Advances in Genetic Algorithms and Natural Language Processing

Research in Computing Science

Series Editorial Board

Research in Computing Science es una publicación trimestral, de circulación internacional, editada por el Centro de Investigación en Computación del IPN, para dar a conocer los avances de investigación científica y desarrollo tecnológico de la comunidad científica internacional. **Volumen 134,** noviembre 2017. Tiraje: 500 ejemplares. *Certificado de Reserva de Derechos al Uso Exclusivo del Título* No.: 04-2005-121611550100-102, expedido por el Instituto Nacional de Derecho de Autor. *Certificado de Licitud de Título* No. 12897, *Certificado de licitud de Contenido* No. 10470, expedidos por la Comisión Calificadora de Publicaciones y Revistas Ilustradas. El contenido de los artículos es responsabilidad exclusiva de sus respectivos autores. Queda prohibida la reproducción total o parcial, por cualquier medio, sin el permiso expreso del editor, excepto para uso personal o de estudio haciendo cita explícita en la primera página de cada documento. Impreso en la Ciudad de México, en los Talleres Gráficos del IPN – Dirección de Publicaciones, Tres Guerras 27, Centro Histórico, México, D.F. Distribuida por el Centro de Investigación en Computación, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, México, D.F. Tel. 57 29 60 00, ext. 56571.

Editor responsable: *Grigori Sidorov, RFC SIGR651028L69*

Research in Computing Science is published by the Center for Computing Research of IPN. **Volume 134**, November 2017. Printing 500. Printed in Mexico City, in the IPN Graphic Workshop – Publication Office.

Advances in Genetic Algorithms and Natural Language Processing

María de Lourdes Martínez Villaseñor (ed.)

Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2017

ISSN: 1870-4069

Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal
Unidad Profesional "Adolfo López Mateos", Zacatenco
07738, México D.F., México

http://www.rcs.cic.ipn.mx
http://www.ipn.mx
http://www.cic.ipn.mx

Indexed in LATINDEX, DBLP and Periodica

Printing: 500

Printed in Mexico

ISSN 1870-4069

Editorial

Este volumen de la revista *Research in Computing Science* contiene artículos seleccionados relacionados con algoritmos evolutivos para la optimización con restricciones y artículos que utilizan técnicas de procesamiento natural para el razonamiento automático. Se seleccionaron cuidadosamente 15 artículos que fueron revisados por lo menos por dos miembros del comité revisor tomando en cuenta la originalidad, contribución científica en el campo de la Inteligencia Artificial y la calidad cada artículo.

En lo que se refiere a la optimización evolutiva con métodos inspirados en la naturaleza, este volumen incluye seis artículos de aplicación de algoritmos de optimización evolutiva en diferentes problemas como el problema del empacado y bin embalaje, creación de modelo innovador para aparadores comerciales, clasificación de arritmias cardiacas, optimización de rutas de transporte y selección de líderes en redes inalámbricas de sensores

De igual manera en este volumen se pueden encontrar cuatro artículos de análisis y modificación de algoritmos evolutivos para optimización en los que se presentan: a) mejoras a la búsqueda armónica para la optimización numérica con restricciones b) mejoras en la evolución diferencial para la optimización de mecanismos cuatro barras, c) análisis de medidas de diversidad para la optimización evolutiva y d) una propuesta de un nuevo operador de cruce basado en el operador de cruce simulado.

La minería de opiniones y análisis de sentimientos mediante técnicas de procesamiento de lenguaje natural y minería de textos ha cobrado gran relevancia en los últimos años dada su versatilidad de uso en muy diversas aplicaciones.

En este volumen, se podrán encontrar seis artículos que utilizan técnicas de procesamiento natural para el razonamiento automático. Dos artículos realizan minería de opiniones en particular para la detección de engaño y decidir la polaridad en un tópico. La minería de textos se utiliza en un artículo para abordar el tema de registros inespecíficos de diagnóstico en una sala de urgencias. En el cuarto artículo se aplican técnicas de representación distribucional para la generación automática de resúmenes en múltiples documentos. El quinto artículo presenta la generación automática de recursos léxicos a la medida del dominio. El sexto discute la minería de datos en redes sociales.

