos. Por lo tanto, la sinergia en el diseño de nuevos algoritmos de minería de datos masivamente paralelos bajo un paradigma de computación en GPU es un nuevo campo de investigación y aplicación con alto interés científico y comercial. Prueba de esta tendencia es el elevado número de publicaciones científicas en este campo (scopus 2012: 2726, 2011: 2441, 2010: 1795, 2009: 1217, 2008: 807). Los primeros artículos científicos publicados sobre la computación de propósito general en GPU y sus aplicaciones recibieron una gran acogida y actualmente poseen cientos de citas [2–4]. Rápidamente su aplicación se ha extendido a grandes campos de la inteligencia artificial, visión artificial, metaheurísticas y minería de datos, donde numerosos autores de prestigio como W. B. Langdon han tomado la computación en GPU como un nuevo paradigma para la resolución de problemas [5–8].

3 Objetivos

El objetivo de esta tesis doctoral es el estudio, análisis, diseño, implementación y evaluación de algoritmos de minería de datos en GPUs. El desarrollo de la tesis es multidisciplinar puesto que se deberá conocer y aplicar los conocimientos relativos al diseño hardware de las GPUs, junto a la resolución e innovación en las demandas de soluciones algorítmicas a problemas de minería de datos. Específicamente, se desglosan dos grandes objetivos:

- La paralelización de algoritmos ya existentes con un elevado coste computacional para lograr su ejecución en un tiempo razonable.
- El diseño y propuesta de nuevos algoritmos, ideados ya bajo el paradigma de computación masivamente paralela en GPU, para lograr mejores resultados que algoritmos del estado del arte en las tareas de minería de datos.

4 Metodología y Plan de Trabajo

En plan de la trabajo de la tesis doctoral se enmarca bajo las líneas de actuación de la beca predoctoral perteneciente al programa de formación del profesorado universitario (FPU) del Ministerio de Educación en el periodo abarcado entre Diciembre de 2011 y Diciembre de 2015. El plan de trabajo previsto se divide de la siguiente forma:

- Primer año: revisión bibliográfica completa de la investigación desarrollada en computación en GPUs y minería de datos, incluyendo modelos de clasificación y asociación. Identificación de los retos y problemas abiertos en las técnicas de minería de datos.

- Segundo y tercer año: desarrollo e implementación de algoritmos de minería de datos en GPU. Estudios experimentales. Publicaciones científicas.
- Cuarto año: aplicación de su funcionamiento en distintos marcos de trabajo y aplicaciones reales para su transferencia al mundo empresarial y a la sociedad. Redacción definitiva de la memoria de tesis.

En cuanto a la metodología, se seguirá el método científico para lograr los objetivos mencionados:

- Establecimiento de la hipótesis: desarrollo de nuevos algoritmos en GPU para la extracción de conocimiento, mejora y adaptación de los algoritmos disponibles para problemas específicos, y análisis de los mecanismos para comparar dichos algoritmos.
- Recopilación de datos: obtención de bases de datos públicas y actuales sobre las que extraer conocimiento.
- Validar las hipótesis y las observaciones: evaluar la calidad y rendimiento de los algoritmos de minería de datos para la extracción de conocimiento respecto a las bases de datos utilizadas.
- Readaptación de la hipótesis inicial tomando en cuenta los resultados obtenidos. Esto implicará la modificación y ajuste de los algoritmos y los mecanismos de análisis acerca de su comportamiento como resultado de validaciones llevadas a cabo y de la experiencia acumulada.

5 Relevancia

El interés y la importancia del trabajo desarrollado en esta tesis doctoral será muy relevante en el ámbito científico de la minería de datos. Ofrece soluciones reales, prácticas y de bajo coste a los nuevos problemas y retos en minería de datos. Los ingente cantidad de publicaciones anuales en este campo demuestra la utilidad de la computación en GPU en técnicas de inteligencia artificial que requieren altos costes computacionales, bien por la complejidad algorítmica o por el gran tamaño de los conjuntos de datos. Temas punteros hoy en día como son *big data* y *cloud computing* incluyen soluciones tecnológicas basadas en procesamiento paralelo y distribuido con GPUs.

Los resultados obtenidos en los dos primeros años de investigación han demostrado alcanzar un gran rendimiento en la aceleración de algoritmos de minería de datos ya existentes. Concretamente, hemos logrado acelerar la ejecución de algoritmos evolutivos de reglas de clasificación [9, 10] y de minería de reglas de asociación [11]. Además, el uso de GPUs nos ha permitido el desarrollo de nuevos modelos masivamente paralelos [12], con los que hemos conseguido resultados mejores que los algoritmos del estado del arte. Recientemente, también hemos logrado buenos resultados en la aceleración de algoritmos de reglas de clasificación de datos multi-instancia, así como en la paralelización de algoritmos de discretización de datos.

En la actualidad continuamos investigando y desarrollando con gran interés nuevos modelos masivamente paralelos en GPU sobre otras tareas de minería de datos. Es el caso de la clasificación multi-etiqueta, donde además del problema de la dimensionalidad, existen otros problemas abiertos como el desbalanceo de las etiquetas. Además, nos estamos introduciendo en otras áreas de la inteligencia artificial, como es la visión artificial, para acelerar algoritmos de *markerless human motion capture*, donde también estamos obteniendo buenos resultados.

La presencia de GPUs en clusters y en millones de ordenadores por todo el mundo permite el desarrollo de aplicaciones de investigación y comerciales que aceleren el cómputo en GPU. De una forma totalmente transparente para el usuario, es posible convertir su ordenador en una plataforma de cómputo muy potente para la resolución de problemas de inteligencia artificial, ingeniería, biomedicina, física, química, etc. De esta forma, se logra la transferencia del desarrollo logrado por la investigación a la sociedad.

Referencias

1. G. Luque and E. Alba, *Parallel Genetic Algorithms: Theory and Real World Applications*, ser. Studies in Computational Intelligence. Springer, 2011.
2. J. Owens, D. Luebke, N. Govindaraju, M. Harris, J. Krger, A. Lefohn, and T. Purcell, "A survey of general-purpose computation on graphics hardware," *Computer Graphics Forum*, vol. 26, no. 1, pp. 80–113, 2007.
3. J. Owens, M. Houston, D. Luebke, S. Green, J. Stone, and J. Phillips, "GPU computing," *Proceedings of the IEEE*, vol. 96, no. 5, pp. 879–899, 2008.
4. S. Che, M. Boyer, J. Meng, D. Tarjan, J. Sheaffer, and K. Skadron, "A performance study of general-purpose applications on graphics processors using CUDA," *Journal of Parallel and Distributed Computing*, vol. 68, no. 10, pp. 1370–1380, 2008.
5. K.-L. Fok, T.-T. Wong, and M.-L. Wong, "Evolutionary computing on consumer graphics hardware," *IEEE Intelligent Systems*, vol. 22, no. 2, pp. 69–78, 2007.
6. W. Langdon and A. Harrison, "GP on SPMD parallel graphics hardware for mega bioinformatics data mining," *Soft Computing*, vol. 12, no. 12, pp. 1169–1183, 2008.
7. W. Langdon, "Large scale bioinformatics data mining with parallel genetic programming on graphics processing units," *Studies in Computational Intelligence*, vol. 269, pp. 113–141, 2010.
8. L. Jian, C. Wang, Y. Liu, S. Liang, W. Yi, and Y. Shi, "Parallel data mining techniques on Graphics Processing Unit with Compute Unified Device Architecture (CUDA)," *Journal of Supercomputing*, pp. 1–26, 2011.
9. A. Cano, A. Zafra, and S. Ventura, "Speeding up the evaluation phase of GP classification algorithms on GPUs," *Soft Computing*, vol. 16, no. 2, pp. 187–202, 2012.
10. A. Cano, J. Olmo, and S. Ventura, "Parallel Multi-Objective Ant Programming for Classification Using GPUs," *Journal of Parallel and Distributed Computing*, vol. 73, no. 6, pp. 713–728, 2013.
11. A. Cano, J. M. Luna, and S. Ventura, "High Performance Evaluation of Evolutionary-Mined Association Rules on GPUs," *Journal of Supercomputing, In press*, 2013.
12. A. Cano, A. Zafra, and S. Ventura, "Parallel evaluation of Pittsburgh rule-based classifiers on GPUs," *Neurocomputing, In press*, 2013.

PhD Thesis proposal:
Studying Permutation-based Combinatorial Optimization Problems and Proposing Efficient Estimation of Distribution Algorithms

Josu Ceberio

Advisors: Alexander Mendiburu, Jose A. Lozano
Intelligent Systems Group,
Department of Computer Science and Artificial Intelligence,
University of the Basque Country UPV/EHU,
Donostia, 20018 Gipuzkoa, Spain
josu.ceberio@ehu.es

Keywords: Estimation of Distribution Algorithms, Permutation-based Combinatorial Optimization Problems, Probability Models on Rankings

1 Introduction

In the last 20 years, metaheuristic algorithms [9] have attracted the attention of the artificial intelligence community, postulating as a solid alternative to operational research methods in the resolution of combinatorial optimization problems. Characterized by the absence of mathematical formalizations, metaheuristics are able to obtain very competitive results (optimal in some cases), in affordable computation times. These methods make few assumptions about the optimization problem to be solved, and thus, they can be applied to a wide variety of optimization problems, even in the cases where the function to be optimized has no mathematical expression or it can only be approximately calculated.

In this thesis proposal, we focus on a set of metaheuristics called estimation of distribution algorithms (EDAs) [6] that belong to the family of evolutionary algorithms (EAs). The main characteristic of EAs is the use of techniques inspired by the natural evolution of species. In nature, species change across time; individuals evolve, adapting to the characteristics of the environment. This evolution leads to individuals with better characteristics. The same idea is translated to the world of computation, where an individual represents a particular solution for the problem to be solved, a population comprises several individuals, and different operators such as crossover, mutation and selection techniques are used to make the individuals (solutions) evolve (improve). The most popular reference of these types of algorithms are Genetic Algorithms (GAs) [2]. As an extension to GAs, EDAs were introduced in the field of EAs.

Based on machine learning techniques, at each iteration, instead of using recombination and mutation operators, EDAs estimate a joint probability dis-

tribution associated with the set of most promising individuals, trying to explicitly express the interrelations between the different variables of the problem. Sampling the probabilistic model generated in the previous generation, a new population of solutions for the problem is created. The algorithm stops iterating and returns the best solution found across the generations when a certain stopping criterion is met, such as a maximum number of generations/evaluations, homogeneous population, or lack of improvement in the solutions. Based on this general framework, several EDA approaches have been developed in the last years [4,5,7], where each one learns a specific probabilistic model that conditions the behavior of the EDA from the point of view of complexity and performance.

EDAs have demonstrated their validity in the solution of real problems from diverse fields [5,8]. However, there is a type of problem for which EDAs have not been extensively developed and whose solutions are coded naturally as permutations; we refer to the *permutation-based combinatorial optimization problems* (PCOPs). Travelling Salesman, Quadratic Assignment or Flowshop Scheduling are examples of this category of problems. These problems present a real challenge for EDAs, and particularly for the typical compact and factorized probability models implemented by these, since in most cases these models cannot capture the *mutual exclusivity constraints* associated with the permutations.

2 Related work

Many works in the literature [1, 5, 10–12] try to solve permutation problems based on EDA type approaches. However, we see that in most of the cases, EDAs designed for integer and real-valued domains are adapted to deal with permutation-based problems. On the one hand, some papers propose learning discrete domain models such as univariate, bivariate [12] or probabilistic graphical models [11], and modifying the sampling step in order to guarantee feasible solutions. On the other hand, some works suggest translating the permutation problem to the real-valued domain [1], or to a subset of natural values [10], which, due to the redundancy associated with the codification, lead EDAs to a poor performance. Nonetheless, the main drawback of the previous approaches is that these EDAs do not learn a probability distribution over a permutation space, but a distribution over an integer or real-values space, and thus, these models do not efficiently summarize the irregularities contained in the permutations.

Due to the lack of suitable proposals for solving PCOPs, and motivated by the growing interest seen with respect to probability models on rankings for modelling permutation functions, this PhD thesis proposal focuses on the introduction of these models to the framework of EDAs. In this sense, Thurstone order statistics models, models induced by paired comparisons, distance-based ranking models or the multistage ranking models define a wide variety of probability models which could be applied in EDAs for solving PCOPs.

Alternatively, a number of papers focus [3] on the use of the "low-frequency" terms of a Fourier decomposition in order to codify probability distributions over permutations compactly discarding high order marginal information. Although

these methods can only be used with small size instances, such approaches could be used to study the geometry of the problems/instances by analyzing the number of Fourier terms that are needed to give a reliable codification of the probability distribution associated to the problem. This study can enhance the understanding of this type of problems, identifying the critical characteristics that the models should take into account when implementing EDAs.

3 Aims and Objectives

The main objectives of the research are detailed at following:

- To review and compare the existing EDAs proposed in the literature for solving PCOPs.
- To study probability models on rankings and to introduce a new model in the framework of EDAs for solving PCOPs.
- To propose an EDA-based approach that implements one of the models studied in the previous objective, and outperforms the state-of-the-art of a selected PCOP.
- To study PCOPs in depth in order to determine the relevant characteristics to take into account when designing new EDAs.

4 Working plan of the project

The working plan will be oriented to take a step forward in each of the proposed aims. The research contributions will be summarized in articles to submit to referred journals of the JCR (preferably to the first quartile), sending in some cases preliminary versions of these papers to the most prestigious conferences in the area. The working plan of the proposed objectives is detailed below:

- Objective 1: To review and compare the existing EDAs proposed in the literature for solving PCOPs.
  - Task 1: Review the state-of-the-art works on EDAs and choose the most representative approaches to include in the study. Obtain the source codes or try to implement the algorithms following the specifications of the respective papers.
  - Task 2: Choose a subset of different permutation problems which is representative enough of the state-of-the-art in order to evaluate the performance of the EDAs considered in the previous task.
  - Task 3: Design an experimental study to evaluate the EDAs over the permutation problems selected in the previous step.
  - Task 4: Determine which are the most successful probability models and identify the associated properties.
  - Task 5: Write a scientific report.
- Objective 2: To study probability models on rankings and to introduce a new model in the framework of EDAs for solving PCOPs.

  - Task 1: Review the probabilistic models for permutation domains proposed in the Machine Learning and Statistical literature.
  - Task 2: Analyze and classify the existing probability models on rankings in order to determine the applicability of them in the framework of EDAs.
  - Task 3: Propose a new EDA based on a probability model on ranking.
  - Task 4: Evaluate the efficiency of the proposed EDA in the benchmark used in the previous objective, and draw conclusions.
  - Task 5: Write a scientific report.
- Objective 3: To propose an EDA-based approach that implements one of the models studied in the previous objective, and outperforms the state-of-the-art of a selected PCOP.
  - Task 1: Choose a state-of-the-art permutation-based problem which has been extensively studied.
  - Task 2: Review the state-of-the-art of the selected problem and study the best performing approaches.
  - Task 3: Develop an EDA-based approach. Hybridization techniques will be considered in order to propose a competitive algorithm.
  - Task 4: Carry out an experimental analysis and determine the contribution of the new EDA in the approach.
  - Task 5: Write a scientific report and draw conclusions about the applicability of probability models on rankings for solving these type of problems.
- Objetive 4: To study PCOPs in depth in order to determine the relevant characteristics to take into account when designing new EDAs.
  - Task 1: Choose a wide set of the most representative permutation-based problems, and define their characteristics in detail.
  - Task 2: Determine a list of meaningful descriptors to categorize the permutation-based problems.
  - Task 3: Study the statistical properties of the Boltzmann distribution associated to the permutation problems, such as L-decomposability, conditional independence, etc.
  - Task 4: Study the codification of probability distributions using the low order terms of the Fourier decomposition in each problem and determine the order of the terms needed to codify feasible distributions.
  - Task 5: From task 3 and 4 draw conclusions about the permutation problems and propose the most effective models for each problem.
  - Task 6: Write a scientific report.

5 Relevance

Little work has been done with regard to permutation-based problems from the perspective of EDAs. In most cases, other metaheuristics have been preferred due to the difficulty of efficiently modelling permutations. This work intends to extend the applicability of EDAs to permutation domain problems, proposing models that perform efficiently even in large scale instances.

Additionally, we aim to contribute to the understanding of permutation-based problems, beyond the *mutual exclusivity constraint* addressed so many times. By studying these problems in terms of probabilistic modelling, we aim to shed some light on understanding the behavior of the existing EDAs over PCOPs, in order to propose more efficient approaches.

References

1. P. A. N. Bosman and D. Thierens. Crossing the road to efficient IDEAs for permutation problems. In *GECCO*, 2001.
2. D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison/Wesley, Reading MA, 1989.
3. J. Huang, C. Guestrin, and L. Guibas. Fourier theoretic probabilistic inference over permutations. *Journal of Machine Learning Research*, 10:997–1070, 2009.
4. P. Larrañaga and J. A. Lozano. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Kluwer Academic Publishers, 2002.
5. J. A. Lozano, P. Larrañaga, I. Inza, and E. Bengoetxea. *Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms (Studies in Fuzziness and Soft Computing)*. Springer, 2006.
6. H. Mühlenbein and G. Paaß. From Recombination of Genes to the Estimation of Distributions I. Binary Parameters. In *Lecture Notes in Computer Science 1411: Parallel Problem Solving from Nature - PPSN IV*, pages 178–187, 1996.
7. M. Pelikan, D. E. Goldberg, and F. G. Lobo. A survey of optimization by building and using probabilistic models. *Comp. Opt. and Applications*, 21(1):5–20, 2002.
8. M. Pelikan, K. Sastry, and E. Cantú-Paz. *Scalable Optimization via Probabilistic Modeling: From Algorithms to Applications (Studies in Computational Intelligence)*. Springer, 2006.
9. C. R. Reeves. *Modern Heuristic Techniques for Combinatorial Optimization Problems*. Blackwell Scientific Publications, 1993.
10. T. Romero and P. Larrañaga. Triangulation of Bayesian networks with recursive Estimation of Distribution Algorithms. *International Journal of Approximate Reasoning*, 50(3):472–484, 2009.
11. S. Tsutsui and M. Miki. *Recent Adv. in Sim. Evol. and Learning*, chapter Using Edge Histogram Models to Solve Permutation Problems with Probabilistic Model-Building Genetic Algorithms. Advances in Natural Computation Series. 2004.
12. S. Tsutsui, M. Pelikan, and D. E. Goldberg. Node Histogram vs. Edge Histogram: A Comparison of PMBGAs in Permutation Domains. Technical report, Medal, 2006.

Tratamiento de la dimensionalidad, desbalanceo e información de dependencia en clasificación multietiqueta

Doctorando: Francisco Charte Ojeda
Directores: Antonio Jesús Rivera Rivas y María José del Jesus Díaz

Dep. de Informática, Universidad de Jaén
Campus Las Lagunillas s/n — 23071 - Jaén
francisco@fcharte.com,{arivera,mjjesus}@ujaen.es
http://simidat.ujaen.es

Keywords: clasificación multietiqueta, preprocesamiento, desbalanceo, dimensionalidad.

1 Introducción

La clasificación es una de las tareas más importantes y estudiadas en Minería de Datos. En un escenario de clasificación tradicional el objetivo es establecer una relación entre un espacio de entrada o características X^n hacia un conjunto finito de clases o etiquetas L, donde $L = l_1, l_2, \ldots, l_k$. Cuando $|L| = 2$ se dice que el clasificador es binario. Si $|L| > 2$ se precisa un clasificador multiclase. En cualquier caso cada instancia pertenece exclusivamente a una de esas clases.

Cuando cada muestra del conjunto de datos se asocia no a una etiqueta sino a un conjunto de ellas $Y \subseteq L$, aparece el término de *Conjunto de Datos Multietiqueta* y el campo de la *Clasificación Multietiqueta* [1]. La clasificación binaria y multiclase se pueden considerar casos especiales de problemas multietiqueta donde el número de etiquetas asignadas a cada instancia es 1.

Si bien los primeros problemas en clasificación multietiqueta estaban relacionados con la categorización de textos [2], el campo de aplicación de esta técnica se ha diversificado bastante. En los últimos años se ha usado clasificación multietiqueta para la anotación semántica de imágenes y vídeo [3], categorización de música [4], identificación funcional de genes [2], etc.

Las vías de investigación en el campo de la clasificación multietiqueta [1] se han concentrado hasta el momento fundamentalmente en los dos siguientes enfoques:

- **Transformación de los datos**: Partiendo del dataset multietiqueta original se obtiene un dataset multiclase o múltiples datasets binarios que pueden ser procesados mediante clasificadores tradicionales. Puede tener inconvenientes de pérdida o degeneración de información, al ser una simplificación del problema original en la que no se tienen en cuenta importantes aspectos específicos de este tipo de datasets.

– **Adaptación de métodos**: La mayoría de los esfuerzos se han puesto en la adaptación de algoritmos ya existentes con el objeto de poder tratar varias etiquetas por instancia, pero sin tener en cuenta todas las características de este problema y la necesidad de un tratamiento más especifico, con capacidad para abordarlo de manera integral.

Asimismo existen múltiples propuestas [5–7] de solución a este problema basadas en el uso de *ensembles* de clasificadores binarios o multiclase. Como ocurre en clasificación tradicional, en general los *ensembles* exhiben un mejor comportamiento global que los clasificadores individuales.

2 Hipótesis de partida

Un primer paso para abordar con éxito el problema de la clasificación multietiqueta consiste en extraer las principales características que presenta, planteando un conjunto de hipótesis sobre ellas.

Dimensionalidad: Una primera peculiaridad de los datasets multietiqueta es la alta dimensionalidad que presentan. Así, si se analizan conjuntos de datos multietiqueta [8] se puede observar la gran cantidad de instancias y de características que contienen. Esta dimensionalidad se ve incrementada por el hecho de disponer cada instancia de múltiples etiquetas (miles en ocasiones), una característica especifica, inexistente en clasificación tradicional, que suscita una hipótesis: la aplicación de una técnica de selección de etiquetas podría permitir reducir el espacio de salida y, con ello, mejorar la efectividad de los clasificadores.

Desbalanceo: En segundo término es necesario considerar el desbalanceo, extremo en muchos casos, en la representación que tiene cada etiqueta en este tipo de conjuntos de datos [9]. El desbalanceo en clasificación binaria es un problema profusamente estudiado y es conocida su influencia en el comportamiento de los clasificadores. En el contexto de la clasificación multietiqueta este problema adquiere mayor complejidad, al existir múltiples etiquetas que pueden aparecer conjuntamente en las instancias y con niveles de representación muy distintos. Se presenta la hipótesis de si podrían mejorarse los resultados a través del preprocesamiento de estos datasets, con técnicas de eliminación de instancias y sobremuestreo diseñadas para el campo multietiqueta.

Dependencias entre etiquetas: Otro aspecto a tener en cuenta en este tipo de clasificación es la correlación que suele existir entre etiquetas o, lo que es lo mismo, la probabilidad de que un cierto conjunto de dos o más etiquetas aparezca al presentar distintos conjuntos de entradas al clasificador. Si las etiquetas de un dataset multietiqueta fuesen totalmente independientes entre sí, seria posible dividirlo en un conjunto de datasets binarios y obtener la predicción para cada una de ellas por separado sin pérdida de rendimiento, pero se sabe que esto generalmente no es así [10]. Un clasificador multietiqueta, por tanto, debería diseñarse bajo la hipótesis de la existencia de dependencias entre las etiquetas, extrayendo y usando dicha información a fin de obtener mejores resultados.

3 Objetivos

Tal y como se ha mencionado, los conjuntos multietiqueta se caracterizan por su alta dimensionalidad en un nuevo espacio, el de etiquetas, así como por presentar en general un alto nivel de desbalanceo entre etiquetas y la existencia de dependencias entre estas. Todos ellos son aspectos que habrá que tener en cuenta para diseñar soluciones que aborden este tipo de problemas. En consecuencia, para mejorar la calidad de los modelos obtenidos han de seguirse básicamente dos caminos, no excluyentes entre sí: el preprocesamiento de los datos, a fin de reducir la dimensionalidad y corregir en la medida de lo posible el desbalanceo, y la optimización del algoritmo de aprendizaje aprovechando la información de dependencia entre etiquetas.

El objetivo general es conseguir, utilizando técnicas de computación flexible, nuevos modelos que afronten el problema de clasificación multietiqueta en diferentes fases. Para ello dicho objetivo se dividirá en los hitos siguientes:

- **Reducción de la dimensionalidad en el espacio de etiquetas**: Que la reducción del espacio de salida puede redundar en una mejora de rendimiento en clasificación y eficiencia del clasificador es algo probado [11], pero bajo premisas muy concretas de presencia de dispersión que no siempre se dan. En el campo de la computación flexible encontramos alternativas a las técnicas usadas hasta ahora, basadas en lógica nítida y lógica difusa, sistemas asociativos y métodos probabilísticos que permitirían inferir unas etiquetas a partir de la presencia de otras. Con ellas seria posible aportar un método menos restrictivo para reducir el espacio de salida con el que ha de trabajar el clasificador, aplicable de manera general a datasets multietiqueta.
- **Reducción del nivel de desbalanceo**: Las propuestas [9] existentes en la literatura se ciñen a la adaptación de algoritmos de clasificación específicos y se usan para solventar casos muy concretos, asumiendo como un hecho la existencia de desbalanceo en todo dataset multietiqueta pero sin realizar una medición que lo confirme. Es necesario identificar cuál es el nivel de desbalanceo en este tipo de datasets, aportando una o más medidas adaptadas al problema multietiqueta, y ofrecer algoritmos de preprocesamiento específicos con capacidad para equilibrar el nivel de representación de las etiquetas en cualquier dataset de este tipo. De esta forma se permitirá el posterior uso del clasificador multietiqueta que mejor se adapte a cada problema concreto, sin estar sujetos a un algoritmo concreto.
- **Clasificación aprovechando la información de dependencia entre etiquetas**: La información sobre dependencias entre etiquetas ha sido utilizada [5] principalmente en algoritmos de clasificación multietiqueta basados en mecanismos de transformación de los datos, pero raramente como elemento fundamental en el diseño de un clasificador multietiqueta. Hasta el momento no existe una definición formal sobre la interpretación de esta información de correlación [12], ni un mecanismo de referencia sobre cuál sería su utilización. La tercera aportación consistirá en diseñar un clasificador multietiqueta que, mediante la hibridación de modelos de reglas y de

RBFNs, tenga en cuenta la correlación entre etiquetas. Las redes neuronales de funciones de base radial destacan por su eficiencia en la obtención de resultados, así como por la interpretabilidad de los modelos obtenidos. La información de dependencia puede ser aprovechada en la fase de diseño de la red a fin de mejorar su rendimiento en clasificación.

4 Metodología y plan de trabajo

La consecución de los objetivos propuestos exige la definición de una metodología y un plan de trabajo en el que se entremezclan hitos teóricos con otros de tipo práctico, como variante del método científico clásico. La metodología se compondra de los hitos siguientes:

- Revisión bibliográfica del estado del arte relativo a la clasificación multietiqueta, de métodos de reducción de dimensionalidad y reducción de datos, y de técnicas de computación flexible.
- Análisis de la influencia de aspectos característicos de los datasets multietiqueta en la distribución de los datos.
- Diseño de nuevos modelos para las diferentes fases que determinan el problema de clasificación multietiqueta.
- Obtención de observaciones mediante la aplicación de los nuevos modelos a conjuntos de datos estándar, evaluando la efectividad, el rendimiento y la interpretabilidad de los resultados.
- Contraste de la hipótesis mediante la comparación de los resultados obtenidos con los generados por los métodos mas relevantes descritos en la bibliografíay readaptación de la hipótesis, si procede, a la vista del contraste realizado previamente si este exige la modificación del modelo.

En cuanto al plan de trabajo, parte de los objetivos enumerados en la sección 3 han sido ya completados con los siguientes resultados:

- **Dimensionalidad en el espacio de etiquetas**: Se ha diseñado un modelo de reducción de la dimensionalidad en el espacio de etiquetas basado en la obtención de reglas de asociación fuertes. Mediante un algoritmo de minería de reglas de asociación se analiza la correlación entre etiquetas, obteniendo un conjunto de reglas fuertes que permiten la eliminación de un subconjunto de etiquetas, el entrenamiento de un clasificador multietiqueta con ese dataset simplificado y su posterior recuperación mediante inferencia. El modelo ha sido probado experimentalmente con múltiples datasets y algoritmos de clasificación multietiqueta obteniendo buenos resultados.
- **Corrección del nivel de desbalanceo entre etiquetas**: Se ha definido un conjunto de medidas específicas para determinar el nivel de desbalanceo en datasets multietiqueta. También se han diseñado dos métodos de preprocesamiento básicos para la eliminación y generación de instancias de manera aleatoria que han arrojado resultados interesantes.

El resto de los objetivos se obtendran de acuerdo a la siguiente planificación:

1. Está finalizándose el diseño de un algoritmo de generación de instancias sintéticas más sofisticado, basado en las características de los datasets multietiqueta, cuyo objetivo es facilitar su preprocesamiento para corregir el desbalanceo. Este nuevo método no se limita a replicar muestras seleccionadas aleatoriamente, sino que utiliza información sobre la distribución de las etiquetas en el dataset para producir nuevas instancias de etiquetas minoritarias.
2. A continuación se evaluará experimentalmente la influencia de dicho método de preprocesamiento en los resultados de clasificación, empleando para ello todos los datasets disponibles públicamente y analizando su influencia en múltiples algoritmos de clasificación multietiqueta.
3. Se analizarán y evaluaran distintos mecanismos para integrar la información de dependencia entre etiquetas, obtenida mediante reglas de asociación como en los pasos previos o bien por otros medios, en la fase de diseño de un clasificador multietiqueta.
4. Finalmente, se desarrollará un nuevo algoritmo de clasificación multietiqueta basado en el mejor mecanismo determinado en el paso previo y se procederá a su evaluación experimental respecto a otras propuestas existentes. Por su eficacia como métodos de aproximación universal y su respuesta local, las redes neuronales basadas en RBF pueden resultar la herramienta ideal para este tipo de tarea.

5 Relevancia

El número de publicaciones relativas a clasificación multietiqueta ha crecido de manera muy importante en los últimos años, reflejando así el creciente interés por aportar soluciones a este problema que cada vez encuentra más aplicaciones en el mundo real.

Desde una perspectiva científica, las aportaciones descritas en el apartado 3 son relevantes en cuanto abordan aspectos poco estudiados hasta el momento en este campo, como son la reducción de la dimensionalidad en el espacio de salida o el tratamiento del desbalanceo en datasets multietiqueta, con el objetivo de ofrecer soluciones suficientemente genéricas como para que puedan ser utilizadas sin que importe la aplicación concreta o el clasificador que mejor se adapte a dicha aplicación. Las propuestas existentes en la actualidad están, en general, fuertemente ligadas a su uso en un clasificador concreto o para resolver una problemática especifica. Incorporando la información de dependencia entre etiquetas en la fase de diseño del clasificador se espera optimizar el rendimiento de este, obteniendo un mejor rendimiento respecto a propuestas existentes.

La aplicación de estas propuestas en entornos reales es prácticamente inmediata, pudiendo contribuir a la mejora de sistemas de clasificación automática de páginas web, imágenes, textos y, en general, todo tipo de recursos electrónicos para cuya categorización se requiere un mecanismo capaz de asignar múltiples etiquetas no excluyentes entre si.

Bibliografía

1. G. Tsoumakas, I. Katakis, I. Vlahavas, Mining Multi-label Data, in: O. Maimon, L. Rokach (Eds.), Data Mining and Knowledge Discovery Handbook, Springer US, Boston, MA, 2010, Ch. 34, pp. 667–685.
2. M.-L. Zhang, Multilabel Neural Networks with Applications to Functional Genomics and Text Categorization, IEEE Trans. Knowl. Data Eng. 18 (10) (2006) 1338–1351.
3. M. Boutell, J. Luo, X. Shen, C. Brown, Learning multi-label scene classification, Pattern Recognit. 37 (9) (2004) 1757–1771.
4. A. Wieczorkowska, P. Synak, Z. Raś, Multi-Label Classification of Emotions in Music, in: Intelligent Information Processing and Web Mining, Vol. 35, 2006, Ch. 30, pp. 307–315.
5. J. Read, B. Pfahringer, G. Holmes, E. Frank, Classifier chains for multi-label classification, Mach. Learn. 85 (2011) 333–359.
6. G. Tsoumakas, I. Vlahavas, Random k-labelsets: An ensemble method for multilabel classification, in: J. Kok, J. Koronacki, R. Mantaras, S. Matwin, D. Mladenič, A. Skowron (Eds.), Machine Learning: Proc. ECML 2007, Vol. 4701 of LNCS, 2007, pp. 406–417.
7. G. Tsoumakas, I. Katakis, I. Vlahavas, Effective and Efficient Multilabel Classification in Domains with Large Number of Labels, in: Proc. ECML/PKDD Workshop on Mining Multidimensional Data, 2008, pp. 30–44.
8. G. Tsoumakas, E. S. Xioufis, J. Vilcek, I. Vlahavas, MULAN multi-label dataset repository.
   URL `http://mulan.sourceforge.net/datasets.html`
9. J. He, H. Gu, W. Liu, Imbalanced multi-modal multi-label learning for subcellular localization prediction of human proteins with both single and multiple sites, PloS one 7 (6) (2012) 7155.
10. L. Tenenboim-Chekina, L. Rokach, B. Shapira, Identification of label dependences for Multi-Label classification, in: Proc. 2nd Int. Workshop on Learning from Multi-Label Data (MLD'10), 2010, pp. 53–60.
11. D. Hsu, S. Kakade, J. Langford, T. Zhang, Multi-label prediction via compressed sensing, in: Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, A. Culotta (Eds.), Advances in Neural Information Processing Systems 22, 2009, pp. 772–780.
12. M.-L. Zhang, Z.-H. Zhou, A review on multi-label learning algorithms, IEEE Transactions on Knowledge and Data Engineeringdoi:10.1109/TKDE.2013.39.

Moderación Automática de Comentarios y Reputación de Usuarios en Sitios Web Sociales

Doctorando: Jorge de la Peña Sordo
jorge.delapenya@deusto.es
Directores: Dr. Igor Santos Grueiro y Dr. Pablo García Bringas
{isantos,pablo.garcia.bringas}@deusto.es

S^{3}Lab, DeustoTech Computing, Universidad de Deusto
Avenida de las Universidades 24, 48007, Bilbao, España

Keywords: Aprendizaje automático, recuperación de información, data mining, web mining, categorización de texto, procesado lenguaje natural.

1. Introducción

La generación de contenidos por parte de los usuarios supone uno de los mayores valores que la era de la información, tal y como se la conoce hoy en día, aporta a la sociedad. De esta manera, los antiguos portales web, en los que existían un plantel de profesionales que generaba el contenido, se han convertido en plataformas de difusión de la creatividad de los usuarios, los cuales generan contenidos de manera cada vez más profesional. El crecimiento de este tipo de plataformas, organizado en torno a comunidades, es evidente. Prueba de ello es el crecimiento que distintas redes sociales y comunidades han sufrido últimamente.

En particular, diversos sitios web sociales de noticias como *Digg*[1] o *Menéame*[2] son muy populares en este ámbito. Estos sitios trabajan de manera sencilla e intuitiva: los usuarios envían enlaces de artículos en línea, y otros usuarios de este sistema los valoran positiva o negativamente mediante votaciones. Los artículos más votados favorablemente aparecen, finalmente, en la portada [1].

Esta tesis se enfoca en la web *Menéame*, la cual ya dispone de un método para moderar y así poder filtrar automáticamente comentarios e historias; sin embargo, se basa en los votos de otros usuarios y, por lo tanto, no puede ser objetivo, con lo que podrían aparecer grupos de presión y usuarios o grupos trol desvirtuando el sistema. En la misma línea, hay enfoques para filtrar *spam* en comentarios [2,3], donde se propone un método basado en diversas opiniones y características sintácticas para filtrar automáticamente mensajes de *spam* en comentarios, dentro del sitio web de *Amazon*[3]. Para evitar estos problemas, en el contexto de la categorización de usuarios, se han realizado diferentes trabajos empleando las características de las pulsaciones del usuario en el teclado [4] o su

[1] http://digg.com
[2] http://www.meneame.net
[3] http://www.amazon.com

actividad en los registros [5,6]. Sin embargo, no existen trabajos específicos que se encarguen de la problemática de las noticias sociales y sus peculiaridades.

Por ello, esta investigación busca mitigar el salto existente entre la plataforma y el usuario, ya que el volumen de contenido generado en ciertas Websites no permite que se pueda hacer un control manual. Por lo tanto, se propone un sistema de moderación automática y escalable que permita a la plataforma tener una mejor información sobre sus usuarios y sobre el uso que hacen de la plataforma. En el mismo sentido, los usuarios estarán más protegidos de agravios y vejaciones, lo que hará que su estancia en la plataforma sea más placentera.

Para su realización se diferencian 4 tareas principalmente: el estudio de la problemática, en la cual se contextualiza el problema. Como siguiente tarea, la categorización de comentarios; entrenamiento y sistema de categorización. La categorización de usuarios como tercera tarea, haciendo uso de las características de los perfiles de los mismos, se realizará un sistema que los clasifique. Por último, se generará un sistema para controlar y predecir la reputación y el comportamiento del usuario.

Así, esta investigación busca paliar uno de los problemas derivados del uso de las comunidades en Internet: la gestión responsable de contenido que los usuarios suben a la misma. Dado el carácter prematuro de la tecnología, es una oportunidad única de contribuir al desarrollo de estas comunidades y fomentar el aprovechamiento de la sociedad de la información por parte de aquella parte de la ciudadanía que corre peligro de sufrir una exclusión social digital.

2. Hipótesis

La creación de contenido en Internet ha venido sufriendo una rápida transformación en cómo se genera el contenido. Si bien antaño el webmaster era el único responsable del contenido de su sitio, a día de hoy, el contenido es generado también por los usuarios del sitio. En muchos casos, ésto puede suponer un problema para los dueños legítimos del sitio, ya que resulta laborioso hacer una moderación manual de todos los contenidos que los usuarios aportan.

En concreto, existen sitios web, en los que gran parte del contenido es generado por los usuarios. Dentro de este tipo de sistemas están las redes sociales como *Facebook*[4] o *Tuenti*[5], las cuales buscan potenciar la comunicacin entre los usuarios. Otro ejemplo, son los sitios de noticias sociales como *Digg* o su análogo español *Menéame*. Por último, existe una gran cantidad de formas de promocionar la imagen de marca de las empresas a través de estos sistemas. *Tuenti* ofrece las páginas, que permiten que las entidades que lo contraten tengan un espacio privilegiado dentro de la propia red social, permitiendo la creación de un espacio de la empresa con mucha visibilidad, a la que ya se han adherido empresas como *Coca-Cola*, *HP* o *Movistar*. En la misma medida, existen las páginas de *Facebook*, que permiten a cualquier empresa crear una página, dotándole de una

[4] `http://www.facebook.com`
[5] `http://www.tuenti.es`

serie de herramientas para realizar la monitorización del uso que de la misma hacen los usuarios.

Por otro lado, se encuentran otros servicios sociales como los denominados sistemas de recomendación. Estos sistemas permiten obtener a los usuarios, de mano de otros miembros de la comunidad, valoraciones sobre distintos productos. Este tipo de sistemas, como por ejemplo *Stumbleupon*[6] con un carácter más general o *Minube*[7] orientando a la generación de una comunidad de viajes, tienen un poder de influencia en los usuarios, repercutiendo de forma muy directa en la reputación de la empresa.

Sin embargo, a diferencia de lo que ocurría anteriormente, el control que tiene la empresa sobre estos contenidos es completamente nulo, ya que son los usuarios y consumidores los que generan estos contenidos. A la luz de estos acontecimientos, se constituye la hipótesis fundamental de partida sobre la cual se basa esta tesis:

> "*Es posible determinar, gestionar y predecir el comportamiento de los usuarios en diferentes plataformas web mediante el empleo de minería de opiniones, técnicas de aprendizaje automático, métodos de recuperación de información y series temporales.*"

3. Objetivos

Con la hipótesis de partida planteada, hemos procedido a identificar el objetivo principal de esta investigación:

Objetivo principal 1 *Desarrollar un sistema gestor y predictor de reputaciones de usuarios en sitios web sociales mediante la categorización automática de usuarios y sus comentarios.*

Este objetivo principal, a su vez, puede ser expresado en diferentes objetivos específicos, con el propósito de cumplir tal objetivo principal:

Objetivo específico 1 *Desarrollar un método para categorizar comentarios publicados por usuarios en sitios web sociales.*

Objetivo específico 2 *Diseñar y desarrollar nuevas técnicas para la clasificación y seguimiento de usuarios.*

Objetivo específico 3 *Implementar métodos eficaces para la gestión de la reputación de los usuarios y la posterior predicción de su comportamiento.*

[6] `http://www.stumbleupon.com`
[7] `http://www.minube.com`

El primer objetivo incluye todas aquellas acciones que el sistema necesita para la categorización de comentarios, incluyendo el entrenamiento del módulo y un sistema de categorización. En cuanto al segundo objetivo, utilizando las características de los perfiles de los usuarios y en función de los comentarios que éstos publican, se realizará un sistema que clasifique a los usuarios en alguna de las categorías establecidas previamente. Finalmente, en el tercer objetivo específico, se generará un sistema para la gestión de la reputación de los diferentes usuarios, el cual pueda predecir comportamientos futuros de los mismos.

4. Metodología y Plan de trabajo

La coordinación y control técnico y de calidad se realizará durante todas las fases que comprende el trabajo. Pese a ello, se define una fase de trabajo inicial en la que se planificarán los aspectos más importantes del mismo. A continuación se detallan las fases o hitos en los que se ha planificado la investigación. Al tratarse de un sistema complejo, se ha estructurado de la siguiente manera:

- **Fase 1: Estudio de la problemática.**
  En esta fase de trabajo, se contextualizará el problema, identificando los diferentes hitos y su importancia, además de realizar estudios teóricos sobre las diversas técnicas para realizar los objetivos marcados en la investigación. Los objetivos de esta fase de trabajo son:
  1. Determinar el caso de estudio sobre el que construir el sistema.
  2. Estudiar las diferentes técnicas para el desarrollo del trabajo.
  3. Recopilar y etiquetar los datos.
- **Fase 2: Categorización de comentarios.**
  En esta fase, se desarrollará un prototipo para la categorización automática de comentarios. Las categorías que se han valorado van en función de tres diferentes informaciones para el administrador del sitio o *community manager*. En primer lugar, se diseñará un sistema para saber si un comentario hace referencia a la información del sitio o a otro usuario. En segundo lugar, se evaluará la posibilidad de evaluar el grado de aporte del comentario. Por último, se ha considerado que puede resultar de utilidad, el establecer el grado de polémica que ese comentario suscita. A pesar de ser un sistema de filtrado, se ha valorado la opción de dejar abiertos los comentarios a filtrar y que el administrador del sitio o *community manager*, sea quien decida que comentarios son o no válidos para su sitio. Los objetivos de esta fase de trabajo son:
  1. Desarrollar un sistema para la categorización automática de comentarios en sitios sociales.
  2. Establecer las diferentes categorías para este sistema.
- **Fase 3: Categorización de usuarios.**
  Esta fase de trabajo está orientada a la categorización y el seguimiento de los usuarios. Para ello, es importante catalogar correctamente los comentarios que éstos han realizado. Las categorías de los usuarios vienen a informar

al administrador del sitio diversas acciones que el usuario realiza o no. Por ejemplo, si el usuario aporta o no, si el usuario comenta aportes, si el usuario difunde aportaciones de otros, si el usuario es comunicativo con otros usuarios, si el usuario acosa o no a otros usuarios, etc. Utilizando esta información, se modelará un comportamiento base del usuario que será mejorado como reputación en la fase 4. En concreto, esta fase tiene los siguientes objetivos:

1. Implementar un nuevo método de categorización de usuarios.
2. Desarrollar una nueva técnica para el seguimiento de los usuarios.