Agradezco el trabajo realizado por los miembros de la Sociedad Mexicana de Inteligencia Artificial en el soporte para la preparación de este volumen.

Los procesos de envío, revisión y selección de artículos así como la preparación de memorias del congreso fueron realizados de manera gratuita por el sistema EasyChair (www.easychair.org).

María de Lourdes Martínez Villaseñor
Editora invitada
noviembre de 2017

ISSN 1870-4069

Table of Contents

Page

ISSN 1870-4069

Evolución diferencial con memoria de parámetros para la optimización de mecanismos de cuatro barras

Maury-Faney Zapata-Zapata[1], Efrén Mezura-Montes[2], Edgar-Alfredo Portilla-Flores[3]

[1] Laboratorio Nacional de Informática Avanzada (LANIA A. C.), Xalapa, Veracruz, México

[2] Universidad Veracruzana, Xalapa, Veracruz, México

[3] Instituto Politécnico Nacional, CIDETEC, Ciudad de México, México

mzapata.mca15@lania.edu.mx, emezura@uv.mx, aportilla@ipn.mx

Resumen. L-SHADE es un algoritmo basado en evolución diferencial que adapta automáticamente los parámetros relacionados con la mutación (F) y la cruza (CR) a través de una memoria de parámetros, y además controla el tamaño de la población (NP) usando una función lineal. En este trabajo se incorpora una estrategia de manejo de restricciones a L-SHADE con el fin de optimizar tres casos de estudio de un mecanismo de cuatro barras y se llama C-LSHADE a este nuevo enfoque. En la experimentación, se compara el desempeño de C-LSHADE contra un algoritmo DE/rand/1/bin del estado del arte que utiliza un enfoque de parámetros aleatorios para resolver problemas mecánicos de optimización. Los resultados obtenidos demuestran que C-LSHADE es capaz de llegar a mejores o iguales soluciones que DE/rand/1/bin, empleando un porcentaje considerablemente menor de evaluaciones de la función objetivo y liberando al usuario de la tarea de ajustar los parámetros F, CR y NP.

Palabras clave: Optimización, evolución diferencial, control de parámetros, mecanismo de cuatro barras, L-SHADE.

Differential Evolutión with Parameter Memory to Optimize Four-Bar Mechanisms

Abstract. L-SHADE is an algorithm based on differential evolution wich automatically adapts the parameters related to mutation (F) and crossover (CR) using a parameter memory. L-SHADE also controls the population size (NP) with a linear function. In this work, a constraint-handling technique is added to L-SHADE to optimize three four-bar

mechanisms and this new approach is called C-LSHADE. During the experimentation, the performance of C-LSHADE is compared against a state-of-the-art DE/rand/1/bin algorithm that uses a random parameter approach to solve mechanical optimization problems. The obtained results suggest that C-LSHADE is able to find better or equal solutions than DE/rand/1/bin, using a considerably lower percentage of evaluations of the objective function while keeping the user from adjusting the parameters F, CR y NP.

Keywords: Optimization, differential evolution, parameter control, four-bar mechanism, L-SHADE.

1. Introducción

Los algoritmos evolutivos (AEs) se han convertido en una técnica popular para resolver problemas complejos de optimización debido a su facilidad de uso y robustez. El éxito de dichas técnicas puede ser encontrado en la literatura especializada [3,13]. No obstante, a pesar del auge que han tenido, uno de los problemas importantes que influye en la obtención de buenos resultados cuando se aplica un AE, es la configuración de sus parámetros [4].

De acuerdo con [4], existen dos formas de abordar la problemática mencionada: (1) el ajuste de parámetros y (2) el control de parámetros. El ajuste de parámetros es la estrategia que se practica comúnmente, que consiste en hallar buenos valores para los parámetros antes de ejecutar el algoritmo y después correr el algoritmo con los valores seleccionados, los cuales se mantienen constantes durante todo el proceso. Por otra parte, en el control de parámetros, los valores se modifican durante la ejecución del algoritmo de tal forma que se promueva una búsqueda exitosa de la solución.