- **Fase 4: Reputación y predicción del comportamiento de los usuarios.**
  En esta fase se trabajará en la reputación del usuario, con el objeto de predecir cuáles son las acciones futuras (comentarios) de los usuarios. Partiendo de la información sobre los usuarios que el sistema de la fase de trabajo anterior ofrece, se establecerán ciertas métricas para obtener así un modelo de reputación del usuario. Este modelo de reputación servirá para saber cómo afectan los usuarios a los sitios en los que comentan y aportan. Además, la reputación, tanto de los usuarios como de los sitios no es estática. Por tanto, en este punto se modelará, de un modo temporal, los comentarios emitidos por los usuarios y la reputación asociada a los mismos. Los objetivos de esta fase son los siguientes:
  1. Adecuar los métodos de time-series para la predicción temporal.
  2. Realizar una prueba de concepto sobre predicción de acciones futuras de usuarios.
  3. Generar un modelo de reputación para los usuarios.
- **Fase 5: Difusión de los resultados.**
  En esta fase de la investigación se difundirán los resultados del trabajo realizado en importantes foros internacionales como conferencias relevantes y revistas de alto impacto. Los objetivos de esta fase de trabajo son los siguientes:
  1. Identificar los diferentes foros de difusión tanto desde el punto de vista académico como desde punto de vista divulgativo.
  2. Difundir la investigación realizada, maximizando su impacto en la sociedad de la información.

5. Relevancia

La red de redes, Internet, es cada vez mucho más social. De hecho, se lleva ya unos cuantos años dentro del paradigma conocido como la Web 2.0. El término está asociado a aplicaciones Web que facilitan el compartir información, la interoperabilidad, el diseño centrado en el usuario [7] y la colaboración en la World Wide Web. Un sitio Web 2.0 permite a los usuarios interactuar y colaborar entre sí como creadores de contenido generado por usuarios en una comunidad virtual, a diferencia de sitios web donde los usuarios se limitan a la observación pasiva de los contenidos que se ha creado para ellos. De esta forma, ya desde tiempo atrás, este tipo de forma de observar la Web viene creciendo de forma desmesurada.

A pesar de que el mundo se mueve en una época en la que el contenido multimedia está comenzando a ser muy importante, el contenido textual sigue siendo mucho mayor que el anterior. Además, todos los Websites con generación de contenido social están totalmente repletos de comentarios, ideas y opiniones de las diferentes personas que pasaron por el sitio web social. Estos comentarios pueden ser tanto benignos como malignos. Pero incluso, los comportamientos de los usuarios pueden ser puntuales o sobre un tema en concreto, o incluso pueden ser mucho más dirigidos.

Al igual que sucede en las relaciones sociales, en el mundo real se pueden encontrar el mismo tipo de comportamientos que en la red, con lo que puede haber problemas de reputación y de comportamientos ilícitos. Es por ello, que a pesar de que Internet es una red totalmente libre, no se puede vulnerar a las personas. Por lo tanto, debido a la problemática comentada que se basa fundamentalmente en el crecimiento de la compartición de contenido en la Web y a que cada vez hay un mayor número de personas que se encargan de generar dicho comportamiento, se necesitan herramientas que sean capaces de detectar las situaciones extremas y que permitan llevar un control automatizado sobre lo que está sucediendo con el contenido y los usuarios en el respectivo sitio Web.

Por otra parte, el volumen de contenido generado existente en ciertos Websites no permite que se pueda hacer un control manual. Es por eso, que en este punto entra a jugar la propuesta de tesis denominada como Moderación Automática de Comentarios y Reputación de Usuarios en Sitios Web Sociales. Así, se intenta desarrollar un sistema que sea capaz de realizar la categorización automática de usuarios y sus comentarios, mediante el empleo de algoritmos de aprendizaje automático supervisado y la predicción de series temporales, para crear un sistema que gestione la reputación de éstos en sitios web sociales.

Referencias

1. Lerman, K.: User participation in social media: Digg study, in: Proceedings of the 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Workshops, IEEE Computer Society, 2007, pp. 255–258.
2. Jindal, N.—Liu, B.: Review spam detection, in: Proceedings of the 16th international conference on World Wide Web, ACM, 2007, pp. 1189–1190.
3. Jindal, N.—Liu, B.: Opinion spam and analysis, in: Proceedings of the international conference on Web search and web data mining, ACM, 2008, pp. 219–230.
4. Ypma, A.—Heskes, T.: Automatic categorization of web pages and user clustering with mixtures of hidden markov models, WEBKDD 2002-Mining Web Data for Discovering Usage Patterns and Profiles (2003) 35–49.
5. Elahi, S.—dAquin, M.—Motta, E.: Who wants a piece of me? reconstructing a user profile from personal web activity logs, in: International ESWC Workshop on Linking of User Profiles and Applications in the Social Semantic Web, 2010.
6. Nasraoui, O.—Soliman, M.—Saka, E.—Badia, A.—Germain, R.: A web usage mining framework for mining evolving user profiles in dynamic web sites, Knowledge and Data Engineering, IEEE Transactions on 20 (2) (2008) 202–215.
7. Características de la Web 2.0: http://www.techpluto.com/web-20-services/

Sistema multiagente para la asignación de recursos en entornos distribuidos

Doctorando: Fernando De la Prieta[1]
Directores: Javier Bajo Pérez[2] y Sara Rodríguez González[1]

[1] Universidad de Salamanca, Departamento de Informática y Automática
Plaza de la Merced s/n, Salamanca – España
[2] Universidad Politécnica de Madrid, Departamento de Inteligencia Artificial
Bloque 2, Despacho 2101, Campus Montegancedo, Boadilla del Monte, Madrid – España
{fer, srg}@usal.es ; {jbajo@fi.upm.es}

Keywords. Cloud Computing, Multi-agent system, negociación

1 Introducción

La industria tecnológica está haciendo un gran esfuerzo para desarrollar el paradigma Cloud Computing[1]. El número de plataformas de Cloud Computing tanto privadas, como públicas está creciendo rápidamente [4]. El rápido crecimiento está motivado por los intereses económicos que subyacen alrededor de las características puramente computacionales o tecnológicas [6]. Las plataformas Cloud desde un punto de vista externo, ofrecen tres tipos de servicios ampliamente conocidos (Software, Plataforma e Infraestructura). Y, desde un punto de vista interno, los servicios que ofrecen se consideran elásticos, gracias a que las tecnologías subyacentes (virtualización, granjas de servidores, servicios web, etc.) han alcanzado su madurez.

La mayoría de las plataformas Cloud Computing conocidas hacen un gran énfasis en proporcionar un servicio elástico de infraestructura (a través de la virtualización del hardware) [4], sin tener en cuenta los servicios de nivel superior como plataforma y software. Éstas plataformas tampoco hacen un énfasis especial en la optimización de los recursos hardware. Esta limitación en las plataformas actualmente existentes, que podemos denominar de primera generación, ha constituido una barrera para su implantación. Incluso, dada la popularidad que ha alcanzado el término, se está haciendo popular el término "*fake Cloud Computing*"[2].

En este sentido, esta Tesis se plantea como una de las primeras aproximaciones de los Sistemas Multiagente basados en Organizaciones Virtuales [12] en el desarrollo de una plataforma Cloud Computing completa y funcional. A lo largo del desarrollo, se investigará en como gestionar automáticamente las necesidades computacionales, es decir, la infraestructura hardware subyacente, para satisfacer la demanda de servicios a diferentes niveles

[1] http://searchcloudcomputing.techtarget.com/photostory/2240149038/Top-10-cloud-providers-of-2012/1/Introduction

[2] It's hosting, dammit: Fed up with fake cloud providers. In: InfoWorld. http://www.infoworld.com/d/cloud-computing/its-hosting-dammit-fed-fake-cloud-providers-206384

2 Hipótesis inicial

A nivel académico, la mayoría de trabajos existentes en el estado del arte están relacionados con el uso ilimitado de recursos que disponen las plataforma Cloud Computing comerciales, más que en el desarrollo de técnicas y herramientas específicamente orientadas a la mejora del paradigma en su conjunto. Entre los trabajos que se pueden encontrar en este segundo grupo, destacan los trabajos relacionados con la provisión de recursos de infraestructura [4], o el desarrollo de algún servicio específico en Cloud, los ampliamente denominados *cloud-based services* [5]. En conclusión, el desarrollo de los entornos Cloud ha estado liderado por el sector empresarial. Por ello, una vez que las plataformas cloud de primera generación están ampliamente implantadas, es el momento de que el entorno académico dirija sus esfuerzos al desarrollo de la siguiente evolución de este tipo de plataformas.

En esta línea, el principal objetivo de este trabajo es la **investigación y desarrollo de un sistema multiagente basado en organizaciones virtuales para la asignación dinámica de recursos en entornos Cloud**. Pese a que hasta el momento los agentes inteligentes no han tenido un peso destacado en el entorno Cloud, e incluso a priori los enfoques de ambos paradigmas pueden parecer opuestos, un análisis más detallado demuestra que ambos paradigmas son complementarios, siendo posible identificar un buen número de sinergias. Mientras que por un lado, los entornos Cloud Computing puede cubrir las necesidades computacionales de persistencia de información y el potencial de computo que requieren los MAS en diferentes aplicaciones como minería de datos, gestión de servicios complejos, etc. Por otro lado, los MAS pueden ser utilizados para el diseño de entornos Cloud Computing mucho más eficientes, escalables y adaptables que los existentes actualmente. El uso de MAS en el diseño de sistemas del núcleo de sistemas Cloud Computing permite crear plataformas cloud con características inteligentes como pueden ser la autonomía o el aprendizaje, entre otras.

3 Objetivos

Uno de los primeros retos que surge en este trabajo es integrar un MAS en un entorno Cloud, incorporando las características de estos tipos de sistemas (organizaciones, comunicación, normas, etc.), pero manteniendo la alta disponibilidad. En el marco del proyecto de tesis, se fija como objetivo la obtención de una plataforma de agentes para su uso en entornos de alta disponibilidad.

Un entorno Cloud Computing se caracteriza por su infraestructura y los servicios que se ofrecen. Mientras que la infraestructura aporta recursos computacionales al entorno, los servicios consumen estos recursos. El principal reto radica en el hecho en que se debe mantener el nivel de calidad (QoS) de los servicios con independencia de su demanda, lo que implica que sea necesario gestionar dinámicamente los recursos computacionales asignados a cada servicio. En este sentido, el estado del arte presenta trabajos en los que la toma de decisiones se realiza de (i) forma centralizada [7]; o (ii) no se tiene en cuenta la demanda del servicio [2]; o (iii) se estimar el uso [11]; o (iv) simplemente se planifica los recursos [8]. Por lo que, en cualquiera de los

caso, actualmente no existe una solución óptima al problema de reasignación de recursos, y mucho menos que ésta sea eficiente y ahorre costes. Este hecho hace que la obtención de una arquitectura multi-agente basada en organizaciones virtuales, con capacidades de reorganización y adaptación, resulte especialmente adecuada para ser aplicada en estos entornos.

Desde una perspectiva formal, el problema se plantea desde dos puntos de vista, una vista *micro*, en la que la reasignación de recursos se realiza en cada servicio o dentro de un servidor físico; y a nivel *macro*, reasignación de recursos a nivel global del entorno cloud, lo que podría implicar la migración de máquinas virtuales, encendido o apagado de servidores físicos, etc. El problema se puede modelar en términos de organizaciones virtuales de agentes inteligentes que se auto-adaptan y reorganizan en función de las necesidades del contexto, donde cada agente gestiona recursos o servicios y se coordina con el resto de agentes de la organización.

En primer lugar, a nivel *micro* existen diferentes trabajos que satisfacen el problema, ya sea la reasignación de recursos entre las máquinas virtuales que aloja un servidor [10], o el balanceo entre las diferentes máquinas virtuales asignadas a un recurso. En el marco del trabajo, se propone como sub-objetivo el estudio, uso y optimización de los algoritmos ya existentes para alcanzar cotas de eficiencia mayores, a la vez que se reducen los tiempos y recursos computacionales del propio algoritmo.

En segundo lugar, a nivel *macro* no existen referencias relevantes en el estado del arte, más allá de los algoritmos comentados previamente. Para solucionar este problema abierto, se propone que los diferentes agentes que formen el sistema dialoguen, y compartan información, y lleguen a acuerdos sobre el reparto de los recursos existentes en el entorno Cloud. Siguiendo esta metodología es posible distribuir la toma de decisiones a lo largo del entorno Cloud, reduciendo los cuellos de botella de los algoritmos centralizados, y al mismo tiempo haciendo posible que las decisiones las tomen aquellos componentes que necesitan los recursos computacionales. En este sentido, los primeros trabajos, ya se han realizado con resultados satisfactorios [9], por lo que se plantea como subobjetivo en el marco del trabajo de tesis seguir los esfuerzos realizados en esta línea de investigación.

Finalmente, el último subobjetivo de la tesis será aislar la toma de decisiones de la infraestructura subyacente ya que los entornos cloud son altamente heterogéneos. Utilizando el concepto de artefacto propuesto por Ricci et al. [1] es posible aislar el sistema multiagente (toma de decisiones) de la heterogeneidad del sistema, lo que en si mismo constituye una gran ventaja con respecto a los sistemas actuales que dependen en gran medida de los entornos de virtualización (KVM, Xen, OpenVZ, etc.).

4 Metodología y Plan de Trabajo

En una Tesis donde el principal componente de trabajo está relacionado con la investigación, parece adecuado plantear una metodología que permita establecer objetivos alcanzables en un tiempo fijado, evaluar los resultados de forma conjunta entre el doctorando y los tutores, para posteriormente discutir el siguiente nivel de objetivos atendiendo a los resultados obtenidos.

Para ello se plantea una organización basada en una indagación original planificada, realizando una revisión del estado del arte, planteando novedades objetivas sobre el estado del arte, diseñando y desarrollando un nuevo sistema, que será evaluado en casos de estudio reales. Dado que se espera obtener una arquitectura software, el desarrollo de prototipos en los que se incorporen los objetivos perseguidas gradualmente resulta fundamental Esto permitirá alcanzar la solución en el tiempo estimado. Los aspectos que requieran la evaluación de técnicas, así como la propuesta de alternativas adecuadas al tipo de problema que se aborda, se plantean en paralelo y serán incorporadas en el siguiente prototipo.

El método de investigación que se seguirá es conocido como *action−research*, uno de los más habituales en el ámbito de los sistemas de información y las TIC. Este método se caracteriza por estar orientado a la acción y al cambio, focalizarse en un problema y Generar conocimiento de investigación relevante.

Los hitos que se plantean en la presente tesis son los siguientes:

- *Revisión detallada del estado del arte e identificación de novedades*.
- *Diseño de un entorno de pruebas.* Este hito ya se ha superado, y aunque se podría haber utilizado un entorno de simulación, como por ejemplo simcloud[3] , se ha preferido el desarrollo de una plataforma propia: +Cloud.
- *Inclusión de los principios del paradigma de agentes en el entorno Cloud.*
- *Inclusión del concepto de entorno y uso de artefactos.*
- *Diseño de un sistema multiagente específico.* Ya se disponen de resultados iniciales [3].
- *Investigación y mejora de algoritmos locales de asignación de recursos.*
- *Investigación y mejora de algoritmos de asignación de recursos a nivel global*. que como ya se ha enunciado se han realizado algunos trabajos iniciales [9].
- *Integración y evaluación de los algoritmos propuestos*.

No cabe duda que para el desarrollo de un entorno Cloud Computing es necesario disponer de un entorno de pruebas adecuado. Así, en este sentido, el grupo de investigación BISITE de la Universidad de Salamanca ha desplegado una completa granja de servidores de última generación, con capacidad de virtualización por hardware y 16Gb de RAM por equipo. Todos ellos conectados punto a punto mediante una red 10Gbps. Gracias a esta infraestructura será posible validar todos las innovaciones que se obtengan en el marco de este proyecto de Tesis.

5 Relevancia

En cuanto al impacto en la sociedad, la gestión de la información y el trato con el cliente resultan fundamentales, ya que es quien mayores beneficios puede extraer del avance del Cloud Computing. La nube puede tener un impacto muy importante en la economía española, cimentada en el sector servicios, que representa el 50% del Producto Interior Bruto y el 43% de la población ocupada, según los últimos datos del

[3] http://simcloud.com/

INE (Instituto Nacional de Estadística). El paradigma resulta especialmente atractivo para las pymes. Éstas no disponen de los recursos necesarios para invertir en grandes infraestructuras de datos, por lo que la nube les ofrece la oportunidad de equiparar sus capacidades tecnológicas a las de las grandes empresas. No obstante, las pymes no son las únicas interesadas. La crisis económica también está teniendo un impacto en las grandes empresas, que ven limitadas sus inversiones tecnológicas, lo cual puede inclinar la balanza hacia los servicios de Cloud Computing.

A nivel académico, este trabajo de tesis es susceptible de ser relevante, ya que es uno de los primeros trabajos que utiliza un sistema multiagente para gobernar y automatizar un entorno Cloud Computing, y propone la incorporación de organizaciones virtuales auto-adaptativas al paradigma cloud, integrando de esta forma la computación social y el cloud computing. Así mismo, este trabajo es uno de los primeros que propone la utilización de tecnologías de acuerdo con nuevos mecanismos de negociación que utiliza razonamiento basado en casos para reasignar recursos de forma que se mantenga la calidad del servicio constante. Prueba del interés existe están los primeros trabajos publicados en el marco de la tesis [9][3]. Así como los proyectos de investigación que sustentan este trabajo: Cloud-IO[4] y Smartmedical[5].

Referencia

1. A. Ricci, M. Viroli, and A. Omicini. 2007. Give agents their artifacts: the A&A approach for engineering working environments in MAS. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems* (p. 150). ACM
2. D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff and D. Zagorodnov. 2009. The eucalyptus open-source cloud-computing system. In *Cluster Computing and the Grid, 2009. CCGRID'09. 9th IEEE/ACM International Symposium on* (pp. 124-131). IEEE.
3. F. De la Prieta, S. Rodríguez, J. Bajo and J.M. Corchado. 2013. A Multiagent System for Resource Distribution into a Cloud Computing Environment. In *Advances on Practical Applications of Agents and Multi-Agent Systems* (pp. 37-48). Springer Berlin Heidelberg.
4. Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman, S., Youseff, L. and Zagorodnov, D. The Eucalyptus Open-Source Cloud-Computing System. Cluster Computing and the Grid, 2009. CCGRID '09. 9th IEEE/ACM International Symposium. pp:124 - 131 May, 2009)
5. P. Mell and T. Grance. The NIST definition of Cloud Computing. In: NIST Special Publication 800-145. Sep. 2011.
6. R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg and I. Brandic. Cloud Computing and emerging IT plaforms: Vision, hype and reality for delivering computing as the 5th utility. In: Future Generation Computer Systems. Volume 25. Issue 6, June 2009. Pages. 599-616.
7. R. Buyya, S. K. Garg, and R. N. Calheiros. 2011. SLA-oriented resource provisioning for cloud computing: Challenges, architecture, and solutions. In *Cloud and Service Computing (CSC), 2011 International Conference on* (pp. 1-10). IEEE.

[4] http://www.cloud-io-project.com/

[5] http://smartmedical.es/

8. R. Han, L. Guo, M. Ghanem and Y. Guo. 2012 . Lightweight resource scaling for cloud applications. In *Cluster, Cloud and Grid Computing (CCGrid), 2012 12th IEEE/ACM International Symposium on* (pp. 644-651). IEEE.
9. S. Heras, F. De la Prieta, V. Julian, S. Rodríguez, V. Botti, J. Bajo and J.M. Corchado. 2012. Agreement technologies and their use in cloud computing environments. *Progress in Artificial Intelligence*, *1*(4), 277-290.
10. X. Miao, & J. Han. 2011. The Design of a Private Cloud Infrastructure Based on XEN. In Distributed Computing and Applications to Business, Engineering and Science (DCABES), 2011 Tenth International Symposium on (pp. 160-164). IEEE.
11. Z. Gong, X. Gu, and J. Wilkes. 2010. Press: Predictive elastic resource scaling for cloud systems. In *Network and Service Management (CNSM), 2010 International Conference on* (pp. 9-16). IEEE.
12. S. Rodríguez, Y. De Paz, J. Bajo and J.M. Corchado. Social-based Planning Model for Multi-agent Systems. Expert Systems with Applications 38 (10), pp. 13005-13023. 2011.

Energy Efficient Resource Usage in Data Centers: GreenDoop

Ph. D. Student: Damián Fernández
Advisors: Alejandro Fernández-Montes, Juan A. Ortega

Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla, Sevilla, España
damianfdezcerero@gmail.com, {afdez,jortega}@us.es

1 Introduction

The growing interest in energy efficiency because of cloud computing has led big companies to enhance its efforts in reducing power consumption of data center infrastructures due to the increase in data centers energy consumption in the past ten years, which is 1.5% of worldwide energy consumption [1,2]. Thanks to these efforts and researchers results, current energy consumption trends have slowed down in the past five years.

However, many of these efforts have been directed to areas unrelated to software, such as cooling [3], hardware scaling [4] or power distribution [5]. This work, in contrast, focuses on software systems that support data centers, such as parallel computing systems or distributed file systems in order to develop a software product that minimizes the energy consumption.

Therefore the proposal to improve energy efficiency of these resources must be built starting from an existing software ecosystem. We are planning to choose Hadoop as the base system due to it is Apache v2 licensed, it is becoming a *de facto* standard, many important companies are contributing to improve it [6] and because it is based on Google File System [7], what makes it a very interesting approach for real infrastructures deployed over current data centers.

2 Hypothesis

In order to create a real software system which improves data centers energy efficiency while affecting as little as possible performance and reliability, we have to choose an existing distributed computing framework to work with. To do this, we need the system to be open source, high-availability and widespread. That system is Apache Hadoop, and our goal is to modify it applying the designed energy efficiency policies.

2.1 Modifications to Hadoop

In order to minimize the number of machines running needed to operate correctly by applying load balancing, data organization and switching on/off policies in

Hadoop-based data centers, we have to make modifications to Hadoop modules, taking into account both major uses of these data centers: serving data and processing jobs.

Hadoop Distributed File System (HDFS) is responsible for the first task. Yet Another Resource Neogitator (YARN), the job scheduling and cluster resource management module, is responsible for the latter. The purpose of YARN is running MapReduce applications [8].

Global Modifications

- **Switching on/off Manager Node**: We propose a global node that performs all tasks related to the switching on/off machines and resource management tasks derived from this. This node works close to the Namenode (HDFS) and ResourceManager (YARN) to mantain proper data center functioning. This is achieved by coordinating all nodes and new policies relying on the capabilities of the systems which are being coordinated.

Modifications to HDFS These modifications to HDFS intend to minimize the number of machines running needed to maintain redundancy benefits and data reliability. In order to achieve this goal, we will apply –among others– these modifications:

- **Restructuration of the Data Center**: We propose to divide the data center into a number of sectors equals to the number of replicas of each block that are housed in order to mantain the least running sectors as possible.
- **Balancing Policy**: In order to increase machines' CPU use and switching off of idle machines, we propose a tasks balancing policy to increase CPU and memory usage. This enables Switching Manager Node to switch off idle machines while filling each server until a threshold is reached.
- **Buffer Nodes**: To face block modifying while replicas are switched off problem, the easiest option is to turn on the machines that are affected and replicate block modifications. However, this pattern can spend more energy than mantaining machines in idle state.
  Therefore we propose to setup Buffer Nodes that mantain block modifications, mapping machines with changes applied or managing block versioning until a system returns to be consistent. This consistency is achieved by performing tasks that replicate changes which can be run periodicly.
- **Cache System**: This new cache system will improve data center response time when a machine fails by storing a new replica of most relevant blocks in buffer nodes (if they are not already stored).

Modifications to YARN

- **Workload Prediction Node**: Although the Switching on/off node can give neccesary resources in response to requests if the available resources are not

enough, data center performance and response time gets worse. In addition, the task arrival pattern may incur in the switching off and on of many machines, stressing the hardware and even spending more energy than keeping machines in idle state. According to queuing theory, the arrival of events to a set of resources is described by exponential distribution and the arrival of consecutive events to a set of resources is described by a Gamma distribution. Our proposal is to start from these distributions to fit the situation: to minimize energy consumption while maintaining the best possible performance and bringing the hardware stress to a minimum. We will rely on, adapt and improve policies developed for energy efficiency in the execution of tasks in grid computing environments [9].

3 Objectives

GreenDoop is intended as a set of software proposals for energy efficiency in data centers. This proposals starts from a comprehensive study of the state of the art and current market solutions. In addition, GreenDoop has the objective of transferring the obtained results by creating a highly configurable and useful enterprise tool. The expected results include:

- **Comprehensive survey of the state of the art**: In this paper we study different solutions proposed for the problem of data centers' high energy consumption.
- **Hadoop modifications impact analysis**: In order to analyze how proposed modifications affect Hadoop we will experiment and simulate different environments, measuring many metrics such as I/O response time, hard drive failure rate, data loss rate, etc.
- **Workload Prediction**: It is especially interesting to predict the data center workload based on a queuing model in different use cases such as batch processing, real-time requirements, etc.
- **Create GreenDoop**: A key part of this work is to create a tool that could significantly reduce current data centers' energy consumption.

4 Methodology and Work Plan

Our dissertation work plan has a total duration of approximately three years. This work plan is divided into six main groups of activities as shown in the Gantt chart in Figure 1

- **Detailed study of existing solutions**. We will start analyzing current data center energy-efficiency problems and solutions, their lacks and possible improvements.
- **Energy efficiency policies development**. In this period we will propose abstract energy-efficient policies based on existing distributed file systems workaround.

- **Publications**. We intend to disseminate the results obtained in our PhD Thesis during the entire work plan.
- **GreenDoop development**. Once energy efficiency policies are developed, our focus is to build a Hadoop extension software to deploy these policies, named GreenDoop.
- **Deploying and testing GreenDoop on a real data center**. In order to demonstrate our research results feasibility we will focus on test GreenDoop and its impact in data center power consumption, reliability and performace.
- **Thesis writing**

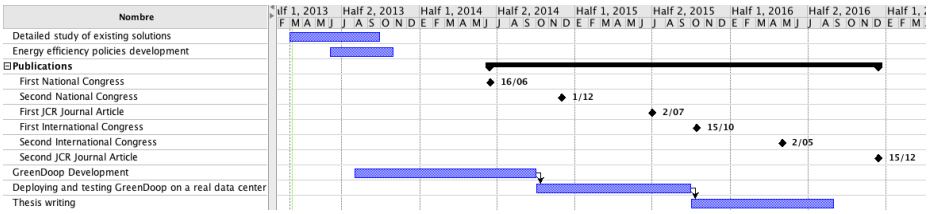


Fig. 1. Work plan Gantt chart

5 Relevance

The predicted energy savings have been studied theoretically and have been simulated in a virtual environment. These energy savings depend on the parameters used and the needs of each user. These parameters include:

- **Number of always-on sectors (R_{on})**: This configurable parameter represents the number of always-available replicas of each block derived from the restructuration of the data center. Less sectors on implies less energy consumed.
- **Total number of sectors (R_t)**: This parameter represents the total number of replicas of each block that data center stores. Hadoop default value for this parameter is three.
- **Time between system-level replications (T_{br})**: This configurable parameter represents the time between system-level replications. During this time the system is in an inconsistent state, relying on buffer nodes functionality to maintain the reliability of the data. When this time is ellapsed or buffer nodes maximum capacity is achieved, replication tasks are executed to get the system back to its consistent state and remove buffer nodes data.
- **Time needed to replicate (T_{nr})**: This time represents the average time that a node needs to copy its modified blocks to switched off nodes that contain out-of-date replicas of those blocks.

- **Time to switch on a node ($T_{off \to on}$)** and **Time to switch off a node ($T_{on \to off}$)**.
- **Power to switch between on/off state ($P_{off \leftrightarrow on}$)**, **Power to maintain a node in on state (P_{on})** and **Power to maintain switched off a node (P_{off})**.
- **Power cicle time ($T_{on \leftrightarrow off}$)**= $T_{off \to on} + T_{on \to off}$.
- **Sectors off (R_{off})**= $R_t - R_{on}$.
- **Total number of nodes (M_t)**: Total number of worker machines in the data center.
- **Number of buffer nodes (M_b)**: Number of buffer nodes used. This number depends on user needs and buffer policy.
- **Number of machines on (M_{on})**: Number of machines that belong to always-on sectors. This number is described by $M_{on} = \frac{R_{on} \cdot M_t}{R_t}$.
- **Number of machines off (M_{off})**: Number of machines that belong to switched off sectors. This number is described by $M_{off} = M_t - M_{on}$.

Disregarding energy consumption resulting from the execution of tasks, which is unpredictable, the energy consumption of a Hadoop base system is computed as the power needed to maintain on all the machines multiplied by the operation time T as described in the following equation (1):

$$E_{hd} = P_{on} \cdot M_t \cdot T \tag{1}$$

In order to compute energy consumed by our Hadoop-modified system GreenDoop, we have to compute several consumptions. First, we compute the energy consumed by the nodes of the sectors that are always on (2):

$$E_1 = P_{on} \cdot M_{on} \cdot T \tag{2}$$

Second, we compute energy consumed by machines of the sectors that are switched off (3):

$$E_2 = P_{off} \cdot M_{off} \cdot T \tag{3}$$

Third, we compute energy consumed by Buffer nodes (4):

$$E_3 = P_{on} \cdot M_b \cdot T \tag{4}$$

Fourth, we compute energy consumed by switching on and off nodes when system-level replications occur (5):

$$E_4 = T_{on \leftrightarrow off} \cdot P_{on \leftrightarrow off} \cdot M_{off} \cdot \frac{T}{T_{br}} \tag{5}$$

Fifth, we compute energy consumed by nodes while replicating (6):

$$E_5 = T_{nr} \cdot P_{on} \cdot M_{off} \cdot \frac{T}{T_{br}} \tag{6}$$

Summarizing, the energy consumed by GreenDoop is: $E_{gd} = E_1 + E_2 + E_3 + E_4 + E_5$. In order to compare energy consumed by GreenDoop and Hadoop we need to compute the ratio between them as described in the following equation (7):

$$\frac{E_{gd}}{E_{hd}} = \frac{R_{on}}{R_t} + \frac{R_{off} \cdot P_{off}}{R_t \cdot P_{on}} + \frac{M_b}{M_t} + \frac{T_{nr} \cdot R_{off}}{T_{br} \cdot R_t} + \frac{T_{on \leftrightarrow off} \cdot P_{on \leftrightarrow off} \cdot R_{off}}{T_{br} \cdot P_{on} \cdot R_t} \quad (7)$$

In order to achieve maximum energy saving and based on a common hypothetical environmet, this equation is parametrized with this values: R_{on}: 1, R_t: 3, T_{br}: 28800 s, T_{nr}: 600 s, T_{on}: 90 s [10], T_{off}: 25 s [10], $P_{off \leftrightarrow on}$: 110 W [10], P_{on}: 100 W [10], P_{off}: 5 W [10], M_t: 300, M_b: 2.

Applying these parameters, GreenDoop system energy consumption would be about 39% of the current Hadoop system.

References

1. Koomey, J.: Worldwide electricity used in data centers. Environmental Research Letters (2008)
2. Koomey, J.G.: Growth in data center electricity use 2005 to 2010. Technical report, Stanford University (2011)
3. El-Sayed, N.: Temperature management in data centers: Why some (might) like it hot. Technical report, Department of Computer Science, University of Toronto (2012)
4. Fan, X.: Power provisioning for a warehouse-sized computer. Technical report, Google Inc. (2007)
5. Femal, M.: Boosting data center performance through non-uniform power allocation. In: Autonomic Computing, 2005. ICAC 2005. Proceedings. Second International Conference on, Dept. of Comput. Sci., North Carolina State Univ. (2005)
6. Borthakur, D.: Apache hadoop goes realtime at facebook. In: Proceedings of the 2011 international conference on Management of data - SIGMOD '11, Facebook (2011)
7. Shvachko, K.: The hadoop distributed file system. In: 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST), Yahoo! (2010)
8. Tomasic, I.: Bigdata and mapreduce with hadoop. In: Proceedings of the 1st International Conference on Cloud Assisted ServiceS. (2012)
9. Fernández-Montes, A.: Energy-Saving Policies in Grid-Computing and Smart Environments. PhD thesis, Universidad de Sevilla (2012)
10. Fernández-Montes, A.: Smart scheduling for saving energy in grid computing. Expert Systems with Applications (2012)

eJason: una Plataforma para el Desarrollo de Sistemas Multiagente Distribuidos Eficientes y Robustos

Doctorando: Álvaro Fernández Díaz
Directores: Dr. Lars-Ake Fredlund y Dr. Clara Benac Earle

Facultad de Informática, Universidad Politécnica de Madrid
{avalor,lfredlund,cbenac}@babel.ls.fi.upm.es
https://babel.ls.fi.upm.es/

Keywords: Lenguajes de programación, Sistemas multiagente distribuidos, Tolerancia a fallos

1. Introducción

El proyecto de tesis presentado persigue el **aprovechamiento de las sinergias** entre las tecnologías dirigidas al desarrollo de **sistemas multiagente** (distribuidos) y aquéllas orientadas al desarrollo de **sistemas distribuidos**. Ambos tipos de sistemas precisan mecanismos de concurrencia y distribución eficientes. Estos mechanismos posibilitan la interacción entre los distintos componentes que conforman un sistema. Por otro lado, los sistemas multiagente afrontan problemas típicos de los sistemas distribuidos, como, por ejemplo, las condiciones de carrera o la propagación de fallos entre distintos componentes del sistema. Por tanto, la robustez de estos tipos de sistemas depende de la existencia de técnicas de tolerancia a fallos comunes, o, al menos, similares, en ambos.

Sin embargo, y, en cierto modo, sorprendentemente, el estado del arte de las tecnologías orientadas a sistemas multiagente (lenguajes de programación y herramientas de desarrollo) evoluciona de manera independiente al estado del arte de los sistemas distribuidos. En nuestra opinión, esta evolución independiente puede dilatar en el tiempo la resolución de determinados problemas comunes. Por citar algún ejemplo, ciertos sistemas distribuidos (especialmente los basados en arquitecturas de actores, exclusivamente reactivos) podrían beneficiarse de la proactividad e indepencia inherentes a las arquitecturas de agentes inteligentes. Por otro lado, los sistemas multiagente pueden beneficiarse de la eficiencia de las técnicas y mecanismos de distribución y tolerancia a fallos disponibles para el desarrollo de sistemas distribuidos.

2. Hipótesis de Partida

Nuestra hipótesis de partida es la siguiente:

"La semejanza entre sistemas multiagente y sistemas distribuidos posibilita un intercambio tecnológico beneficioso para ambos".

Se han llevado a cabo diversas investigaciones preliminares que apoyan nuestra hipótesis inicial:

- En nuestra experiencia, los sistemas multiagente implementados mediante el lenguaje de programación Erlang proporcionan **altas prestaciones**, en términos de baja latencia introducida por eventos de comunicación y escalabilidad con respecto al número de componentes del sistema. En [4], mostramos una comparación entre dos implementaciones, Java y Erlang, de un sistema multiagente. Dicha comparación muestra un ejemplo de cómo Erlang puede superar las prestacion de Java en el desarrollo de sistemas multiagente (altamente concurrentes y distribuidos).
- El lenguaje de programación Jason, específico para sistemas multiagente y basado en la arquitectura BDI, es uno de los más populares para este tipo de sistemas. La implementación principal de este lenguaje ha sido realizada en Java. En [6] proporcionamos una comparación preliminar (utilizando sistemas multiagente básicos) entre las prestaciones de la implementación de Jason en Java y **eJason**, nuestra implementación de un subconjunto relevante de **Jason en Erlang**. Esta comparación muestra cómo un mismo sistema Jason puede ser ejecutado en eJason en una cantidad de tiempo hasta tres órdenes de magnitud inferior a la ejecución en Java-Jason (menos de un segundo y varios minutos, respectivamente). Además, la implementación en eJason permite escalar los sistemas en mayor medida (unos cientos de agentes en Jason y varios cientos de miles en eJason).

3. Objetivos

El objetivo último de nuestro proyecto de tesis es:

"Proporcionar una plataforma de desarrollo de sistemas multiagente distribuidos que sean robustos y se ejecuten de manera eficiente".

Este objetivo principal se divide en los siguientes subobjetivos:

1. Aprovechar las sinergias entre las tecnologías específicas para sistemas multiagente y las tecnologías orientadas a sistemas distribuidos.
2. Minimizar la ratio (dificultad de aprendizaje)/ (beneficio de uso) relativa a la plataforma.
3. Facilitar y fomentar la adopción de arquitecturas multiagente por parte de los desarrolladores de sistemas distribuidos de propósito general.

4. Mejorar las prestaciones (robustez, tiempo de ejecución) de los sistemas multiagente importando, desarrollando y/o perfeccionando técnicas que se utilizan de manera exitosa en los sistemas distribuidos.

Estos objetivos iniciales, relativamente abstractos, se han concretizado durante el transcurso de la investigación, tal como indicamos en la siguiente sección.

4. Metodología y plan de trabajo

La metodología global que se está siguiendo en este proyecto de tesis es el habitual para las tesis doctorales :

1. Preparación de la Tesis
2. Desarrollo del trabajo conducente a la tesis
3. Elaboración de la Tesis

4.1. Preparación de la Tesis

La preparación de la tesis consiste en lo siguiente:

- Elección del tema
- Análisis del Estado del Arte
- Planteamiento de la hipótesis de partida y objetivos

Nótese que, en el proyecto de tesis aquí presentado, esta fase ya ha sido completada.

Como se indica en el apartado anterior, el tema elegido para este proyecto es el aprovechamiento de las sinergias entre las tecnologías de desarrollo de sistemas multiagente y sistemas distribuidos. Durante el análisis del estado del arte, se identificaron una serie de deficiencias en los lenguajes de programación para sistemas multiagente. Estos lenguajes, en su mayoría implementados en Java (por ejemplo Jason, Goal o la plataforma JADE), sufren problemas de escalabilidad y sus prestaciones (eficiencia, robustez), en comparación con lenguajes basados en actores, como Erlang, son inferiores.

Por otro lado, los lenguajes de propósito general, como Erlang, carecen de una capa de abstracción, como la arquitectura BDI, de la que disfruta un elevado número de lenguajes de programación para sistemas multiagente. Esta abstracción permite al programador, entre otras, delegar de manera sencilla en el entorno de ejecución la responsabilidad de llevar a cabo tareas complejas. Ejemplos de estas tareas complejas son la selección de un flujo de ejecución conducente a la consecución de ciertas metas o la incorporación e interpretación de la intencionalidad asociada a los mensajes entre componentes del sistema.

Como punto de partida, se consideraron las prestaciones del lenguaje de programacion para sistemas multiagente Jason y el lenguaje de implementación para sistemas concurrentes, distribuídos y tolerantes a fallos Erlang. Para aprovechar los fortalezas de ambos se barajaron dos posibilidades. La primera consistía en

la elaboración de una serie de bibliotecas Erlang que implementasen los pilares fundamentales de Jason (estructuras BDI como la base de creencias, base de planes, intenciones, ciclo de razonamiento). Esta opción suponía la aparición de una nueva extensión de Erlang cuyo uso requeriría un esfuerzo de aprendizaje relativamente elevado. La segunda opción consistía en implementar y extender Jason utilizando Erlang. De esta manera, un programador Jason no necesita aprender ningún nuevo lenguaje. Esta opción fue la elegida finalmente y la que se está desarrollando actualmente.

4.2. Desarrollo del trabajo conducente a la Tesis

El trabajo de investigación conducente al cumplimiento de los objetivos de este proyecto se divide en las siguiente fases:

1. Diseño de la implementación de Jason en Erlang
2. Implementación de eJason
3. Diseño de las extensiones a Jason
4. Implementación de las extensiones
5. Experimentos

Esta etapa requiere una inversión considerable de tiempo, especialmente la parte de implementación. En la actualidad las dos primeras etapas están prácticamente completadas y también se han realizado algunos experimentos. En este momento se están desarrollando distintas extensiones a Jason. Una de estas extensiones, consistente en técnicas de distribución y tolerancia a fallos ha sido publicada en [5]. Finalmente, resta diseñar nuevos experimentos que validen la tesis de partida.

4.3. Elaboración de la Tesis

La elaboración de la Tesis consiste en presentar de forma escrita el trabajo de investigación realizado para cumplir los objetivos que se plantearon al principio del proyecto de tesis. Puesto se se ha realizado un estudio del estado del arte exhaustivo y se han publicado varios artículos donde se presentan los primeros resultados, entendemos que esta fase del proyecto debería poder realizarse en un menor tiempo que las fases anteriores.

4.4. Plan de trabajo

En el siguiente plan de trabajo se muestra la distribución temporal de las fases del proyecto de tesis descritas anteriormente.

- 2010: Preparación de la Tesis. Erlang como lenguaje de programacion de sistemas multiagente [4].
- 2011: Preparación de la Tesis. Análisis de lenguajes para sistemas multiagente.

- 2012: Desarrollo del trabajo conducente a la tesis. Diseño de la traducción de Jason a Erlang. Primeros experimentos y resultados preliminares [7,6,5].
- 2013 (hasta junio-julio): Implementación completa de Jason a Erlang. Diseño e implementación de extensiones a Jason. Validación de resultados mediante nuevos experimentos.
- 2013 (sept-dic) Elaboración de la tesis.

5. Relevancia

Desde nuestro punto de vista, el cumplimiento del objetivo principal del proyecto de tesis aquí presentado, es decir, desarrollar un lenguaje de programación para sistemas multiagente que tenga buenas prestaciones de concurrencia, distribución y tolerancia a fallos, tiene sin duda relevancia tanto en el campo científico como en el campo industrial.

Respecto al campo científico, los lenguajes de programación para sistemas multiagentes son objeto de interés para la comunidad científica, como demuestra la gran cantidad de artículos relacionados que han sido publicados en algunas de las conferencias y revistas más importantes del área de sistemas multiagente (por ejemplo [2,8,3,1]) . Sin embargo, la mayoría de estos lenguajes no proporcionan soporte nativo a la distribución y tolerancia a fallos de manera simultánea, es decir, precisan la utilización de software de terceros. Nuestra aportación, aumentar el lenguaje Jason con distribución y tolerancia a fallos, sí ofrece esta posibilidad.

Por otro lado, los resultados preliminares de nuestro trabajo son alentadores. Éstos evidencian una mejora considerable tanto de los tiempos de ejecución y como de la escalabilidad de los sistemas multiagente implementados en eJason. Estos resultados han recibido una buena acogida por parte de la comunidad científica, tal y como se manifiesta en la aceptación de diversos artículos científicos [4,7,5,6]. Finalmente, y algo que agradecemos en gran medida pues supone una validación favorable y externa de nuestro proyecto, los dos autores principales de Jason, Jomi Fred Hübner y Rafael H. Bordini, nos han contactado, motu proprio y de manera independiente, para mostrar su apoyo al proyecto aquí descrito.

En relación al campo industrial, la decisión de implementar Jason en Erlang conlleva que los sistemas multiagente resultantes sean muy eficientes, escalables y robustos, lo que los hace muy interesantes desde el punto de vista industrial. De esta manera, se facilita que empresas que ya utilizan Erlang (Amazon, Whatsapp, Facebook, Tuenti), debido a las mencionadas prestaciones, adopten arquitecturas multiagente. Esta adopción, por un lado, permitiría a dichas empresas beneficiarse de las ventajas de las arquitecturas multiagente y, por otro, aumentaría la presencia de las mismas en la industria.