Ambas formas de configuración son válidas, sin embargo, el ajuste de parámetros puede ser inapropiado debido a los siguientes inconvenientes [4]:

- Los parámetros no son independientes, lo que hace prácticamente imposible probar todas las combinaciones diferentes.
- El proceso de ajustar los parámetros requiere mucho tiempo.
- Para un problema dado, los valores de parámetros seleccionados no necesariamente son los óptimos, incluso si el esfuerzo realizado en la configuración fue significativo.

Como solución a los inconvenientes mencionados, se pueden aplicar mecanismos de control que sean lo suficientemente útiles para realizar un buen proceso de configuración de parámetros y liberar a los usuarios de dicha tarea. Sin embargo, desarrollar una técnica de control que adapte los parámetros de forma automática implica un proceso de diseño y por tal motivo, la investigación realizada en este tema se encuentra en su etapa inicial [4]. Es por esto que los investigadores y usuarios de los AEs, aún prefieren realizar un ajuste de parámetros con base en una experimentación previa.

De acuerdo a [5], los estudios en control de parámetros no han sido enfocados a dominios específicos como lo es el de problemas de diseño mecánico. Como ejemplos de este dominio en particular se pueden mencionar los trabajos presentados en [12] y [16]. En ambos casos se realiza un ajuste de parámetros; en el primero, no se explica claramente cómo se llegó a los valores seleccionados y en el segundo se toman valores recomendados en la literatura. También se han propuesto enfoques como los presentados en [1] y [7], en donde los valores de los parámetros relacionados con la cruza y la mutación son generados aleatoriamente en cada ciclo del algoritmo, que si bien no se establecen valores constantes, tampoco se tiene un control sobre el progreso de la búsqueda. Finalmente, en [5], se propone un algoritmo que incorpora una técnica de control con comportamiento senoidal, pero ésta sólo es aplicada al parámetro relacionado con la mutación.

Derivado de lo anterior, surge la motivación de este trabajo, en donde L-SHADE, un algoritmo que incorpora mecanismos de control en los parámetros relacionados con la mutación, cruza y población, y que originalmente fue diseñado para resolver problemas de optimización sin restricciones, es adaptado para solucionar problemas restringidos de diseño mecánico. El objetivo de esta investigación, es promover el uso de dicho algoritmo en la solución de problemas relacionados con mecanismos de cuatro barras, evitando así, que se realice un ajuste manual de parámetros que posiblemente no sea el más adecuado.

El resto del artículo se encuentra organizado de la siguiente manera: en la Sección 2 se describen los problemas de optimización correspondientes a tres casos de estudio de un mecanismo de cuatro barras. Posteriormente, en la Sección 3, se muestra el algoritmo basado en evolución diferencial empleado para estre trabajo, enfatizando las modificaciones realizadas a la versión original para resolver problemas con restricciones. En la Sección 4, se presentan y discuten los resultados obtenidos y en la Sección 5 se exponen las conclusiones derivadas de esta investigación.

2. Declaración de problemas de optimización

En la Figura 1 se muestra un mecanismo de cuatro barras conformado por: barra de referencia ($\bar{r_1}$), barra de entrada ($\bar{r_2}$), acoplador ($\bar{r_3}$) y barra de salida o balancín ($\bar{r_4}$). Para efectos del análisis, se utilizan dos sistemas de coordenadas; el primero, se encuentra fijo al mundo real (OXY) y el segundo es para referencia ($O_2X_rY_r$), en donde (x_0, y_0), es la distancia entre los puntos de origen de ambos sistemas, θ_0 es el ángulo de rotación del segundo sistema con respecto del primero, θ_i $(i = 1, 2, 3, 4)$ denota los ángulos de las cuatro barras, y el par de coordenadas (r_{cx}, r_{cy}) definen la posición del acoplador C [2].

La síntesis del mecanismo de cuatro barras se lleva a cabo para calcular la longitud de sus barras, el ángulo de rotación con respecto al sistema OXY, la distancia entre los dos sistemas de coordenadas y el conjunto de ángulos para la barra de entrada que permitirán generar la trayectoria deseada.

El problema de optimización consiste en minimizar la distancia entre el punto de contacto y el punto de precisión predefinido. La función propuesta para dicho

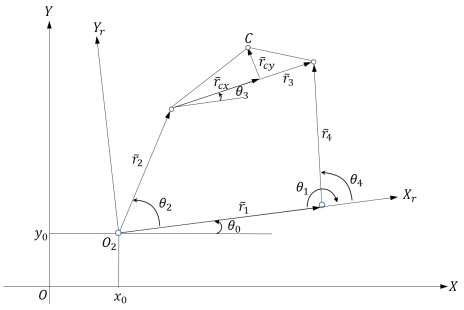


Fig. 1: Mecanismo de cuatro barras. Adaptada de [2].

cálculo es:

$$f(\theta_2^i) = \sum_{i=1}^{N}[(C_{xd}^i - C_x^i)^2 - (C_{yd}^i - C_y^i)^2], \tag{1}$$

en donde C_x^i y C_y^i son las coordenadas del punto C, mientras que C_{xd}^i y C_{yd}^i son las coordenadas del punto de precisión que define la trayectoria deseada. Más detalles del análisis del mecanismo mostrado, pueden ser encontrados en [2,6].

En este trabajo se obtuvo la síntesis dimensional de tres casos de estudio distintos, en donde el punto C del mecanismo debe seguir una trayectoria de puntos cartesianos previamente definidos. A continuación, se describe cada caso de estudio.

2.1. Caso de estudio 1: seguimiento de una trayectoria lineal vertical, sin sincronización previa - CE1

En el primer caso de estudio, tomado de [6], el acoplador debe pasar por los puntos de precisión $\Omega = \{(20,20),(20,25),(20,30),(20,35),(20,40),(20,45)\}$, los cuales se encuentran alineados verticalmente. El vector de variables de diseño es: $\boldsymbol{p} = [r_1, r_2, r_3, r_4, r_{cx}, r_{cy}, \theta_0, x_0, y_0, \theta_2^1, \theta_2^2, \theta_2^3, \theta_2^4, \theta_2^5, \theta_2^6]^T$, en donde las primeras cuatro variables corresponden a las barras del mecanismo, las dos siguientes definen la posición del acoplador, θ_0, x_0 y y_0 indican la orientación del sistema $O_2X_rY_r$ con respecto al sistema OXY, y las variables restantes corresponden a la secuencia de ángulos de la barra de entrada ($\bar{r}_2$).

Los límites superior e inferior de las variables de diseño se encuentran definidos por: $r_1, r_2, r_3, r_4 \in [0, 60]$, $r_{cx}, r_{cy}, x_0, y_0 \in [-60, 60]$ y $\theta_0, \theta_2^1, \theta_2^2, \theta_2^3, \theta_2^4, \theta_2^5, \theta_2^6 \in [0, 2\pi]$

Se considera el problema de optimización mono-objetivo descrito de la Ecuación 2 a la 11:

$$min f(\boldsymbol{p}) = \sum_{i=1}^{N}[(C_{xd}^i - C_x^i)^2 - (C_{yd}^i - C_y^i)^2], \qquad \boldsymbol{p} \in \mathbb{R}^{15}. \tag{2}$$

Sujeto a:

$$g_1(\boldsymbol{p}) = r_1 + r_2 - r_3 - r_4 \leq 0, \tag{3}$$
$$g_2(\boldsymbol{p}) = r_2 - r_3 \leq 0, \tag{4}$$
$$g_3(\boldsymbol{p}) = r_3 - r_4 \leq 0, \tag{5}$$
$$g_4(\boldsymbol{p}) = r_4 - r_1 \leq 0, \tag{6}$$
$$g_5(\boldsymbol{p}) = \theta_2^1 - \theta_2^2 \leq 0, \tag{7}$$
$$g_6(\boldsymbol{p}) = \theta_2^2 - \theta_2^3 \leq 0, \tag{8}$$
$$g_7(\boldsymbol{p}) = \theta_2^3 - \theta_2^4 \leq 0, \tag{9}$$
$$g_8(\boldsymbol{p}) = \theta_2^4 - \theta_2^5 \leq 0, \tag{10}$$
$$g_9(\boldsymbol{p}) = \theta_2^5 - \theta_2^6 \leq 0. \tag{11}$$