Referencias

1. Rafael H. Bordini, Mehdi Dastani, Jürgen Dix, and Amal El Fallah-Seghrouchni, editors. *Multi-Agent Programming: Languages, Platforms and Applications*, volume 15 of *Multiagent Systems, Artificial Societies, and Simulated Organizations*. Springer, 2005.
2. Rafael H. Bordini and Jomi Fred Hübner. Semantics for the jason variant of agentspeak (plan failure and some internal actions). In *Proceedings of the 2010 conference on ECAI 2010: 19th European Conference on Artificial Intelligence*, pages 635–640, Amsterdam, The Netherlands, The Netherlands, 2010. IOS Press.
3. Rafael H. Bordini, Michael Wooldridge, and Jomi Fred Hübner. *Programming Multi-Agent Systems in AgentSpeak using Jason (Wiley Series in Agent Technology)*. John Wiley & Sons, 2007.
4. Álvaro Fernández Díaz, Clara Benac Earle, and Lars-Ake Fredlund. Implementing a multiagent negotiation protocol in erlang. In *Proceedings of the 9th ACM SIGPLAN workshop on Erlang*, Erlang '10, 2010.
5. Álvaro Fernández Díaz, Clara Benac Earle, and Lars-Åke Fredlund. Adding distribution and fault tolerance to jason. In *Workshop on Programming based on Actors, Agents, and Decentralized Control 2012(AGERE! 2012)*, Tucson, Arizona (USA), October 21-22 2012.
6. Álvaro Fernández Díaz, Clara Benac Earle, and Lars-Åke Fredlund. ejason: an implementation of jason in erlang. In *10th International Workshop on Programming Multi-Agent Systems (ProMAS 2012)*, Valencia, Spain, June 5 2012.
7. Álvaro Fernández Díaz, Clara Benac Earle, and Lars-Åke Fredlund. Erlang as an implementation platform for bdi languages. In *Eleventh ACM SIGPLAN Erlang Workshop 2012(Erlang 12)*, Copenhagen, Denmark, September 14 2012.
8. Koen V. Hindriks, Wiebe van der Hoek, and M. Birna van Riemsdijk. Agent programming with temporally extended goals. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '09, pages 137–144, Richland, SC, 2009. International Foundation for Autonomous Agents and Multiagent Systems.

Linear programming with Interval Type-2 fuzzy constraints

Juan C. Figueroa-García[1]
Advisor: Germán Hernández[2]

[1] Universidad Nacional de Colombia
jcfigueroag@udistrital.edu.co *
[2] Universidad Nacional de Colombia
gjhernandezp@gmail.com **

Abstract. This work is focused in proposing a linear programming model with Interval Type-2 fuzzy constraints. It is expected to find an optimal solution to this problem using Interval Type-2 fuzzy sets, fuzzy linear programming concepts, and convex optimization methods.

1 Introduction and Motivation

Many decision making problems have to deal with uncertainty in the parameters of the problem, so the analyst must face up with uncertainty (probabilistic and/or possibilistic) in different ways. In an LP problem, all its parameters (costs, technological coefficients and constraints) can contain uncertainty, so many approaches can be used when uncertain parameters appear. As usual, as more uncertainty sources are considered, more complex the model and the method used to achieve a solution are. A typical uncertainty source comes from the concept of a constraint, which is commonly assumed as probabilistic. But, when the constraints of the problem are defined by experts and they use non-probabilistic information; what is the correct way to address the problem?

Recently, another type of uncertainty, called linguistic uncertainty has been considered. In this, the uncertainty about different perceptions of a concept, like the one arising when you have multiple experts with equally valuable opinions that affects the definition of the constraints of an LP problem, is modeled. To do so, Interval Type-2 Fuzzy Sets (IT2FS) seem to be an appropriate representation to deal with, so we propose its use to define the constraints of an FLP problem as IT2FS through a general model and a method for finding a solution.

An IT2FS is a more complex uncertainty measure which needs more complex representations than classical fuzzy sets, so a good way for reducing its complexity is by using a Type-reduction strategy that consists on finding a set embedded into a Type-2 fuzzy set, in order to apply convex optimization techniques.

---

* Juan Carlos Figueroa is Assistant Professor in the Universidad Distrital Francisco José de Caldas, Bogotá - Colombia.

** Germán Hernández is Associate Professor in the Universidad nacional de Colombia, Bogotá - Colombia.

In practical applications such as financial, supply chain, markov chains, control, etc, the analyst needs to make a decision based on the decision variables x of the system, so our proposal is oriented to find their optimal values x^* under Type-2 fuzzy uncertainty. We expect to use convex optimization techniques to ensure its interpretability and simplicity, which is highly desirable by decision makers.

2 Description of the proposal

2.1 Previous works

The idea to involve other kind of uncertainties to LP problems comes from previous previous works (Figueroa and Hernández [5], and Figueroa [4]) which are based on the idea of finding a Type-1 fuzzy set[3] embedded into a Type-2 fuzzy set. Our proposal is intended to be more general in the sense that it is based on the well known Bellman-Zadeh [1] decision-making principle, so we expect to find a more robust solution.

Classical works on fuzzy linear programming (Zimmermann [14], Zimmermann and Fullér [15], Verdegay [13], Delgado, Verdegay and Vila [2], Lodwick and Jamison [7]) are focused in finding a solution of an FLP using a single fuzzy set per uncertain parameter. Our proposal tries to involve *Type-2 fuzzy constraints* to a *LP* problem using previous findings, and the uncertainty coming from the knowledge of multiple experts, represented by Type-2 fuzzy sets.

2.2 Hypothesis

Given the problem of addressing multiple experts opinions about a constraint of an LP problem, there is a necessity of dealing with multiple fuzzy sets given by the experts of the system. In this way, the initial hypothesis of the proposal is:

How to involve Type-2 fuzzy uncertainty to LP problems?

This hypothesis is based on the idea that there is the possibility of having equally valuable experts who are defining the constraints of an LP problem, and they are not in agree with only one fuzzy set, so this situation generates a linguistic uncertainty source which can be modelled using Type-2 fuzzy sets. Some basic concepts about interval type-2 fuzzy sets are shown as follows.

2.3 Interval Type-2 fuzzy constraints

Interval type-2 fuzzy sets allow to model lunguistic uncertainty, i.e. the uncertainty about different perceptions and concepts. Mendel (See [10, 12, 11, 6]), and Melgarejo (See [9, 8]) provided formal definitions of IT2FS, and Figueroa [3, 5] proposed an extension of the FLP to include constraints with linguistic uncertainty represented by IT2FS. Now, the concept of an interval type-2 fuzzy constraint is shown as follows

[3] In this work, a Type-1 fuzzy set is referred as a classical fuzzy set.

Definition 1 (Interval type-2 fuzzy constraint) *Consider a set of constraints of an FLP problem defined as an IT2FS $\tilde{b}$ defined on the closed interval $\tilde{b}_i \in [\underline{b}_i, \overline{b}_i]$, $\{\underline{b}_i, \overline{b}_i\} \in \mathbb{R}$ and $i \in \mathbb{N}_m$. The membership function which represents $\tilde{b}_i$ is:*

$$\tilde{b}_i = \int_{b_i \in \mathbb{R}} \left[\int_{u \in J_{b_i}} 1/u \right] \Big/ b_i, \;\; i \in \mathbb{N}_m, J_{b_i} \subseteq [0,1] \tag{1}$$

Note that $\tilde{b}$ is bounded by both *Lower* and *Upper* primary membership functions, namely $\underline{\mu}_{\tilde{b}}(x)$ with parameters $\check{\underline{b}}$ and $\hat{\underline{b}}$ and $\overline{\mu}_{\tilde{b}}(x)$ with parameters $\check{\overline{b}}$ and $\hat{\overline{b}}$. Now, the Footprint of Uncertainty *(FOU)* of each set $\tilde{b}$ is a measure of uncertainty of a fuzzy set. A graphical representation of $\tilde{b}_i$ is shown in Figure 1

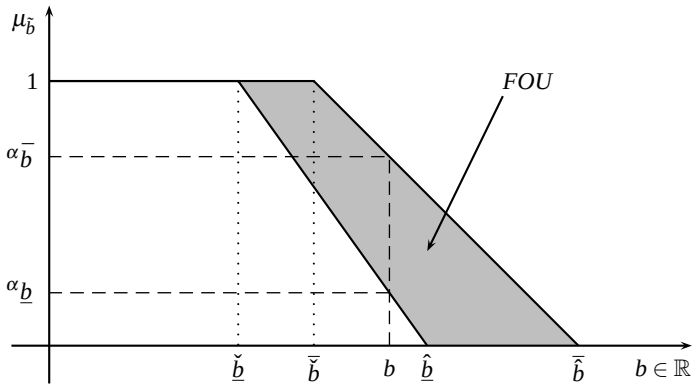


Fig. 1. IT2FS constraint $Ax \precsim \tilde{b}$

In this Figure, $\tilde{b}$ is an IT2FS with linear membership functions $\underline{\mu}_{\tilde{b}}$ and $\overline{\mu}_{\tilde{b}}$. A particular value b projects an interval of infinite membership degrees $u \in J_b$, as follows

$$J_b \in \left[{}^{\alpha}\underline{b}, {}^{\alpha}\overline{b} \right] \; \forall \, b \in \mathbb{R} \tag{2}$$

where J_b is the set of all possible membership degrees associated to $b \in \mathbb{R}$.

2.4 A mathematical programming model with interval type-2 fuzzy constraints

Given the concept of an IT2FS constraint and the definition of an FLP, an uncertain constrained FLP model (IT2FLP) can be defined as follows:

$$\begin{gathered} \max_{x \in X} z = c'x + c_0 \\ s.t. \\ Ax \precsim \tilde{b} \\ x \geqslant 0 \end{gathered} \tag{3}$$

where $x, c \in \mathbb{R}^m$, $c_0 \in \mathbb{R}$, $A \in \mathbb{R}^{n \times m}$. $\tilde{b}$ is an IT2FS vector defined by two primary membership functions $\underline{\mu}_b$ and $\overline{\mu}_b$. $\precsim$ is a Type-2 fuzzy partial order.

3 Objectives

3.1 General Objective:

Define an optimization model that allows to solve LP problems where its constraints includes linear Interval Type-2 fuzzy sets.

3.2 Specific Objectives:

- Formally define the concept of a solution of an Interval Type-2 fuzzy LP problem.
- Develop a formal framework to solve LP problems with Interval Type-2 constraints.
- Develop an optimization method for an LP model with IT2FS constraints.
- Validate the developed method through both simulated and study cases relating to industrial engineering sciences like logistics and production planning.

In this way, the main idea of our proposal is to find an optimal solution in terms of $x \in X$ under type-2 fuzzy uncertainty, based on the Bellman-Zadeh decision making principle and FLP methods.

3.3 Methodology

The following methodological scheme is proposed:

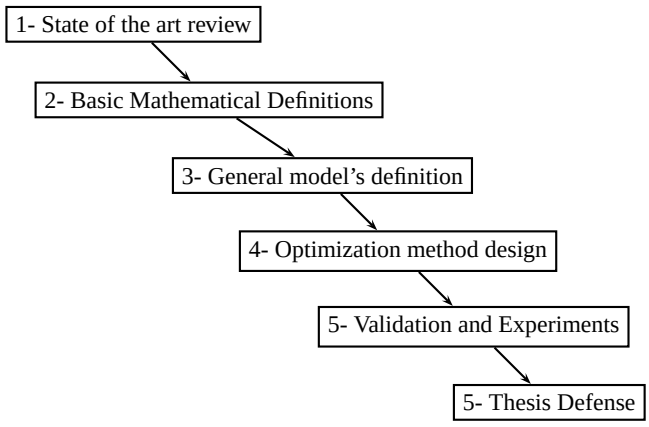


Fig. 2. Proposed Methodology

3.4 Value added

The main idea of this proposal is to obtain optimal values of $x \in \mathbb{R}$ given uncertain constraints $\tilde{b}$. In general, the proposal tries to find a global optima based on the decision-making framework provided by Bellman and Zadeh, applied to FLP problems with interval type-2 fuzzy constraints.

On the other hand, the proposal is inteneded to solve problems when having opinions of multiple experts represented by multiple fuzzy sets, using interval type-2 fuzzy sets as a single representation of their knowledge.

4 Relevance

The relative recent use of type-2 fuzzy sets in control and classification problems has motivated its extension to optimization problems, including LP ones. The potential of type-2 fuzzy sets for representing collective knowledge of multiple experts about a parameter becomes into a novel application to optimization problems.

In this way, decision-making techniques based on type-2 fuzzy sets is the next step in fuzzy decision-making. Many practical applications (such as financial, hierarchical analysis, production planning, etc) show ambiguity among experts and uncertainty, so the design of optimization methods for helping decision making under type-2 fuzzy uncertainty is an interesting topic to be covered.

References

1. R. E. Bellman and L. A. Zadeh, *Decision-Making in a Fuzzy Environment*, Management Science **17** (1970), no. 4, 141–164.
2. M. Delgado, J.L. Verdegay, and M.A. Vila, *A general model for fuzzy linear programming*, Fuzzy Sets and Systems **29** (1989), 11–29.
3. Juan Carlos Figueroa, *Interval type-2 fuzzy linear programming: Uncertain constraints*, IEEE Symposium Series on Computational Intelligence, IEEE, 2011, pp. 1–6.
4. _____, *A general model for linear programming with interval type-2 fuzzy technological coefficients*, 2012 Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS), IEEE, 2012, pp. 1–6.
5. Juan Carlos Figueroa and Germán Hernández, *Computing optimal solutions of a linear programming problem with interval type-2 fuzzy constraints*, Lecture Notes in Computer Science **7208** (2012), 567–576.
6. Qilian Liang and Jerry M. Mendel, *Interval type-2 fuzzy logic systems: Theory and design*, IEEE Transactions on Fuzzy Systems **8** (2000), no. 5, 535–550.
7. Weldon A. Lodwick and K. David Jamison, *Theoretical and semantic distinctions of fuzzy, possibilistic, and mixed fuzzy/possibilistic optimization*, Fuzzy Sets and Systems **158** (2007), no. 1, 1861–1872.
8. Miguel Melgarejo, *Implementing Interval Type-2 Fuzzy processors*, IEEE Computational Intelligence Magazine **2** (2007), no. 1, 63–71.
9. Miguel Alberto Melgarejo, *A Fast Recursive Method to compute the Generalized Centroid of an Interval Type-2 Fuzzy Set*, Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS), IEEE, 2007, pp. 190–194.

10. Jerry Mendel, *Uncertain rule-based fuzzy logic systems: Introduction and new directions*, Prentice Hall, 1994.
11. Jerry M. Mendel and Robert I. John, *Type-2 fuzzy sets made simple*, IEEE Transactions on Fuzzy Systems **10** (2002), no. 2, 117–127.
12. Jerry M. Mendel, Robert I. John, and Feilong Liu, *Interval type-2 fuzzy logic systems made simple*, IEEE Transactions on Fuzzy Systems **14** (2006), no. 6, 808–821.
13. J. L. Verdegay, *Fuzzy mathematical programming*, Fuzzy Information and Decision Processes (M.M. Gupta and E. Sanchez, eds.), North-Holland, Amsterdam, 1982, pp. 231–237.
14. H. J. Zimmermann, *Fuzzy programming and Linear Programming with several objective functions*, Fuzzy Sets and Systems **1** (1978), no. 1, 45–55.
15. H. J. Zimmermann and Robert Fullér, *Fuzzy Reasoning for solving fuzzy Mathematical Programming Problems*, Fuzzy Sets and Systems **60** (1993), no. 1, 121–133.

Hacia un sistema de marketing dirigido más eficaz y personalizado en redes sociales

Patxi Galán-García, Dr. Carlos Laorden Gómez, and Dr. Pablo García Bringas

DeustoTech Computing - S^{3}Lab, University of Deusto,
Avenida de las Universidades 24, Bilbao, Spain
{patxigg,claorden,pablo.garcia.bringas}@deusto.es

1 Introducción

Hoy en día, ¿quién no tiene uno o varios perfiles en alguna de las redes sociales (RRSS) existentes? Esta situación de comunicación social genera gran cantidad de información, siendo mucha de ella de carácter personal y estando accesible para cualquiera en Internet.

Esta información es clave para los intereses del marketing, pero su procesamiento no es trivial. Con este trasfondo, parece claro que el siguiente paso en la evolución de la comercialización va parejo con las nuevas tecnologías, especialmente con las RRSS. El problema de esta situación es no caer en el spam debido al bajo coste que supone su uso.

La solución a este problema es la personalización de los anuncios. Este paradigma ofrece nuevas oportunidades a empresas anunciantes donde, los usuarios finales no estarán saturados con anuncios no deseados, si no con ofertas que realmente les interesen logrando un mayor impacto. Para lograr este objetivo, es necesario obtener información sobre el usuario pero, por defecto, a los usuarios no les gusta dar dicha información, por lo que encontrar otros sistemas para obtenerla es vital.

Con estos hechos, se quiere postular una metodología para obtener los temas de las conversaciones de los usuarios en RRSS y sistemas de mensajería instantánea (SMI) para generar anuncios más personalizados y eficientes.

2 Hipótesis de partida

Los retos a los que se enfrentarán los sistemas publicitarios del futuro serán uno de los puntos de atención de la comunidad científica. Por ello, se ha establecido una hipótesis fundamental para la tesis: *Es posible generar una metodología capaz de mejorar la detección de las temáticas en las conversaciones y textos cortos para mejorar el marketing dirigido dentro de los sistemas de redes sociales y mensajería instantánea.*

Esta hipótesis trata de demostrar si es posible adaptar las campañas de marketing de manera personalizada, mediante el análisis de las conversaciones entre usuarios para maximizar el impacto y alcance de los anuncios.

3 Objetivos

El primer objetivo es **procesar las conversaciones** obtenidas de las RRSS y SMI, para después obtener los análisis morfológicos, sintácticos, semánticos y pragmáticos. Para lograrlo, es necesario procesar y filtrar una gran cantidad de conversaciones. El problema es que no todas están correctamente estructuradas y escritas. Los SMI generan una gran cantidad de información, pero hay algunos problemas como *lenguaje coloquial* (alteraciones del idioma que son adoptadas por el uso cotidiano), *abreviaturas* (contracciones o abreviaturas de palabras), *errores ortográficos* (términos no escritos correctamente), *más de un interlocutor* (es importante detectar todos los usuarios que toman parte en la conversación), *texto dividido* (fragmentación de las frases en diferentes líneas por la introducción de texto nuevo o más interlocutores). El segundo objetivo es **extraer los temas de las conversaciones**. Para lograr obtener los intereses del usuario mediante el análisis de sus conversaciones, se diseñará una metodología capaz de analizar y extraer el conocimiento que estamos buscando en dichas conversaciones, teniendo en cuenta algunos problemas de PLN como la ambigüedad. La metodología estará formada por 2 enfoques principales. El primero, recibirá las conversaciones, previamente etiquetadas, para entrenar los algoritmos de aprendizaje automático *Machine Learning (ML)* que construirán los clasificadores para después, clasificar nuevas conversaciones. Como nuestro interés se centra en la búsqueda de lo que los usuarios quieren en tiempo real, las conversaciones etiquetadas deben reflejar los cambios de temáticas que suelen sufrir las conversaciones. De esta manera, cada instancia etiquetada incluira varios temas, dependiendo de la conversación mantenida entre los usuarios. Después de adquirir la base de conocimiento, se utilizarán ML para clasificar las nuevas conversaciones no etiquetadas, teniendo en cuenta que esta clasificación tiene que ser capaz de mostrar los cambios de temáticas del diálogo para validar la precisión de nuestro método. Este enfoque es nuestro método off-line. En el segundo enfoque, se recurrirá a la base de conocimiento generada. La diferencia radicará en que, en vez de clasificar conversaciones completas, se extraerán las temáticas de cada frase en tiempo real. La idea es adaptar nuestro método para analizar, en tiempo real, las conversaciones de RRSS o SMI. Con este enfoque, trataremos de deducir los intereses de los usuarios según lo que hablan. Este enfoque es nuestro método on-line.

Resumiendo, si aplicamos nuestro método off-line, una vez recolectadas las conversaciones para generar campañas de marketing, obtendremos campañas para usarlas a medio plazo con las necesidades que el usuario ha estado pidiendo. Este tipo de publicidad, es la que más se usa hoy en día. Por otro lado, si aplicamos nuestro método on-line para generar ofertas a los usuarios, estaríamos usando la información más actualizada sobre sus intereses, mejorando el impacto y aumentando la predisposición del usuario a consumir los productos. El tercer objetivo es la **generación de campañas publicitarias personalizadas**. La idea es usar los temas obtenidos de las conversaciones analizadas para ofrecer a los usuarios ofertas personalizadas. Y, el cuarto objetivo es **divulgar las**

campañas generadas por Internet usando herramientas que nos ofrecen las RRSS para compartir.

4 Metodología y plan de trabajo

Para validar la hipótesis que se ha presentado, es necesario identificar y detallar ciertos hitos previos. Estos objetivos de carácter teórico y práctico que conforman la metodología básica, son fundamentales para terminar la parte que concierne a la investigación y tesis que aquí se presenta.

1. Revisar el estado del arte y obtener, leer, clasificar y revisar las publicaciones sobre la categorización de documentos, centrándonos en textos cortos, especialmente en conversaciones. A su vez, asistir a foros científicos que trabajen en el área en el que se desempeña la investigación.
2. Analizar las diferentes fuentes de datos para la construcción de una base de conocimiento, centrándonos en aspectos como el formato y decidir cuál utilizar, para despues diseñar un sistema capaz de unificar los datos de la fuente de datos seleccionada.
3. Generar las bases de conocimiento unificando las temáticas que ellas contengan basándose en la fuente de datos previamente seleccionada.
4. Diseñar un sistema de votos para evaluar los resultados de cada nuevo módulo que se implemente para la detección de temáticas.
5. Realizar experimentos con el sistema y los diferentes métodos y módulos desarrollados para validar los resultados obtenidos con data-sets previamente etiquetados.
6. Presentar y publicar los resultados obtenidos a la comunidad científica mediante foros de debate como congresos y seminarios.

La Figura.1 muestra gráficamente las 4 etapas del flujo final del sistema, tomando como referencia los objetivos principales de la futura tesis.

La primera etapa ya posee ciertos módulos desarrollados y probados como por ejemplo, el sistema de organización, corrección y traducción de palabras

Figura 1. Personalización de anuncios analizando conversaciones de usuarios en RRSS.

desde el lenguaje de los mensajes cortos (SMS) al lenguaje castellano. El sistema recupera el texto introducido, llamado Unidad Conversacional (UC) y lo analiza, obteniendo el texto en castellano. Este módulo tiene en cuenta 2 partes. La primera es que el usuario que escribe, puede hacerlo en una o más líneas, siendo todo una misma frase y la segunda es que, la persona que escribe puede hacer una pregunta o una sentencia y esperar antes de escribir otra sentencia.

Este sistema de recuperación de información, empieza a trabajar cuando la persona termina de escribir. Utiliza un contador de tiempo para la espera. Si la persona no escribe en cierto tiempo, se determina que no va a escribir más. La frase que ha generado se transforma en una UC y se almacena para su análisis previa aplicación de métodos de limpieza de caracteres y repeticiones.

Para el corrector ortográfico de algunos términos SMS, usamos un diccionario personalizado y la distancia de Levenshtein [1]. Un ejemplo de un mensaje SMS en ingles es: "lo Patxi wassup?", siendo su traducción "Hello Patxi, what's up?". Con esta técnica hemos obtenido buenos resultados en el análisis de textos desestructurados [2].

En el área de detección de temáticas, la validación es complicada debido a que depende de las diferentes interpretaciones que una persona le asigne. Como nosotros necesitamos una gran cantidad de datos etiquetados, nos hemos decantado por el formato AVE [3] por su facilidad a la hora de analizar y generar.

En los primero experimentos, hemos generado la base de conocimiento con artículos específicos de Wikipedia en castellano[1] y de la Real Academia de la Lengua Española (RAE[2]). Para la gestión de los diferentes análisis sintácticos, hemos utilizado la herramienta OpenSopurce *FreeLing* [4]. Esta herramienta nos permite obtener los diferentes significados, sentidos, valores y géneros de las palabras en la oración en varios idiomas. Por último, hemos utilizado la herramienta lucene [5] para almacenar todo el conocimiento que hemos generado, dividido en temáticas y contexto. La metodología específica para el desarrollo de la investigación parte de, una vez seleccionada la fuente de datos, descargarlos y procesar los diferentes artículos con la herramienta FreeLing para después, almacenar los propios datos y los resultados sintácticos y semánticos en el índice creado con lucene. De estos datos, previamente se han eliminado los *stop-words* [6], se ha aplicado *stemming* [7] para obtener la raíz de la palabra, se ha obtenido el sentido de la palabra dentro del contexto de la frase y, finalmente se ha almacenado en la base de conocimiento para su posterior uso en la extracción de temáticas. La gestión de los aspectos sintácticos de la oración cooren por cuenta de la herramienta FreeLing. Su objetivo es hacer el análisis sintáctico y la desambiguación de las oraciones. Nuestro sistema guarda las palabras de la frase en "contenedores sintácticos". Este contenedor almacena toda la información semántica y la desambiguación sintáctica acerca de su elemento principal en relación con el contexto de la frase analizada. Hemos tratado de obtener las temáticas de la oración utilizando dos bases de conocimiento o índices, creados con la herramienta lucene, diferentes. El primer índice contiene información ac-

[1] http://es.wikipedia.org
[2] http://www.rae.es

erca de las temáticas y el segundo sobre los sentidos de las palabras que contienen dichas temáticas. Decidimos separar la información acerca de los sentidos y las temáticas, para dar más importancia a estas últimas, ya que son las temáticas las que estamos buscando para deducir los intereses y necesidades de los usuarios mediante la inferencia de los temas dentro de la conversación. Por lo tanto, los sentidos sólo se utilizan para ayudar en la elección tema.

Si bien estos experimentos han demostrado una mejora en la detección de los temas dentro de textos cortos, el conocimiento que tenemos es demasiado pequeño y sólo puede clasificar conversaciones bajo unos pocos temas generales.

Actualmente, estamos experimentando con otros métodos para mejorar los resultados. Por ejemplo, la indexación de la información y las relaciones de DBpedia[3] para aumentar nuestra base de conocimientos y también estamos etiquetando manualmente muchos SMS para validar los resultados. Además, en una primera aproximación utilizando estas técnicas, hemos desarrollado un chat-bot, llamado *Negobot* [2], para detectar si el sujeto involucrado en una conversación tiene tendencias pedófilas.

5 Relevancia

Este trabajo de investigación y futura tesis, se dirige a mejorar las campañas de marketing actuales, añadiendo a la ecuación información actualizada sobre los intereses y las necesidades de los usuarios utilizando técnicas de PLN. Este nuevo aporte debería aumentar el impacto de los anuncios ofrecidos y mejorar sus posibilidades de éxito convirtiendose en marketing viral y boca a boca para mejorar su impacto.

References

1. T. Okuda, E. Tanaka, and T. Kasai, "A method for the correction of garbled words based on the Levenshtein metric," *Computers, IEEE Transactions on*, vol. 100, no. 2, pp. 172–178, 1976.
2. C. Laorden, P. Galán-García, I. Santos, B. Sanz, J. Gómez-Hidalgo, and P. Bringas, "Negobot: A conversational agent based on game theory for the detection of paedophile behaviour," in *Proceedings of the 5th International Conference on Computational Intelligence in Security for Information Systems (CISIS)*, 2012, in press.
3. A. Penas, Á. Rodrigo, V. Sama, and F. Verdejo, "Overview of the answer validation exercise 2006," *Evaluation of Multilingual and Multi-modal Information Retrieval*, pp. 257–264, 2007.
4. X. Carreras, I. Chao, L. Padró, and M. Padró, "Freeling: An open-source suite of language analyzers," in *Proceedings of the 4th LREC*, vol. 4, 2004.
5. E. Hatcher, O. Gospodnetic, and M. McCandless, "Lucene in action," 2004.
6. W. Wilbur and K. Sirotkin, "The automatic identification of stop words," *Journal of information science*, vol. 18, no. 1, pp. 45–55, 1992.
7. J. Lovins and M. I. O. T. C. E. S. LAB., *Development of a stemming algorithm*. MIT Information Processing Group, Electronic Systems Laboratory, 1968.

[3] http://dbpedia.org

A Distributed Dynamic Norm Monitoring Framework for Evolving Institutions

Ignasi Gómez-Sebastià and Javier Vázquez-Salceda

Universitat Politècnica de Catalunya
Software Department (LSI)
c/Jordi Girona 1-3, E08034, Barcelona, Spain
{igomez, jvazquez}@lsi.upc.edu

Keywords: Multi-Agent Systems, Normative Systems, Monitoring, Governance

1 Introduction

Agent-based systems [10] are an alternative for implementing open, autonomous and dynamic computational systems. Bringing some organisational structure into autonomous systems is a way for regulating the behaviour of the different autonomous components to ensure their objectives are aligned with the holistic objectives of the system, effectively taming the complexity of the system. At the same time, in scenarios where multi-agent systems are applied to systems with an overall holistic goal, it is not desirable that an agent's autonomous and emergent behaviour diverges from the overall goal of the system.

Artificial social systems define an abstract social level over computational systems. Such level models the multi-agent system as a society of entities, defining structured patterns of behaviour that facilitate and enhance the coordination of agent activities [9]. The social structure is composed by roles, interaction rules, norms and a communication language. Norms (the most relevant element in the scope of this work) define the expected patterns of behaviour, imposing guidelines on the patterns of interaction among agents and facilitating them.

An Electronic Institution [9] is the model of an human institution with a norm specification provided in some machine-readable formalism, where the institution itself acts as a norm provider and enforcer. Such specifications are composed of regulative and non-regulative components. Regulative rules explain what should happen ideally (*e.g.*, the winner of an auction will pay for the product before leaving the auction room) and are typically represented in terms of deontic concepts [7] such as obligations, permissions and prohibitions.Constitutive rules (also known as counts-as rules) are able to transform low level facts perceived from the environment into high-level interpreted institutional facts. A Normative Multi-Agent System is a set of autonomous agents that must comply to social norms. Normative multi-agent systems are typically used to design electronic institutions.

2 Related work

The word *governance* derives from a Greek verb that means '*to steer*' and implies assuring that an organisation provides a good pattern of results while avoiding an undesirable pattern of bad circumstances. When governance is applied to Normative Systems results and circumstances are instantiated into states of the normative environment. Therefore, governance on Normative Systems can be seen as checking patterns of states of the normative environment.

One of the requirements when implementing Normative Systems is being able to assess, at runtime, the state of the normative environment (e.g., a norm has been violated, a sanction has been fulfilled, etc.). Some existing lines of research (*e.g.*, [1] [8]) already try to tackle this issue on some scenarios.

However, more complex scenarios can be defined, mainly scenarios with, at least, one of these properties. 1) The number of elements to take into account per time unit is too high. In other words, for every unit of time, too many norms have to be taken into account. This can happen either because the number of norms is high, or because at every time unit many new predicates (i.e., facts) come into the system to be interpreted. When many new predicates come into the system, many new norm instances may be activated and processed, even when the number of abstract norms is not specially high. Therefore, the system may not be able to compute the state of the normative environment at runtime. 2) The normative context, that defines the normative environment, is not static. The normative context expands and contracts as new norms are added to the institution and removed from it respectively. The system must be able to continue computing the state of the normative environment at runtime. Furthermore, the system must keep on producing states that are consistent with the modifications performed on the normative context (e.g., if a norm has been removed from the normative context, it might not make sense to compute its violation as part of the state of the normative environment from now on).

3 Objectives

Existing frameworks tend to fail when trying to apply them to complex scenarios as the ones above. In this thesis proposal we aim to tackle this problem, enriching and extending existing lines of research, with two main objectives:

- To formalise and develop an extended normative framework and architecture that can cope with scenarios where, due to the cost of computing the state of the normative environment, the number of elements to take into account per time unit is very high. We tackle this issue by distributing the computation of the normative environment on the cloud. The plan is to formalize a generic distribution method that can be used independently from the particular implementation of the framework we choose. Then, instantiate this formal method on a prototype and empirically test it on scenarios with a heavy load of events per unit of time.

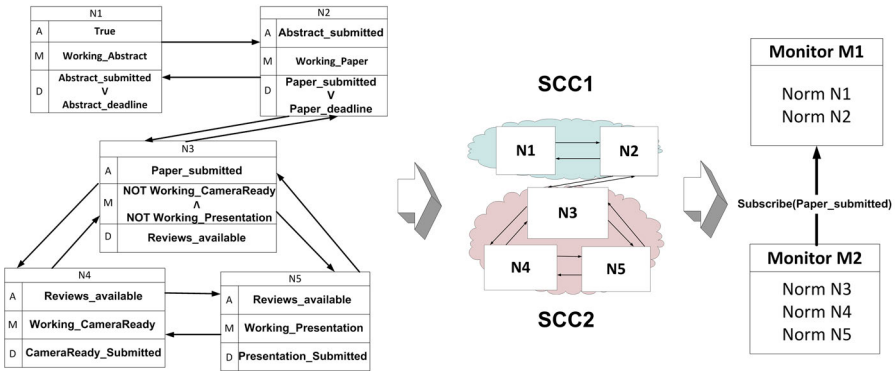


Fig. 1. Graph with norm dependencies, strongly connected components(SCC) and monitor distribution

– To extend the framework in the first objective to cope with scenarios where the normative context is dynamic, therefore it can expand and contract. Both operations are performed at runtime, without having to stop computing the normative state. The normative states computed are consistent with the expansion and contraction operations. We tackle this issue by formally defining the operations to be supported in order to allow for expanding and contracting the normative context. Then, we expand the formal framework, mainly providing support for a more expressive norm life cycle. Next, we formally implement expansion and contraction operations on the existing framework. Finally, we outline an architectural design that improves existing prototypes providing support for institutional agents.

The proposed research will effectively open further lines of research, such as support for multi-context normative environments (e.g., agents operating in Regional, National and International normative contexts at the same time) and contexts able to measure their performance and expand or contract consequently (e.g., detecting a particular norm is counter-productive for the overall system objectives and remove it from the normative context).

4 Methodology and Work Plan

Our proposal is to take an existing monitoring framework for normative contexts [1] and provide both theoretical and technological extensions, so it can cover both of our objectives.

Regarding distribution of the computational context, we have provided a proposal [4] that consists in performing an off-line analysis of the normative framework detecting dependencies between the different norms. Then, we model the normative context as a graph. As seen in *Figure* 1 nodes are norms and there is a directed node between two norms when they are dependent. Next,

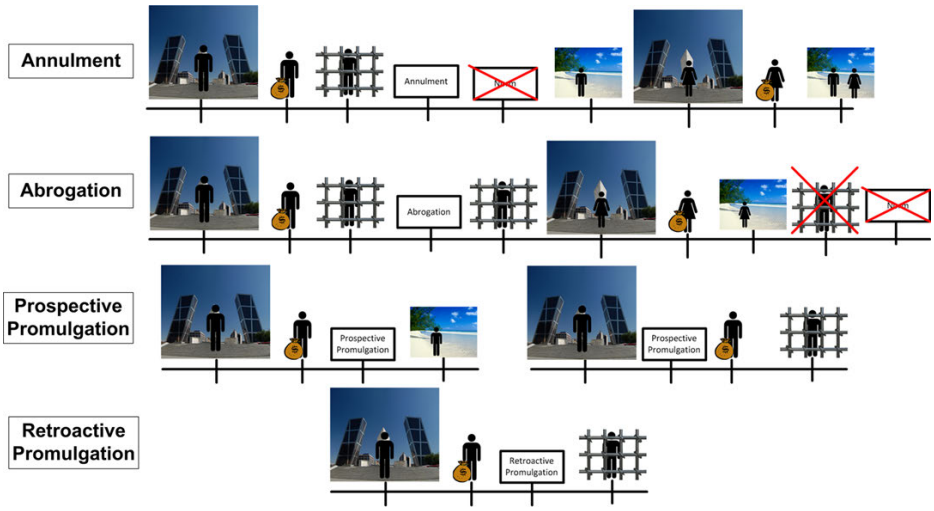


Fig. 2. Operation time lines

we perform a network analysis on the graph known as Strongly-Connected-Component detection [3]. textitFigure 1 shows the results of this analysis. The resulting graph known as *component graph* allows for reducing the original problem (a whole normative context) into several sub-problems (one per Strongly-Connected-Component) plus one more problem on the component graph (coordination among the different Strongly-Connected-Components). If such sub-problems are distributed among different computational nodes we have effectively distributed the computational process associated to the original problem, as seen in *Figure* 1. The *Strongly-Connected-Components* analysis guarantees that: 1) The formulas with a high number of information dependencies among them are grouped on the same Strongly Connected Component, therefore, they will be on the same sub-normative environment. 2) Information dependencies among formulas in different sub-normative environments are minimized.

Regarding the support for monitoring electronic normative environments that are dynamic (able to evolve through time as regulations change to adapt to new situations and behaviours) we have provided a proposal [5] consisting in four operations.. We present two operations for expanding and contracting the normative environment, effectively supporting adding new norms and removing existing ones. Each of these two operations comes in two forms, *Ex Tunc*(*i.e.*, from the outset) and *Ex Nunc* (*i.e.*, from now on) that will retroactively change the normative consequences of actions committed prior to the existence of the norm(from the outset) , or affect only actions committed after the existence of the norm respectively(from now on).

Figure 2 provides a graphical example of the operations. The scenario is based on the norm *'If an agent occupies a place in the management board of a*

national bank, it is forbidden for the agent to borrow money from the bank'. The examples of the four operations are as follows: *Annulment*: The agent occupies a place in the management board of a national bank and borrows money from the bank. The norm is violated and a sanction is enforced (agent goes to jail). Then, the norm is annulled. Sanctions are de-enacted (agent is released from jail and gets the money back) and the norm can not be violated any more. Therefore, another agent with a place in the management board can borrow money from the bank without legal consequences. *Abrogation*: The agent occupies a place in the management board of a national bank and borrows money from the bank. The norm is violated and a sanction is enforced (agent goes to jail). Then, the norm is abrogated. The norm can not entail further legal consequences but it remains in the system. Therefore, another agent with a place in the management board can borrow money from the bank without legal consequences. Once the sanctions applied to the violation of the norm have been fulfilled (*i.e.,* the first agent did his time in jail) the norm is effectively removed from the system. *Prospective Promulgation*: The agent occupies a place in the management board of a national bank and borrows money from the bank. The norm is promulgated prospectively. As the norm was not in place when the violation occurred, no sanction is applied. If the violation had occurred after norm promulgation, the agent would be sanctioned. *Retroactive Promulgation*: The agent occupies a place in the management board of a national bank and borrows money from the bank. The norm is promulgated retroactively. Even though the norm was not in place when the violation occurred, a sanction is applied.

We provide a formal model for the operations and propose a formal extension of the base normative monitoring framework that will allow it to support normative context modifications at run-time. That is, the normative environment can be expanded and contracted without having to stop the monitoring process. Furthermore, all expansions and contractions performed leave the normative context in a consistent state (*e.g.*, if a norm was violated, it will remain violated, unless an Ex Tunc contraction of the norm has been performed). The chapter includes informal descriptions of the actions to be performed on the normative context for each of the four supported operations. Then, formal algorithms implementing the operations are provided.

5 Relevance

Regarding distribution of the computational context, the main contribution to the state of the art is providing a formal method for distributing normative environments among a group of normative monitors on a cloud of computational resources. The basic idea behind this procedure is reducing a complex problem to N more simple sub-problems and distribute the computational resources associated to the problem solving. This would allow for applying normative monitoring processes to scenarios where either the number of events or the number of production rules (i.e., norms) to be taken into account is high.

Regarding support for dynamic normative contexts, the main contributions to the state of the art of proposals are: 1) A formal method for expanding and contracting normative contexts at run-time. The method mixes the expressivity of two existing lines of work [6] [2] effectively supporting both prospective and retroactive context expansion and contraction. 2) Formal descriptions of the four operations to be supported. Formal high-level algorithms for implementing them. 3) Analysis on the extension of the norm life-cycle required for supporting the operations. 4) Architectural design of the system to be implemented in order for supporting dynamic normative contexts.

Having support for dynamic normative contexts opens two main lines of research: 1) Adaptable normative contexts, able to insert or remove norms into the system depending on how the system evolves with respect to its high level objectives. 2) Support for multi-context monitoring, where the monitoring process can jump from one normative context to another just by applying the expansion and contraction operations. The first step on this line of research implies looking into norm meta-information (*i.e.*, metrics and heuristics about their performance,). Then, we can apply norms to a simulation framework (agent-based micro-simulation) and compare simulation results in scenarios where different norm combinations are applied.

References

1. S. Alvarez-Napagao, H. Aldewereld, J. Vázquez-Salceda, and F. Dignum. Normative monitoring: semantics and implementation. In *Coordination, Organizations, Institutions, and Norms in Agent Systems VI*, pages 321–336. Springer, 2011.
2. G. Aucher, D. Grossi, A. Herzig, and E. Lorini. Dynamic context logic and its application to norm change. *Normative Multi-Agent Systems*.
3. R. Bloem, H. N. Gabow, and F. Somenzi. An algorithm for strongly connected component analysis in n log n symbolic steps. In *Formal Methods in Computer-Aided Design*, pages 56–73. Springer, 2000.
4. I. Gómez-Sebastià, S. Alvarez-Napagao, and J. Vázquez-Salceda. A distributed norm compliance model. *CCIA*, 232:110–119.
5. I. Gómez-Sebastia, S. A. Napagao, J. V. Salceda, and L. O. Felipe. Towards runtime support for norm change from a monitoring perspective.
6. G. Governatori and A. Rotolo. Changing legal systems: Legal abrogations and annulments in defeasible logic. *Logic Journal of IGPL*, 18(1):157–194, 2010.
7. D. Grossi, H. Aldewereld, J. Vázquez-Salceda, and F. Dignum. Ontological aspects of the implementation of norms in agent-based electronic institutions. *Computational & Mathematical Organization Theory*, 12(2-3):251–275, 2006.
8. N. Oren, S. Panagiotidi, J. Vázquez-Salceda, S. Modgil, M. Luck, and S. Miles. Towards a formalisation of electronic contracting environments. In *Coordination, Organizations, Institutions and Norms in Agent Systems IV*, pages 156–171. Springer, 2009.
9. J. Vázquez-Salceda. The role of norms and electronic institutions in multi-agent systems applied to complex domains. the harmonia framework. *AI Communications*, 16(3):209–212, 2003.
10. M. Wooldridge, N. R. Jennings, et al. Intelligent agents: Theory and practice. *Knowledge engineering review*, 10(2):115–152, 1995.

Gene clustering with $\mathcal{K}$-Formal Concept Analysis

José María González-Calabozo
Directores: Carmen Peláez-Moreno[1] and Francisco J. Valverde-Albacete[2]

[1] Dpto. de Teoría de la Señal y de las Comunicaciones.
Universidad Carlos III de Madrid
Avda. de la Universidad, 30. Leganés 28911. Spain
jmgc,carmen@tsc.uc3m.es
[2] Departamento de Lenguajes y Sistemas Informáticos
Univ. Nacional de Educación a Distancia, c/ Juan del Rosal, 16. 28040 Madrid, Spain
fva@lsi.uned.es

Keywords: Gene Expression Data analysis, biclustering, coclustering, conceptual clustering

1 Introduction and Related Work

Genetics is a science with growing importance due to the promise, among others, of personalized health treatment at low costs or the deciphering of metabolic operation at the cell level.

Gene expression analysis is the basic task of interpreting co-regulation between genes implicit in a set of gene expression measurements, where co-regulation is the simultaneous increase or decrease of the expression level of several genes or groups of them. It is hypothesized that such Gene Expression Data (GED) measurements can provide information about how the cell reacts to specific *(operating) conditions*, e.g. what kind of cell it is, whether it is under stress, etc. Such data adopt the form of a *gene expression profile* for each condition.

Gene expression measurement. Microarray chips are able to measure concentration levels for thousand of known mRNA chains present in a cell in just one run, and they can be used to analyze cells from different experiments. The output of each of these microarray chip measurements is a *profile* of mRNA concentration levels—where each row identifies a gene—referred to as a*(n experimental) condition (of expression)*. Each condition is related to the operating mode of the cells being measured e.g. a different tissue sample, or cell environment, etc. GED from these conditions are thus put together as a *Gene Expression Array* (GEA) where cells reflect the expression value of a gene in a row in the particular conditions of the column.

Costly measurements make the number of gene profiles available low: hence a typical database has thousand of genes and just a dozen of different (expression) conditions. However, this problem is being addressed by means of open data initiatives and crowdsourcing projects: Many of these GEA have been made publicly available, with the expectation that sharing these data will raise the

pace of discovery and lower its costs. In this thesis we will be using conditions from these databases.

The interest of GED lies in the *relative* expression level for the same gene in different conditions. Absolute values of mRNA levels are less interesting because there is not a straightforward relationship among concentrations at different conditions and the number of proteins synthesized or how these proteins affect the cell. Therefore the focus is on how the gene expression levels change from one experiment to other. We describe as *underexpression* the phenomenon of a joint decrease in gene expression values within groups of genes and as *overexpression* the opposite process of increase in such levels.

Note that GED profiles are indirect measures of gene activity through the measurement of the concentration levels of different mRNA chains (the sequence which codifies one protein at ribosomes) under different expression conditions, what compounds the problem of interpretation.

Furthermore, gene expression is a noisy, hardly repeatable process even in controlled conditions. Real-valued GED needs to be normalized because the key information appears in the changes in gene expression not in their absolute value. Thus **different data-contitioning strategies** must be defined and tested, as part of the exploration process.

Clustering for GED analysis. Clustering GED is, at present, possibly one of the most popular techniques to distinguish diseased from healthy tissue, or to identify genes responsible for some particular disease, among many other applications. However, traditional clustering techniques are mostly effective when searching for global patterns in data [1], but they lose performance when looking for a specific subset of conditions among them.

Also, clustering GED is further complicated by high noise in the samples and the low dimensionality of conditions compared with the number of genes to classify.

An alternative–widely used at present—is to use biclustering algorithms [2] —where the search is performed simultaneously by rows and columns [3,4,5]— used in exploratory mode. The idea is to find more robust clusters by combining the evidence in both genes and conditions.

Conceptual clustering for GED analysis. Formal Concept Analysis (FCA) is an analysis method whose origins date back to 1982 [6], but it was only in the 90's when it reappeared in several areas like linguistics, psychology, artificial intelligence, information retrieval or knowledge management [7]. FCA goes beyond biclustering in providing mutually supported evidence for gene and condition clusters in the form of isomorphic order lattices with more structure than mere biclusters, also in an exploratory setting. For this reason it has also been termed a *conceptual clustering*.

FCA uses lattice theory and Galois connections to generate two order lattices induced by a boolean relation: assuming the triple (G, M, I) a set G of n

objects—genes—, a set M of p attributes—expression conditions—and a boolean relation $I \in 2^{n \times p}$ between them, we defined the *polars* of this connection as:

$$A' = \{m \in M | (g,m) \in I \ \forall g \in A\} \qquad B' = \{g \in G | (g,m) \in I \ \forall m \in B\} \quad (1)$$

Then, a *formal concept* is a pair of a subset of objects $A \subseteq G$ and an associated subset of attributes of $B \subseteq M$, that are mutual images by the polars, that is $(A,B) = (A,A') = (B',B)$. There is an order relation between concepts that creates a complete (order) lattice $\mathfrak{B}(G,M,I)$ called the *concept lattice* of (G,M,I), that is, a conceptual bi-clustering in both genes and attributes.

There have been several proposals to use FCA to conceptually cluster genes [8,9,10], but all these techniques have problems when discretising gene expression levels to boolean values as needed by standard FCA algorithms. FCA is also known to be extremely sensitive to boolean noise in the data.

Also due to the lop-sided quality of GED data new **novel visualization techniques** for concept lattices targeted at easing gene identification tasks should be developed, or at least adapted from standard FCA visualization techniques.

$\mathcal{K}$-Formal Concept Analysis. One of the main limitations of FCA is that it only works with boolean data. In $\mathcal{K}$-Formal Concept Analysis, however, an object can have an attribute in a certain degree, that is, assuming $R \in \mathcal{K}^{n \times p}$ as incidence matrix representing GED, the value at $R_{ij} = \lambda$ indicates that object i has the attribute j in degree λ, or in our case, that a particular gene i is under or over-expressed to degree λ for the condition j . In this framework, $\mathcal{K}$ the algebra to express the degrees is an *idempotent semifield* with special properties.

In *$\mathcal{K}$-FCA*, like in *FCA*, the set of objects is given by a row vector $x \in \mathcal{X} \cong \mathcal{K}^{1 \times n}$ with row vectors in a *left $\mathcal{K}$-semimodule*[11]. The set of attributes is given by a column vector $y \in \mathcal{Y} \cong \mathcal{K}^{p \times 1}$ as column vectors in a *right $\mathcal{K}$-semimodule*. An incidence matrix $R \in \mathcal{K}^{n \times p}$ relates objects with attributes. The triplet $(G,M,R)_{\mathcal{K}}$ is called a *$\mathcal{K}$-valued context*. On vector spaces over idempotent semifields one may create a Galois connection whose polars are,

$$\overset{-}{\varphi}y = \bigvee \{x \in X | \langle y|x \rangle_R \leq \varphi\} \qquad x^{-}_{\varphi} = \bigvee \{y \in Y | \langle x|y \rangle_R \leq \varphi\}$$

where $\langle y|x \rangle_R = y^t Rx$ is just a scalar product defined between spaces.

A *formal φ-concept* of the context $(G,M,R)_{\mathcal{K}}$ is a pair $(x,y) \in X \times Y$ such that $x^{-}_{\varphi} = y$ and $\overset{-}{\varphi}y = x$. Therefore x is the *extent*, y is the *intent* and φ its lower degree of existence. The structural lattice $\underline{\mathfrak{B}}(G,M,I^{\varphi}_R)$ of a *$\mathcal{K}$-concept Lattice* $\underline{\mathfrak{B}}_{\varphi}(G,M,R)$ is the *Concept Lattice of the context* (G,M,I^{φ}_R) [12, def. 8].

In order to address all these problems, the purpose of this thesis is to develop and test **exploratory analysis algorithms** based in $\mathcal{K}$-Formal Concept Analysis, to carry out the conceptual clustering task on GED.

2 Objectives

We gather again the objectives stated in the previous section,

- To design and test different data-conditioning strategies for GED.
- To develop new visualization techniques targeted at easing GED exploratory tasks.
- To cast the gene co-regulation problem as conceptual clustering with $\mathcal{K}$-Formal Concept Analysis.

3 Methodology and working plan

The process which gives as a result the concept lattice is divided into three steps: first, the gene expression data is extracted, and next some data-conditioning is necessary prior to running the conceptual clustering algorithm. The first milestone of this thesis consists on finding the conditioning that produces the best clustering results for GED analysis.

After this normalization, the proposed clustering algorithm is executed on the data. This $\mathcal{K}$-Formal Concept Analysis and its visual exploration are the main focuses of this thesis and will be referred as to milestones 2 and 3.

Milestone 1: Data-conditioning. GED concentrations are highly variable due to the vast number of factors influencing expression in a particular type of experimental condition. Some data-conditioning is required to normalize the gene expression matrix rows in order to be able to measure the proper change in concentration between different experiments.

Note that the issue of **finding a suitable data-conditioning process** adequate for biclustering algorithms, in the context of semifield algebra, the maths behind $\mathcal{K}$-Formal Concept Analysis, has not been addressed by the community.

The first milestone of this thesis is to **obtain a basic, motivated data-conditioning process for GED to be afterwards processed by $\mathcal{K}$-Formal Concept Analysis.**

Milestone 2: Conceptual Clustering. Our chosen incidence matrix $R \in \mathbb{R}^{n \times p}$ is the gene expression matrix after normalization where objects are genes, and attributes are experimental conditions. Thus to complete the analysis it is necessary to select an adequate complete idempotent semifield $\mathcal{K}$.

We will **obtain different KFCA exploration pipelines on two semirings the max-plus $\overline{\mathbb{R}}_{\max,+}$ and min-plus $\overline{\mathbb{R}}_{\min,+}$ algebras**, that have previously proven adequate for confusion data analysis.

Max-plus algebra is expected to be suited to identifying underexpresed genes since the *φ-formal concepts* of this analysis will contain genes with lower expression levels for those experiments. Analogously, the min-plus algebra will be used to identify over-expressed genes.

The visualization facilities at this point will be the standard ones for FCA, but we will use the interaction with the data to **improve the data conditioning in the previous Milestone**.

Milestone 3: Data visualization technique. This thesis intends to **obtain different visualization devices and techniques to help a researcher find the desired information easily**.

To obtain all the possible φ-*formal concepts* it is necessary to explore all distinct values of $R \in \mathbb{R}^{n \times p}$ for the $\overline{\mathbb{R}}_{\max,+}$ and $\overline{\mathbb{R}}_{\min,+}$ domain. $\overline{\mathbb{R}}_{\max,+}$ exploration will sweep all the R values in the range $\varphi \in [\min(R), 0]$ where $\min(R)$ is the minimum value of matrix R. The sweep range for the $\overline{\mathbb{R}}_{\min,+}$ domain will be $\varphi \in [0, \max(R)]$ where $\max(R)$ is the maximum value of matrix R. For each value of φ there will be a different lattice yielding a long sequence of lattices.

Lattices can be constructed and presented in Hasse diagrams, but these should be consistent throughout different φ values to show how this threshold affects their behaviour: diagrams should present the concepts *at the same position for different* φ and their size should be proportional to the number of genes. It is also interesting to show the evolution of gene appearance with φ, which genes appear earlier or how persistent they are within a given concept for some predefined attributes.

Figure 1 shows a preliminary device to visualise a lattice of over-expressed genes. It is easy to pinpoint the more populated concepts which have more genes, and the partial order between concepts is evident. For instance, it is easy to see that the concept that belongs to the root genes grown on selenate has for a daughter the more general concept of roots (control and selenate). New forms of visualizations will be developed focusing on a concept and how it changes in the exploration range. A gene could even be tracked and compared with some other genes to see their evolution with φ. No user studies for these devices are

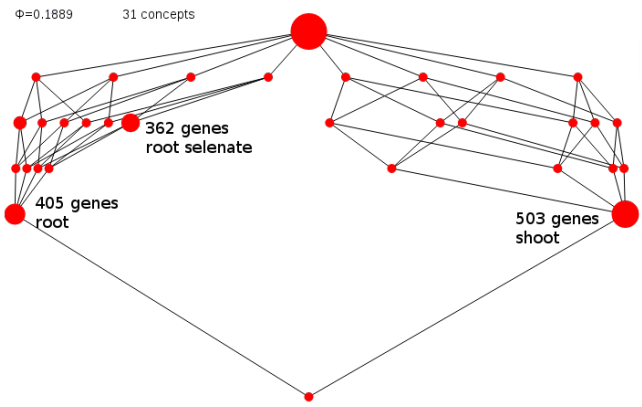


Fig. 1. This analysis consists of 8 samples of *Arabidopsis thaliana*. Two samples are root control and two shoot control tissues. The other samples belong to plants grown on selenate-rich environment: there are two samples from the roots and other two from the shoots. This lattice shows the concepts that appear in a $\overline{\mathbb{R}}_{\min,+}$ for a given threshold value of $\phi = 0.1889$.

intended in this thesis.

4 Relevance

Foremost, the results of our analysis can be used in clinical applications to identify genes which can cause diseases, or to know how the environment affects cell epigenetics. It can be used to identify key genes and also to classify tissues in an unsupervised mode based only on their expression profile. Although we are presenting a novel biclustering method as applied to GED analysis, yet it could, in principle, be used in any data analytics fields.

Furthermore, the new data visualization techniques evolved from Hasse diagrams will help identify key genes and important clusters, and also bring about a better understanding of the cluster components.

This thesis intends to provide a public web service that implements the described algorithm and visualization schemes to help researchers analyse their own data linked to detailed gene descriptions. Source code will be made available.

References

1. de Souto, M.C.P., Costa, I.G., de Araujo, D.S.a., Ludermir, T.B., Schliep, A.: Clustering cancer gene expression data: a comparative study. BMC bioinformatics **9** (January 2008) 497
2. Gupta, N., Aggarwal, S.: MIB: Using mutual information for biclustering gene expression data. Pattern Recognition **43**(8) (August 2010) 2692–2697
3. J. A. Hartigan: Direct Clustering of Data Matrix. Journal of the American Statistical Association **67** (1972)
4. Roy, S., Bhattacharyya, D.K., Kalita, J.K.: CoBi: Pattern Based Co-Regulated Biclustering of Gene Expression Data. Pattern Recognition Letters (March 2013)
5. Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: a survey. IEEE/ACM Trans. Comput. Biol. Bioinformatics **1** (2004)
6. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Berlin, Heidelberg (1999)
7. Priss, U.: Formal Concept Analysis in Information Science. Review of Information Science **40**(1989) (2002) 1–22
8. Gebert, J., Motameny, S., Faigle, U., Forst, C.V., Schrader, R.: Identifying Genes of Gene Regulatory Networks Using Formal Concept Analysis. Journal of Computational Biology **15**(2) (2008) 185–194
9. Motameny, S., Versmold, B., Schmutzler, R.: Formal Concept Analysis for the Identification of Combinatorial Biomarkers in Breast Cancer. Gene (2008)
10. Kaytoue, M., Kuznetsov, S.O., Napoli, A., Duplessis, S.: Mining gene expression data with pattern structures in formal concept analysis. Information Sciences **181**(10) (2010) 1989–2001
11. Golan, J.S.: Semirings and their Applications. Kluwer Academic (1999)
12. Valverde-Albacete, F.J., Peláez-Moreno, C.: Galois Connections between Semimodules and Applications in Data Mining. Formal Concept Analysis (2007) 181–196

Contributions to weakly supervised classification problems

Jerónimo Hernández-González

Advisors: Iñaki Inza and Jose A. Lozano
Intelligent Systems Group, University of the Basque Country UPV/EHU
P. Manuel de Lardizabal 1, 20018 Donostia-San Sebastián, Spain
Emails: {jeronimo.hernandez,inaki.inza,ja.lozano}@ehu.es

Keywords: Generalized definition of supervision, Weakly supervised learning, Bayesian network classifiers

1 Introduction

In supervised classification, the aim is to reproduce the inherent categorizing behavior of a real problem of interest. This behavior can be represented as a function Π that, given an example of the problem, assigns a value to the class variable C from a set of possible categories $\mathcal{C}$ (or class labels). The space of examples $\mathcal{X}$ is defined by means of a set of measurable features, a.k.a. predictive variables $(X_1, \ldots, X_n)$. Assuming the existence of a probability distribution $p(\mathbf{x}, c)$ underlying the problem which determines the lowest error rate that can be achieved by any classifier (*Bayes error*), the function Π returns the label c that shows the largest posterior probability $p(c|\mathbf{x})$ for the provided example $\mathbf{x}$.

In this context, the term *supervised* refers to the class information provided together with the training instances. According to the classical definition of supervised classification, the problem examples —a set of class-labeled instances $(\mathbf{x}_i, c_i)$, with $\mathbf{x_i} = (x_{i1}, \ldots, x_{in}) \in \mathcal{X}$ and $c_i \in \mathcal{C}$— are used to build a classifier f (*learning stage*) that tries to reproduce the categorizing behavior (Π) of the problem. Thus, f will be able to predict the class labels of a new unlabeled instance (*prediction stage*).

However, this definition does not cover other frameworks of classification presented by the machine learning community during the last years, which go beyond the classical description of (semi-)supervised problems.

2 Hypothesis

Similar to [3], this PhD thesis project considers a generalized definition of supervision based on three features extracted from these novel problems:

– Alternative information of supervision —class uncertainty— in the data provided at the learning stage [5, 6] and/or the prediction stage [7, 9], in contrast to the classical labeled $(\mathbf{x}_i, c_i)$ or unlabeled $(\mathbf{x}_i)$ examples.

Single Instance		Multiple Instances	
Single Label	Multiple Labels	Single Label	Multiple Labels
$\Pi : \mathcal{X} \to \mathcal{C}$	$\Pi : \mathcal{X} \to 2^{\mathcal{C}}$	$\Pi : 2^{\mathcal{X}} \to \mathcal{C}$	$\Pi : 2^{\mathcal{X}} \to 2^{\mathcal{C}}$

Table 1. Four definitions of function Π depending on the instance-label relationship.

– Underlying instance-label relationship —alternative definitions of function Π (Tab. 1)— of the problem [2, 12, 13].

In the field of weakly supervised problems, the state-of-the-art is large and in part confusing. Researchers of different communities have derived similar, or even the same, non-standard classification problems from different background applications, developing their own proposals without knowing each other. In this context, we *propose* a *novel* taxonomy based on the three characteristics exposed above, which may lead to a better understanding of the field, revealing similarities and differences between weakly supervised problems —currently under revision in the Journal of Machine Learning Research (JMLR).

In the light of this idea of generalized supervision, in this thesis we project to deal with specific non-standard classification problems. Thus, motivated by real applications, we will try to provide a solution to the problems of *learning from label proportions* (LLP) and *learning from multi-dimensional crowds* (MDC).

The *learning from label proportions* (LLP) is a weakly supervised classification problem where the examples are provided unlabeled and grouped in *bags* —or disjoint sets of examples—. Although the labels of the instances in a bag are actually known, the individual pairing relation $(\mathbf{x}, c)$ is lost and only the proportions of instances in the bag that belong to each class label are known. The dataset, in addition to a set of m unlabeled instances, specifies the distribution of instances in b bags ($D = \mathbf{B}_1 \cup \mathbf{B}_2 \cup \cdots \cup \mathbf{B}_b$ where $\mathbf{B}_i \cap \mathbf{B}_j = \emptyset, \forall i \neq j$). Each bag $\mathbf{B}_i$ groups m_i instances, where $\sum_{i=1}^{b} m_i = m$, and the related p_{ic} denotes the proportion of instances in $\mathbf{B}_i$ which have class label c, with $\sum_{c \in \mathcal{C}} p_{ic} = 1$.

That the label proportions provide relevant class information, useful to improve the accuracy of the learnt classifiers, is shown in our paper recently published in the *Pattern Recognition* journal [5], where we propose a competitive algorithm based on the Structural EM (SEM) strategy to learn Bayesian network classifiers (BNC) from this kind of data. One of the main *contributions* was the modification of the classical SEM in order to build a robust method that considers joint label assignments; that is, only the individual label assignments which globally fulfill the label proportions of its bag are considered, reducing the number of possible assignments from $|\mathcal{C}|^{m_i}$ to $m_i! / \prod_{c \in \mathcal{C}} (m_i \cdot p_{ic})!$.

Our interest in the LLP problem is due to its application to the problem of embryo selection in *assisted reproductive technology*. As expected, the next step will be to apply the developed technologies to this real problem in collaboration with the *Unidad de Reproducción Asistida* of the *Hospital Donostia* (Spain).

In the *learning from multi-dimensional crowds* (MDC), the examples are provided without the true labels (a.k.a. gold-standard), and only the labels assigned by multiple (non-expert) annotators are available. As in a multi-dimensional (MD) classification problem [1] there are several class variable $(C_1, \ldots, C_d)$, the

label space $\mathcal{C} = \mathcal{C}_1 \times \cdots \times \mathcal{C}_d$ is composed of all the combinations of label assignments to the d class variables —i.e. label configurations, $\mathbf{c}_i = (c_{i1}, \ldots, c_{id}) \in \mathcal{C}$. In contrast to the typical MD example $(\mathbf{x}_i, \mathbf{c}_i)$, an example of a crowdly labeled MD problem $(\mathbf{x}_i, \mathbf{A}^i)$ provides a $(t \times d)$-matrix $\mathbf{A}^i$ such that A^i_{ac} indicates the class label predicted for class variable C_c by annotator L_a.

In the unidimensional framework, the ability of learning from *crowds* has been shown [11, 10]. However, the MD framework involves additional difficulties that require a serious study of reliability measures, gold-standard estimation, relations between classes and/or annotators, etc. Our first approach to the MDC problem, a SEM method that estimates the reliability of each annotator for predicting each class variable, has been accepted for publication in *CAEPIA'13*.

The motivating real applications (text, image or video labeling) come from the field of multi-label [12] learning. In our solution, they are transformed to the multi-dimensional framework: the multi-label learning paradigm can be described as a multi-dimensional problem in which there are as many binary class variables as class labels in the multi-label problem [1, 12]. Following this idea, the thesis project will consider to *test* the proposed method(s) in a real application[1].

Certainly, additional challenges that require an adequate solution arise when dealing with *real* weakly supervised data. For example, the existing theoretically founded frameworks for *evaluation* have been mainly developed under the assumption of full information of supervision in training —i.e. the real class label(s) of each example are known. But in the training stage of a weakly supervised problem, the information of supervision may be incomplete. Only when simulated datasets are used, this problem can be skipped by separating a subset of full-labeled instances for validation [5, 8]. However, when dealing with *real* datasets, this tricky solution can not be applied. Thus, it may be interesting to *analyze* from an abstract viewpoint the theoretical implications of using weakly supervised data for evaluation.

In the real applications of the specific problems presented above (LLP and MDC), we are not aware of any framework to evaluate with the real weakly labeled datasets. In [4], we presented a very basic ad-hoc solution for the LLP problem, together with a comparison with the classical simulated technique. Extending this idea, a necessary fair (and theoretically founded) framework for evaluation with weakly labeled data could be developed.

3 Objectives

- To present a taxonomy which integrates the non-classical supervised classification problems, structuring and improving the understanding of the field.
- To study the learning from label proportions (LLP) and multi-dimensional crowds (MDC) problems and propose algorithms to learn Bayesian network classifiers from the specific kind of data.
- To analyze a real application of the LLP and MDC problems and present proposals to deal with both.

[1] Crowdly annotated data can be easily obtained from the Web (E.g. mturk.com)

– To propose a specific framework for evaluation with the weakly supervised data of the LLP and MDC problems and a general theoretical analysis.

4 Methodology and Work Plan

We present a general description of the methodology followed in order to fulfill the objectives of this PhD thesis: Once an objective is selected, the efforts are focused in addressing the challenge and reaching a solution, which will be written and sent to a referred journal of the JCR for evaluation (preferably to one of the first quartile). A preliminary version of the results may be presented in a conferences of the field. In detail, the presented objectives can be elaborated as:
Objective 1: To present a taxonomy which integrates the non-classical supervised classification problems:

Task 1 State-of-the-art revision of the weakly supervised classification problems.
Task 2 Extraction of characteristics of the problems and identification of the relevant ones taking into account its influence in the kind of supervision provided.
Task 3 Write a proposal of taxonomy that structures the literature and may lead to a better understanding of the field.

Objective 2: To study the LLP problem and propose algorithms to learn BNC from this kind of data (similar for the MDC problem):

Task 1 State-of-the-art revision for this and related problems, and analysis of their mathematical models. State-of-the-art revision of Bayesian network classifiers and techniques to learn from data with missing values.
Task 2 Implementation of a *novel* framework to test artificial LLP datasets, where a wider range of LLP scenarios of increasing complexity can be obtained.
Task 3 *Implementation* and evaluation of the methods designed for LLP problems.
Task 4 Write a paper where the process followed until reaching the objective is described and the obtained results are analyzed.

Objective 3: To analyze a real application of the LLP problem and present a proposal to deal with it (similar for the MDC problem):

Task 1 State-of-the-art revision for previous proposal to solve the chosen real application of the LLP problem: the embryo selection problem.
Task 2 Data collection and analysis together with the domain experts.
Task 2 Study and analysis of the problems related with the use of weakly labeled data in evaluation. To *propose* a specific framework for the LLP problem.
Task 3 *Application* of our LLP techniques and the evaluation framework to the real problem. Analysis of the results in collaboration with the domain experts.
Task 4 Write paper(s), from either an applied or methodological point of view, describing the process followed until reaching the objective.

Objective 4: To propose a framework for evaluation with weak supervision:

Task 1 State-of-the-art revision frameworks of evaluation in the case of classical supervision, and related work.

Task 2 General theoretical analysis of the implications of using weakly supervised data for evaluation. Study in the specific problems (LLP and MDC).
Task 3 To *propose* a framework for method evaluation in the specific weakly supervised classification problems (LLP and MDC).
Task 4 Write paper(s) to present the developed framework(s) and the theoretical analysis that underlies the proposal.

5 Relevance

Although the number of presented problems is sharply increasing, a *taxonomy* of weakly supervised problems that organizes the field has never been presented. Additionally, as an intrinsic technique to supervised classification, a specific framework to *evaluate* weakly supervised classification methods is required for the sake of fairness. Both ideas rely on the fact that the majority of these are *real* problems that can not be evaluated as a classical supervised problem.

Regarding the specific problems (LLP and MDC), both have been chosen given their applicability. Currently, in an *ongoing project* with the *Unidad de Reproducción Asistida* of the *Hospital Donostia* (Spain), we study the problem of embryo selection in assisted reproductive techniques as a LLP problem [5].

With researchers of the *Plymouth Marine Laboratory* (UK), we have recently started a new project for studying the biomass and fish population of the sea which could be modeled as a LLP regression problem [8].

Taking advantage of the crowd information obtained from the Web is a hot topic in machine learning research. Up to our knowledge, the ML/MD framework that we present is a *novel* paradigm with multiple real applications.

References

1. Bielza, C., Li, G., Larrañaga, P.: Multi-dimensional classification with Bayesian networks. International Journal of Approximate Reasoning 52(6), 705–727 (2011)
2. Dietterich, T.G., Lathrop, R.H., Lozano-Pérez, T.: Solving the multiple instance problem with axis-parallel rectangles. Artificial Intelligence 89(1–2), 31–71 (1997)
3. García-García, D., Williamson, R.C.: Degrees of supervision. In: Advances in Neural Information Processing Systems 25 (NIPS 2011). pp. 897–904 (2011)
4. Hernández-González, J., Inza, I.: Learning naive Bayes models for multiple-instance learning with label proportions. In: Proc. of 14th CAEPIA. pp. 134–144 (2011)
5. Hernández-González, J., Inza, I., Lozano, J.A.: Learning Bayesian network classifiers from label proportions. Pattern Recognition (2013), in press
6. Jin, R., Ghahramani, Z.: Learning with multiple labels. In: Advances in Neural Information Processing Systems 15 (NIPS 2002). pp. 897–904 (2002)
7. Kuncheva, L.I.: Full-class set classification using the Hungarian algorithm. International Journal of Machine Learning and Cybernetics 1(1–4), 53–61 (2010)
8. Musicant, D.R., Christensen, J.M., Olson, J.F.: Supervised learning by training on aggregate outputs. In: Proceedings of the IEEE-ICDM. pp. 252–261 (2007)
9. Ning, X., Karypis, G.: The set classification problem and solution methods. In: Proceedings of the SDM 2009. pp. 847–858 (2009)

10. Raykar, V.C., Yu, S., Zhao, L.H., Valadez, G.H., Florin, C., Bogoni, L., Moy, L.: Learning from crowds. Journal of Machine Learning Research 11, 1297–1322 (2010)
11. Snow, R., O'Connor, B., Jurafsky, D., Ng, A.Y.: Cheap and fast - but is it good? evaluating non-expert annotations for natural language tasks. In: Proceedings of the Conference on Empirical Methods in NLP. pp. 254–263 (2008)
12. Zhang, M.L., Zhou, Z.H.: A review on multi-label learning algorithms. IEEE Transactions on Knowledge and Data Engineering (2013), in press
13. Zhou, Z.H., Zhang, M.L., Huang, S.J., Li, Y.F.: Multi-instance multi-label learning. Artificial Intelligence 176(1), 2291–2320 (2012)

Optimización de Redes de Sensores Inalámbricos mediante Computación Inteligente

Doctorando: Jose M.Lanza-Gutierrez[1]
Director: Juan A.Gomez-Pulido[1]

Universidad de Extremadura
Departamento de Tecnología de Computadores y Comunicaciones
Escuela Politécnica, Campus Universitario s/n, 10003 Cáceres (España)
{jmlanza, jangomez}@unex.es

Keywords: computación evolutiva, optimización multiobjetivo, redes de sensores inalámbricos, routers

1. Introducción

Este proyecto de tesis aborda el estudio y optimización en el marco de una de las tecnologías de mayor proyección futura dentro del área de las comunicaciones inalámbricas: las redes de sensores inalámbricos [1] (WSNs, del inglés Wireless Sensor Networks), redes que constituyen un tópico de investigación y desarrollo de gran actualidad en los ámbitos académico e industrial.

Las redes de sensores inalámbricos aportan importantes ventajas que las presentan como una buena alternativa a otros tipos de redes tradicionales: utilización de dispositivos económicos, ausencia de cableado, bajos costes de despliegue, etc. Estos factores, entre otros, permiten que las WSNs puedan ser desplegadas en entornos en los que el despliegue de otro tipo de redes sería totalmente inviable. De esta forma, podemos encontrar una larga lista de aplicaciones. tanto en entornos civiles como en militares (agricultura intensiva, monitorización medioambiental, control de tráfico, operaciones de rescate, vigilancia, ...) [2].

Sin embargo, no todo son ventajas en la utilización de las WSNs, ya que también presentan importantes inconvenientes que aún no han sido resueltos. En esta tesis centramos nuestro esfuerzo en tratar de optimizar algunos de los factores que resultan más atractivos para la industria, como son aquellos que afectan a la calidad de servicio ofrecida y a los costes económicos asociados.

En la literatura actual podemos encontrar un gran número de publicaciones que tratan de optimizar las WSNs. La mayoría de ellas estudian las WSNs desde un punto de vista tradicional, donde las redes están compuestas de una serie de sensores que captan información acerca de su entorno y son alimentados por baterías, y por un nodo central que recopila toda la información recibida por los sensores. Como es de esperar, al estar los sensores alimentados por baterías, estas redes tienen una fuerte dependencia del consumo energético. Cada vez que un sensor captura un paquete de información debe enviarlo hacia el nodo colector, lo que implica un desgaste energético. A su vez, es habitual que los sensores se

encuentren a varios saltos del nodo central, por lo que el paquete pasaría por sensores intermedios, lo que genera aún un consumo mayor.

En los últimos años ha surgido una novedosa línea de investigación, en la que unos nuevos dispositivos especializados en tareas de comunicación han sido incorporados a las WSNs: los routers [2]. Los routers alivian la carga de trabajo de los sensores en tareas de retransmisión, lo que afecta positivamente a aspectos tales como el tiempo de vida de la red, los costes y la calidad de servicio ofrecida, entre otros. Permitiendo además que el tamaño de las WSNs pueda ser incrementado sustancialmente, y por tanto su campo de aplicación. A lo largo de este proyecto de tesis, abordaremos el problema del posicionamiento eficiente de estos nuevos dispositivos a las WSNs tradicionales.

2. Trabajos relacionados

El posicionamiento eficiente de WSNs ha sido definido a lo largo de la literatura como un problema de optimización np-completo [3], por lo que podemos encontrar un gran número publicaciones tratando de aportar soluciones. Puesto que la resolución de un problema np-completo requiere de un esfuerzo computacional muy elevado, es necesario la utilización de técnicas no habituales con el propósito de facilitar su resolución. En este trabajo hemos optado por utilizar la computación evolutiva [4] (*EC*, del inglés *Evolutionary Computation*). Este conjunto de técnicas avanzadas de computación, se caracteriza por ofrecer resultados de calidad en tiempos de cómputo reducido; sin embargo, el número de trabajos que utilizan estas técnicas para optimizar las WSNs es muy reducido, menos aun si consideramos entre problema desde un punto de vista multiobjetivo, lo cual supone una importante novedad añadida.

Si nos centramos en la optimización de las WSNs desde el punto de vista tradicional podemos encontrar un gran número de publicaciones. Por ejemplo, utilizando heurísticas podemos destacar a [3, 5–7], así como utilizando la EC a [8–13]. Optimizando las WSNs con routers podemos encontrar un menor número de publicaciones, por ejemplo utilizando heurísticas a [14–16], y utilizando la EC un menor volumen aún, por ejemplo a [17] y a [18].

3. Objetivos

El principal objetivo de este proyecto de tesis doctoral reside en el despliegue eficiente de routers en WSNs, teniendo en cuenta el mayor número posible de parámetros de la red (cobertura ofrecida, consumo energético, robustez, costes de despliegue), escenarios reales, métodos determinísticos y un amplio conjunto de técnicas de computación avanzadas.

Para alcanzar este objetivo principal es necesario abordar previamente una serie de sub-objetivos, como son: la profundización en el conocimiento de las WSNs y el estado del arte actual, el estudio de las diversas técnicas de optimización multiobjetivo disponibles, la aplicación de estas técnicas al despliegue eficiente de WSNs, la creación de nuevos métodos algorítmicos que puedan ser

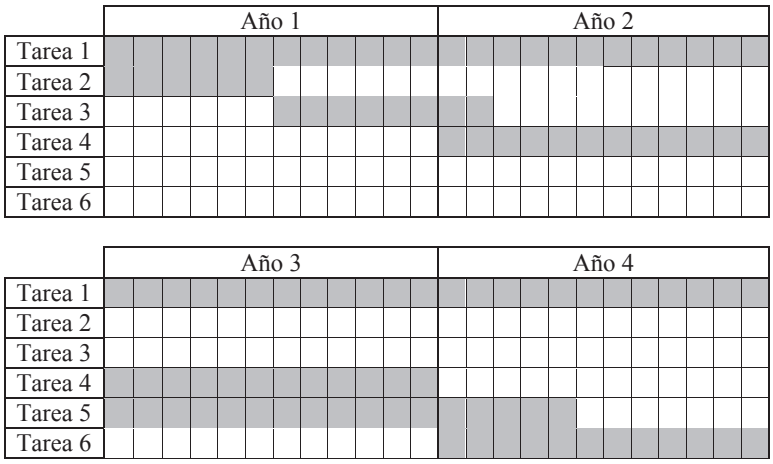


Figura 1. Cronograma del proyecto de tesis.

extrapolados a otros problemas de optimización, la utilización de la computación paralela (tanto en memoria compartida como en distribuida), así como la difusión intensiva de los resultados de la investigación mediante publicaciones en conferencias de prestigio (IEEE, ACM) y revistas con factor de impacto (ISI), con el objetivo de contribuir al avance del conocimiento y producir trabajo posteriormente utilizable y citado por otros autores. Estos sub-objetivos se traducirán en las tareas a desarrollar en esta tesis.

4. Metodología y Plan de trabajo

Todo proyecto de tesis debe seguir una correcta planificación para asegurar que todos y cada uno de los objetivos propuestos puedan alcanzarse. En la figura 4 se puede observar el cronograma que define cada una de las etapas por que pasará este proyecto, cuyas tareas describimos a continuación:

- Tarea 1: Documentación, diseminación de resultados y formación. Esta tarea es un conjunto de acciones que tienen como fin la preparación y consecución de una tesis doctoral. Incluye el estudio constante, la actualización de conocimientos y tecnologías, y la publicación de resultados mediante la asistencia a congresos científicos y la publicación de artículos en revistas de prestigio.
- Tarea 2: Definición del problema del despliegue eficiente de redes de sensores inalámbricos. Búsqueda bibliográfica de los diferentes modelos de caracterización de una red inalámbrica de sensores, modelos energéticos a utilizar y definición de los diversos factores susceptibles de ser optimizados en una red esta tipología: definición de las funciones objetivo.

- Tarea 3: Estudio de los diversas técnicas de resolución de problemas np-complejos, profundizando en la computación evolutiva. Dentro de esta disciplina contemplamos los dos subtipos más importantes: los algoritmos evolutivos y los algoritmos basados en inteligencia colectiva. Puesto que el problema tratado en este proyecto es multiobjetivo, se estudiarán las diversas medidas de calidad multiobjetivo existentes en la literatura.
- Tarea 4: Implementación de las metaheurísticas estudiadas en la tarea 2 aplicadas al despliegue eficiente de las WSNs caracterizadas en la tarea 3. Se comenzará utilizando los algoritmos estándares: NSGA-II [19] y SPEA2 [20], y se continuará por novedosos algoritmos basados en inteligencia colectiva (GSA [21], ABC [22], ...).
- Tarea 5: Utilización de la computación paralela (distribuida y de memoria compartida) con el fin de disminuir los tiempos de cómputo necesarios.
- Tarea 6: Propuesta de una metaheurística propia con los conocimientos obtenidos en las etapas anteriores. El objetivo es que los conocimientos adquiridos a la hora de resolver este problema puedan ser extrapolados a la resolución de otros problemas de optimización multiobjetivo.

Los medios necesarios para la realización de este proyecto son: un computador personal, un cluster de computación, el acceso a medios bibliográficos y la financiación necesaria para la asistencia a congresos.

5. Relevancia

Tanto la optimización multiobjetivo, como el despliegue eficiente de redes de sensores inalámbricos poseen una buena acogida en la comunidad científica. Ejemplo de ello es el gran número de publicaciones que podemos encontrar en foros de primer nivel (revistas indexadas y congresos internacionales de prestigio), tal y como vimos en la sección 2 (trabajos relacionados).

Estos trabajos pueden clasificarse en torno a dos principales líneas de investigación: la optimización de las WSNs tradicionales, y la optimización de las WSNs en las que se incorporan routers, esta última más novedosa y surgida con posterioridad. La mayoría de estos trabajos se centran en la utilización de heurísticas (técnicas especialmente adaptadas al problema que se desea resolver) monoobjetivo. Sin embargo, este problema de optimización, al igual que otros muchos, posee una clara concepción multiobjetivo, puesto que es habitual tener en cuenta múltiples factores a la hora de desplegar una red de esta tipología, como el consumo energético, la robustez o los costes de despliegue, entre otros.

Al igual que en otras áreas, hay una clara tendencia hacia la utilización de metaheurísticas (técnicas de resolución de problemas muy generales) multiobjetivo, la cual aún no ha sido muy explotada en la optimización de WSNs, y que por lo tanto, es una interesante línea de investigación aplicada al novedoso campo del posicionamiento eficiente de routers en WSNs.

Una muestra de la relevancia académica que aporta este proyecto de tesis es la buena acogida que han tenido los trabajos publicados durante las primeras

etapas de ésta, en la que se ha obtenido un total de 4 publicaciones en congresos nacionales, 2 congresos internacionales, 2 capítulos de libros de difusión internacional y una revista con factor de impacto ISI.

Es importante destacar que el estudio de las WSNs no solo tiene una gran transcendencia en el ámbito académico. Podemos encontrar grandes empresas en España (Telefonica, General Dynamics, Tecnatom, Lidax, Técnicas Reunidas) que desarrollan aplicaciones comerciales sobre las WSNs. Es por este motivo, por el que se estudiarán las vías necesarias para transferir este conocimiento hacia el ámbito industrial, por ejemplo mediante colaboraciones.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el Ministerio de Economía y Competitividad y el FEDER (Fondo Europeo de Desarrollo Regional), bajo el proyecto TIN2012-30685 (proyecto BIO), y por el Gobierno de Extremadura, a través de la ayuda GR10025 al grupo TIC015.

Referencias

1. Dargie, W., Poellabauer, C.: Fundamentals of Wireless Sensor Networks: Theory and Practice. Wiley (2010)
2. Mukherjee, J.Y.B., Ghosal, D.: Wireless sensor network survey. Computer Networks **52** (2008) 2292 – 2330
3. Cheng, X., Narahari, B., Simha, R., Cheng, M., Liu, D.: Strong minimum energy topology in wireless sensor networks: Np-completeness and heuristics. IEEE Transactions on Mobile Computing **2** (2003) 248–256
4. Zitzler, E.: Evolutionary algorithms for multiobjective optimization: Methods and applications (1999)
5. Cardei, M., Du, D.Z.: Improving wireless sensor network lifetime through power aware organization. Wireless Networks **11** (2005) 333–340
6. Zhang, H., Hou, J.: Maintaining Sensing Coverage and Connectivity in Large Sensor Networks. Ad Hoc & Sensor Wireless Networks **1** (2005)
7. Liu, L., Hu, B., Li, L.: Energy conservation algorithms for maintaining coverage and connectivity in wireless sensor networks. IET Communications **4** (2010) 786–800
8. Konstantinidis, A., Yang, K.: Multi-objective energy-efficient dense deployment in wireless sensor networks using a hybrid problem-specific moea/d. Applied Soft Computing **11** (2011) 4117–4134
9. Konstantinidis, A., Yang, K.: Multi-objective k-connected deployment and power assignment in wsns using a problem-specific constrained evolutionary algorithm based on decomposition. Computer Communications **34** (2011) 83–98
10. Martins, F., Carrano, E., Wanner, E., Takahashi, R., Mateus, G.: A hybrid multiobjective evolutionary approach for improving the performance of wireless sensor networks. IEEE Sensors Journal **11** (2011) 545–554
11. Hu, X.M., Zhang, J., Yu, Y., Chung, H.H., Li, Y.L., Shi, Y.H., Luo, X.N.: Hybrid genetic algorithm using a forward encoding scheme for lifetime maximization of wireless sensor networks. IEEE Transactions on Evolutionary Computation **14** (2010) 766–781

12. Jia, J., Chen, J., Chang, G., Tan, Z.: Energy efficient coverage control in wireless sensor networks based on multi-objective genetic algorithm. Journal of Computers and Mathematics with Applications **57** (2009) 1756–1766
13. Jia, J., Chen, J., Chang, G., Wen, Y., Song, J.: Multi-objective optimization for coverage control in wireless sensor network with adjustable sensing radius. Journal of Computers and Mathematics with Applications **57** (2009) 1767–1775
14. Wang, Q., Xu, K., Takahara, G., Hassanein, H.: Device placement for heterogeneous wireless sensor networks: Minimum cost with lifetime constraints. IEEE Transactions on Wireless Communications **6** (2007) 2444–2453
15. Han, X., Cao, X., Lloyd, E.L., Shen, C.C.: Fault-tolerant relay node placement in heterogeneous wireless sensor networks. IEEE Transactions on Mobile Computing **9** (2010) 643–656
16. Xu, K., Hassanein, H., Takahara, G., Wang, Q.: Relay node deployment strategies in heterogeneous wireless sensor networks. IEEE Transactions on Mobile Computing **9** (2010) 145–159
17. Zhao, C., Chen, P.: Particle swarm optimization for optimal deployment of relay nodes in hybrid sensor networks. In: Proceedings of IEEE CEC. Volume 1. (2007) 3316–3320
18. Perez, A., Labrador, M., Wightman, P.: A multiobjective approach to the relay placement problem in wsns. In: Proceedings of IEEE WCNC. Volume 1. (2011) 475–480
19. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast elitist multi-objective genetic algorithm: Nsga-ii. IEEE Transactions on Evolutionary Computation **6** (2000) 182–197
20. Zitzler, E., Laumanns, M., Thiele, L.: Spea2: Improving the strength pareto evolutionary algorithm. Technical report, Computer Engineering and Networks Laboratory (TIK), ETH Zurich (2001)
21. Rashedi, E., Nezamabadi-pour, H., Saryazdi, S.: Gsa: A gravitational search algorithm. Information Sciences **179** (2009) 2232 – 2248
22. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. J. of Global Optimization **39** (2007) 459–471

Metaheuristic Strategies for Scheduling Problems with Uncertainty

PhD Student: Juan José Palacios Alonso
PhD Supervisor: M.Camino Rodríguez Vela and Inés González Rodríguez

University of Oviedo and University of Cantabria
palaciosjuan@uniovi.es, crvela@uniovi.es, ines.gonzalez@unican.es
www.di.uniovi.es/iscop

1 Introduction

Scheduling problems have formed an important body of research during the last decades. A *scheduling problem* consists in scheduling a set of jobs $\{J_1, \ldots, J_n\}$ on a set of physical resources or machines $\{M_1, \ldots, M_m\}$. Each job J_i is composed of m tasks or operations $\{\theta_{i1}, \ldots, \theta_{im}\}$ with processing time p_{ij}. At the same time, we usually have constraints that establish that two task belonging to the same job cannot overlap their execution in time and that each task requires the uninterrupted and exclusive use of one of the machines for its whole processing time. Depending on the additional constraints we define, we may obtain different families of problems. The most popular in the literature are the job shop (JSP), the open shop (OSP) and the flow shop (FSP) but there exists also variants of them as the flexible job shop (FJSP) among others. Commonly, the objective function to optimise is the earliest time in which all jobs can be finished or *makespan*. However many other objectives may be optimised, being the most popular the *tardiness*, the *idleness* and the *total flow time*.