2.2. Caso de estudio 2: seguimiento de una trayectoria no alineada, con sincronización previa - CE2

En este caso de estudio se realiza una sincronización prescrita y la trayectoria es no alineada, la cual consta de los puntos: $\Omega = \{(3,3),(2.759,3.363),(2.372,3.663),(1.890,3.862),(1,355,3.943)\}$. El vector de variables de diseño es: $\boldsymbol{p} = [r_1, r_2, r_3, r_4, r_{cx}, r_{cy}]^T$. Las variables x_0, y_0, $\theta_0 = 0$, y la sincronización prescrita está dada por la siguiente secuencia de ángulos:

$$\theta_2^i = \left\{ \frac{2\pi}{12}, \frac{3\pi}{12}, \frac{4\pi}{12}, \frac{5\pi}{12}, \frac{6\pi}{12} \right\}. \tag{12}$$

Los límites superior e inferior de las variables de diseño se encuentran definidos por: $r_1, r_2, r_3, r_4 \in [0, 50]$ y $r_{cx}, r_{cy} \in [-50, 50]$.

En este caso, el problema de optimización mono-objetivo se describe de la Ecuación 13 a la 17:

$$minf(\boldsymbol{p}) = \sum_{i=1}^{N} [(C_{xd}^i - C_x^i)^2 - (C_{yd}^i - C_y^i)^2], \qquad \boldsymbol{p} \in \mathbb{R}^6, \tag{13}$$

sujeto a:

$$g_1(\boldsymbol{p}) = r_1 + r_2 - r_3 - r_4 \leq 0, \tag{14}$$
$$g_2(\boldsymbol{p}) = r_2 - r_3 \leq 0, \tag{15}$$
$$g_3(\boldsymbol{p}) = r_3 - r_4 \leq 0, \tag{16}$$
$$g_4(\boldsymbol{p}) = r_4 - r_1 \leq 0. \tag{17}$$

2.3. Caso de estudio 3: generación de movimiento delimitado por un conjunto de pares de puntos - CE3

Finalmente se tiene el tercer caso de estudio, tomado de [2], en donde se consideran diez pares de puntos de precisión ($K = 10$). El vector de variables

de diseño es: $\boldsymbol{p} = [r_1, r_2, r_3, r_4, r_{cx}, r_{cy}, \theta_0, x_0, y_0, \theta_2^1, ..., \theta_2^{10}]^T$, con los límites superior e inferior definidos como: $r_1, r_2, r_3, r_4 \in [0, 60]$, $r_{cx}, r_{cy}, x_0, y_0 \in [-60, 60]$ y $\theta_0, \theta_2^1, ..., \theta_2^{10} \in [0, 2\pi]$.

La función de este caso es diferente a la de los problemas anteriores, ya que considera la distancia del acoplador a los dos puntos de precisión de cada par de coordenadas. Los pares de puntos pueden ser encontrados en [2] y el problema de optimización mono-objetivo se describe de la Ecuación 18 a la 31:

$$min f(\boldsymbol{p}) = \sum_{i=1}^{K}[(C_{1xd}^i - C_x^i)^2 - (C_{1yd}^i - C_y^i)^2] + \sum_{i=1}^{K}[(C_{2xd}^i - C_x^i)^2 - (C_{2yd}^i - C_y^i)^2],$$
$$\boldsymbol{p} \in \mathbb{R}^{19}, \tag{18}$$

Sujeto a:

$$g_1(\boldsymbol{p}) = r_1 + r_2 - r_3 - r_4 \leq 0, \tag{19}$$
$$g_2(\boldsymbol{p}) = r_2 - r_3 \leq 0, \tag{20}$$
$$g_3(\boldsymbol{p}) = r_3 - r_4 \leq 0, \tag{21}$$
$$g_4(\boldsymbol{p}) = r_4 - r_1 \leq 0, \tag{22}$$
$$g_5(\boldsymbol{p}) = \theta_2^1 - \theta_2^2 \leq 0, \tag{23}$$
$$g_6(\boldsymbol{p}) = \theta_2^2 - \theta_2^3 \leq 0, \tag{24}$$
$$g_7(\boldsymbol{p}) = \theta_2^3 - \theta_2^4 \leq 0, \tag{25}$$
$$g_8(\boldsymbol{p}) = \theta_2^4 - \theta_2^5 \leq 0, \tag{26}$$
$$g_9(\boldsymbol{p}) = \theta_2^5 - \theta_2^6 \leq 0, \tag{27}$$
$$g_{10}(\boldsymbol{p}) = \theta_2^6 - \theta_2^7 \leq 0, \tag{28}$$
$$g_{11}(\boldsymbol{p}) = \theta_2^7 - \theta_2^8 \leq 0, \tag{29}$$
$$g_{12}(\boldsymbol{p}) = \theta_2^8 - \theta_2^9 \leq 0, \tag{30}$$
$$g_{13}(\boldsymbol{p}) = \theta_2^9 - \theta_2^{10} \leq 0. \tag{31}$$