In classical scheduling problems all input data are assumed to be well defined and all constraints are assumed to be hard, which is not so common in real-life applications. To reduce the gap between theory and practice, this thesis focuses on solving scheduling problems considering that uncertainty and vagueness. For instance, we shall consider uncertain task durations as well as flexible due-date constraints.

2 Related Work

The complexity of scheduling problems means that practical approaches to solving them usually involve heuristic strategies. Metaheuristics have been extensively used and evolved during the last decades and in consequence there exists a wide range of them [12]. Among many others, we highlight a tabu search in [8] for the JSP, a local search in [7] for the flexible FJSP, and a hybridized ACO algorithm in [3] for the OSP.

In the last years, *fuzzy sets* have emerged as a very interesting tool to model uncertainty and have been used in different manners. This approach appears to

be more flexible as it needs less information from a human expert or past data for modelling a problem and it involves a lesser computational cost. Some attempts have been made to extend metaheuristics designed for the deterministic problem to the case where uncertain durations are modelled via fuzzy intervals, for instance a genetic algorithm in [11] and a local search strategy in [10] for the fuzzy JSP; or a PSO in [9] for the fuzzy OSP. When managing uncertainty, robustness and stability are important concepts to take in account. A solution is robust if it performs relatively well under a wide range of possible problem realizations [1]. However in the literature we find many discrepancies between different authors when applying this concept to scheduling problems. For instance it can be measured as the variation of the objective function depending on the real scenario, but also as the variation of the starting time of each single operation with respect to the predicted value or the best possible performance in the worst scenario.

3 Objectives

The main goal for this thesis is to efficiently solve different scheduling problems taking in account the uncertainty pervading real world problems. To achieve this goal, we may establish the following more detailed objectives:

- **Uncertainty model:** Having adopted fuzzy sets as a framework for dealing with uncertainty and vagueness, we intend to propose and develop analytic measures for optimality in this fuzzy framework, as well as studying the search spaces and related concepts such as dominance, criticality, etc.
- **Robustness and Stability:** When dealing with uncertainty, robustness and stability are important concepts to take in account. Unfortunately, there is no consensus in the definitions of these concepts and associated measures. Therefore the objective is to acquire knowledge of the different proposals in the literature, analyse them and adopt definitions for using during the thesis. Here we intend to fix metrics for these concepts that allow us to improve solutions accordingly.
- **Design of meta-heuristics:** The objective is to design different algorithms for solving fuzzy scheduling problems based on the adopted uncertainty model. This is not trivial, as the search spaces are now different and the properties of the deterministic problems may not hold in the fuzzy case. In addition, we plan to design multi-objective algorithms for optimising at the same time classical objective functions and the robustness of the solutions.

4 Methodology

In this section we describe the methodology to achieve each of the previous objectives. In addition, we briefly expose the preliminary results obtained in the early stages of the thesis.

4.1 Uncertainty model

The starting point for this objective is to make a review of the literature in order to know different techniques to deal with uncertainty and vagueness. The main source of uncertainty we shall consider is in task durations. The crudest representation for uncertain processing times would be a human-originated confidence interval. If some values appear to be more plausible than others, a natural extension is a fuzzy interval or a fuzzy number. For the thesis we adopt the use of *triangular fuzzy numbers* or *TFN* denoted as $N = (n^1, n^2, n^3)$, where $[n^1, n^3]$ defines an interval of possible values and n^2 the most plausible value in it. The number of authors using this model is increasing, which suggests that it is a good representation model. A first issue to be addressed in this framework is the arithmetic of TFNs and, in particular, the two operations needed in scheduling: addition and maximum. Extending the addition on real numbers to TFNs is straightforward, but this is not the case for the maximum. Indeed, the set of TFNs is not closed under the maximum and its computation can be computationally expensive. The main approaches in the literature are based on approximating the maximum by a TFN [4] [6]; we need to analyse the soundness and advantages and dissadvantages of each approach. Another issue is to define a total ordering in the set of TFNs, in order to optimise fuzzy objective values. In principle, we shall use the expected value to this end, taking into account its relationship with various ranking methods used by other authors.

A source of vagueness may be due-dates, as they can be flexible. A common approach for modelling flexible due-dates is to use a fuzzy set with linear decreasing membership function representing a satisfaction level for the ending date of a job. If task durations are modelled as TFNs, the completion time of a job is also a TFN, so we measure the satisfaction level using the *agreement index* [11].

To study the different search spaces of fuzzy scheduling problems where durations are modelled as TFNs, we have started to analyse and adapt schedule generation schemes, in special the well-known Giffler & Thompson algorithm (G&T), to those problems. So far we have seen that a minor change in the way we adapt the G&T has a great impact on the associated search space and we have identified different search spaces. We are currently studying their properties, with special emphasys on their dominance (i.e., if they contain at least one optimal solution).

4.2 Robustness and Stability

Regarding robustness and stability, the first step is to make a review of the different definitions and measures that are present in the literature and study the relation between those concepts and the adopted uncertainty model. However, so far we have found many discrepancies in the way these concepts are defined and measured in the scheduling literature. Our starting point has been the semantics for fuzzy schedules from [5], by which solutions should be understood as a-priori solutions, also called predictive schedules, providing a fuzzy interval of

possible values for the starting time of each task. We have used it as a basis to naturally adapt the definition of ϵ-robustness from [2] and associated metrics to the fuzzy framework. For evaluating the robustness of a predictive schedule we perform it on different scenarios using a MonteCarlo simulation and take the largest obtained ϵ value as robustness measure. In addition, when evaluating the performance of an algorithm we also run it without considering the uncertainty, taking as real duration for each task the most probable one (the modal value). Until now we have seen that solutions obtained considering the uncertainty are more robust than if we do not take it in account. Nevertheless, we are still studying different measures of robustness and in the near future we intend to start studying stability concepts.

4.3 Design of Metaheuristics

The plan is to start studying the most relevant algorithms published for deterministic scheduling problems and try to adapt and develop different extensions of them to the fuzzy framework. Based on the ideas and results obtained with these first approaches and the experience of our research group in designing metaheuristics for scheduling problems, we intend to develop new strategies to efficiently solve fuzzy scheduling problems. Our preliminary experience tell us that extend metaheuristics to the fuzzy framework is actually a challenge. Previously to the beginning of the thesis we had already developed a couple of metaheuristics for solving the fuzzy OSP and presented in IWINAC and Fuzz-IEEE conferences. During the firsts steps of the thesis we have developed different multi-objective approaches based on our previous experience. The first approach optimises the well known objective functions makespan and tardiness which in our case are both TFNs. To deal with both objectives, we have designed a swarm based algorithm with lexicographical goal programming which have obtained satisfactory results. Following a similar methodology, we have adapted our algorithm to deal with flexible due-dates instead of deterministic ones, considering the agreement index as objective function. The obtained results are again positive when comparing with other algorithms. The first approach has been presented in 2012 at the RCRA workshop and an extension of it, including a robustness analysis, is currently under review in a JCR indexed journal. The second approach has been sent to a journal and is currently under revision. Recently we have also implemented different search algorithms, including a Scatter Search, for the fuzzy OSP. Unfortunately, results are not competitive enough and it is open to future work how they can be improved.

We have also implemented different search algorithms for the fuzzy job shop problem. We have developed a tabu search algorithm based on the neighbourhood structures for the fuzzy framework proposed in [10], and then combined it with a genetic algorithm, obtaining very competitive results with the state-of-the-art both in quality and computational cost. This work has been already presented at the IWINAC 2013 conference.

Currently we are working on the flexible job shop problem with uncertainty, which is a topic of interest for many researchers. We have designed a neighbour-

hood for the fuzzy problem and developed a tabu search which we combine again with a genetic algorithm. The results clearly outperform the current state of the art for the fuzzy problems, and are also competitive in the most common crisp benchmarks. We are also working on a co-evolutionary algorithm with specific local search for each populations. Here the preliminary results are promising but we still testing and developing the algorithm.

For evaluating the results we shall obtain during the development of the thesis, we use the most extended benchmarks in the literature. For the fuzzy job shop, there are several problem instances of common use, but this is not the case for other fuzzy scheduling problems, with no specific benchmarks available for researchers. When this is the case, following a common approach in the literature, we generate a set of problems from the best-known benchmarks for the deterministic problem by transforming the original crisp processing times into TFNs such that their modal value corresponds to the original duration. It is desirable that the generated TFNs holds that $n^3 - n^2 \geq n^2 - n^1$ because in this case we can use the best known lower bound for the crisp problem as a lower bound for the fuzzy instance. This allows to better assess the quality of the obtained solutions.

For the future, once we adopt a measure for robustness, we intend to develop multi-objective algorithms using that measure as an objective function. Here we plan to adopt lexicographical goal programming with different hierarchies between the objective functions and dominance-based algorithms, and compare the obtained results so we can find the best way to optimise a classical objective function having robust solutions. In addition, based on the fact that multi-core processors are widely extended, parallel computing is one of our main targets.

5 Relevance

Fuzzy scheduling has been a topic of interest for many researchers during the last years. However it is still a relatively new line of research and there are many properties that have not been yet well-studied nor defined. For instance, even if many authors are adopting fuzzy numbers to deal with uncertainty, they are using different approaches and arithmetic. Also, it is well-known the importance of the different search spaces when facing complex problems, however in the case of fuzzy scheduling they have not been deeply studied yet, as well as schedule generator schemes.

Robustness and stability have not been well-defined yet for the scheduling problems as different authors propose different definitions. As a consequence, most researchers in fuzzy scheduling focus on optimise classical objectives, as makespan, but not on optimising the robustness or the stability of the obtained schedules, which are critical aspects when dealing with uncertainty. This supports the relevance of looking for an adequate measure of robustness and stability and designing multi-objective algorithms optimising not only a classical objective function, but also its robustness.

Meta-heuristics are important for solving fuzzy scheduling problems as they are NP-hard problems. Good meta-heuristics can yield good solutions for our problems in a short period of time, however the introduction of uncertainty in the problem makes it harder and a deep study is needed to develop new strategies. It is also remarkable that multi-core processors are widely extended nowadays, however few authors design meta-heuristics that exploit that computing capacity, therefore we consider relevant the design of parallel algorithms for solving our problems using that technology.

Acknowledgements

This research has been supported by the Spanish Government under research grant FEDER TIN2010-20976-C02-02.

References

1. Aytung, H., Lawley, M.A., McKay, K., Shantha, M., Uzsoy, R.: Executing production schedules in the face of uncertainties: A review and some future directions. European Journal of Operational Reseach 161, 86–110 (2005)
2. Bidot, J., Vidal, T., Laboire, P.: A theoretic and practical framework for scheduling in stochastic environment. Journal of Scheduling 12, 315–344 (2009)
3. Blum, C.: Beam-ACO—hybridizing ant colony optimization with beam search: an application to open shop scheduling. Computers & Operations Research 32(6), 1565–1591 (2005)
4. Fortemps, P.: Jobshop scheduling with imprecise durations: a fuzzy approach. IEEE Transactions of Fuzzy Systems 7, 557–569 (1997)
5. González Rodríguez, I., Puente, J., Vela, C.R., Varela, R.: Semantics of schedules for the fuzzy job shop problem. IEEE Transactions on Systems, Man and Cybernetics, Part A 38(3), 655–666 (2008)
6. Lei, D.: Solving fuzzy job shop scheduling problems using random key genetic algorithm. International Journal of Advanced Manufacturing Technologies 49, 253–262 (2010)
7. Mastrolilli, M., Gambardella, L.: Effective neighborhood functions for the flexible job shop problem. Journal of Scheduling 3(1), 3–20 (2000)
8. Nowicki, E., Smutnicki, C.: An advanced tabu search algorithm for the job shop problem. Journal of Scheduling 8, 145–159 (2005)
9. Palacios, J.J., González-Rodríguez, I., Vela, C.R., Puente, J.: Particle swarm optimisation for open shop problems with fuzzy durations. In: Proceedings of IWINAC 2011, Part I. Lecture Notes in Computer Science, vol. 6686, pp. 362–371. Springer (2011)
10. Puente, J., Vela, C.R., González-Rodríguez, I.: Fast local search for fuzzy job shop scheduling. In: Proceedings of ECAI 2010. pp. 739–744. IOS Press (2010)
11. Sakawa, M., Kubota, R.: Fuzzy programming for multiobjective job shop scheduling with fuzzy processing time and fuzzy duedate through genetic algorithms. European Journal of Operational Research 120, 393–407 (2000)
12. Talbi, E.G.: Metaheuristics. From Design to Implementation. Wiley (2009)

Detección de emociones expresadas con la voz mediante caracterización biométrica

Doctorando: Daniel Palacios-Alonso
Director: Victoria Rodellar

Departamento de Arquitectura y Tecnología de Sistemas Informáticos.
Facultad de Informática. Universidad Politécnica de Madrid
victoria@fi.upm.es, daniel@junipera.datsi.fi.upm.es

Keywords: Firma Glotal, Temblor Emocional, Estrés Emocional, Parámetros Biomecánicos Fonatorios

1. Introducción

Al hablar generamos sonidos, lo que vulgarmente denominamos voz, dichos sonidos transportan intrínsecamente información y ruido o dicho de otra forma, sonido con o sin significado, haciendo más ardua la tarea de la asociación de estados emocionales a ciertos patrones de voz. Podríamos decir que las emociones suscitadas a través de la voz pertenecen a la parte significante de la voz. Existen muchos factores que intervienen en la producción de la voz, tales como, aspectos neurológicos, psicológicos, socio-culturales y fisiológicos.

Generalmente, se ha estipulado que existen tres formas de expresar emociones, la primera sería emoción corporal, otra gestual y finalmente vocal. Las dos primeras pueden ser detectadas sin mucha dificultad por la tensión muscular generada, ademas de poder asociar un estado neutro por la carencia de dicha tensión. Distinto es la detección de emociones en la voz, puesto que se carece de un estado neutro de partida y por tanto hace más complicado la detección de la misma. Véase que un silencio puede ser la expresión manifiesta de un estado emocional, por tanto se parte que el silencio también aporta significado emocional.

2. Hipótesis de partida y trabajos relacionados

A lo largo de los últimos años muchos investigadores han estudiado multitud de parámetros para intentar identificar la relación directa entre dichos parámetros y la asociación con los diferentes estados emocionales, aunque no hay acuerdo, ni evidencia probada de tales hechos. En el siguiente artículo [1] se presenta un resumen de las emociones a estudio, los parámetros de búsqueda, resumen de bases de datos emocionales y los métodos de clasificación utilizados.

Muchos autores han intentado categorizar cuántas emociones existen. Plutchick [2] propuso 4 pares de emociones primarias y opuestas, **alegría/tristeza, confianza/disgusto, miedo/ira** y **sorpresa/anticipación**. A partir de la

combinación de las 4 tuplas primarias se genera una amalgama con todas las emociones conocidas en el lenguaje y que representó en forma de flor. Se puede decir que generó un autómata emocional donde los estados eran las emociones definidas y las variables de entrada o *inputs* eran la valencia (*valence*) y la excitación (*arousal*), las cuales hacían transitar a los diferentes estados emocionales. La valencia especifica cómo de positivo o negativo es un hecho, mientras que la excitación cuantifica la intensidad.

El enfoque inicial en este trabajo es centrarse en la búsqueda de las emociones en hispanohablantes, dejando en un segundo plano diferentes lenguas y culturas, parámetros que se han encontrado imprescindibles en la clasificación emocional. Se han encontrado dos trabajos previos con bases de datos emocionales en lengua española. El primero contiene cuatro emociones: tristeza, felicidad, ira y estado neutral. El corpus consistía en tres pasajes cortos de cuatro o cinco frases, quince frases cortas y treinta palabras aisladas producidas por un actor masculino. Los registros de voz fueron validados subjetivamente por quince personas, obteniendo una identificación del 85 %. La segunda base de datos contenía siete emociones y fue realizada por ocho actores, cada uno de los cuales leía dos textos expresando tres emociones distintas, obteniendo un total de 336 registros. Los registros de voz fueron validados mediante un test de percepción por 1054 estudiantes, obteniendo solamente una identificación adecuada 34 registros de los 336, es decir el 10 %. Como se puede comprobar la obtención de datos es crucial para poder identificar y catalogar emociones. Además, existen bases de datos emocionales en otros idiomas, tales como danés, alemán, inglés, japonés, hebreo, sueco, chino, ruso, eslovaco, hindi y otros varios [3].

Hay que destacar que en los trabajos realizados con anterioridad han centrado sus esfuerzos en la caracterización paramétrica basada en la *fonética* y en la *prosodia*.

3. Objetivos

Nuestro objetivo es ir mas allá e intentar relacionar la **firma biométrica de la voz con la actividad neuronal producida al hablar**. Nuestros trabajos previos han demostrado que la estimación de rigidez biomecánica en la voz con respecto la dispersión y ciclicidad están directamente relacionada con la diferenciación emocional [4], al igual que el estrés generado al defender una idea expresada con sinceridad y su opuesta [5].

Para llegar a lograr el objetivo principal será necesario conseguir los siguientes hitos:

- Generación de una aplicación multiplataforma para la obtención y valoración de muestras espontáneas de calidad.
- Búsqueda de **nuevos parámetros** a partir de las muestras seleccionadas y selección de un conjunto de entrenamiento inicial.
- Generación de un modelo definitivo paramétrico a través de n-ciclos con diferentes algoritmos de clasificación supervisada.
- Retroalimentación con nuevas muestras y generación de conclusiones.

4. Metodología y plan de trabajo

La base fundamental de este trabajo son los: **datos reales de calidad no manipulados** y **datos bien clasificados**, por tanto como primer objetivo de esta tesis es la obtención de datos de calidad, para ello se implementará una aplicación para la toma y clasificación de emociones. A continuación se detalla en mayor detalle.

4.1. Aplicación

Actualmente, la mejor forma de llegar al usuario final es a través de Internet. Basándonos en esta idea se realizará una aplicación web para la *recogida y evaluación* de muestras masivas. En una primera etapa, la aplicación se habilitará para la **recogida** de muestras. Para ello, se hará uso de guiones usados con anterioridad por otros investigadores [6], realizando algunas mejoras. El guión del que se parte consta de una encuesta con una serie de preguntas polémicas que serán valoradas de 1-7, siendo 1 *completamente en desacuerdo* y 7 *completamente de acuerdo* con la pregunta y a su vez valorar de 1-7 los sentimientos que le provoca, si son muy *indiferentes* o *muy fuertes*. Se escogerán aquellas preguntas en las que el usuario este totalmente en desacuerdo (1) o de acuerdo (7) y su sentimiento sea muy fuerte (7). Una vez escogida la pregunta a realizar, el usuario será grabado defendiendo su opinión personal sobre la pregunta (idea consistente) y acto seguido se le pedirá que defienda la idea completamente contraria (idea inconsistente), provocando desconcierto y una serie de patrones a estudio.

Obtenidas las muestras se habilitará la segunda etapa, **evaluación** de las mismas por distintas personas a las que han realizado las grabaciones y totalmente ajenas al proceso de adquisición, ésto se hará así para evitar posibles subjetividades en el proceso de evaluación.

Como se puede observar la idea concebida no solo aporta un número indeterminado de muestras en origen, si no además un conjunto inicial de aprendizaje no supervisado realizado por la evaluación de sujetos anónimos de emociones ajenas y adquiridas en otro momento en el tiempo. Una vez finalizado estas etapas se escogerán todas aquellas emociones que hayan tenido los *ratios* más elevados y con mayor grado de aceptación generalizada.

Es lógico pensar que toda aplicación basada en la red de redes tiene que estar sustentada por el diseño e implementación de una base datos creada a tal efecto, la cual poseerá un modelo de entidad-relación e integridad referencial que recogerá toda la información de la aplicación.

4.2. Búsqueda de parámetros

Muchos trabajos se han realizado en la discretización de una onda o muestra de sonido, ejemplos de estos parámetros son la *frecuencia fundamental*, diferentes formantes, el tono (*pitch*), la amplitud, etcétera [7]. Ninguno de ellos ha aportado información tangencial, tan solo adicional, por lo que se concluye que hay que

buscar en otro tipo de parámetros que aporten luz al objetivo propuesto. Tal y como se adelantó en la sección anterior alguno de los parámetros a estudio seria *body stiffness*, *cover stiffness, cycle 1, cycle 2*, dispersión de los anteriores, entre otros. El número del que se parte son la extracción de 65 parámetros, acústicos y biomecánicos, que serán anotados en la base de datos creada con anterioridad. Estos y otros parámetros albergan primeros indicios de una novedosa parametrización y diferencial respecto al paradigma tratado de manera recurrente y carente de resultados obvios.

4.3. Clasificación de parámetros

Una vez realizado la selección de parámetros a estudio se procederá a la clasificación de las emociones obtenidas a través de algoritmos de clasificación. La extracción de parámetros se realiza mediante una aplicación desarrollada en Matlab, a partir de los cuales se puede proceder a la identificación de las emociones utilizando distintos algoritmos de clasificación. Como se presenta en algunos artículos , como por ejemplo [8], Gaussianas, SVM, K-Means, árboles decisionales y redes bayesianas son algunos de los algoritmos preferidos por la comunidad científica para clasificar este tipo de elementos, aunque con unos resultados parciales y de firma no muy clara. No se descarta la utilización de nuevos algoritmos de clasificación, como un enfoque de clasificación de k-means pero utilizando redes bayesianas no paramétricas.

4.4. Diagrama de Gantt

A continuación se presenta un diagrama de Gantt para explicar los tiempos asignados para cada una de las tareas.

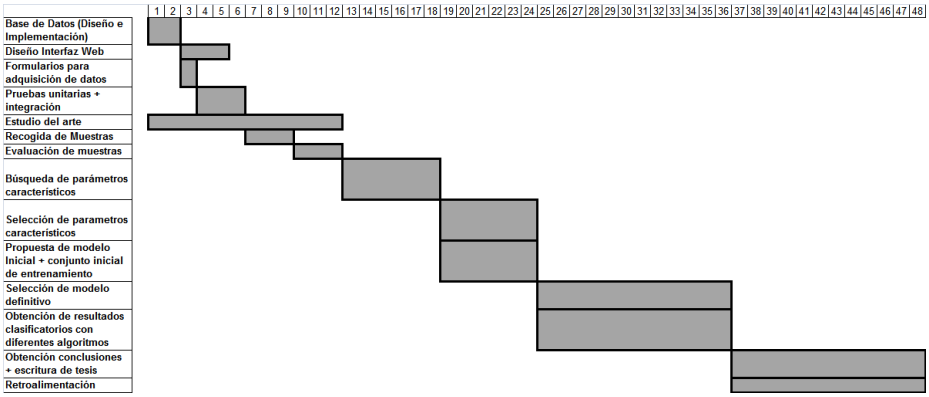


Figura 1. Diagrama de Gantt de tareas a realizar expresados en meses

5. Relevancia

A continuación se presenta algunas de las ideas que sugiere la relevancia del citado proyecto y las múltiples posibilidades que ofrece el lograr alcanzar la meta.

5.1. Investigadora

Desde el punto de vista investigador, se elaboraría un modelo completo de los parámetros característicos de la emoción, lo que se podría denominar *genoma vocal emocional*. Ofreciendo líneas futuras de estudio y de profundización. Dados los limitados resultados anteriores basados en aproximaciones acústicas, nuestro enfoque será utilizar parámetros biomecánicos basados en la estimación de la huella vocal y la correlación con la actividad neuro-motora.

Como línea futura se podría extrapolar el presente trabajo a otras lenguas, realizando colaboraciones con universidades extranjeras, dando a éste proyecto ámbito internacional de gran impacto.

5.2. Nivel de aplicación

La detección de emociones en la voz tiene muchas aplicaciones potenciales como por ejemplo vídeo juegos de próxima generación con los que interactuar de forma dinámica, conversores texto/voz, traductores automáticos de idiomas, aplicaciones forenses, *call centers*, ayuda en servicios de fraude telefónico, etcétera. Adaptación a las nuevas necesidades de los usuarios y entendiendo mejor sus preferencias. Por ejemplo, reacción eficaz ante llamadas telefónicas de auxilio.

5.3. Transferencia a la sociedad

Del presente trabajo podría derivarse la realización de varias patentes relacionadas con: Protocolos para la evocación de emociones vía web. Métodos de caracterización de emociones. Métodos de clasificación. Así como registros de distintos programas software en la realización de interfaces inteligentes hombre/máquina.

Referencias

1. Ramakrishnan, S.: Recognition of Emotion from Speech: A Review, in Speech Enhancement, Modeling and Recognition-Algorithms and Applications, Publisher InTech (2012)
2. Plutchick, R.: The Psychology and Biology of Emotion. Harper Collins Publishers (1994)
3. Ververidis, D. and Kotropoulos, C.: A Review of Emotional Speech Databases. Proc. 9th Panhellenic Conference on informatics (PCI), pp. 560-574, Thessaloniki, Greece, (2003)

4. Rodellar,V., Palacios-Alonso, D., Bartolomé, E., Gómez, P.: Vocal Fold Stiffness Estimates for Emotion Description in Speech. Proceedings of BIOSIGNALS 2013 – 6th International Conference on Bio-inspired Systems and Signal Processing, pp. 112-119 (2013)
5. Rodellar, V., Palacios-Alonso, D., Bartolomé E., Gómez-Vilda, P.: Emotional stress detection in contradictory versus self-consistent speech by means of voice biometrical signature. Proceedings of IWINAC 2013 - 5th International Work-Conference on the Interplay Between Natural and Artificial Computation, Vol. I, pp. 74-82, June 2013
6. Arciuli, J., Villar, G., Mallard, D.: Lies, Lies and More Lies. 31st Annual Meeting of the Cognitive Science Society (CogSci 2009), pp. 2329-2334 (2009)
7. Hasrul, M., Hariharan, M., Sazali, Y.: Human Affective (Emotion) Behavior Analysis using Speech Signals: A Review. Proceedings of the 2012 International Conference on Biomedical Engineering (ICoBE), February (2012), pp. 217 - 222,(2012)
8. Tarasov, A., Delany, S.:Benchmarking Classification Models for Emotion Recognition in Natural Speech: a Multi-Corporal Study. EmoSPACE Workshop (in conjunction with IEEE FG 2011 conference), (2011).

Sistema Multi-agente para modelar Procesos de Consenso en Toma de Decisión en Grupo a Gran Escala usando Técnicas de Soft Computing

Doctorando: Iván Palomares Carrascosa
Director: Luis Martínez López

Grupo de Investigación "Sistemas Inteligentes basados en Análisis de Decisión Difuso" (Sinbad^2). Departamento de Informática, Universidad de Jaén.
e-mail: ivanp@ujaen.es
http://sinbad2.ujaen.es

Palabras clave: Toma de Decisiones; Procesos de Alcance de Consenso; Sistemas Multi-agente; Soft Computing

1. Introducción

La *Toma de Decisiones* (TD) es un proceso habitual en las actividades cotidianas de los seres humanos, cuando nos enfrentamos a situaciones en las que existen varias alternativas, y debemos decidir cuál de ellas es la mejor o cuál llevar a cabo. Los llamados problemas de *Toma de Decisión en Grupo* (TDG), caracterizados por la participación de múltiples expertos con diferentes puntos de vista, han adquirido especial importancia e interés investigador dentro del área de la TD en las últimas décadas [4]. Tradicionalmente, los problemas de TDG se han resuelto aplicando únicamente un proceso de selección de alternativas [2], en el que se escoge la mejor alternativa o subconjunto de ellas, sin tener en cuenta el nivel de acuerdo existente entre los expertos. Una consecuencia de ello es la posibilidad de tomar decisiones que no sean aceptadas como buenas por parte de algunos expertos, porque consideren que sus preferencias no han sido tenidas en cuenta. Por esta razón, el estudio de los *Procesos de Consenso* en la resolución de problemas de TDG se ha convertido en un importante campo de investigación en el ámbito de la TD [3, 7]. Habitualmente, en los procesos de consenso los expertos discuten y modifican sus preferencias, bajo la supervisión de una figura humana conocida como *moderador*, con el fin de aproximar dichas preferencias y alcanzar un alto nivel de acuerdo antes de tomar una decisión [7].

Entre los resultados de las numerosas investigaciones llevadas a cabo en el ámbito de los procesos de consenso, cabe destacar:

- La definición de un gran número de *modelos teóricos de consenso*, para ayudar a llevar a cabo dichos procesos [3].
- La definición de diferentes *medidas de consenso*, es decir, indicadores para medir el nivel de acuerdo alcanzado entre los expertos [4], normalmente basados en el empleo de medidas de similitud entre expertos y operadores de agregación de preferencias [1].

- El desarrollo de *Sistemas de Apoyo al Consenso* (SAC), que son sistemas basados en técnicas inteligentes que implementan los modelos de consenso propuestos, con el fin de automatizar las tareas llevadas a cabo por el moderador humano y eliminar la posible subjetividad del mismo.

2. Hipótesis de partida

Los modelos y enfoques hasta ahora propuestos para llevar a cabo procesos de consenso se han centrado normalmente en problemas de TDG en los que participa un reducido número de expertos. Sin embargo, debido a la reciente evolución de nuevos paradigmas tecnológicos y entornos que hacen posible la participación de un elevado número de individuos en procesos de TD (tales como redes sociales, e-democracia, etc.), los llamados problemas de *TDG a Gran Escala* han adquirido una relevancia cada vez mayor en los últimos años. Los resultados obtenidos hasta el momento a partir de estudio de los procesos de consenso no son suficientes en estas situaciones, ya que aparecen nuevos problemas y aspectos que requieren un mayor estudio para la mejora de procesos de discusión por parte de un elevado número de expertos. Algunos de estos problemas son los siguientes:

- La necesidad de *supervisión constante por parte de los expertos* involucrados en el problema de TDG, para revisar y modificar sus preferencias durante el proceso de consenso. Cuando el número de expertos participantes es elevado, este hecho puede suponer un elevado coste temporal invertido en alcanzar un acuerdo, así como la posibilidad de que algunos expertos eventualmente pierdan motivación e interés en el problema a abordar.
- La mayoría de los SAC propuestos hasta la fecha se centran únicamente en procesos de consenso con un reducido número de expertos. Sin embargo, los problemas de TDG a gran escala requieren de una alta escalabilidad para facilitar el manejo de grandes cantidades de información relativa a las preferencias de los expertos. Por esta razón, surge la necesidad de *desarrollar SAC basados en arquitecturas altamente escalables*, que sean capaces de dar soporte a este tipo de problemas de forma efectiva.
- La existencia de *subgrupos de expertos que intentan manipular el proceso de discusión* desviando la opinión colectiva a su favor, es frecuente en muchos de problemas de TDG llevados a cabo mediante procesos de consenso en la vida real. Dichos comportamientos suelen dificultar el alcanzar un acuerdo. Además, en problemas de TDG a gran escala, identificar y gestionar estos comportamientos puede resultar especialmente complejo sin la ayuda de enfoques y herramientas adecuados para tal fin.
- Conocer la visión o *actitud de los expertos participantes hacia el alcance de consenso* en cada problema a abordar, es un aspecto importante a tener en cuenta para optimizar el proceso de consenso y adaptarlo a dicha actitud. Al tratar con grandes grupos, determinar y reflejar dicha actitud en el proceso a través de las medidas de consenso utilizadas, puede resultar complejo sin la elección una medida adecuada para ello.

- La necesidad de *conocer y entender la situación actual del problema* de forma simple, a partir de las posiciones de los expertos según sus preferencias, puede ayudar en el proceso de discusión del consenso. Hasta ahora, los modelos matemáticos subyacentes a los modelos y SAC existentes eran suficientes para obtener e interpretar esta información con cierta facilidad, debido al reducido número de participantes en los problemas. Sin embargo, la gran cantidad de información utilizada en problemas de TDG a gran escala hace necesario el empleo de herramientas de apoyo basadas en la representación visual de dicha información, que permitan hacerla más interpretable y así monitorizar fácilmente el estado actual del problema.

La constante evolución y los retos actuales presentes en los problemas de TDG a gran escala, nos llevaron al comienzo de esta investigación a plantearnos la siguiente **hipótesis de partida**: *Los modelos de consenso y SAC actuales no son capaces de satisfacer las necesidades actuales de los problemas de TDG a gran escala. Los procesos de consenso llevados a cabo por dichos modelos y sistemas son excesivamente rígidos, debido principalmente a la alta supervisión requerida por los expertos humanos. Por ello, es necesario flexibilizar el proceso de consenso mediante mecanismos de semi-supervisión, y optimizarlo teniendo en cuenta la actitud de los expertos. Además, el manejo de grandes grupos de expertos acentúa la necesidad de diseñar sistemas basados en arquitecturas altamente escalables, distribuidas y capaces de soportar el manejo de grandes cantidades de información de manera efectiva, así como de detectar comportamientos contrarios al alcance de consenso y monitorizar el proceso de discusión, mediante herramientas que nos ayuden a ello.*

3. Objetivos

En base a la hipótesis de partida expuesta, el **propósito** inicialmente fijado en esta Tesis Doctoral fue el de desarrollar un SAC basado en la tecnología de *sistemas multi-agente* [8], caracterizada por su alta escalabilidad y capacidades que computación distribuida, que permitiese implementar diversos modelos de consenso nuevos y existentes, junto con diversas técnicas de soft computing para la mejora y automatización de los procesos de consenso en TDG a gran escala. Para conseguir este propósito, se consideraron los siguientes **objetivos**:

1. Desarrollo de un **modelo de autonomía semi-supervisada** que permita conseguir un alto grado de **automatización** en las tareas llevadas a cabo por los expertos humanos, reduciendo así el coste temporal invertido en las mismas. Dicho modelo ha de permitir a los expertos humanos delegar en agentes inteligentes para la discusión y modificación de sus preferencias de forma autónoma, con el fin de reducir el coste invertido en los procesos de consenso con grandes grupos, aunque en determinadas ocasiones en las que el sistema proponga cambios importantes en las opiniones, los agentes pueden solicitar la supervisión del experto humano correspondiente.

2. Integración en el SAC de mecanismos para **detectar y penalizar comportamientos contrarios al consenso**, es decir, situaciones en las que un experto o subgrupo de expertos con intereses similares no están dispuestos a modificar sus opiniones iniciales hacia un acuerdo global, e intentan desviar la opinión colectiva a su favor.
3. **Integración**, en los modelos de consenso implementados en el SAC, **de la actitud del grupo hacia el consenso**, que viene determinada por la importancia que los expertos dan a sus convicciones propias respecto a la importancia dada al alcance de consenso. Dicha actitud debe reflejarse en las medidas de consenso utilizadas para determinar el nivel de acuerdo alcanzado, con el fin de optimizar el proceso de consenso en función de la actitud adoptada por el grupo en un problema determinado.
4. Construcción de una **herramienta gráfica de monitorización**, que facilite un análisis visual de las preferencias y posiciones de acuerdo/desacuerdo entre los expertos durante el proceso de consenso.

4. Metodología y Plan de Trabajo

La **metodología** a seguir para el desarrollo de la investigación conducente a esta tesis doctoral abarca una vertiente teórica y una vertiente práctica.

En la **vertiente teórica**, partiendo de un estudio profundo de los modelos y enfoques de consenso existentes, se estudió en primer lugar su posible adaptación a un contexto basado en sistemas multi-agente. Además, se han revisado en profundidad diversas técnicas de soft computing, toma de decisiones bajo incertidumbre, modelado de preferencias y tecnologías multi-agente. En base a las técnicas revisadas, se han propuesto los modelos y herramientas necesarios para alcanzar los objetivos anteriormente propuestos. Por último, se ha definido un operador de agregación, y una medida de consenso basada en dicho operador, que permite integrar la actitud del grupo de expertos en el proceso de consenso.

En la **vertiente práctica**, nos hemos planteado construir las siguientes herramientas:

(i) Un SAC basado en una plataforma multi-agente, que desarrolle las ideas propuestas en la vertiente teórica y permita implementar múltiples modelos de consenso de forma flexible.
(ii) Un mecanismo de semi-supervisión basado en modelos de negociación, para dotar a los agentes inteligentes del SAC con un alto grado de autonomía para llevar a cabo de forma efectiva las tareas de supervisión de preferencias normalmente llevadas a cabo por los expertos humanos durante el proceso de consenso.
(iii) Un mecanismo para la detección y manejo de comportamientos grupales contrarios al consenso.
(iv) Una herramienta de monitorización para facilitar el análisis visual del problema durante el proceso de consenso.

La Tesis Doctoral se está desarrollando según el siguiente **plan de trabajo**:

T1. Puesta al día bibliográfica: Revisión de los modelos y enfoques de consenso presentes en la literatura para problemas de TDG, centrada en problemas que tienen lugar en ambientes de incertidumbre. Estudio de metodologías, arquitecturas y aplicaciones de los sistemas multi-agente. Revisión de técnicas de soft computing a utilizar en el desarrollo de la tesis.

T2. Desarrollo de las propuestas. Se trata del núcleo de la tesis y comprende:

i) Definir y proponer una arquitectura multi-agente altamente escalable, capaz de reemplazar al moderador humano en los procesos de consenso, y construir un SAC basado en dicha arquitectura.

ii) Proponer un modelo de autonomía semi-supervisada, basado en modelos de negociación entre agentes inteligentes, para conseguir un alto grado de automatización de las tareas realizadas por los expertos humanos en los procesos de consenso.

iii) Definir un operador de agregación capaz de reflejar la actitud del grupo hacia el consenso, y definir modelos de consenso que integren dicha actitud en el proceso y permitan una optimización del mismo.

iv) Definir una metodología para detectar y manejar comportamientos grupales contrarios al consenso, mediante técnicas de clustering difuso [6]. Integrar dicha metodología en los modelos de consenso definidos.

v) Propuesta de una herramienta de monitorización, basada en mapas autoorganizativos [5], para analizar visualmente el proceso de consenso.

5. Relevancia

Se pretende que las soluciones alcanzadas en esta investigación sean aplicadas a problemas reales de TDG basados en *e-democracia* y *redes sociales*, y que sea ampliable a otros campos y áreas, como por ejemplo entornos de comercio electrónico para compras en grupo (*e-marketplaces*) y *sistemas de recomendación en grupo*. También se pretende llevar a cabo la *transferencia* del SAC desarrollado para su utilización en entornos académicos, empresariales y organizativos, así como en administraciones locales.

A nivel científico, actualmente se han divulgado y publicado los resultados hasta ahora obtenidos en varias *revistas internacionales* de impacto, tales como: Soft Computing, Knowledge-based Systems, IEEE Transactions on Fuzzy Systems, Expert Systems with Applications, International Journal of Uncertainty, Fuzziness and Knowledge-based Systems, etc; así como en *congresos y workshops* de ámbito nacional e internacional, como ISKE, PAAMS, ISDA, AGOP, ESTYLF, LFSC (EUSFLAT), etc.

Publicaciones actuales relacionadas con el Proyecto de Tesis:

1. I. Palomares and L. Martínez. A Semi-Supervised Multi-Agent System Model to support Consensus Reaching Processes. *IEEE Transactions on Fuzzy Systems*, Accepted on June 26th, 2013.
2. I. Palomares, L. Martínez, and F. Herrera. A Consensus Model to Detect and Manage Non-cooperative Behaviors in Large Scale Group Decision Making. *IEEE Transactions on Fuzzy Systems*, In Press, 2013.

3. I. Palomares, J. Liu, Y. Xu, and L. Martínez. Modelling experts' attitudes in Group Decision Making. *Soft Computing*, 16(10):1755-1766, 2012.
4. I. Palomares, R.M. Rodríguez, and L. Martínez. An attitude-driven web consensus support system for heterogeneous group decision making. *Expert Systems with Applications*, 40(1): 139-149, 2013.
5. I. Palomares, P.J. Sánchez, F.J. Quesada, F. Mata, and L. Martínez. COMAS: A Multi-agent System for Performing Consensus Processes. *Advances in Intelligent and Soft Computing*, 91: 125-132, Springer (Berlin), 2011.

Referencias

1. G. Beliakov, A. Pradera, and T. Calvo. *Aggregation Functions: A Guide for Practitioners*. Springer, 2007.
2. F. Herrera, E. Herrera-Viedma, and J. Verdegay. A sequential selection process in group decision making with linguistic assessments. *Information Sciences*, 85(1995):223–239, 1995.
3. E. Herrera-Viedma, J.L. García-Lapresta, J. Kacprzyk, M. Fedrizzi, H. Nurmi, S. Zadrozny, and (Eds.). *Consensual Processes. Studies in Fuzziness and Soft Computing*, volume 267. Springer, 2011.
4. J. Kacprzyk. Group decision making with a fuzzy linguistic majority. *Fuzzy Sets and Systems*, 18(2):105–118, 1986.
5. T. Kohonen. *Self-organizing maps*. Heidelberg: Springer, 1995.
6. W. Pedrycz. *Knowledge-based Clustering*. Wiley, 2005.
7. S. Saint and J. R. Lawson. *Rules for Reaching Consensus. A Modern Approach to Decision Making*. Jossey-Bass, 1994.
8. M. Wooldridge. *An introduction to Multiagent Systems (2nd Ed.)*. John Wiley and Sons, 2009.

Modelo para la generación de cursos virtuales en sistemas de gestión de aprendizaje usando tecnologías de la web semántica

Doctorando: Danilo Pástor-Ramírez[1]

Director: Jovani Jimenez-Builes[2]

Grupo de Investigación en Inteligencia Artificial en Educación

Departamento de Ciencias de la Computación y de la Decisión. Facultad de Minas.

Universidad Nacional de Colombia. Medellín. Cra. 80 x Cl. 65. Barrio Robledo.

dpastor@espoch.edu.ec[1]

jajimen1@unal.edu.co[2]

1 Introducción

Actualmente nuestra sociedad demanda el uso de las tecnologías de información y comunicación y las Instituciones de Educación Superior (IES) están incluyendo en sus procesos de enseñanza la instrucción mediada por el computador. Las IES están usando el e-learning como una alternativa para lograr los objetivos educativos sus sistemas de aprendizaje [1]. El E-learning es un ambiente de aprendizaje que usa medios electrónicos como herramienta para mejorar la interacción y comunicación con los estudiantes [2]. Las herramientas de E-learning pueden estar compuestas de Entornos virtuales de Aprendizaje (VLE en inglés) o también llamados Sistemas de Gestión de Aprendizaje (LMS en inglés), los cuales son entornos que fueron diseñados para gestionar cursos online, recursos educacionales, docentes y estudiantes.

Matthews [3] considera a la web semántica como un conjunto de tecnologías, herramientas y estándares que son parte de un sistema que ayuda a que la web tenga significado en sus contenido. En el estudio realizado por Martin et. al. [4] indica que se espera incrementar la tendencia en los próximos años de aplicaciones semánticas en la educación. Varios estudios se han realizado de la web semántica aplicados al E-learning, siendo estos más frecuentes con el uso de Ontologías para representar el conocimiento en distintos contextos de los procesos de enseñanza-aprendizaje (e.g [5], [6]).

De acuerdo Aroyo y Dicheva [7] mencionan que los sistemas educacionales basados en la web semántica necesitan tener funcionalidades donde el contenido deba ser

interoperable, colaborativo, intercambiable y re-usable, es decir, que exista compatibilidad en la creación estructurada y unificada de recursos didácticos, en este sentido, se nota que existe requerimientos de diseño VLEs pero en forma estructurada y estandarizada. De la misma forma Koper [8] manifiesta que a través de la web semántica se puede apoyar a los docentes a realizar sus tareas que consumen mucho tiempo y son repetitivas por cada curso virtual que se genere. Todo esto implica, que exista la necesidad de implementar algún tipo de sistema que produzca los cursos virtuales a través de las tecnologías de la web semántica.

2 Hipótesis de partida

Frente al gran crecimiento de la oferta de educación virtual, a través de los VLEs, especialmente en el ámbito universitario, se evidencia que los docentes universitarios se apoyan, cada vez más, de los VLEs, sin embargo, según Mahdizadeh et al. [9] menciona que el E-learning no está bien integrado a los procesos de enseñanza de alto nivel y que los docentes solo usan las capacidades de las herramientas de los VLEs en forma superficial. En este sentido, Stewart et al. [10] sugieren investigar que contenido debería ser presentado, como el contenido debe ser bien presentado y que actividades deberían ser ofrecidas y que combinación de información deberían ser mejor presentados en los VLEs.

Varios estudios han sido realizados de la web semántica aplicados al E-learning, siendo estos más frecuentes al uso de Ontologías para representar el conocimiento en distintos contextos de enseñanza/aprendizaje. Cuéllar et al. [11] desarrollan un procedimiento para lograr la integración de diferentes sistemas E-learning y su gran aporte es que extraen la información de la base de datos del LMS y lo asocian a una Ontología, por lo que, esta investigación abre posibilidades de aplicar a ámbitos de VLEs. Huang et al. [12] propone un novedoso enfoque de E-learning semántico donde integra el contenido, el proceso de aprendizaje y la personalidad del aprendiz, destacando que este trabajo que considera es aspecto técnico y el pedagógico, sin embargo, no utiliza las actuales tecnologías de la web semántica. Castellanos-Nieves et al. [13] describen un enfoque en el que usan tecnologías de la web semántica para apoyarse en la evaluación de preguntas abiertas en cursos virtuales, enfocándose únicamente a una parte importante de un VLE que son las Evaluaciones. Existen otros trabajos que no tienen relación con tecnologías de la web semántica, pero sugieren que se debería investigar más en la especificación de un modelo que permita crear cursos virtuales de calidad para los LMSs [14].

Al ser los LMS ampliamente utilizados en el ámbito del E-learning y de estos uno de los más reconocidos es Moodle, en donde los usuarios generan cursos virtuales y también recursos educativos, se está tendiendo a convertirse en un repositorio de datos, provocando una falta de normalización, lo que dificulta su interoperabilidad y reusabilidad [15]. Finalmente Wang y Hsu [16] proponen un sistema que provee un entorno para producir material de enseñanza personalizados en base a ciertos reque-

rimientos y recomienda diseñar un procedimiento de creación de cursos para generar material de enseñanza.

Después de la revisión de la literatura analizada, se observa, *que no se conoce un mecanismo que permita la generación de cursos virtuales en sistemas de gestión de aprendizaje de forma estructurada y estandarizada, usando tecnologías de la web semántica y considerando aspectos tecnológicos y pedagógicos propios del diseño de entornos de aprendizaje virtual***.**

A través de las tecnologías de la web semántica tales como el lenguaje para definir las ontologías, el lenguaje de consulta de contenidos, un motor de inferencia y considerando las teorías de aprendizaje así como también el diseño y las estrategias de enseñanza adecuados en el diseño de ambientes virtuales de aprendizaje, se propone un modelo que permita la generación de cursos virtuales en sistemas de gestión de aprendizaje. Este modelo, permitirá crear en forma estructurada y unificada los cursos virtuales, permitiendo obtener diseños novedosos y en menor tiempo que los producidos convencionalmente y a su vez beneficiará a la usabilidad de los aprendices.

3 OBJETIVOS

3.1 Objetivo General

Crear un modelo para la generación de cursos virtuales en sistemas de gestión de aprendizaje usando tecnologías de la web semántica.

3.2 Objetivos Específicos

- Caracterizar los sistemas de gestión de aprendizaje y su convergencia con las tecnologías de la web semántica.
- Determinar los factores que influyen en el modelamiento de la generación de cursos considerando las teorías de aprendizaje así como también el diseño y las estrategias instruccionales.
- Diseñar un mecanismo que permita la creación de cursos virtuales empleando las tecnologías de la web semántica y que se adapten a un sistema de gestión de aprendizaje.
- Diseñar una arquitectura que integre los componentes de la creación de cursos, las plantillas, recursos de aprendizaje y recursos de los sistemas de gestión de aprendizaje.
- Desarrollar los módulos necesarios para construir automáticamente plantillas de cursos en concordancia con el modelo planteado.
- Evaluar la efectividad el proceso de construcción de cursos con los docentes y la usabilidad de los cursos personalizados con los estudiantes

4 Metodología y plan de trabajo

La metodología se compone de seis fases que corresponde a su respectivo objetivo específico y las describo a continuación:

Fase 1.- Revisión de la literatura existente sobre entornos de aprendizaje virtual y su aplicabilidad con las tecnologías de la web semántica, complementando con teorías y estrategias de enseñanza virtual.

Fase 2: Se realiza un análisis de los modelos de construcción de cursos virtuales y se determinará los factores que influyen a la hora de construirlos. El resultado será la obtención de las variables que intervendrán en el modelo propuesto.

Fase 3: Se establecerá el mecanismo que permita crear los cursos virtuales de forma automática tomando en consideración los parámetros obtenidos en la fase anterior.

Fase 4: Se planteará una arquitectura que permita la integración de los componentes que formarán framework, tomando en cuenta que se desarrollará en un sistema de gestión de aprendizaje genérico y utilizando las tecnologías de la web semántica.

Fase 5: Se desarrollará un módulo de software como parte del CMS que mediante un tipo asistente se pueda crear plantillas de cursos virtuales de manera automática,

Fase 6: Se llevará a cabo un procesos de evaluaciones a docentes mediante técnicas cualitativas y además se realizará un test de usabilidad en las aulas personalizadas a los estudiantes.

5 Relevancia

Fundamentalmente la mayor contribución de esta investigación es la propuesta de un modelo que permita la construcción de cursos virtuales en sistemas de gestión de aprendizaje utilizando tecnologías de la web semántica y considerando las teorías de aprendizaje así como también el diseño y las estrategias de enseñanza. El trabajo propuesto es innovador, ya que, basado en la bibliografía revisada, no se encontraron modelos que permitan generar cursos virtuales de forma automática y además utilizando las variables propuestas como las tecnologías de la web semántica y de los aspectos tecnológicos y pedagógicos del diseño instruccional.

Además, producto de la propuesta se espera lograr los siguientes aportes:

- El aporte teórico será el modelo de construcción de cursos virtuales, ya que, es un nuevo enfoque en relación a los encontrados en la revisión del estado del arte.

- El aporte práctico será el desarrollo de un módulo integrado a un CMS que permita la generación automática de cursos virtuales tipo plantillas, usando las tecnologías de la web semántica.
- El aporte empírico será la validación tanto del producto generado, es decir, la eficacia del proceso de creación de los cursos virtuales por parte de los docentes y por otro lado la usabilidad del curso virtual por parte de los estudiantes que se beneficiarán del mismo.

Referencias

1. P.-C. Sun, R. J. Tsai, G. Finger, Y.-Y. Chen, y D. Yeh, «What drives a successful e-Learning? An empirical investigation of the critical factors influencing learner satisfaction», *Comput. Educ.*, vol. 50, n.º 4, pp. 1183-1202, may 2008.
2. A. Sangrà, D. Vlachopoulos, y N. Cabrera, «Building an inclusive definition of e-learning: An approach to the conceptual framework», *Int. Rev. Res. Open Distance Learn.*, vol. 13, n.º 2, pp. 145-159, feb. 2012.
3. B. Matthews, «Semantic web technologies», *E-Learn.*, vol. 6, n.º 6, p. 8, 2005.
4. S. Martin, G. Diaz, E. Sancristobal, R. Gil, M. Castro, y J. Peire, «New technology trends in education: Seven years of forecasts and convergence», *Comput. Educ.*, vol. 57, n.º 3, pp. 1893-1906, nov. 2011.
5. B. Vesin, M. Ivanović, A. Klašnja-Milićević, y Z. Budimac, «Protus 2.0: Ontology-based semantic recommendation in programming tutoring system», *Expert Syst. Appl.*, vol. 39, n.º 15, pp. 12229-12246, nov. 2012.
6. W.-C. Shih, C.-T. Yang, y S.-S. Tseng, «Ontology-based content organization and retrieval for SCORM-compliant teaching materials in data grids», *Future Gener. Comput. Syst.*, vol. 25, n.º 6, pp. 687-694, jun. 2009.
7. L. Aroyo y D. Dicheva, «The new challenges for e-learning: The educational semantic web», *Educ. Technol. Soc.*, vol. 7, n.º 4, pp. 59–69, 2004.
8. R. Koper, «Use of the Semantic Web to Solve Some Basic Problems in Education: Increase flexible, distributed lifelong learning; decrease teacher's workload», *J. Interact. Media Educ.*, vol. 2004, n.º 1, 2004.
9. H. Mahdizadeh, H. Biemans, y M. Mulder, «Determining factors of the use of e-learning environments by university teachers», *Comput. Educ.*, vol. 51, n.º 1, pp. 142-154, ago. 2008.
10. C. Stewart, C. Bachman, y S. Babb, «Replacing professor monologues with online dialogues: A constructivist approach to online course template design», *J. Online Learn. Teach.*, vol. 5, n.º 3, pp. 511–522, 2009.
11. M. P. Cuéllar, M. Delgado, y M. C. Pegalajar, «A common framework for information sharing in e-learning management systems», *Expert Syst. Appl.*, vol. 38, n.º 3, pp. 2260-2270, mar. 2011.
12. W. Huang, D. Webster, D. Wood, y T. Ishaya, «An intelligent semantic e-learning framework using context-aware Semantic Web technologies», *Br. J. Educ. Technol.*, vol. 37, n.º 3, pp. 351–373, 2006.
13. D. Castellanos-Nieves, J. T. Fernández-Breis, R. Valencia-García, R. Martínez-Béjar, y M. Iniesta-Moreno, «Semantic Web Technologies for supporting learning assessment», *Inf. Sci.*, vol. 181, n.º 9, pp. 1517-1537, may 2011.
14. G. Y. Aguirre Isabel, «Diseño de un modelo Pedagogico - Dicdactico para el Aprendizaje en Linea», presentado en XIII Encuentro Internacional VirtualEduca, Panama, 2012.

15. C. P. B. Moreno, J. A. C. Hernández, D. C. V. Bravo, y P. A. Gaona, «Indexación Semántica de Contenidos Digitales para la referenciación y presentación de recursos digitales mediante el desarrollo de un modelo Ontológico».
16. H.-C. Wang y C.-W. Hsu, «Teaching-Material Design Center: An ontology-based system for customizing reusable e-materials», *Comput. Educ.*, vol. 46, n.º 4, pp. 458-470, may 2006.

Aprendizaje de redes bayesianas híbridas con mixturas de funciones base truncadas

Doctoranda: Inmaculada Pérez-Bernabé
Director: Antonio Salmerón Cerdán

Departamento de Matemáticas, Universidad de Almería, Ctra. Sacramento s/n, 04120 Almería, España,
{iperez, antonio.salmeron}@ual.es

Keywords: Redes bayesianas híbridas, mixturas de funciones base truncadas, aprendizaje automático

1. Introducción e hipótesis de partida

Las redes bayesianas [1] comenzaron a popularizarse a finales de la década de los 80 como una herramienta para el tratamiento de la incertidumbre en Inteligencia Artificial. Éstas constituyen una representación estructurada del conocimiento basada en un grafo dirigido acíclico donde cada vértice representa una variable aleatoria y cada arco indica la existencia de dependencia probabilística entre las variables que une. Una de las principales virtudes de las redes bayesianas es que permiten definir algoritmos eficientes de razonamiento que sacan partido de la estructura del grafo subyacente para realizar los cálculos de forma local [2,3,4]. No obstante, la mayor parte de los algoritmos existentes en la literatura están enfocados a problemas donde todas las variables de la red son de tipo cualitativo o discreto, lo que limita el ámbito de aplicación de este tipo de modelos.

Una solución pasa por discretizar las variables continuas, pero este proceso conlleva una pérdida de información, por lo que surge de forma natural el estudio de redes bayesianas donde puedan coexistir variables de tipo discreto y continuo, las llamadas *redes bayesianas híbridas*. La primera solución vino por la adopción del llamado modelo *condicional Gaussiano* [5], bajo el que subyace la suposición de que la distribución conjunta sobre las variables continuas es una normal multivariante. Además de esta restricción, las redes bayesianas de tipo condicional Gaussiano tienen la limitación de que no admiten estructuras donde variables discretas tengan padres continuos, lo que provoca que algunas relaciones no puedan ser representadas.

Con idea de resolver las limitaciones del modelo condicional Gaussiano, surgió el llamado modelo de *mixtura de exponenciales truncadas (MTEs)* [6], compatible con los mismos esquemas de inferencia exacta desarrollados para variables discretas. Desde la aparición de las MTEs, se han desarrollado numerosos trabajos en el ámbito de las redes bayesianas híbridas, destacando la introducción del modelo de *mixturas de polinomios (MOPs)* [7]. Tanto MTEs como MOPs han

sido recientemente generalizadas por el modelo de *mixturas de funciones base truncadas (MoTBFs)* [8].

2. Objetivos

El objetivo principal de la investigación a realizar es la obtención de métodos eficientes y efectivos para el aprendizaje automático de redes bayesianas híbridas basadas en MoTBFs. Este objetivo principal se desglosa en los siguientes subobjetivos:

- **Subobjetivo 1:** Desarrollo de algoritmos para la estimación de MoTBFs univariantes a partir de datos.
- **Subobjetivo 2:** Desarrollo de algoritmos para la estimación de MoTBFs condicionadas a partir de datos.
- **Subobjetivo 3:** Desarrollo de un método para la incorporación de conocimiento a priori en el proceso de aprendizaje de MoTBFs.
- **Subobjetivo 4:** Desarrollo de un algoritmo de aprendizaje de la estructura de la red particularizado al caso de MoTBFs.
- **Subobjetivo 5:** Desarrollo de un paquete software que implemente los métodos citados en los objetivos anteriores.

Todos los subobjetivos se corresponden con **aportaciones originales**. En lo relativo a los subobjetivos 1, 2 y 4, existen trabajos relativos al caso de modelos MTE, pero no esquemas válidos para el ámbito general de la MoTBFs. El objetivo 3 es totalmente novedoso en el ámbito de las redes bayesianas híbridas, En cuanto al subobjetivo 5, la idea es implementar un paquete en el entorno `R` [9], donde no existe hasta el momento ninguna funcionalidad para el manejo de redes híbridas más allá del modelo condicional Gaussiano.

3. Metodología y plan de trabajo

De forma genérica, la metodología a seguir durante el desarrollo de este proyecto de tesis se corresponde con el habitual método científico:

- formulación de hipótesis, que en nuestro caso implica el desarrollo de algoritmos para el aprendizaje automático de redes híbridas;
- recogida de observaciones: en este caso, disponer de datos reales sobre los que aplicar los algoritmos desarrollados;
- contraste de las hipótesis con las observaciones, es decir, evaluar los avances que proporciona el empleo de los nuevos métodos;
- finalmente, readaptación de las hipótesis iniciales a la luz de los resultados obtenidos.

Hemos dividido el plan de trabajo en **cinco tareas**, relacionadas con los objetivos mencionados en la sección 2, distribuidas en un período de tres años. Son las siguientes:

- **Tarea 1. Estimación de densidades MoTBF univariantes**. Plantearemos el problema de la estimación de parámetros como un programa de optimización, siguiendo la filosofía de los llamados M-estimadores [10]. Basaremos la elección del número de funciones base en un criterio de verosmilitud penalizada tipo BIC [11].
  Temporización: M1-M18.
  Hito 1: Alcance del subobjetivo 1, y publicación de los resultados en una revista JCR.
- **Tarea 2. Estimación de densidades MoTBF condicionadas**. Tomando como base la llamada estructura de árbol mixto [12], desarrollaremos un esquema para la estimación de densidades MoTBF condicionadas, que hará uso del método de estimación univariante desarrollado en la tarea 1.
  Temporización: M12-M18.
  Hito 2: Alcance del subobjetivo 2, y publicación de los resultados en una revista JCR. Esta tarea puede simultanearse con la tarea 1, y los resultados publicarse en un único trabajo.
- **Tarea 3. Incorporación de conocimiento a priori**. Seguiremos una filosofía similar al enfoque Bayesiano en estimación de parámetros. Sin embargo, en este caso, por la naturaleza semi-paramétrica de las densidades MoTBF, la especificación de información a priori no parece inmediata. Adoptaremos por tanto un esquema basado en la combinación de densidades, una a priori y otra estimada a partir de los datos usando los algoritmos construidos en las tareas anteriores.
  Temporización: M19-M24.
  Hito 3: Alcance del subobjetivo 3 y publicación de los resultados en una revista JCR.
- **Tarea 4. Aprendizaje estructural de redes híbridas con distribución MoTBF**. Comenzaremos explorando una alternativa simple, como es la aplicación de un algoritmo de aprendizaje de la estructura basado en tests de independencia, como el algoritmo PC, para luego estimar las densidades condicionadas correspondientes a dicha estructura. En una segunda fase, perseguiremos la definición de un estadístico específico para MoTBFs y estudiaremos su distribución en el muestreo, con la idea de construir un contraste de hipótesis específico.
  Temporización: M25-M30.
  Hito 4: Alcance del subobjetivo 4 y publicación de los resultados en una revista JCR.
- **Tarea 5. Integración del software en un paquete de** `R`. Esta tarea consistirá en recopilar el software desarrollado en las tareas anteriores para conformar un paquete de `R` con calidad suficiente como para ser puesto a disposición de la comunidad de usuarios de dicho entorno de análisis de datos.
  Temporización: M31-M36.
  Hito 5: Alcance del subobjetivo 5 y publicación de los resultados en una revista JCR tipo *Journal of Statistical Software*.

4. Relevancia

Existen innumerables campos de aplicación en los que coexisten variables de tipo discreto y continuo y que en la actualidad no son abordados de una forma totalmente satisfactoria desde el punto de vista de las redes bayesianas fundamentalmente debido a dos razones: por un lado, el todavía insuficiente desarrollo metodológico de las redes bayesianas híbridas, y por otro lado, la no disponibilidad de software que permita abordar tales aplicaciones por parte de usuarios de diferentes ámbitos. Pensamos que el presente proyecto aborda los dos aspectos mencionados, y supondrá un avance significativo no solo del aspecto metodológico, sino de la aplicabilidad de estas técnicas fuera del ámbito de la I.A.

Agradecimientos

El trabajo a realizar en esta tesis está subvencionado por el Ministerio de Economía y Competitividad, a través del proyecto TIN2010-20900-C04-02 y por la Junta de Andalucía mediante el proyecto P11-TIC-7821, con cofinanciación FEDER.

Referencias

1. Pearl, J.: Probabilistic reasoning in intelligent systems. Morgan-Kaufmann (San Mateo) (1988)
2. Jensen, F., Lauritzen, S., Olesen, K.: Bayesian updating in causal probabilistic networks by local computation. Computational Statistics Quarterly **4** (1990) 269–282
3. Madsen, A., Jensen, F.: Lazy propagation: a junction tree inference algorithm based on lazy evaluation. Artificial Intelligence **113** (1999) 203–245
4. Shenoy, P., Shafer, G.: Axioms for probability and belief function propagation. In Shachter, R., Levitt, T., Lemmer, J., Kanal, L., eds.: Uncertainty in Artificial Intelligence 4, North Holland, Amsterdam (1990) 169–198
5. Lauritzen, S.: Propagation of probabilities, means and variances in mixed graphical association models. Journal of the American Statistical Association **87** (1992) 1098–1108
6. Moral, S., Rumí, R., Salmerón, A.: Mixtures of truncated exponentials in hybrid Bayesian networks. In: ECSQARU'01. Lecture Notes in Artificial Intelligence. Volume 2143. (2001) 135–143
7. Shenoy, P., West, J.: Inference in hybrid Bayesian networks using mixtures of polynomials. International Journal of Approximate Reasoning **52** (2011) 641–657
8. Langseth, H., Nielsen, T., Rumí, R., Salmerón, A.: Mixtures of truncated basis functions. International Journal of Approximate Reasoning **53** (2012) 212–227
9. R Development Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. (2011) ISBN 3-900051-07-0.
10. Bai, Z., Wu, Y.: General m-estimation. Journal of Multivariate Analysis **63** (1997) 119–135

11. Schwarz, G.: Estimating the dimension of a model. Annals of Statistics **6** (1978) 461–464
12. Moral, S., Rumí, R., Salmerón, A.: Approximating conditional MTE distributions by means of mixed trees. In: ECSQARU'03. Lecture Notes in Artificial Intelligence. Volume 2711. (2003) 173–183

Scalable and Efficient Algorithms in Machine Learning for High-Dimensional Data

PhD candidate: Diego Peteiro-Barral
Supervisors: Bertha Guijarro-Berdiñas and Oscar Fontenla-Romero

Dept. of Computer Science, University of A Coruña, Spain
{dpeteiro, cibertha, ofontenla}@udc.es
http://www.lidiagroup.org

Keywords: Machine Learning, Big Data, Scalability

1 Introduction

Some of the best-known data mining methods were developed and tested on datasets such as those available in the UCI Machine Learning Repository. It is certainly not self-evident that they will all scale up to much larger datasets with acceptable runtimes and memory requirements. The most obvious answer to this problem is to take a sample from a large dataset and use that for data mining. However, Catlett work [1] shows that induction from subsets of data decreases accuracy. Despite the advantages of certain sampling strategies — speed-ups and improving the accuracy of the classifier over random sampling in noise-free domains— Catlett concludes that they are not a solution to the general problem of scaling up to very large datasets. To add to this there is the consideration that some application areas are concerned with the discovery of new knowledge, where discarding a large proportion of the data is very risky [2].

Real-world problem domains related to large-scale machine learning —these "big learning" problems— arise in many domains including bioinformatics, astronomy, recommendation systems, social networks, computer vision, web search and online advertising. The big learning setting has attracted intense interest across industry and academia, with active research spanning diverse fields ranging from machine learning and databases to large scale distributed systems and programming languages.

2 Hypotheses

The volume of data today poses the most serious problem. Traditional methods typically made the assumption that the data is memory resident. This assumption is no longer tenable. In this section we highlight some of the outstanding research issues for designing and implementing large-scale mining methods [3]:

High dimensionality. In general, the complexity of different mining algorithms may not be linear in the number of dimensions, and new parallel and distributed

methods are needed to handle large number of attributes. Current methods are able to (perhaps) handle data in the gigabyte range, but are not suitable for terabyte-sized data. Even a single scan for these databases is considered expensive. Most current algorithms are iterative, and scan data multiple times. In general, minimizing the number of data scans is paramount. Another factor limiting the scalability of most mining algorithms is that they rely on in-memory data structures.

Distributed data mining (DDM). Implementation of data mining ideas in distributed computing environments is thus becoming crucial for ensuring system scalability as data continues to grow in size and complexity. If the time complexity of the learning algorithm is more than linear in the number of samples, processing small, fixed-size samples concurrently can reduce the complexity to linear. A general model for DDM is provided by the *collective data mining* framework introduced by Provost [4]. The model has three layers: 1) The sample selection procedure partitions the data S into subsamples S_i, one for each of the p nodes available. 2) For each node there is a corresponding learning algorithm L_i which runs on the corresponding subsample S_i and generates a concept description C_i. 3) The concept descriptions C_i are then merged by a combining procedure to form a final concept description C.

Data location. If all the data is together in one very large dataset, we can distribute it on to p nodes. An entirely different situation arises when the data may inherently belong to different datasets on different nodes, for example in different parts of a company or even in different cooperating organizations. Today's large-scale data sets are usually logically and physically distributed, requiring a decentralized approach to mining. Moreover, there are many popular data mining applications that deal with sensitive data, such as people's medical and financial records. The central collection of such data is not desirable as it puts their privacy into risk.

Data skew. One of the problems adversely affecting balancing in distributed data is data skew. Data skew primarily refers to a non uniform distribution in a dataset. The term data skew encompasses several related but distinct effects. Walton et al. [5] classify the effects of skewed data distribution on a parallel execution, distinguishing intrinsic skew from partition skew. Intrinsic skew is skew inherent in the dataset. It is a property of the data –e.g., there are more citizens in Paris than in Waterloo. Partition skew occurs on parallel and distributed implementations when the data is not evenly distributed between nodes –e.g., samples may be partitioned by clustering attribute, in user specified ranges.

Measurability. For scaling up learning algorithms, the issue is not as much one of speeding up a slow algorithm as one of turning an impracticable algorithm into a practicable one. The crucial issue is seldom "how fast" you can run on a certain problem but instead "how large" a problem can you (feasibly) deal with [4]. But how to measure scalability? It is difficult to determine how does a method behave and compare with the other ones in terms of test error, training time and memory requirements, which are the practically relevant criteria. That

is, what metrics are suitable? Clearly, algorithm assessment is a multi-objective problem and there is no way we can come with a single fair total order [6].

3 Objectives

The general aim of this thesis is to propose solutions for scalable machine learning to be applicable to big volumes of data, using *distribution* as the way to achieve it, i.e., the approach to scaling is to avoid the need to run algorithms on very large datasets by partitioning the data. For the purposes of this proposal we will start by assuming that the approach adopted is to use a distributed local area network of personal computers, as for many organizations this will be a much cheaper and more realistic option [2]. It consists of a number of nodes and an interconnection network which connects nodes and supports the transfer of data between nodes. A node in an independent unit, consisting of processor and local memory. Besides, the algorithms generated will have to be able to manage some restrictions that arise when working in distributed environments:

- Preserving the privacy of data allocated at different sites. It is typically not feasible to share or communicate data at all. We want to release aggregate information about the data, without leaking individual information about participants. One approach is to alter the data before delivering it. The second approach assumes that sites cooperate to learn the global data mining results without revealing the data at their individual sites. Local models are built at each site, and are then merged/combined via various methods. Therefore, raw data is never interchanged among locations.
- Efficiency in the communications (reduced costs of communication between sites).
- Managing the heterogeneity of data and the influence in learning when the distinct classes of data are not equally represented in the distinct partitions or, even, when the partitions have different representative prototypes for a same class. This is an important stage that have not been considered enough in distributed learning.

Because of these restrictions, the classical machine learning methods can not be extended in a direct and trivial form, since they presuppose the existence of all data in only one location. Regarding the process of learning the thesis will tackle the three types of problems that can be resolved: classification, regression and uni-class classification. Although they share a common algebraic frame, it is necessary to take into account their own peculiarities in order to obtain the highest efficiency in each task.

4 Methodology and Work Plan

For the achievement of the objectives the general methodology based in the classical scientific method will be used involving the following tasks:

- Exploratory Phase. Preparation of the necessary theoretical frame by analyzing the relevant scientific literature to establish the starting point for the work: advantages and problems of the current algorithms, space and temporal complexity and restrictions to be considered by the new models.
- Research. Proposal of the theoretical development to new machine learning solutions.
- Experimentation design, to determine the proofs conditions to allow us to contrast the hypotheses, and the measurements under which the algorithms will be compared as in this field a simple measure of precision is not enough to measure scalability.
- Design and implementation of the algorithms to be compared
- Contrasting and verification of the hypothesis by means of performance measures and statistical methods to determine significant differences among results.

Regarding the working plan, the research issues and challenges, presented have been receiving our attention in the past recent years. They have been addressed and solution techniques have already been evaluated and proposed with the following results:

Evaluating the effectiveness of a multi-objective problem. As mentioned, in distributed learning, real-world problems usually consider several performance measures. A multi-criteria problem is then formulated using a set of alternatives and criteria. In the decision-making process criteria are identified, weights are given to each criterion to reflect its relative importance, and weighted preference scores are derived based on the criteria weights and criteria score. The ultimate result of this process is a ranking list of alternatives. Our achievements so far in this direction establish a framework to compare the scalability of algorithms and are gathered in the following list of publications:

1. D. Peteiro-Barral, B. Guijarro-Berdiñas: A Study on the Scalability of Artificial Neural Netwoks Training Algorithms Using Multiple-Criteria Decision-Making Methods. ICAISC 2013.
2. R. Méndez, B. Remeseiro, D. Peteiro-Barral and M. G. Penedo: Multi-criteria evaluation of class binarization and feature selection in tear film lipid layer classification. ICAART (2) 2013: 62:70.
3. D. Peteiro-Barral, B. Guijarro-Berdiñas, B. Pérez-Sánchez, O. Fontenla-Romero: A comparative study of the scalability of a sensitivity-based learning algorithm for artificial neural networks. Expert Syst. Appl. 40(10): 3900-3905 (2013).
4. V. Bolón-Canedo, D. Peteiro-Barral, B. Remeseiro, A. Alonso-Betanzos, B. Guijarro-Berdiñas, A. Mosquera González, Manuel G. Penedo, N. Sánchez-Maroño: Interferential Tear Film Lipid Layer Classification: An Automatic Dry Eye Test. ICTAI 2012: 359-366.
5. V. Bolón-Canedo, D. Peteiro-Barral, A. Alonso-Betanzos, B. Guijarro-Berdiñas, N. Sánchez-Maroño: Scalability Analysis of ANN Training Algorithms with Feature Selection. CAEPIA 2011: 84-93.

Distributed learning algorithms for classification and regression problems. In order to preserve data-privacy we follow the approach to interchange

local learnt models instead of raw data among individual sites. As local models, neural networks have been used. Our achievements so far are the following:

1. D. Peteiro-Barral, B. Guijarro-Berdiñas: A survey of methods for distributed machine learning. Progress in AI 2(1): 1-11 (2013).
2. D. Peteiro-Barral, B. Guijarro-Berdiñas, B. Pérez-Sánchez, O. Fontenla-Romero: A distributed learning algorithm based on two-layer artificial neural networks and genetic algorithms. ESANN 2011.
3. D. Peteiro-Barral, B. Guijarro-Berdiñas, B. Pérez-Sánchez: Distributed learning on nonuniform class-probability distributions based on genetic algorithms and artificial neural networks. IEEE SSCI HIMA 2011: 54-60.
4. D. Peteiro-Barral, B. Guijarro-Berdiñas, B. Pérez-Sánchez: Dealing with "Very Large" Datasets - An Overview of a Promising Research Line: Distributed Learning. ICAART (1) 2011: 476-481.

When Distribution is Part of the Semantics. Investigating data skew we realized that most DDM approaches are not suitable for our purposes. They aim to infer a global model and try to approximate the results one would get from a single joined data source. Thus, they treat distribution as a technical issue only and ignore that the distribution itself may have a meaning and that models depend on the context in which they were derived. There might not really exist a single model that describes the distributed data, but two or more groups of models. Therefore, one should first explore the dependencies of local models instead of a straightforward integration. Our achievements so far in this direction are the following:

1. D. Peteiro-Barral, B. Guijarro-Berdiñas: An Analysis of Clustering Approaches to Distributed Learning on Heterogeneously Distributed Datasets. KES 2012: 69-78.
2. D. Peteiro-Barral, B. Guijarro-Berdiñas, B. Pérez-Sánchez: Learning from Heterogeneously Distributed Data Sets Using Artificial Neural Networks and Genetic Algorithms. JAISCR 2(1): 1-21 (2012).
3. D. Peteiro-Barral, B. Guijarro-Berdiñas, B. Pérez-Sánchez: On the Effectiveness of Distributed Learning on Different Class-Probability Distributions of Data. CAEPIA 2011: 114-123.

At present, we are working in two new distributed algorithms. The first is based on frontier vector quantization and information theory that implements a genetic algorithm (GA) to combine the models inferred from distributed sites. In this case, the complete theoretical framework has been proposed, the algorithm has been contrasted and we are currently working on improving its performance:

- D. Peteiro-Barral, B. Guijarro-Berdiñas: A Distributed Learning Algorithm Based on Frontier Vector Quantization and Information Theory. ICANN 2013.
- D. Peteiro-Barral, B. Guijarro-Berdiñas: Improving the Scalability of a Distributed Learning Algorithm Based on Frontier Vector Quantization and Information Theory. CAEPIA 2013.
- D. Peteiro-Barral, B. Guijarro-Berdiñas: A Distributed Partition-Aware Learning Algorithm based on Island Model Genetic Algorithms. ECML PKDD 2013.

The second is a one-class multi-task classification algorithm using one-class Support Vector Machines (SVM) as local models. In this case, the theoretical

framework is almost finished and we are currently working on finding the best scheme to reduce communications cost among partitions.

Finally, two already open questions is 1) how to detect partition data skew by anomaly detection using one-class classifiers in order to recognize the distribution itself and learn accordingly, and 2)how to evaluate the scalability of DDM algorithms beyond scale-up, size-up, and speed-up: how long does it take to reach a given performance, how does the computational effort increases with dataset size, and which amount of data is needed for reaching a given accuracy.

5 Relevance

In year 2000, the total quantity of information in the World Wide Web (WWW) was calculated to be between 25 and 50 terabytes. In 2005, these calculations reached the 600 terabytes. At present, the quantity of information is almost incalculable. In general, the advances in the ICT have contributed to the proliferation of big databases, usually distributed, whose automatic analysis is of high interest. However, the great majority of current Machine Learning algorithms are unable to handle this new situation. In order to give solutions to these new problems, a new field of research has arisen that occupies of large-scale or Big Data learning. The development of new methods of machine learning that allow to improve the efficiency and the computational performance of the current ones, in addition to expanding their potential fields of application, will suppose a significant progress in the field of the Artificial Intelligence.

Acknowledgments. This research has been supported in part by the Secretaría de Estado de Investigación of the Spanish Government through the research project TIN 2012-37954; and by the Consellería de Industria of the Xunta de Galicia through the research projects CN2011/007 and CN2012/211; all of them partially funded by FEDER funds of the European Union. Diego Peteiro-Barral acknowledges the support of Xunta de Galicia under *Plan I2C* Grant Program.

References

1. Jason Catlett. *Megainduction: Machine Learning on Very Large Databases*. PhD thesis, University of Sydney, 1991.
2. Max Bramer. *Principles of Data Mining*. Springer, 2013.
3. Mohammed J Zaki and Ching-Tien Ho. *Large-scale parallel data mining*, volume 1759. Springer Verlag, 2000.
4. Foster Provost. Distributed data mining: Scaling up and beyond. *Advances in distributed and parallel knowledge discovery*, pages 3–27, 2000.
5. Christopher B Walton, Alfred G Dale, and Roy M Jenevein. A taxonomy and performance model of data skew effects in parallel joins. In *Proceedings of the International Conference on Very Large Data Bases*, 1991.
6. S. Sonnenburg, V. Franc, E. Yom-Tov, and M. Sebag. PASCAL Large Scale Learning Challenge. http://largescale.first.fraunhofer.de/workshop/, 2008.

Optimización de la gestión energética en vehículos eléctricos mediante Sistemas Inteligentes Híbridos

Héctor Quintián[a], Emilio Corchado[b]

[a]Doctorando, Departamento de Informática y Automática, Universidad de Salamanca
hector.quintian@usal.es
[b]Director, Departamento de Informática y Automática, Universidad de Salamanca
escorchado@usal.es

1 Resumen

En el mundo actual casi todos los dispositivos funcionan con energía eléctrica, desde pequeños accesorios, teléfonos móviles, ordenadores, electrodomésticos del hogar hasta la gran maquinaría industrial. La electricidad está presente por tanto en casi todas las partes y lugares, y por tanto la generación renovable de energía es una gran comodidad. El último tipo de maquinaría que se resiste a este importante cambio son aquellas relacionadas con el transporte. A parte de otras razones (políticas, sociales, infraestructurales, etc.) el principal problema tecnológico se encuentra en el almacenamiento de energía a bordo de vehículos. Las soluciones electroquímicas son muy caras y pesadas comparadas con las soluciones actuales, siendo esta última característica muy importante para maquinaria móvil. Es por ello que el éxito de la implantación del vehículo eléctrico (VE) depende en gran medida de la situación actual de las baterías electromecánicas, sus dimensiones, coste, autonomía, ciclo de vida, etc.

Actualmente existen requerimientos urgentes[1] para reducir de forma radical la dependencia europea de las importaciones de petróleo así como reducir las emisiones de carbón en el transporte un 60% en el 2050, e incrementar la movilidad de los ciudadanos europeos. Estos requisitos implican un cambio dramático en la eficiencia del sistema de transporte europeo, así como el desarrollo y la implementación de nuevos y sostenibles combustibles y sistemas de propulsión.

Basado en todos estos factores este proyecto de tesis se centra en el siguiente concepto: **Como usar este caro y limitado recurso de la forma más eficiente para contribuir al progreso tecnológico.**

1.1 Hipótesis de partida

En este proyecto, la evaluación en el campo de la gestión energética inteligente en entornos de baterías para VE, se realizará atendiendo a aspectos cruciales, como

[1] WHITE PAPER – Roadmap to a Single European Transport Area – Towards a competitive and resource efficient transport system /* COM/2011/0º44 FINAL*/ - http://eurlex.europa.eu/LexUriServ/LexUriServ.do?uri=COM:2011:0144:FIN:EN:HTML

© Springer-Verlag Berlin Heidelberg 2011

duración de la carga, estrategias para frenado regenerativo, disposición de celdas y caras, costes, densidad de energía [kWh/kg], densidad de potencia [kW/kg], rapidez de carga, rapidez de descarga, número de ciclos de carga/descarga antes del primer fallo y antes del fallo completo del sistema de almacenamiento inteligente, pérdida de energía en condiciones de reposo, envejecimiento o pérdida progresiva de características con él, etc. Los datos requeridos para llevar a cabo tanto la fase de optimización como la de monitorización y control, serán proporcionados gracias a **la colaboración con las empresas INMOTIA y REPLENTIA**.

1.2 Investigaciones previas

En la actualidad, debido a que el sector de las energías renovables así como el VE son prioridades fijadas en "HORIZONTE 2020" de la EU, empieza a haber investigaciones sobre baterías eléctricas, VE, e infraestructuras para los VE (red de sistemas de recarga de VEs). Entre otras cabría destacar los siguientes proyectos europeos (FP7). **ELVA**: Este proyecto, genera, investiga y analiza los conceptos de diseño innovadores para vehículos eléctricos (http://www.elva-project.eu). **EASYBAT**: La misión principal es hacer frente a los nuevos cambios en la integración de las nuevas baterías complejos mediante la definición de nuevos conceptos para la inserción inteligente de las baterías y, en particular, mediante el desarrollo de interfaces genéricas para vehículos eléctricos (http://www.easybat.eu). **LABOHR**: tiene como objetivo desarrollar sistemas de baterías de ultra-alta energía para aplicaciones de automoción (http://labohr.eu). **MAENAD**: Investiga capacidades avanzadas para facilitar el desarrollo de VE eficientes y asequibles (http://www.maenad.eu).

Diversos proyectos nacionales y europeos, publicaciones y registros de propiedad, abalan la experiencia del director de este proyecto de tesis tanto en el **campo de Inteligencia computacional y optimización de procesos industriales y energéticos**: SoftComp, IT4Innovations, FOEHN, QUICK, OPERA (http://opera.usal.es), MIDAS (http://midas.usal.es) y MAGNO, como en el **campo de la visualización de datos y dispositivos móviles**: "Desarrollo de Sistemas Distribuidos con Dispositivos Inalámbricos para Entornos Dinámicos con Tecnología de Agente" y "Desarrollo de aplicaciones autónomas para dispositivos inalámbricos basadas en tecnología de agente", **como en el del VE**: "Investigación, desarrollo e innovación en sistemas de comunicaciones e inteligencia artificial aplicados a vehículos eléctricos" La experiencia en dichos proyectos garantizará la consecución de los objetivos de este proyecto.

2 Objetivos

Este proyecto con un alto grado de interdisciplinariedad, consiste en la investigación, desarrollo e implementación de un sistema de gestión de almacenamiento inteligente de energía para VEs aplicando modelos de inteligencia computacional y algoritmos de control e identificación, con dos objetivos principales que se desarrollaran de forma paralela:

a) Optimización del almacenamiento energético de baterías para VE formadas por arrays de celdas, variando entre 200 y 6000 unidades para alcanzar una mayor eficiencia energética asociada a una reducción radical de los costes. La obtención de este modelo requiere las siguientes etapas:

1º) Se analizará e investigará la aplicación de arquitecturas neuronales basadas en aprendizaje no supervisado [1] para analizar los datos proporcionados por las empresas colaboradoras que definen el caso de estudio. Los modelos existentes se mejorarán mediante la investigación y aplicación de ensembles y modelos de fusión de la información con el fin de optimizar los procesos de visualización/clasificación [2] [3] [4], así como de obtener modelos más estables.

2º) Se investigará el potencial de los sistemas conexionistas anteriores en tareas de selección de características. También se compararán y aplicarán modelos de selección de características, como por ejemplo, tipo *filter* y tipo *wrapper* [5] [6]. Se investigarán además nuevos métodos así como modelos híbridos.

3º) Se investigarán sistemas de identificación para modelado de procesos [7], lo que permitirá realizar tareas orientadas a la supervisión de procesos y optimización de la gestión y almacenamiento energéticos de las baterías.

b) **Monitorización y control del VE mediante dispositivos móviles.**

4º) Desarrollo e integración de aplicaciones en dispositivos móviles (tabletas, *smartphones*, etc.) orientadas al usuario final (interacción vehículo-conductor, gestor de flota, etc.) para llevar a cabo las siguientes tareas: Monitorización de datos propios del VE, control remoto táctil/voz sobre vehículo, gestión de flotas.

Los resultados obtenidos en este proyecto serán probados en VEs de INMOTIA.

3 Metodología y plan de trabajo

T.1 Análisis del estado del arte: Análisis del estado del arte de forma continua en el tiempo, sobre el campo de los modelos híbridos inteligentes aplicados a la optimización de procesos de gestión y almacenamiento energético, así como de los dispositivos móviles integrados en vehículos y diversos aspectos de comunicación entre ellos.

HITO 1: Documento de recopilación de información relevante para el proyecto.

T.2 Estudio de viabilidad: Se llevará a cabo en la segunda tarea un estudio de viabilidad en base a su implementación al campo que nos ocupa.

HITO 2: Documento de estudio de viabilidad y alternativas.

T.3 Estudio y diseño del sistema: Esta tarea abarca el análisis y diseño del sistema inteligente de este proyecto (herramienta software), en colaboración con las empresas **INMOTIA y REPLENTIA**. Una vez realizado el estudio y análisis detallado de requisitos, se pasará al diseño y desarrollo del software informático.

HITO 3: Documento de estudio y diseño del sistema.

T.4 Análisis del Conjunto de Datos Inicial y Diseño de Experimentos: Se analizará el comportamiento de las muestras de datos, llamadas prototipo de ensayo sobre el software. Se aplicarán algoritmos estadísticos: Análisis de Componentes Principales; modelos conexionistas: EPP [8]; y modelos de clasificación y visualización no su-

pervisados, empleando ensembles [9] y métodos de fusión (WeVos) [10], para analizar la estructura interna de los conjuntos de datos a tratar, para conocer si el caso de estudio proporcionado por las **empresas colaboradoras** es suficientemente descriptivo y no se han producido errores en la toma de los datos.

HITO 4: Documento de estudio y comparación sobre modelos.

HITO 5: Documento de estudio de la validez de los datos.

T.5 Diseño e implementación de la capa de comunicación entre dispositivos móviles y VE: Esta tarea se centra en el diseño e implementación de la capa de comunicación entre dispositivos móviles y el CAN BUS del VE, teniendo en cuenta todos los elementos hardware de **bajo coste** (plataformas ARM, arduino, etc.) sobre los que se investiga en la tarea T3 en colaboración con INMOTIA. Dicha capa será validada sobre bancos de prueba proporcionados por la empresa colaboradora INMOTIA.

HITO 6: Documento de diseño e implementación de la capa de comunicación.

HITO 7: Documento de validación de la capa de comunicación.

T.6 Análisis, diseño e implementación de interfaz gráfica multiplataforma para monitorización, control de VE y gestión de rutas en dispositivos móviles: Esta tarea comprende el análisis, diseño e implementación de un sistema software multiplataforma usable, para mejorar la interacción vehículo-conductor a través de dispositivos móviles logrando una experiencia de usuario agradable y satisfactoria.

HITO 8: Documento de análisis, diseño e implementación del software.

T.7 Selección de características para tareas de filtrado/procesado: Se aplicarán y compararán distintos tipos de modelos y algoritmos para realizar trabajos de selección de características (*feature selection*) [5] [6] para tareas de pre-procesado. Su objetivo será conocer el grado de información de las variables de entrada y salida del modelo, así como eliminar las variables de entrada que no sean suficientemente informativas.

HITO 9: Documento de estudio y comparación de técnicas de Pre-procesado

T.8 Identificación y modelado del sistema y aplicación de técnicas metaheurísticas para optimizar el proceso: En esta tarea se lleva inicialmente una comparación exhaustiva de las técnicas de identificación para optimización de procesos [11]. También en esta fase se compararán diferentes técnicas metaheurísticas de optimización.

HITO 10: Base de conocimiento: relativa al modelado, la programación, las técnicas, la optimización y las aplicaciones.

T.9 Investigación y Aplicación de los modelos y algoritmos seleccionados al caso de estudio: En esta tarea se lleva a cabo la contrastación a través de la experiencia, es decir, una vez realizados los experimentos, diseñaremos los modelos que definan al sistema. El resultado será deducir una serie de reglas que recojan toda la investigación realizada en esta fase, de forma que quede recogido todo el conocimiento para ser automatizado en la tarea siguiente.

HITO 11: Deducción de las reglas de todo el conocimiento de la investigación.

T.10 Automatización de los procesos: software de optimización energética: En esta tarea se pretende que los modelos de sistemas dinámicos y los algoritmos de optimización obtenidos, sean instrumentos de toma de decisión, para propuestas de

optimización de procesos de almacenamiento de energía. El software permitirá diseñar experimentos, modelar, simular, predecir y optimizar el proceso.

HITO12: Software de optimización de sistemas de gestión de energía.

T.11 Construcción de prototipo para el caso de estudio: Se obtendrá un prototipo que será probado y evaluado tanto en bancos de prueba como en VEs de INMOTIA.

4 Relevancia

Los resultados obtenidos del estudio del comportamiento entre las celdas de baterías (hasta ahora muy poco investigado), así como el modelado de un sistema de gestión energética del VE junto con el desarrollo de una plataforma hardware/software para la monitorización y control del VE, abrirá nuevas vías de investigación relacionadas con el VE, así como con el campo de las energías renovables.

Por otro lado, el estudio sobre nuevos algoritmos para el análisis de la estructura interna del conjunto de datos basados en ensembles y selección de características basados en *wrappers, filters* y sistemas híbridos, permitirá una mejor comprensión y análisis de los conjuntos de datos en este y otros sectores.

La propuesta presentada permitirá medir, monitorizar y controlar los consumos energéticos para conseguir una mayor eficiencia y ahorro energético y una reducción del impacto medioambiental en la industria y los servicios.

Por último los resultados y modelos de gestión energética obtenidos serán validados sobre VE reales (INMOTIA), lo que facilitar su transferencia al sector automovilístico y por lo tanto a la sociedad en general. Además la aplicación al sector energético (REPLENTIA), podrá tener un impacto directo sobre el conjunto de sociedad, mediante una reducción de costes y contaminación.

Bibliografía

1. Corchado, E., Mata, A., Baruque, B., Corchado, JM, Pérez, B.A. Topology-preserving system for environmental models forecasting.Int. J. Comp. Math., 88(9), pp.1979-1989, 2011.
2. Galar, M., Fernández, A., Barrenechea, E., Bustince, H., Herrera, F. An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes. Pattern Recognition, 44(8), pp. 1761-1776, 2011.
3. Bolón, V., Sánchez, N., Alonso-Betanzos, A. A review of feature selection methods on synthetic data. Knowl. Inf. Syst., 34(3), pp. 483-519, 2013.
4. Ruiz, R., Riquelme, J.C., Aguilar, J.S., García, M. Fast feature selection aimed at high-dimensional data via hybrid-sequential-ranked searches. Expert Syst. Appl. 39(12), pp. 11094-11102 2012.
5. Ferreira, A.J., Figuereido, M.A.T. Efficient feature selection filters for high-dimensional data. Pattern Recognition Letters, 33(13), pp. 1794-1804, 2012.
6. Yu, J., A hybrid feature selection scheme and self-organizing map model for machine health assessment. Applied Soft Computing, 11(5), pp. 4041-4054, 2011.
7. Sedano, J., Curiel, L., Corchado, E., de la Cal, E., Villar, J.R. A soft computing method for detecting lifetime building thermal insulation failures. Integrated Computer-Aided Engineering, 17, pp. 103-115, 2010.

8. Corchado, E., Perez, J. A three-step unsupervised neural model for visualizing high complex dimensional spectroscopic data sets. Pattern Anal. Appl., 14(2), pp. 207-218, 2011.
9. Baruque, B., Corchado, E., Yin, H.: The S2-Ensemble Fusion Algorithm. Int. J. Neural Syst. 21(6), pp. 505-525, 2011.
10. Corchado, E., Baruque, B. WeVoS-ViSOM: An ensemble summarization algorithm for enhanced data visualization. Neurocomputing, 75(1), pp. 171-184, 2012.
11. Corchado, E., Sedano, J., Curiel, L., Villar, J.R. Optimizing the operating conditions in a high precision industrial process using soft computing techniques. Expert Systems, 29(3), pp. 276-299, 2012.

A Recommendation Framework Based on the Analytic Network Process and its Application in the Semantic Technology Domain

Student: Filip Radulovic - fradulovic@fi.upm.es
Supervisors: Raúl García-Castro, Asunción Gómez-Pérez

Ontology Engineering Group
Department of Artificial Intelligence
Facultad de Informática
Universidad Politécnica de Madrid
Spain

Key words: Multiple criteria decision making, Recommendation, Semantic Technologies

1 Introduction

When users face a certain problem needing a product, service, or action to solve it, selecting the best alternative among them can be a difficult task due to the uncertainty of their quality. This is especially the case in the domains where users do not have an expertise, like for example in Software Engineering.

Multiple criteria decision making (MCDM) methods are methods that help making better decisions when facing the complex problem of selecting the best solution among a group of alternatives that can be compared according to different conflicting criteria. In MCDM problems, alternatives represent concrete products, services or actions that will help in achieving a goal, while criteria represent the characteristics of these alternatives that are important for making a decision.

Various MCDM methods have been developed to this date (e.g., PROMETHEE [1], Analytic Hierarchy Process (AHP) [2], Analytic Network Process (ANP) [3], and others). Although MCDM methods are widely accepted and used in many different fields for recommendation problems, different MCDM methods are suitable for different kind of problems, and no method is considered to be the best to be applied in every decision making problem [4].

Quality models provide a common framework for quality specification and evaluation by specifying a consistent terminology for quality and by providing guidance for its measurement, and they have been successfully used to date with MCDM methods in recommendation problems. Some MCDM methods (e.g., the AHP and ANP) require the recommendation problem to be formalized into a model, which consists of the alternatives and criteria, and is regarded as one of the main advantages of such methods. In the Software Engineering field, and in the case of the AHP which requires a hierarchical structured model, hierarchical

software quality models (e.g., ISO 9126 [5] or SQuaRE [6]) are very convenient, and different authors have used software quality models based on the ISO 9126 together with the AHP.

Semantic technologies provide new ways to express in machine-processable formats knowledge and data that can be exploited by software agents, and we have seen an exponential growth of these technologies in recent years. Different types of semantic technologies exist, which can be used for different tasks, and which fulfill different requirements. For example, ontology editors are used for implementing ontologies, while ontology matching tools are used for mapping concepts of one ontology to concepts of other.

In the field of the Semantic Web, developers often have to use various tools that belong to different types of semantic technologies, and there exist a large number of tools for every type of technology already. Selecting the right tool or set of tools can help developers to perform their tasks more efficiently, and using the high quality tools will improve the quality of the products. Therefore, the selection of appropriate tools that best suit developers' needs is of crucial importance in the development of the emerging area of the Semantic Web.

The main goal of this thesis is to develop a theoretical recommendation framework based on a MCDM approach, and to apply it in the domain of semantic technologies. One characteristic of semantic technologies is that the quality criteria among them are mutually dependent. When it comes to the MCDM field, the ANP is a method that has been specifically designed to take into account the dependencies between the criteria in the decision process. Therefore, the ANP has been taken as a basis for the recommendation framework to be developed.

We have performed an extensive analysis of the previously mentioned topics in order to analyze the state of the art and to identify the insufficiencies. We found that the following insufficiencies are present:

- *Lack of an approach in the Analytic Network Process that will enable automatic comparisons of alternatives based on real evaluation results.*
  In MCDM recommendation problems, comparisons between alternatives are usually performed by experts in the field, or by users themselves. In some cases, however, it is difficult for ordinary users to make comparisons because of their lack of expertise. Furthermore, although expert opinion is highly valuable in many cases, it can be subjective and expensive, and not every expert may have expertise in every alternative. Also, those cases when new alternatives are added require additional expert comparisons.
  In those cases where the evaluation of alternatives has already been performed and evaluation results are available, it can be useful to ground the comparisons of alternatives in such evaluation results.
- *Lack of an approach in the Analytic Network Process that will enable recommendations based on user quality requirements.*
  The model of the problem is usually developed by experts, and it includes every criteria that are relevant for the recommendation of the best alternatives. Sometimes, however, not every criteria are relevant to the problem that a user faces, for which he or she can have specific quality requirements.

In these cases, constructing models and comparing alternatives separately for each user would be a resource consuming task.
- *Lack of an open source software support for the ANP method, including the support for the evaluation results and user quality requirements.*
- *Lack of an application that will provide users with semantic technology recommendations based on their needs and real evaluation results.*
  As it is often the case that when solving certain problems users have to use various semantic technologies that belong to different types, selecting the right technologies for solving a problem can be a difficult task, especially for less experienced users.

2 Hypotheses

In this thesis, we state the following hypotheses:

- **H1**: The algorithm for the automatic comparison of alternatives is able to produce a set of outputs that, for a given criteria and a set of evaluation results, defines the difference among the alternatives with a degree equivalent to a difference in the evaluation results. In order to accept or reject this hypothesis, we plan to test the algorithm on random evaluation datasets with various statistical distributions and observe the produced outputs.
- **H2**: The algorithm for the automatic construction of the ANP network from user quality requirements can produce recommendations that are closer to user's needs in terms of the number of satisfying requirements. In order to accept or reject this hypothesis, we plan to compare the recommendations obtained with and without the use of the algorithm.
- **H3**: The quality model is independent of the MCDM method that is planned to be used for the recommendations. As in our case we use the ANP, in order to accept or reject this hypothesis we plan to fin other examples where quality models with the same structure are used with different MCDM methods.
- **H4**: Instantiation of the proposed ANP recommendation framework in the domain of semantic technologies can produce correct recommendations. In order to accept or reject this hypothesis, we plan to compare the recommendations obtained with the proposed recommendation framework with a previously defined benchmark/golden standard, which can be extracted from the SEALS European project.

3 Objectives

This thesis aims to provide solutions to the insufficiencies of the State of the Art mentioned in the introduction. In particular, as the main contributions it will provide:

1. A generic recommendation framework built on top of the ANP, which will be able to ground the recommendations in evaluation results and take into account users' needs. The framework is shown on Figure 1.

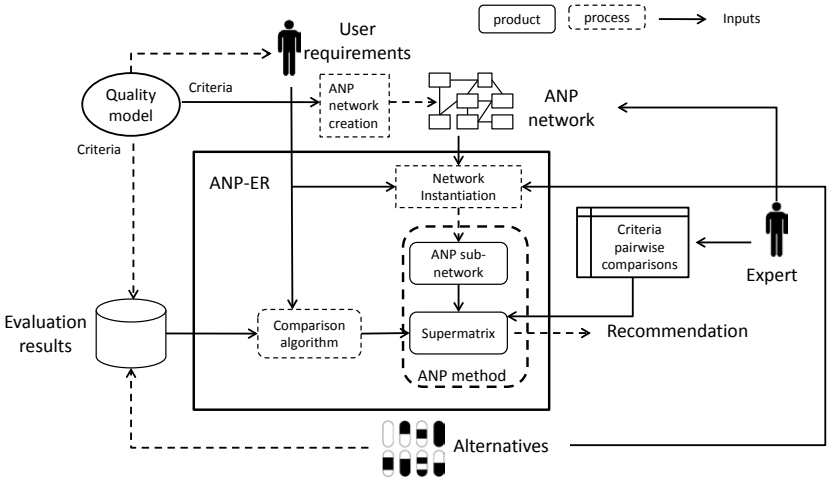


Fig. 1: ANP-based recommendation framework to be developed in the thesis.

2. A software support for the recommendation framework in terms of a Java API and a generic web application for the recommendations.
3. An instantiation of the ANP recommendation framework based on the comparison algorithm in the domain of semantic technologies through a web application for the recommendation of semantic technologies according to users' needs. This web application will exploit the results of semantic technology evaluations produced in the SEALS European project [7].

To achieve the previously specified goals, we will perform the following tasks:

- Develop a domain independent ANP-based comparison algorithm which exploits real evaluation results for automatic comparison of the alternatives based on user requirements. Instead of relying on expert opinions, the comparisons will be based on the results of the process of evaluation of the alternatives. Furthermore, the comparisons in the algorithm will be based on user quality requirements, i.e., the comparison of the two alternatives will depend on a threshold (a minimum quality that a user wants to be satisfied) specified by the user.
- Develop an algorithm for the automatic construction of the ANP network from the user quality requirements. Such a network can be constructed from a broader ANP network that is defined for a particular domain and, because it is grounded in user quality requirements, will be specific for the problem that a particular user faces.
- Develop a software support for the ANP method based on the comparison algorithm, which includes generic programming interface, and a generic Web application for the recommendations.
- Develop a quality model for semantic technologies, which will provide a basis for the specification of user quality requirements.

- Construct an ANP network for the domain of semantic technologies, starting from the semantic technology quality model.
- Perform the pairwise comparisons for the criteria in the ANP network for the semantic technologies, which is a requirement for the ANP method. Alternatives will be compared automatically using the comparison algorithm.
- Implement a web application for the recommendation of semantic technologies.
- Perform an evaluation of the main contributions of this thesis.

4 Methodology And Work Plan

In order to achieve the previously defined objectives, several tasks have to be performed:

- The comprehensive analysis of the state of the art in the field of software quality models will be performed in order to determine which software quality model will be the most suitable to adopt and extend in order to include the particularities of the semantic technology domain. Also, the purpose of this task is to analyze the existing methods for extending the software quality models.
- The analysis of the state of the art in the evaluations of semantic technologies will be performed in order to determine which quality characteristics are the most evaluated, and which quality measures are used to evaluate the quality. One important aspect of this task will be on the process of evaluation and methods for the calculation of quality measures.
- Development of the quality model for the semantic technologies. The most suitable method for extending quality models will be used and an existing software quality model will be extended. In the case that none of the existing methods can be used, a new method for extending software quality models will be developed.
- The comprehensive analysis of the state of the art in the field of multiple criteria decision making will be performed. The results of the state of the art analysis will help us to find the most suitable MCDM method that can be used for the recommendation of semantic technologies, and to identify existing gaps. This task has already been performed, and the ANP is adopted in this thesis.
- The adopted multiple criteria decision making method will then be extended to fit the requirements of the semantic technology domain, and a theoretical recommendation framework will be defined. The theoretical framework will be based on the user quality requirements and real evaluation results.
- The theoretical framework will be implemented into a semantic technology recommendation system, which will be realized in the form of a Web application.
- The proposed semantic technology quality model and the recommendation framework will be evaluated in order to verify the results of the performed work.

5 Relevance

This thesis will provide a valuable extension of the ANP method upon which a research community can rely on when facing specific problems of recommendations in which the key focus is on evaluation of quality of alternatives, and providing to wider range of users a possibility to specify their requirements.

Selecting an appropriate software product is not an easy task and it is regarded as a very important problem. For example, it is a well known fact that selection of ERP systems is highly sophisticated process [8] which requires deep analysis of available software, and where wrong decision can lead not just to significant financial loss and failed investment, but can also have negative influence on companys business. The recommendation framework that is going to be developed in this thesis will be a useful tool for industry practitioners that will help them to provide recommendations in any domain, and that will help users of the framework to find the best solutions for their problems. Furthermore, the web application for the recommendation of semantic technologies will help semantic web developers to choose the best tools for their tasks.

Acknowledgments

This work has been supported by the SEALS European project (FP7-238975) and by the EspOnt project (CCG10-UPM/TIC-5794) co-funded by the Universidad Politécnica de Madrid and the Comunidad de Madrid.

References

1. Brans, J., Vincke, P.: A preference ranking organization method. Management Science **31** (1985) 647–656
2. Saaty, T.: Decision making with the Analytic Hierarchy Process. International Journal of Services Sciences **1** (2008) 83–98
3. Saaty, T.: Fundamentals of the Analytic Network Process - Dependence and feedback in decision-making with a single network. Journal of Systems science and Systems engineering **13** (2004) 129–157
4. Guitouni, A., Martel, J.: Tentative guidelines to help choosing an appropriate MCDA method. European Journal of Operational Research **109** (1998) 501–521
5. ISO: ISO/IEC 9126-1:2001, Software engineering – Product quality – Part 1: Quality model. Technical report, International Organization for Standardization (2001)
6. ISO: ISO/IEC 25010-CD - JTC1/SC7, Software Engineering - Software product Quality Requirements and Evaluation (SQuaRE). Technical report, International Organization for Standardization (2008)
7. García-Castro, R., Esteban-Gutiérrez, M., Gómez-Pérez, A.: Towards an infrastructure for the evaluation of semantic technologies. In: eChallenges, 2010, IEEE (2010) 1–7
8. Hecht, B.: Choose the right ERP software. Datamation-Highlands Ranch **43** (1997) 56–61

A multi-modal and data mining based emotion detection approach to build an affective open standards-based user model to support adaptive learning environments

PhD Student: Sergio Salmeron-Majadas

Advisors: Jesus G. Boticario and Olga C. Santos

aDeNu Research Group. Artificial Intelligence Dept. Computer Science School. UNED
C/Juan del Rosal, 16. Madrid 28040. Spain
ssalmeron@bec.uned.es; jgb@dia.uned.es; ocsantos@dia.uned.es

Keywords: Data mining, Affective Computing, Emotions

1 Introduction

The importance of emotions in our lives is a factor computing has never dealt with until a few years ago affective computing started its developments in the mid 90s [1]. When carrying out a cognitive process, affective states play a key role [2] and this is something that should be taken advantage of when providing affective feedback to students [3]. The work we are carrying out aims at fostering learning performance in current adaptive systems in terms of personalized emotional support. Thereof the first and foremost essential step to offer that adaptive emotional support is to detect the learners' affective states, which is still an open research question [4].

On the proposed research, which is framed in the MAMIPEC project [5], we are following a multimodal approach to detect emotional changes. which is based on processing data coming from different information sources. Once this step has been given the information inferred needs to be added to the user model so it can be useful when providing feedback. The proposed feedback to be given in our work will be given in form of recommendations, so, a recommender system will need to be adapted to this new information inputs, expecting better of learning performance. To make this possible, emotional modeling will be needed, building, with this modeling, the bridges between the data processing and the feedback elaboration system.

2 Hypotheses

The main topic of this work concerns a field where a wide research is currently being done: the importance of affective states in learning [6] and their relations with other user characteristics, which is something long discussed in the literature [7]. The goal proposed in this work is being widely pursued, but usually using a single data source (as can be physiological data [8] or facial expression detection through video recordings [9]). Recent works using a multimodal approach are showing that fusioning

© Springer-Verlag Berlin Heidelberg 2011

different data sources can improve the results obtained by using a single data source [10].
Price and intrusiveness are problems that should be addressed when selecting data sources, being proposed other common devices (which can offer much more information than the traditionally obtained from them, as can be mouse movements or keyboard interactions [11]), new popular devices (Microsoft kinect [12]) or new self-made sensors (arduino based home-made sensors [13]) as emotional information sources.

The hypothesis of this work states that data mining techniques on multimodal data sources can detect affective features for learner modeling in educational contexts.

3 Objectives

To do the aforementioned, a data processing methodology is necessary, able to extract the affective information given by diverse data sources. This methodology has to make use of those sources to detect which one offers the most meaningful affective information while using educational resources. To get there some objectives must be set out:

1. Identify data sources used to gather affective information, paying special attention to their intrusiveness.
2. Study existing approaches and techniques proposed to detect affective states in the literature.
3. Characterize the existing affective modeling approaches.
4. Design and create an infrastructure able to provide tools to gather data from the sources considered as valuable.
5. Collect huge amounts of data to carry out the data mining process.
6. Test and improve existing algorithms on emotional information extraction from single data sources, improving them or proposing new ones if needed.
7. Study different information fusion procedures based on data mining algorithms to improve the emotion prediction results from unimodal approaches.
8. Evaluate the results obtained from the data fusion approach adopted.
9. Enrich the User Model in order to reflect the affective information generated by this work that is being used to provide adaptive personalized support.
10. Evaluate the benefits of the addition of affective information to the learner model.

4 Methodology and Work Plan

A step-wise research methodology has been proposed in this PhD, which consists of the following tasks:

1. Study of the state of the art: Regarding the first objectives aforementioned (1, 2 & 3), a study about how the user's emotional dimension has been carried out considering relevant works from related fields (e.g. Human Computer Interaction, Artificial Intelligence in Education, User Modeling Adaptation and Personalization, In-

telligent Tutoring Systems…). Existing approaches to identify affective states to their proposed representations are to be analyzed, with attention to those concerning the educational range and the techniques used (specially data mining).
2. Methodology analysis: analysis and definition to be followed during the elaboration of the Ph.D in the frame of the MAMIPEC project. After first experiments (in a controlled environment) to gather data and confirm the approach's feasibility in its best conditions, a data processing stage is followed by a design proposal, that should be appraised in new experiments, evaluating its results in field experiments with some variations as the use of cheaper and less intrusive sensors can be.
3. This methodology has to regard experiments to gather the required data to apply different data mining approaches in search of a definitive solution proposal.
4. Development of a solution proposal: Design, elaboration and revision of the proposed solution to the suggested problems and the considered hypothesis. This solution has to be assessed and its results evaluated, being used to redefine the solution following an iterative approach. The definitive solution must be proposed taking issues as interoperability and accessibility into account. To this, different existing standards (e.g. IMS-LIP, IMS-A4A, W3C EmotionML) are to be used, extending them if considered needed.
5. Use of the developed solution in the following stages of the project : integrating the resultant information into the a new User Model including affective information to offer adaptive support in the form of recommendations created by the use of the TORMES user centered design methodology [14] and evaluate the learning improvements resultant from this affective based adaptive support. An iterative approach will be applied to improve the results in successive iterations.
6. Compilation of lessons learnt, best practices and open issues: reflecting all the knowledge acquired during the research process and showing it in the PhD Thesis.

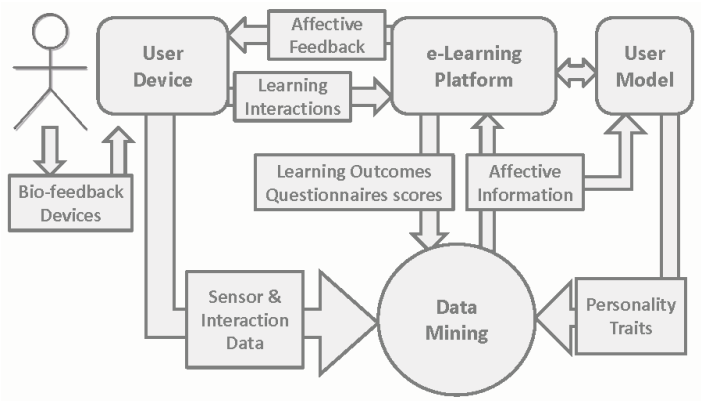


Fig. 1. Proposed architecture to provide emotional feedback

As can be seen in figure 1, the data mining algorithms will be fed with information coming from different data sources (e.g. galvanic skin conductance, temperature, heart rate, breath rate, keystrokes, mouse movements, facial features detection and video, personality questionnaires, self-emotional reports, learning outcomes). The

data mining algorithms output will be used to as an input to enrich the user model and to the learning platform to supply the information needed to provide affective feedback.

Until now, some works have been done. First data gathering activities have already taken place: some experiments have been carried out with participants coming to aDeNu laboratory to solve a series of mathematical tasks while being monitored (the used devices were: heart rate monitor, galvanic skin response sensor, respiration belt, temperature sensor, Microsoft Kinect, webcam, keylogger and mouse tracker) [15]. The first data mining tests are being made with the collected data, but the need of a deeper preprocess of the individual data flows has been detected. Text Mining techniques have been used to emotionally score reports typed by participants when solving a task (set of math problems) and being able to calculate the overall emotional valence of each task proposed has been marked as a goal. Using the text mining scores (filtering out the reports marked as neutral or ambiguous) and the keylogger generated information, a success rate near 70% in predicting positive or negative valence was achieved. To label these reports, two experts where asked to provide an affective labeling (similar to the generated by text mining) [16].

5 Relevance

This works aims to enrich the User Model, being able to take the user emotions into account to provide personalized emotion-based feedback. The impact this approach can have on learners' performance should be evaluated.

The first expected outcome of the presented work is the validation of its hypothesis, meaning this the capability to detect user's affective states changes by means of the use of data mining algorithms over some user data. Other expected contributions can be the development of affective-detection oriented data mining techniques, the enrichment of some standards in order to reflect affective information and offer a standards-based model regarding the affective states.

The current work has also been disseminated in some conferences as EDM, AIED, HCI or UMAP, sharing the first experiments results and open issues found to cope with on the data gathering stage, such as signal synchronization or noise filtering.

Acknowledgments

Authors would like to thank MAMIPEC project (TIN2011-29221-C03-01) colleagues and the Spanish Government for its funding and the FPI grant for the Ph.D student (BES-2012-054522).

References

1. Picard, R.W.: Affective Computing. MIT Technical Report #321. MIT Media Laboratory. (1995)

2. Craig, S., Graesser, A., Sullins, J., Gholson, B.: Affect and learning: an exploratory look into the role of affect in learning with AutoTutor. Journal of Educational Media. 29, 241–250 (2004).
3. Robison, J., McQuiggan, S., Lester, J.: Developing empirically based student personality profiles for affective feedback models. Proceedings of the 10th international conference on Intelligent Tutoring Systems - Volume Part I. pp. 285–29 Springer-Verlag, Berlin, Heidelberg (2010).
4. Picard, R.W.: Affective computing: challenges. International Journal of Human-Computer Studies. 59, 55–64 (2003).
5. Santos, O.C., Boticario, J.G., Arevalillo-Herráez, M., Saneiro, M., Cabestrero, R., del Campo, E., Manjarrés, Á., Moreno-Clari, P., Quirós, P., Salmeron-Majadas, S.: MAMIPEC-Affective Modeling in Inclusive Personalized Educational Scenarios. Bulletin of the IEEE Technical Committee on Learning Technology. 14, 35 (2012).
6. D'Mello, S., Taylor, R.S., Graesser, A.: Monitoring affective trajectories during complex learning. Proceedings of the 29th annual meeting of the cognitive science society. pp. 203–208 (2007).
7. Dennis, M., Masthoff, J., Mellish, C.: Towards a model of Personality, Affective State, Feedback and Learner Motivation. (2012).
8. Jang, E.-H., Park, B.-J., Kim, S.-H., Chung, M., Park, M.-S., Sohn, J.-H.: Classification of Human Emotions from Physiological signals using Machine Learning Algorithms. ACHI 2013, The Sixth International Conference on Advances in Computer-Human Interactions. pp. 395–400 (2013).
9. Cohen, I., Garg, A., Huang, T.S.: Emotion recognition from facial expressions using multilevel HMM. Neural information processing systems (2000).
10. D'Mello, S., Kory, J.: Consistent but modest: a meta-analysis on unimodal and multimodal affect detection accuracies from 30 studies. Proceedings of the 14th ACM international conference on Multimodal interaction. pp. 31–3 ACM, New York, NY, USA (2012).
11. Epp, C., Lippold, M., Mandryk, R.L.: Identifying emotional states using keystroke dynamics. Proceedings of the 2011 annual conference on Human factors in computing systems. pp. 715–724 (2011).
12. Shminan, A.S., Tamura, T., Huang, R.: Student Awareness Model based on student affective response and generic profiles. 2012 International Conference on Information Science and Technology (ICIST). pp. 675 –681 (2012).
13. Coghlan, Niall and Knapp, R. Benjamin (2008) Sensory Chairs: A System for Biosignal Research and Perfor-mance. In: Proceedings of 2008 Conference on New Instruments for Musical Expression (NIME08), 5-7 June, Genoa, Italy (2008).
14. Santos, O.C., Boticario, J.G.: Affective Issues in Semantic Educational Recommender Sys-tems. CEUR workshop proceedings, 896, pp. 71-82 (2012).
15. Santos, O. C., Rodríguez-Ascaso, A., Boticario, J. G., Salmeron-Majadas, S., Quirós, P., Cabestrero, R. Challenges for inclusive affective detection in educational scenarios. In proceedings of HCI International 201 Lecture Notes in Computer Science, in press (2013)
16. Santos, O.C., Salmeron-Majadas, S., Boticario, J.G. Emotions detection from math exercises by combining several data sources. In proceedings of the 16th International Conference on Artificial Intelligence in Education, pp. 742–745 (2013).

Improving recent episodic memory through location-based serious games

PhD student: Jesús A. Sánchez

Advisors: Juan A. Álvarez-García, Juan A. Ortega

Escuela Técnica Superior de Ingeniería Informática, University of Seville, Spain
{jsanchez53, jaalvarez, jortega}@us.es

Keywords: episodic memory impairments (EMI), geolocation, lifelogging

1 Introduction

As World Health Organization reported [8] 35.6 million people were estimated to be living with dementia in 2010 and all the future predictions point that there will be an exponential growth (about 104 millions of people affected in 2050) of this kind of diseases. The most famous one, Alzheimer's disease results in significant episodic memory impairments (EMI). In this article the focus will be put on EMI diseases that force entire families to become caregivers and usually cause familiar dramas [6]. To improve the mental skills of these patients there exists some games such as Mahjong[1] that can assist another treatments to get better results. In this article, a geolocation-based game will be introduced in order to help patients with their memory disorders. The point of this article is centered on the fact of how Lifelogging systems [7] can be turned in memory serious games to enhance people's mental health affected by EMI.

The rest of the paper is organized as follows: previous works will be summarized in section 2. In section 3, the authors' proposal will be introduced and finally relevance will be drawn in section 4.

[1] http://en.wikipedia.org/wiki/Mahjong

2 Related works

Lifelogging systems has been deeply studied since 2005, when prices of GPS begun to decrease, reaching to a large amount of population. Thus, words like life-tracking, lifelogging or digital diaries started to proliferate. Lifelogging systems automatically record a log of a users' personal experience in the form of pictures, sounds, actions, activities, or raw sensor data using wearable or embedded sensors such as cameras, audio recorders, location tracking, and bodily sensors. The idea behind these systems (focused on people with memory impairments) is to gather information about their lifestyle, their behaviors and even the caregivers' behaviors. The aim of these systems is to help patients to remember experiences through clues provided by these kinds of systems. The closest approach to the objective described here is the one from Lee and Dey [5] introducing a system composed by an audio recorder, a wearable camera and a GPS Logger that capture events and after they are used in a "memory exerciser" application.

There are other related works focused on another applications: a personal digital diary, social applications such as a friend finder using geolocations or even predictions over diaries [3].

Location data is one of the most important data in order to evocate recalls. Kalnikaitė study [4] points out that location data combined with images gives a good context to take a look back at the past because patients can go through a large amount of information faster than whether just having only images or only location data (data provides a context to infer where they were). Regarding to useful games used by people who are affected by EMI, there are some studies that show that games like Mahjong [1] can perceptibly help them improving their mental skills such as memory or numeric reasoning.

3 Goals

In this section, the authors' proposal is explained. The hypothesis is if users' memory (normally, aimed to users with a degree of memory degradation) can be improved through a web game. This game will use spatial-temporal registers to perform an analysis over users' data (In this case, Google Latitude is being used but I could be change in any moment).The application gathers and processes the past locations of

the users, using Google Latitude[2]. Processing step determines the meaningful locations for the user. After that, routines are extracted in order to prepare a questionnaire with interesting questions about places and days where and when his behavior was usual or unusual. Once the user answers the questionnaire proposed by the web game a mark will be assigned and will be stored to compare to his previous results. Although this kind of games are recommended to any kind of user, it is not necessary to be affected by EMI to realize the enhancements, indeed, there are a some studies where they point out that this kind of games would be useful to enhance another mental issues like removing cognitive fog after a chemotherapy treatment. In any case, the eventual user will be facing a series of questions about his/her last locations. They could be asked questions like *"Where was the last time where you have been in Brazil?"* or *"What places have you visited last Wednesday?"*

After this sequence of questions, the user can be rated, and this rate would be stored in a database, so by this way an eventual progress of the patient (user) could be shown later.

Under this serious game, there exist different algorithms for extracting significant locations, routines, selecting meaningful questions and giving feedback and clues to the user to improve his memory. These are the main goals that we would like to study in deep. Furthermore other applications such as spatial-temporal prediction could be explored.

3.1 Methodology and Work Plan

To achieve these objectives, a methodology or roadmap has been defined. This methodology has a set of milestones defined as follows:

- A study about the state of art of significant location extraction, temporal and spatial routines from logs and or location registers.
- Revision of ranking systems and questions selection for a web-based serious game. These questions are extracted from user's historical and behavioral log.
- An entity model to represent user's memory state in order to allow data mining to detect anomalous behaviors or potential illness.
- Rich user interfaces that encourage users to use the web application frequently.
- Security system to ensure user's privacy.

[2] https://www.google.com/latitude/

- Hypothesis testing between outcome from users that do not use any memory enhancing system and users that do. Hypothesis would be re adapt if necessary.

To reach this objectives, the system is compounded by a set of modules:

- **Web Game:** This module is the view layer of a Model View Controller (MVC) pattern; it contains all the interface logic to show a form where the user will be asked questions about the past.
- **Gathering Layer**: This layer contains all the code associated to gather all the past locations from a given user. It performs requests against Google Latitude Rest API [3] in order to gather the whole users' data. This historical is stored in a database so that it can be processed.
- **Processing Layer**: This layer is compound by three sub layers, significant places recognition, routines recognition and questionnaire generation sub layer. First one receives as input the whole user historical data and returns as output a set of centroids (being a centroid a pair of coordinates -latitude and longitude- that represent a significant place). To obtain it, a hierarchical cluster is used. Criteria to cut the cluster is related to distance and time. The routines recognition sub layer applies algorithms [2] over the centroids and their appearance in the past in order to extract the spatial-temporal routines. Finally, the questionnaire generation sub layer prepares the significant questions selecting some interesting days and places to ask about in order to find the difference between the typical places or days and the unusual ones.
- **Data Layer**: This layer is just a storage service that is consumed by some other layers like Gathering and Processing layers in order to store and retrieve information about users' location.
- **System Core**: This is the engine of the system, the piece that connects all the layers and it has the task of controlling Gathering, Processing and Data layers in order to provide consistent and stable information to the Web Application. This layer has another important tasks such as authentication tasks using OAuth protocol, KML generations and email deliveries.

4 Relevance

The main problem to go forward in this area of study is the inability to reach a large amount of users. The principal reason of that is that all previous studies were based on devices that are not included in the triplet wallet-keys-phone. Microsoft SenseCam[4], a wide variety of GPS sensors and custom devices made for this purpose are an important barrier. This obstacle avoids having a large database of users and their

[3] https://developers.google.com/latitude/
[4] http://research.microsoft.com/en-us/um/cambridge/projects/sensecam/

feedback. This proposal provides a generic system in order to allow an eventual user to train and improve his/her memory just carrying a smartphone. And obviously he/she has to be signed up in our web application.

On the other hand we have a disadvantage; users do not want to share all their past with a new application if they do not obtain enough reward. We think that this system can help the users to improve their memory and we are studying the possibility of working without storing the data although this will consume much more processing time.

Finally we can not show results due to the fact that work is yet in progress using only location data from 6 volunteers although the stored data includes more than 4 years and 500.000 points summing all the users.

5 References

1. Cheng, Sheung-Tak, Alfred C M Chan, and Edwin C S Yu. 2006. “An exploratory study of the effect of mahjong on the cognitive functioning of persons with dementia.” International journal of geriatric psychiatry 21(7): 611–7.
2. Ferrari, Laura, and Marco Mamei. "Classification and prediction of whereabouts patterns from reality mining dataset." Pervasive and Mobile Computing (2012).
3. Ferrari, Laura, Marco Mamei, and Franco Zambonelli. "All-about digital diaries: Opportunities and challenges." IT Professional 13.5 (2011): 37-43.
4. Kalnikaitė, Vaiva, Portobello Street, S Sheffield, Abigail Sellen, Steve Whittaker, and David Kirk. 2010. “Now Let Me See Where I Was : Understanding How Lifelogs Mediate Memory.” 2045–2054.
5. Lee, Matthew L., and Anind K. Dey. "Lifelogging memory appliance for people with episodic memory impairment." Proceedings of the 10th international conference on Ubiquitous computing. ACM, 2008.
6. Lee, Matthew L., and Anind K. Dey. "Using lifelogging to support recollection for people with episodic memory impairment and their caregivers." Proceedings of the 2nd International Workshop on Systems and Networking Support for Health Care and Assisted Living Environments. ACM, 2008.
7. Sellen, By Abigail, and Steve Whittaker. 2009. “Beyond Total Capture : A Constructive Critique of Lifelogging.” 1–12.
8. World Health Organization. 2012. Dementia 112 Dementia: a public health priority. eds. Tarun Dua, Pamela Nichols, and Yutaro Setoya. World Health Organization.

Bio-inspired Computing and Smart Mobility

Doctorando: Daniel H. Stolfi
Director: Enrique Alba

LCC, Universidad de Málaga, España
{dhstolfi,eat}@lcc.uma.es
http://neo.lcc.uma.es

Keywords: Smart Mobility, Smart City, Algoritmos Bioinspirados

1. Introducción

Las ciudades hoy día evolucionan rápidamente. Un gran número de personas residen o están contemplando la posibilidad de trasladar su domicilio habitual a una ciudad, aumentando así la demanda de servicios, lo que supone una nueva fuente de problemas complejos. ***Smart Cities*** es una iniciativa mundial encaminada a mejorar la explotación de recursos en la ciudad de forma que se ofrezcan servicios de mejor calidad a sus habitantes. Esto se encuentra íntimamente relacionado con la recogida de datos provenientes del estado de la ciudad para actuar en consecuencia utilizando nuevas estrategias inteligentes a diferentes niveles: personas, vehículos, transporte, comunicaciones, energía, edificios, almacenamiento de recursos, etc.

El empleo de las tecnologías de la información para resolver los problemas que presenta un ciudad del siglo XXI nos permite combinar técnicas de recogida de datos con algoritmos basados en metaheurísticas, tanto para realizar optimizaciones como para analizar estos datos (de forma anónima) y ponerlos a disposición de las autoridades y ciudadanos para su consulta (*open data*).

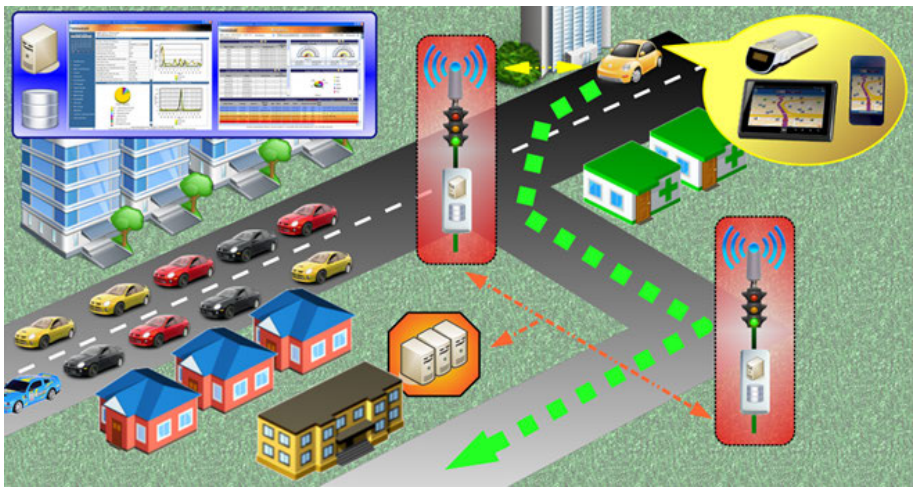


Figura 1. Contexto de esta tesis: *Smart cities*.

Los **algoritmos bioinspirados** permiten encontrar muy buenas soluciones, usualmente la óptima, a problemas complejos en tiempos de cómputo aceptables. No se trata sólo de utilizar las técnicas ya conocidas (GA, ACO ó PSO), sino que además se van a abordar nuevos conceptos basados en la epigenética [1].

La **epigenética** consiste en el mecanismo por el cual se aprecian variaciones en el fenotipo de un organismo sin que se produzcan variaciones del genotipo del mismo. Esto puede producirse por diversos procesos entre los que se encuentran la metilación del ADN y la modificación de las histonas, los cuales tienen su origen tanto en cambios aleatorios como en la influencia del medio ambiente [1] pudiendo influir en la expresión génica. Las variaciones epigenéticas son raramente heredadas por los descendientes del individuo dado que éstas pueden ser incluso revertidas por el mismo entorno. Sin embargo, los cambios en la metilación de una serie de cromosomas podrían ser heredados, e incluso permanecer durante más de una generación [2] si resultan beneficiosos para el individuo.

Durante el desarrollo de esta tesis se aplicarán los mecanismos epigenéticos como técnica bioinspirada para diseñar nuevos algoritmos que sean eficientes en problemas de optimización. Se trata de reflejar la influencia del entorno (representado por las características del problema, el espacio de búsqueda, etc.) sobre la población de un algoritmo evolutivo, realizando cambios en el fenotipo de los individuos que luego puedan ser heredados por sus descendientes o codificar y transmitir reglas de cambio. Si a esta técnica le sumamos el empleo de paralelismo, tanto en la evaluación de escenarios como en la búsqueda de soluciones, dispondremos de un método prometedor para afrontar problemas de diversos ámbitos dentro de una *smart city*.

En concreto, proponemos el empleo de la arquitectura **Red Swarm**, la cual está siendo desarrollada durante esta tesis, para la reducción de los tiempos de viaje [3] y las emisiones de gases de efecto invernadero [4] gracias a la utilización de dispositivos inteligentes conectados a ciertos semáforos, los cuales enlazan vía WiFi con los vehículos, sugiriéndoles nuevas rutas por la ciudad como estrategia para evitar los posibles atascos, accidentes, etc.

Cuando un vehículo se aproxima a un semáforo provisto de un dispositivo Red Swarm se inicia el enlace WiFi y se produce un intercambio de información. El vehículo informa de su destino y el dispositivo le responde con el siguiente trayecto a seguir que podrá conducirlo hacia otro dispositivo Red Swarm o hacia el destino final de su viaje. Con esta técnica y la apropiada configuración calculada *offline* por un algoritmo bioinspirado, se consigue una distribución del tráfico con el fin de evitar atascos, disminuir los tiempos de viaje, reducir las emisiones de gases a la atmósfera, etc.

La evaluación de las configuraciones se realiza utilizando el microsimulador SUMO [5] sobre escenarios reales importados desde OpenStreetMap. Mientras que el algoritmo de cambio de ruta se ha implementado utilizando la interfaz TraCI [6] para modificar el comportamiento de los vehículos desde fuera del simulador. Por último, se ha desarrollado la exportación desde SUMO a GoogleMaps^{TM} y GoogleEarth^{TM} de modo de observar diferentes instantáneas de la simulación en un entorno realista como el que ofrecen estos programas.

2. Hipótesis de Trabajo

La hipótesis de trabajo consiste en el empleo de algoritmos bioinspirados, en especial los nuevos epigenéticos cuya línea queremos arrancar en esta tesis, para resolver los problemas de una *smart city* obteniendo resultados que sean científica y socialmente satisfactorios, posibilitando también su aplicación industrial.

Como apoyo a la misma se cuenta con la experiencia del grupo de investigación en la resolución de problemas de características similares [7,8,9,10,11] cuyos enlaces se presentan en `http://neo.lcc.uma.es`.

Además, el proyecto nacional RoadME (TIN2011-28194) (`http://roadme.lcc.uma.es`) dará soporte a los primeros dos años del desarrollo de esta tesis al tratarse de una línea de investigación estrechamente relacionada con la misma. Asimismo, el grupo de investigación se encuentra en relación con la empresa Telvent así como con el Ayuntamiento de Málaga y la Consejería Andaluza de Fomento y Vivienda. Esto abre la puerta a posibles aplicaciones de las técnicas desarrolladas durante la investigación a problemas reales y cotidianos.

3. Objetivos

El objetivo de esta tesis es el estudio, diseño, implementación y evaluación de soluciones a los problemas que presenta una *smart city*, en especial los relacionados con *smart mobility*, utilizando las metaheurísticas basadas en algoritmos bioinspirados, en particular los epigenéticos. Este objetivo principal puede a su vez dividirse en los siguientes sub-objetivos:

1. Estudio del estado del arte correspondiente a los problemas de transporte inteligente, control de tráfico, optimización de rutas y reducción de tiempos de viajes y emisiones en búsqueda de nuevas aportaciones que realizar.
2. Estudio de las teorías epigenéticas para construir el modelo computacional equivalente aplicado a algoritmos de optimización, teniendo en cuenta la influencia del entorno en el fenotipo de cada individuo de una población.
3. Desarrollo de aplicaciones software aplicando los conocimientos adquiridos durante esta tesis con objeto de resolver problemas de *smart mobility*.
4. Utilización de diferentes tipos de dispositivos (*smartphones*, tabletas) para el desarrollo y despliegue de algoritmos de optimización distribuidos.
5. Análisis de los resultados y verificación de las capacidades teóricas y prácticas que presentan los algoritmos epigenéticos.
6. Difusión de los conceptos, resultados y aplicaciones desarrolladas.

4. Metodología y Plan de Trabajo

Las fases de esta tesis se basan en las establecidas por el **Método Científico** [12] para garantizar una metodología de trabajo rigurosa y bien definida. A continuación se detallan las fases y actividades que realizar en cada una de ellas:

- **Observación**
  1. Se analizarán los algoritmos bioinspirados existentes (ACO, AS, PSO, EA, etc.), así como otras técnicas heurísticas utilizadas actualmente en la resolución de problemas de *smart mobility*. Se estudiarán las mejores soluciones existentes tanto en el ámbito académico como en el industrial.
  2. Se estudiarán los fundamentos de la teoría epigenética (plasticidad del fenotipo, expresión genética, metilación del ADN) y su traducción a operaciones computacionales orientadas a la resolución de problemas de búsqueda y optimización.
  3. Se construirán modelos basados en mapas reales importados desde OpenStreetMap y adaptados para la realización de múltiples simulaciones, detectando los distintos problemas de movilidad.
- **Inducción**
  4. Reconocimiento cualitativo y cuantitativo en los resultados iniciales de las operaciones con más impacto en la calidad de las soluciones.
- **Hipótesis**
  5. Selección de las técnicas epigenéticas más prometedoras estableciendo una hipótesis sobre su posible aplicación en la resolución de problemas.
  6. Diseño e implementación de nuevos algoritmos con la posibilidad de ser extendidos a sistemas paralelos y distribuidos aprovechando los nuevos dispositivos móviles y sistemas operativos (Android, iOS, etc.).
- **Experimentación**
  7. Se estudiarán los algoritmos de forma experimental (simulaciones, aplicaciones, etc.) y matemático, realizando comparaciones estadísticas de resultados e identificando objetivos que admitan falsabilidad, mediante la recolección de datos desde simulaciones de escenarios preestablecidos.
  8. Se interpretarán los datos para plantear conclusiones preliminares encaminando el proceso de investigación hacia su finalización satisfactoria contrastando los resultados con los del estado del arte.
  9. Se determinarán las posibles aplicaciones finales basadas en los nuevos conceptos creados, destinadas a un aumento del desarrollo social y económico de una *smart city*. Nos enfocaremos en el acortamiento de los tiempos de viaje, la reducción de las emisiones de gases de efecto invernadero, la gestión de servicios prioritarios, reducción del nivel de ruido, etc.
- **Tesis/Conclusiones**
  10. Verificación de conclusiones, generalización a otros dominios como trasporte por carretera, aéreo (drones) y marítimo, y análisis global del trabajo realizado desde el punto de vista científico, social y económico.
  11. Establecimiento de relaciones con empresas con objeto de posibles contratos de colaboración y transferencia. Análisis de las posibles patentes fruto de esta investigación.
  12. Realización de estancias de investigación en centros extranjeros de reconocido prestigio en los campos de la tesis.

13. Realización de una web para la presentación de los resultados, documentos y software desarrollado, facilitando las instancias de los problemas, para su reproducción y validación por parte de la comunidad científica.
14. Publicación de los resultados como artículos científicos en conferencias y revistas internacionales de alto impacto.
15. Divulgación a la sociedad en forma de material multimedia (vídeos, animaciones) y noticias en prensa no especializada (periódicos, programas de radio, blogs, redes sociales, etc.).

El trabajo se realizará en el laboratorio 3.3.4 del departamento de Lenguajes y Ciencias de la Computación de la Universidad de Málaga, en donde el grupo de investigación NEO dispone de dos clusters de 16 nodos cada uno interconectados mediante redes Gigabit. Además se dispondrá de *smartphones* y tabletas en los cuales desplegar una arquitectura distribuida de optimización. Mediante las colaboraciones con otros miembros del grupo así como con los visitantes que frecuentemente realizan sus estancias en el laboratorio, se espera ampliar los límites de esta tesis incluyendo la obtención de datos reales desde la administración (Ayuntamiento) junto con colaboraciones con empresas del sector.

Por último cabe destacar que el candidato ha realizado su trabajo de fin de máster "Optimización del Tráfico Rodado en Ciudades Inteligentes" y las siguientes publicaciones relacionadas con la temática de esta propuesta de tesis:

- Red Swarm: Smart Mobility in Cities with EAs en *Genetic and Evolutionary Coputation Conference (GECCO) 2013*
- Reducing Gas Emissions in Smart Cities by Using the Red Swarm Architecture en *XV Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA) 2013*

5. Relevancia

Si bien el presente trabajo se concentra en mejorar la movilidad (*smart mobility*), una ciudad que adolece de atascos también se ve seriamente afectada en su economía (*smart economy*) debido a la pérdida de eficiencia en el transporte. Además, tiempos de trayecto más prolongados aumentan las emisiones de gases de efecto invernadero lo que contradice los principios del medio ambiente inteligente (*smart environment*). Por último, la fluidez del tráfico también afecta a la vida cotidiana de las personas (*smart living*), porque no sufrir atascos y disponer de más tiempo libre, supondrá una mejora en su calidad de vida y salud.

La aplicación de los mecanismos epigenéticos como técnica bioinspirada para el diseño de nuevos algoritmos con el objeto de abordar y resolver los problemas de *smart mobility* de una forma innovadora, como hacemos en esta tesis, representará un importante impacto con gran relevancia tanto en la sociedad como en la empresa y en la administración pública.

Referencias

1. Jaenisch, R., Bird, A.: Epigenetic regulation of gene expression: how the genome integrates intrinsic and environmental signals. Nature genetics **33** (2003) 245–254
2. Gorelick, R.: Evolutionary epigenetic theory. PhD thesis, Arizona State University (2004)
3. Stolfi, D.H., Alba, E.: Red Swarm: Smart Mobility in Cities with EAs. In: ACM Proceedings of the 13th Genetic and Evolutionary Computation Conference (GECCO'13), ACM (2013) In press
4. Stolfi, D.H., Alba, E.: Reducing Gas Emissions in Smart Cities by Using the Red Swarm Architecture. In: Proceedings CAEPIA'13. (2013) In press
5. Behrisch, M., Bieker, L., Erdmann, J., Krajzewicz, D.: SUMO - Simulation of Urban MObility: An Overview. In: SIMUL 2011, The Third International Conference on Advances in System Simulation, Barcelona, Spain (October 2011) 63–68
6. Wegener, A., Hellbrück, H., Wewetzer, C., Lubke, A.: VANET Simulation Environment with Feedback Loop and its Application to Traffic Light Assistance. In: GLOBECOM Workshops, 2008 IEEE, IEEE (2008) 1–7
7. Luque, G., Alba, E.: Parallel Genetic Algorithms: Theory and Real World Applications. Volume 367. Springer (2011)
8. Alba, E.: Parallel metaheuristics: a new class of algorithms. Volume 47. Wiley-Interscience (2005)
9. García-Nieto, J., Alba, E.: Automatic Parameter Tuning with Metaheuristics of the AODV Routing Protocol for Vehicular Ad-Hoc Networks. In: EvoApplications (2). Volume 6025 of Lecture Notes in Computer Science., Springer (2010) 21–30
10. Toutouh, J., Garcia-Nieto, J., Alba, E.: Intelligent OLSR Routing Protocol Optimization for VANETs. Vehicular Technology, IEEE Transactions on **61**(4) (2012) 1884–1894
11. Khouadjia, M.R., Alba, E., Jourdan, L., Talbi, E.G.: Multi-Swarm Optimization for Dynamic Combinatorial Problems: A Case Study on Dynamic Vehicle Routing Problem. In: ANTS Conference. Volume 6234 of Lecture Notes in Computer Science., Springer (2010) 227–238
12. Gauch, H.G.: Scientific method in practice. Cambridge University Press (2003)

Computación Natural en Redes Vehiculares

Doctorando: Jamal Toutouh
Director: Enrique Alba

Lenguajes y Ciencias de la Computación
Universidad de Málaga, España
{eat,jamal}@lcc.uma.es
http://neo.lcc.uma.es/

Keywords: Metaheurísticas, Computación Natural, VANETs, *Smart Mobility*

1. Introducción

El continuo crecimiento de la población mundial y su concentración cada día más acusada en ciudades ponen en jaque la viabilidad del modelo de desarrollo urbano. La iniciativa mundial *Smart Cities* persigue mitigar este problema incrementando la calidad de vida de los ciudadanos, mejorando la eficiencia de los recursos, facilitando la participación ciudadana, y, en definitiva, garantizando el desarrollo sostenible de las mismas. Uno de los ejes fundamentales es la *Smart Mobility*, que trata de paliar los problemas ocasionados por la congestión de las carreteras permitiendo desplazamientos más seguros, cómodos y eficientes.

Los avances en las tecnologías de comunicación inalámbrica han permitido la aparición de las *Vehicular Ad hoc NETworks* (VANET) o redes vehiculares ad hoc [1]. Las VANETs son redes descentralizadas que proveen de una plataforma para el diálogo de vehículos entre sí (*vehicle-to-vehicle* -V2V-) y con los elementos de la infraestructura vial (*vehicle-to-infrastructure* -V2I-) como semáforos, señales de tráfico, etc. Así, los vehículos pueden recibir, procesar y difundir información actualizada sobre distintos aspectos del tráfico (ver Figura 1).

Figura 1. Escenario VANET: Difusión de mensaje de precaución.

Numerosas iniciativas para el estudio y desarrollo de VANETs como *roadME*, *CAR-2-CAR Communication Consortium* y *EUREKA-CELTIC CARLINK* han sido impulsadas por los gobiernos y por la industria, apareciendo nuevas soluciones en el campo de los SIT (Sistemas Inteligentes de Transporte) [1] y la *Smart Mobility*. Fruto de estas iniciativas ha sido el diseño de servicios orientados a la prevención de accidentes, a la mejora de la eficiencia (tiempos de desplazamiento, emisiones de CO_2, etc.) e incluso al entretenimiento de los pasajeros.

El estándar IEEE 802.11p, basado en comunicaciones directas de corto alcance (DSRC), se ha definido expresamente para el acceso al medio inalámbrico en entornos vehiculares (WAVE) [1]. La limitada cobertura del estándar IEEE 802.11p y la alta movilidad de los nodos provoca que los enlaces que se crean durante la comunicación tengan un tiempo de vida muy limitado, lo que complica de forma crítica el correcto intercambio de paquetes (frecuentes cambios de topología y fragmentación de la red). Así, el **encaminamiento** (*routing*) eficiente de paquetes en redes vehiculares es una tarea altamente compleja [2].

Además, las VANETs requieren de un servicio de **difusión** (*broadcasting*) de mensajes para el descubrimiento de nodos cercanos y el envío de información. Sin embargo, en situaciones de tráfico denso (número elevado de conexiones), aparece el problema de tormenta por difusión, congestionándose la red [3].

Cabe destacar la importancia de disponer de una plataforma física para el despliegue de VANETs, es decir, de una **infraestructura** compuesta por nodos fijos (estaciones base) empleados para comunicar nodos móviles con redes estáticas (Internet) y con otros nodos móviles que estén fuera del alcance directo.

La alta volatilidad y el gran dinamismo de las redes vehiculares limita la aplicación directa de protocolos de encaminamiento y difusión ya empleados en otras redes móviles ad-hoc (MANETs) en las comunicaciones V2V, apareciendo diversas líneas de trabajo para el diseño de protocolos específicos [2]. La hipótesis principal que se baraja en esta tesis es que el diseño de protocolos para VANETs se puede plantear como un problema de optimización del rendimiento de la red en el que, por ejemplo, se maximiza la cobertura y la tasa de paquetes enviados y se minimizan los tiempos de entrega. Para la resolución de dicho problema se van a emplear técnicas **metaheurísticas** y de **Computación Natural (CN)** [4], que han demostrado ser eficaces en la resolución de problemas de optimización complejos de diversa naturaleza [5]. Así, el objetivo de este trabajo es el de abrir una línea de investigación en la aplicación de algoritmos como GA, PSO y DE, entre otros, al diseño de protocolos V2V eficientes para VANETs.

Por otro lado, la dificultad del diseño de la infraestructura para proveer de comunicación V2I consiste en la localización óptima de las estaciones base y la selección de sus componentes hardware y software que se usarán . Con ello se persigue reducir los costes de implantación (económicos, sociales y medioambientales), maximizando la cobertura y la robustez de la red. Para abordar el diseño de la infraestructura también se van a emplear técnicas CN como GA, SA y CHC, puesto que ya han demostrado su potencia en el diseño de la infraestructura de tipos de redes similares. Así, el objetivo es ofrecer una metodología basada en CN para abordar el diseño de infraestructuras para VANETs.

La tesis aportará avances en el campo de las VANETs que apoyarán la aparición de nuevos servicios que mejorarán el entorno urbano. Sin embargo, su desarrollo presenta varios desafíos, tales como: ampliar el estado del arte en CN con nuevas técnicas específicas aplicables en el ámbito de las VANETs, seleccionar paradigmas de inteligencia colectiva para inspirar nuevas estrategias de transmisión de paquetes V2V e identificar las estaciones base más adecuadas para proporcionar V2I, que representan líneas de trabajo novedoso y apasionante.

2. Hipótesis de Trabajo

La hipótesis de trabajo consiste en demostrar que los algoritmos de CN ofrecen una herramienta potente para la resolución de los problemas de diseño de VANETs presentados anteriormente. Así, se plantean tres ejes básicos:

- **Optimización *off-line***: Uso de técnicas CN para la configuración eficiente de parámetros de protocolos de difusión y encaminamiento que ya se emplean en MANETs previamente a su despliegue en VANETs [6,7,8].
- **Optimización *on-line***: Como las VANETs operan de forma distribuida, se plantea la utilización de estrategias CN basadas en inteligencia colectiva [9] (ACO, PSO, BCO, etc.) para el diseño de nuevos protocolos.
- **Diseño eficiente de la infraestructura**: Partiendo de la literatura sobre el uso de técnicas CN (GAs, SA, CHC, ...) en el despliegue de redes celulares y redes de sensores [10], se definen estrategias específicas para VANETs.

3. Objetivos

El objetivo troncal de esta tesis consiste en analizar, implementar y evaluar las soluciones propuestas a los problemas de diseño de protocolos e infraestructura en VANETs. Este objetivo consta a su vez de los siguientes sub-objetivos:

O.1 Estudiar en profundidad el estado del arte sobre los problemas que se tratan para identificar las debilidades que luego abordarán nuestras aportaciones.
O.2 Analizar las técnicas de CN en el estado del arte actual, para seleccionar aquellas que mejor se adaptan a los problemas que se tratan en la tesis.
O.3 Definir una metodología que se seguirá en el uso de CN para optimizar protocolos empleados en MANETs para desplegarlos después en entornos vehiculares.
O.4 Analizar diferentes paradigmas de inteligencia colectiva usados en CN para su empleo en el desarrollo de protocolos específicos para VANETs.
O.5 Diseñar infraestructuras eficientes para desplegar VANETs en escenarios geográficos reales, como por ejemplo Málaga, aplicando técnicas de CN.
O.6 Evaluar nuestras propuestas empleando la simulación como herramienta fundamental, incluyendo la validación mediante pruebas de concepto con equipamiento real (vehículos, *tablet PCs*, *smartphones*, ...) y comunicaciones reales.

4. Metodología y plan de trabajo

Para garantizar un desarrollo de la tesis riguroso y bien definido, la metodología se ha basado en las fases del **Método Científico**. En esta sección se presentan dichas fases y se precisan las actividades que se van a llevar a cabo:

- **Observación**

F.1 Estudio del estado del arte en CN, precisando (aunque no de forma excluyente) en las metaheurísticas aplicadas a problemas de comunicaciones, tales como, PSO, DE o GA.

F.2 Análisis de protocolos de encaminamiento y difusión probados en VANETs (AODV, OLSR, DSDV, ...) para identificar sus fortalezas y debilidades.

F.3 Implementación de modelos basados en datos reales (mapas de carreteras, comportamiento de conductores, comunicaciones, ...) para su empleo en tareas de simulación realista de redes vehiculares usando ns-2 y ns-3 [11].

- **Inducción**

F.4 Estudio de las métricas empleadas en telecomunicaciones como la tasa o el tiempo de envío de paquetes y el tamaño de las rutas empleadas, para la definición de experimentos adecuados de evaluación de las soluciones (protocolos y diseño de VANETs) obtenidas por los algoritmos CN.

F.5 Análisis de distintos operadores CN para diseñar nuevos algoritmos mejor adaptados a los problemas que se abordan en la tesis.

- **Hipótesis**

F.6 Obtención de protocolos eficientes para VANETs aplicando CN para optimizar protocolos empleados en MANETs.

F.7 Desarrollo de nuevas estrategias de comunicación V2V (encaminamiento y difusión de paquetes) empleando paradigmas de inteligencia colectiva.

F.8 Diseño de la infraestructura (localización y componentes de las estaciones base) que soportan la comunicación V2I usando CN.

- **Experimentación**

F.9 Análisis experimental y validación estadística tanto de los algoritmos CN propuestos como de las soluciones desarrolladas (mediante pruebas en laboratorio y en entornos reales de los protocolos y de infraestructura).

F.10 Establecimiento de objetivos evaluables (métricas y experimentos) que permitan la comparación de las soluciones propuestas con aquellas existentes, permitiendo a las primeras ser incluidas en el estado del arte.

F.11 Definición de prototipos y de posibles aplicaciones finales en el ámbito de la *Smart Mobility* (avisos de maniobras peligrosas por parte de un vehículo, establecimiento de las rutas eficientes en función del tráfico actual, etc.), para su final aprovechamientos social y económico.

- **Tesis/Conclusiones**

F.12 Verificación cuidadosa de las conclusiones y análisis global del resultado de la tesis desde el punto de vista científico, tecnológico, social y económico.

F.13 Publicación en artículos científicos de los avances en conferencias (*CEC*, *GECCO*, *MSWIM*, ...) y revistas (*Engineering Applications of Artificial Intelligence*, *IEEE Transactions on Vehicular Technology*, ...) de impacto.

F.14 Visitas a centros de investigación en el ámbito de la inteligencia artificial y las comunicaciones como *SnT* (Luxemburgo) y *CENTRIA* (Finlandia).
F.15 Promoción de relaciones con instituciones públicas (Ministerio de Economía y Competitividad, Consejería de Obras Públicas y Transportes de Andalucía, ayuntamientos, ...) e industria (Grupo ETRA, Telvent, *Samsung*, ...) con el fin de establecer entornos de colaboración y transferencia.
F.16 Implementación de aplicaciones VANET para su despliegue en dispositivos móviles y vehículos.
F.17 Estudio de la posibilidad de desarrollar los aportes en forma de patentes.
F.18 Diseño de un portal Web que permita el acceso al material desarrollado durante el trabajo (resultados, software desarrollado y documentación).
F.19 Divulgación global de los avances para un mayor impacto social, a través de la participación eventos y medios de comunicación no especializados.

El trabajo se realizará desde el grupo de investigación NEO (*Networking and Emerging Optimization*) perteneciente a la Universidad de Málaga. Dicho grupo dispone de varios *cluster* de nodos multi-núcleo conectados por redes Gigabit, que se emplearán en la ejecución de los algoritmos de optimización CN y la simulación de los resultados en VANETs. Se tendrá acceso a diferentes tipos de dispositivos móviles con capacidad de comunicación inalámbrica (ordenadores portátiles, *smartphones*, *tablet PCs*, etc.) para la realización de pruebas reales de concepto. La colaboración con expertos con amplia experiencia en el ámbito de la optimización, como lo son los compañeros del grupo y visitantes de otros centros, permitirá abordar los problemas que se tratan con una elevada garantía finalización satisfactoria. A su vez, la visita a otros centros de investigación ampliará los horizontes de la tesis con otros enfoques diferentes al nuestro.

Por último cabe destacar que el candidato[1] ya ha realizado trabajos relacionados con la tesis (aplicación de CN al diseño de protocolos e infraestructura en VANETs), publicando resultados en **tres revistas** indexadas *ISI* [6,7,8] y en **10 congresos** internacionales. Estos trabajos se han realizado en el contexto de **tres proyectos** (**EUREKA-CELTIC CARLINK**, **DIRICOM** y **roadMe**) en los que ha participado el candidato y bajo la posesión de una beca **FPU**.

5. Relevancia

La potencia de las técnicas metaheurísticas y de CN tratando problemas complejos ha intensificado el interés sobre las mismas desde el mundo académico y la industria, puesto que ven en ellas una herramienta útil para obtener resultados eficientes tanto a problemas reales como académicos. Así, el desarrollo de nuevas técnicas de CN y el análisis de su rendimiento marcan unas líneas de investigación importantes en la actualidad. Aplicar CN en el diseño software y hardware de VANETs hace de esta tesis un tema novedoso y de interés por la aplicabilidad final de dicho tipo de redes.

[1] Jamal Toutouh - `http://www.jamal.es`

Desde una perspectiva científica, en este trabajo de investigación se van a definir nuevos problemas de optimización para poder afrontar el diseño eficiente de VANETs, que ofrecen la posibilidad de ser tratados de forma automática e inteligente por medio de técnicas mataheurísticas y CN. Además, se van a proponer nuevos algoritmos CN que van a enriquecer el estado del arte actual.

A nivel de aplicación real del resultado de esta tesis, cabe destacar la importancia de las VANETs para ofrecer servicios de *Smart Mobility* a la ciudadanía (traslados por carretera más eficientes, seguros y cómodos), de los que también se beneficiarán la industria y la administración. El resultado de esta tesis es de vital importancia porque propone diseños eficientes tanto a nivel software como hardware de la plataforma necesaria para implantar dichas redes de comunicación. Así mismo, se van a desarrollar distintos prototipos reales de aplicaciones que ofrecerán servicios para incrementar la seguridad vial o reducir la emisión CO_2 cuya aplicación al mundo real será prácticamente inmediata.

Referencias

1. Hartenstein, H., Laberteaux, K.: VANET Vehicular Applications and Inter-Networking Technologies. Intelligent Transport Systems. John Wiley & Sons, Upper Saddle River, NJ, USA (December 2009)
2. Lee, K.C., Lee, U., Gerla, M.: Survey of routing protocols in vehicular ad hoc networks. Advances in Vehicular Ad-Hoc Networks: Developments and Challenges, IGI Global **21** (2009)
3. Wisitpongphan, N., Tonguz, O.K., Parikh, J., Mudalige, P., Bai, F., Sadekar, V.: Broadcast storm mitigation techniques in vehicular ad hoc networks. Wireless Communications, IEEE **14**(6) (2007) 84–94
4. Blum, C., Roli, A.: Metaheuristics in combinatorial optimization: Overview and conceptual comparison. ACM Comput. Surv. **35**(3) (September 2003) 268–308
5. Alba, E., Blum, C., Asasi, P., Leon, C., Gomez, J.: Optimization Techniques for Solving Complex Problems. Wiley Series on Parallel and Distributed Computing. Wiley (2009)
6. Toutouh, J., Garcia-Nieto, J., Alba, E.: Intelligent OLSR routing protocol optimization for VANETs. Vehicular Technology, IEEE Trans. on **61**(4) (2012) 1884–1894
7. Toutouh, J., Nesmachnow, S., Alba, E.: Fast energy-aware olsr routing in vanets by means of a parallel evolutionary algorithm. Cluster Computing (2012) 1–16
8. García-Nieto, J., Toutouh, J., Alba, E.: Automatic tuning of communication protocols for vehicular ad hoc networks using metaheuristics. Engineering Applications of Artificial Intelligence **23**(5) (2010) 795–805
9. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm intelligence: from natural to artificial systems. Oxford University Press, Inc., New York, NY, USA (1999)
10. Molina, G., Alba, E., Talbi, E.G.: Optimal sensor network layout using multi-objective metaheuristics. J. of Univ. Computer Science **14**(15) (2008) 2549–2565
11. Mccanne, S., Floyd, S., Fall, K.: Network simulator. http://www.nsnam.org/wiki

Author Index

Álamo, Antonio Luis, 259
Álvarez, Ada, 414, 663
Álvarez-Valdés, Ramon, 433

Acevedo-Martínez, Liesner, 624
Agüero, Jorge, 1502
Aghdam, Hamed Habibi, 8
Aguilar-Ruiz, Jesús S., 584
Agustín, Alba, 523
Al-Mamun, Mohammad Abdullah, 1578
Alba, Enrique, 338, 812, 862, 1555, 1557
Albert-Cruz, José, 624
Albisua, Iñaki, 1546
Alcalá-Fdez, Jesús, 882
Alfaro, Rocío, 443
Aliaga, Ignacio, 1489
Alonso, Israel, 1173
Alonso, Maria Teresa, 433
Alonso-Ayuso, Antonio, 523
Alonso-Isla, Borja, 1585
Alonso-López, Guillermo A., 547
Andrés-Romano, Carlos, 424
Andréws-Romano, Carlos, 408
Arbelaitz, Olatz, 19, 29, 1546, 1549
Arenas, M. G., 308, 624, 633, 782
Arnaldo, Ignacio, 318
Arques, Pilar, 39
Arroyo, Javier, 959
Asiain, María José, 1125
Ayala de Diego, Ildefonso, 989
Aznar, Fidel, 39

Baños, Raúl, 278
Bahamonde, Antonio, 1567
Bajo, Javier, 1523
Banga, Julio R., 348
Barrachina, Javier, 1591
Barranquero, Jose, 1551
Barrenechea, Edurne, 1058, 1333, 1468
Baruque, Bruno, 1489
Batalla, Cristina, 443
Bautista, Joaquín, 443, 453
Bedregal, Benjamin, 1125
Bello, Rafael, 594
Bello-García, Antonio, 1413
Benítez, José Manuel, 109
Berrocal-Plaza, Víctor, 49, 852
Blum, Christian, 902
Botella-Rocamora, Paloma, 59
Boticario, Jesus G., 1571
Brezo, Félix, 1597
Brito, Julio, 473, 483
Broz, Diego, 463
Bustince, Humberto, 1125, 1468

Cáceres-Cruz, José, 533
Cózar, Javier, 99, 802
Caballero, Ismael, 1192
Cabrerizo, Francisco J., 1182
Cadenas, José M., 1303
Campos, Vicente, 169, 773
Candelas, David, 189
Cano, Alberto, 288, 443, 1603
Carmona, C. J., 1220, 1313
Carmona, Enrique J., 703
Caro, Raquel, 1068
Carrascosa, Carlos, 1502
Castaño, Oscar, 1323
Castejón, Federico, 703
Castellanos, Ángel, 69
Castillo, Pedro A., 308, 624, 633, 782
Ceberio, Josu, 1553, 1607
Cerrón, Juan, 1028
Chávez, Francisco, 713, 832
Chamorro-Martínez, Jesús, 1038
Charte, Francisco, 1612
Chica, Manuel, 453
Chicano, Francisco, 812, 1555
Chrysostomou, C., 1313
Cigarrán, Juan M., 69
Cleger, Sergio, 79
Clemente, Eddie, 713
Colmenar, J. Manuel, 723
Concha, David, 358
Consoli, Sergio, 368
Contreras, David, 89, 1173
Contreras, Iván, 318, 792
Corberán, Ángel, 541
Corchado, Emilio, 1489
Corchado, Juan M., 1523, 1533

Cordón, Oscar, 453
Cordero, Jorge, 919
Cruz, Carlos, 693, 979
Cruz, Cayetano, 832
Cruz-Ramírez, Manuel, 1343
Cubillo, Susana, 1250
Cuesta-Infante, Alfredo, 723

Díez, Jorge, 1551
Díez-Pastor, José F., 1433
Damas, Sergio, 453
De Armas, Jesica, 473
de la Peña, Jorge, 1618
De la Prieta, Fernando, 1533, 1624
de Las Cuevas, Paloma, 633
De Miguel, Laura, 1125
De Paz, Juan F., 1523
De Santiago, Alondra, 663
Del Castillo, Ester, 1565
del Coz, Juan José, 1551
Del Jesus, M. Jose, 1220, 1313, 1423
del Río, Sara, 109
delaOssa, Luis, 99, 802
Demazeau, Yves, 1533
Diaz-Parra, Ocotlán, 547
Diego, Francisco J., 424, 792
Divina, Federico, 584
Domínguez-Narvaez, Joel A., 547
Dominguez, Antonio, 929
Dozal, León, 713
Duarte, Abraham, 358, 378, 388, 557, 842
Durillo, Juan J., 239

Egea, Jose A., 348
Eguía, Ignacio, 398
Elizondo-Amaya, Mónica, 179
Espín-Estrella, Antonio, 249
Espinilla, M., 1230
Espinosa, Silvia, 1413
Expósito, Airam, 483
Expósito-Izquierdo, Christopher, 493

Fahringer, Thomas, 239
Fajardo, Jenny, 673
Faulin, Javier, 523
Fernández Díaz, Álvaro, 1636
Fernández de Vega, Francisco, 219, 713, 832
Fernández, Alberto, 1333, 1468
Fernández, Alfonso, 189
Fernández, Carlos, 969
Fernández, Damián, 1630
Fernández-Amorós, David, 1182
Fernández-López, David, 358, 378
Fernández-Luna, Juan M., 79
Fernandes, Carlos, 209, 633
Fernandez, Javier, 1125
Ferrer, Javier, 812
Ferri, Cèsar, 119, 1383
Figueroa-García, Juan C., 1642
Frías, Pilar, 1423
Franco, Camilo, 1240
Frutos, Mariano, 259, 463
Fuentes-Fernández, Rubén, 929
Fuentes-Penna, Alejandro, 547

Gámez, José A., 99, 802
Gómez, Daniel, 1115
Gómez, José Rubén, 513
Gómez-Lorente, Daniel, 249
Gómez-Pulido, Juan A., 683
Gómez-Sebastià, Ignasi, 1653
Galán-García, Patxi, 1648
Galar, Mikel, 1028, 1058, 1333, 1468
Gallego, Micael, 773
Galván, Blas, 259
García Sánchez, Pablo, 782
García, Irma, 414
García, Salvador, 149
García, Tatiana, 763
García-Caro, J. Gabriel, 328
García-Fernández, Pedro, 782
García-Gutiérrez, Jorge, 1363
García-Herrero, Jesús, 1513
García-Martínez, Carlos, 199, 902
García-Nieto, José, 812, 862, 1557
García-Osorio, César, 1433
García-Pedrajas, Nicolás, 604
García-Sánchez, Alvaro, 424
García-Sánchez, Pablo, 633
García-Saiz, Diego, 1573
García-Serrano, Ana, 69
García-Valdez, Mario, 219
Garmendia, Luis, 1068, 1077, 1134
Garnica, Oscar, 733
Garrido, M. Carmen, 1303
Gervás, Pablo, 229
Gibert, Karina, 1454
Gil, Consolación, 249, 278
Gil-Cumbreras, Fco. Javier, 584

Giráldez, Raúl, 584, 1373
Gomez-Pulido, Juan A., 743
González del Campo, Ramón, 1068, 1077, 1134
González, Beatriz, 919
González, Jesús, 633
González, Pablo, 1551
González, Pedro, 1220
González-Calabozo, José María, 1659
Gonzalo-Orden, Hernán, 513
Gramajo, S., 1230
Griol, David, 1513
Gromicho, Joaquim, 433
Guerrero, Fernando, 398
Guevara, César, 939
Guijarro-Berdiñas, Bertha, 129
Gurrutxaga, Ibai, 19, 29, 1546, 1549
Gutiérrez, Pedro Antonio, 872, 1343
Gutierrez, Miguel, 1009
Gutierrez, Patricia, 1559

Hatami, Sara, 408
Heradio, Rubén, 1182
Hernández, Pablo, 1250
Hernández-González, Jerónimo, 1561, 1665
Hernández-Orallo, José, 119, 1383
Hernando, Beatriz, 1489
Hernando, Leticia, 822, 1563
Herrera, Francisco, 109, 149, 567, 594, 882, 1105, 1333, 1393, 1468
Herrera, Manuel, 1182
Herrera-Viedma, Enrique, 1182, 1286
Hervás-Martínez, César, 872, 1343
Hidalgo, J. Ignacio, 318, 328, 723, 733, 792
Hougaard, Jens Leth, 1240

Iglesias, José Antonio, 1479
Infante, Javier, 19
Inza, Iñaki, 1561
Irurozki, Ekhine, 1553

Jaén, Juan Ignacio, 643
Jaraíz-Simon, María D., 683
Jiménez, M., 1353
Jimenez-Linares, Luis, 1565
Juan, Angel, 523, 533, 577
Julián, Vicente, 1502
Jurio, Aranzazu, 1058, 1125

Krasnogor, Natalio, 318
Kutanoglu, Erhan, 179

López, Natalia, 298
López, Victoria, 109, 919, 929, 969, 1009
Laguna, Manuel, 842
Lalla-Ruiz, Eduardo, 503
Lamata, María T., 1095
Lanchares, Juan, 733
Lanza-Gutierrez, Jose M., 743, 1671
Lasarguren, Aritz, 1546
Ledezma, Agapito, 1479
Lojo, Aizea, 29
Lopez-Molina, Carlos, 1028
Lozano, Jose A., 822, 1403, 1553, 1561, 1563
Lozano, Manuel, 199, 902
Luaces, Oscar, 1567
Luna, J. M., 753, 1353
Luque, Gabriel, 338
Luque, Mariano, 268

Méndez Babey, Máximo, 463
Méndez, Máximo, 259
Mandow, Lawrence, 139
Martí, José Vicente, 763
Martí, Rafael, 169, 541, 773, 842
Martín H, José Antonio, 939
Martín, Bernardo, 557
Martín, Diana, 882
Martín, Javier, 1144
Martín-Villanueva, Sara, 1373
Martínez, Alejandro, 1303
Martínez, Jorge, 919, 949
Martínez, Luis, 1260
Martínez, Raquel, 1303
Martínez-Álvarez, Francisco, 1323
Martínez-Gavara, Anna, 773
Martínez-Jiménez, Pedro, 1038
Martínez-Plumed, Fernando, 119, 1383
Mas, M., 1153
Masegosa, Antonio D., 673, 1095
Massanet, S., 1163
Mateos-García, Daniel, 1363
Mattioli, Gabriel, 1048
Maudes, Jesús, 1433
Mayor, Gaspar, 1144
Meco, Albert, 959
Melendez, Jaime, 8
Melián-Batista, Belén, 473, 493, 503, 663

Mendiburu, Alexander, 822, 1403, 1553, 1563
Menoyo, Joseba, 999
Merelo, Juan Julián, 209, 219, 308, 624, 633, 782
Meseguer, Pedro, 1559
Millán-Ruiz, David, 318
Mladenovic, Nenad, 368, 388
Molina, Daniel, 594
Molina, José Carlos, 398
Molina, José Manuel, 1513
Monjas, Juan Íñigo, 929
Monserrat, M., 1153
Montemayor, Antonio S., 358
Montero, Javier, 1085, 1115
Montes, Rosana, 1105
Mora, Antonio M., 209, 633
Morata, Miguel, 929
Moreno-Garcia, Juan, 1565
Moreno-Pérez, José A., 368, 473, 483
Moreno-Torres, Jose G., 1393
Moreno-Vega, J. Marcos, 493, 503
Mori, Usue , 1403
Morton, David P., 179
Muguerza, Javier, 19, 29, 1546, 1549

Núñez-Letamendia, Laura, 792
Naranjo, Rodrigo, 959
Navarro, Lilian, 832
Nielsen, Kurt, 1240
Nobel, Javier, 318
Novoa-Hernández, Pavel, 693

Octavio de Toledo, Mar, 969
Oketch, Christine A., 577
Olague, Gustavo, 219, 713
Olivas, Jose A., 1173, 1192, 1200, 1210
Olivera, Ana C., 1557
Olmo, Juan Luis, 288, 1444
Ordieres-Meré, Joaquín, 1413
Ortega, Julio, 278
Ortiz, Victor, 1009
Ourezabal, Mauro J., 1192

Pástor-Ramírez, Danilo, 1695
Pérez, Elena, 567
Pérez, Jesús M., 19, 29, 1546, 1549
Pérez-Bernabé, Inmaculada, 1701
Pérez-de-la-Cruz, José Luis, 139
Pérez-Godoy, M. Dolores, 1423
Pérez-Ortiz, María, 872, 1343
Pérez-Recuerda, Pedro, 1423
Pérez-Rodríguez, Javier, 604
Pacheco, Joaquín, 414, 513
Padrón, Silvia, 533
Pagola, Miguel, 1028
Palacios, Juan José, 1677
Palacios-Alonso, Daniel, 1683
Palomares, Iván, 1260, 1689
Pantrigo, Juan José, 358, 388
Pardines, Inmaculada, 969
Pardo, Eduardo G., 388
Pardo, Juan, 59
Parejo, Jose Antonio, 653
Parras, Manuel, 1423
Parras-Gutierrez, Elisabeth, 782
Parreño, Francisco, 433
Paternain, Daniel, 1058
Peñalver, Alejandro, 1009
Peiró, Juanjo, 541
Pelta, David A., 673, 693
Perona, Iñigo, 29, 1549
Peteiro-Barral, Diego, 129, 1706
Pontes, Beatriz, 1373
Porras, Santiago, 1489
Posada, Marta, 567
Post, Gerhard, 433
Pragnell, Ricardo, 1009
Puentes, Jesus M., 577
Puig, Anna, 89
Puig, Domenec, 8
Pulido, Raúl, 424
Puris, Amilkar, 594

Quesada, Francisco J., 1260
Quevedo, José Ramón , 1567
Quintian, Hector, 1489, 1712

Ríos-Mercado, Roger Z., 179
Rabanal, Pablo, 298
Racero, Jesús, 398
Radulovic, Filip, 1718
Ramírez, Aurora, 892
Ramírez-Quintana, María José, 119, 1383
Rebollo, Miguel, 1502
Rendón, Rafael, 1018
Riera, Daniel, 533
Riquelme, José C., 1323, 1363
Risco-Martín, José L., 723, 733
Rivas, Victor Manuel, 782

Rivera, Antonio J., 1423
Rodríguez, Francisco J., 902
Rodríguez, Inmaculada, 89
Rodríguez, Ismael, 298
Rodríguez, J. Tinguaro, 1085
Rodríguez, Juan J., 1433
Rodríguez, Rosa M., 1569
Rodríguez, Sara, 1533
Rodríguez-Ávila, Humberto, 79
Rodríguez-Cano, Julio C., 79
Rodriguez, Francisco J., 199
Rodriguez-Benitez, Luis, 1565
Romero, Cristóbal, 753, 1444
Romero, Francisco P., 1173, 1192, 1210
Romero, Gustavo, 633
Romero, José Raúl, 643, 753, 892
Romeu, Pablo, 59
Rosa, Agostinho C., 209
Rosete, Alejandro, 673, 882, 979
Rubio, Fernando, 298
Ruiz, Ana Belén, 268
Ruiz-Aguilera, D., 1153
Ruiz-Cortés, Antonio, 653
Ruiz-Montiel, Manuela, 139
Ruiz-Vanoye, Jorge A., 547

Sáez, José A., 1393
Sánchez, Ángel, 557
Sánchez, Ana María, 1105
Sánchez, Jesús A., 1729
Sánchez, Oscar, 919
Sánchez-Monedero, Javier, 872
Sánchez-Oro, Jesús, 358, 378, 842
Sánchez-Pérez, Juan M., 49, 852
Sànchez-Marrè, Miquel, 1454
Saborido, Rubén, 268
Salamó, Maria, 89
Salmeron-Majadas, Sergio, 1724
Sanchis, Araceli, 1479
Sanmartin, Antonio, 969
Santander-Jiménez, Sergio, 614
Santos, Matilde, 939, 949, 959, 989, 999, 1018, 1270
Santos, Olga C., 1571
Sanz, José, 1028
Segura, Sergio, 653
Seker, H., 1313
Sempere, Mireia, 39
Serrano-Guerrero, Jesús, 1192
Sevilla Villanueva, Beatriz, 1454
Sierra, J. Enrique, 1270
Simón-Cuevas, Alfredo, 1200
Solanas, Agusti, 8
Soto-Hidalgo, J.M., 1038
Stolfi, Daniel H., 1734
Suárez-Rodríguez, Amhed, 1200

Tohmé, Fernando, 463
Torra, Vicenç, 1280
Torrens, J., 1153, 1163
Torres, Carmen, 1250
Torres, Isis, 979
Toutouh, Jamal, 1740
Triguero, Isaac, 149, 249
Troncoso, Alicia, 1323
Trujillo, Leonardo, 219

Ureña, Raquel, 1286

Vázquez, Mª Rosario, 1210
Vega-Rodríguez, Miguel A., 49, 614, 683, 743, 852
Ventura, Sebastián, 288, 643, 753, 892, 1353, 1444
Vera, Vicente, 1489
Verdegay, José Luis, 979
Villacorta, Pablo J., 1095
Villanueva, Rafael J., 328
Villanueva-Oller, Javier, 328
Villar, Pedro, 1105
Villarrubia, Gabriel, 1523
Villaverde, Alejandro F., 348
Vitoriano, Begoña, 1085

Whitley, L. Darrell, 1555

Yáñez, Javier, 1115
Yager, Ronald R., 1077, 1134
Yepes, Víctor, 763

Zamora-Martínez, Francisco, 59
Zarrazola, Edwin, 1115
Zato, Carolina, 1523, 1533
Zorrilla, Marta, 1573