3. Enfoque propuesto

3.1. Algoritmo L-SHADE

El algoritmo Evolución Diferencial (DE por sus siglas en inglés) fue desarrollado por Price y Storn en 1995 para ser un optimizador de funciones confiable, versátil y fácil de usar. Desde entonces, DE ha sido probado en competencias como el ICEO (*International Contest on Evolutionary Optimization*) de la IEEE en 1996 y 1997, y en el mundo real en una amplia variedad de aplicaciones [11].

SHADE (Success-History Based DE) [14] es un algoritmo basado en Evolución Diferencial, que adapta los parámetros CR y F utilizando un historial de parámetros exitosos y surge como una versión extendida de JADE [9]. SHADE

obtuvo el cuarto lugar en la Sesión Especial y Competencia de Optimización Mono-Objetivo en Parámetros Reales CEC-2013.

L-SHADE [15] es una versión mejorada de SHADE, ganador de la misma competencia en el año 2014, que incorpora un mecanismo de reducción lineal del tamaño de población (LPSR por sus siglas en inglés) en el cual, de acuerdo a una función lineal, el tamaño de la población disminuye con el paso de las generaciones. En el resto de esta sección se describen las novedades presentadas en L-SHADE.

Asignación de parámetros basada en una memoria histórica

Como se puede observar en la Tabla 1, L-SHADE cuenta con una memoria de H entradas para los parámetros F y CR, denotadas como M_{CR} y M_F respectivamente. El contenido de $M_{CR.k}$ y $M_{F,k}$ $(k = 1, ..., H)$ se inicializa en 0.5. En cada generación g los valores F_i y CR_i empleados por cada individuo $x_{i,g}$, son generados seleccionando un índice o posición r_i de $[1, H]$ y aplicando las siguientes ecuaciones [15]:

$$CR_i = \begin{cases} 0 & si \;\; M_{CR,r_i} = \perp, \\ randn_i(M_{CR,r_i}, 0.1) & en\ caso\ contrario, \end{cases} \tag{32}$$

$$F_i = randc_i(M_{F,r_i}, 0.1). \tag{33}$$

Las expresiones $randn_i$ y $randc_i$ son distribuciones normal y de cauchy con medias M_{CR,r_i}, M_{F,r_i} y varianza de 0.1. En caso de que el valor de CR_i se encuentre fuera de los limites [0,1], será reemplazado por el límite más cercano al valor obtenido. Si el valor $F_i > 1$, F_i se trunca a 1 y cuando $F_i < 0$, se aplica nuevamente la Ecuación 33 hasta obtener un valor aceptable. La forma en que se generan los parámetros CR_i y F_i fue adoptada del algoritmo JADE. En la Ecuación 32, el símbolo $\perp$ hace referencia a un "valor terminal", si M_{CR_i} tiene dicho valor asignado, entonces $CR_i = 0$.

Tabla 1: Memoria Histórica M_{CR}, M_F. Adaptada de [15].

Índice	1	2	...	$H-1$	H
M_{CR}	$M_{CR,1}$	$M_{CR,2}$	...	$M_{CR,H-1}$	$M_{CR,H}$
M_F	$M_{F,1}$	$M_{F,2}$	...	$M_{F,H-1}$	$M_{F,H}$

Mutación de vectores utilizando current-to-*p*best/1/bin

L-SHADE utiliza la estrategia de mutación current-to-$p$best/1 (Ecuación 34) propuesta originalmente en [9], la cual es una variante de current-to-best/1, que incorpora un porcentaje p encargado de controlar qué tan rápido converge el algoritmo.

- current-to-pbest/1

$$v_{i,g} = x_{i,g} + F(x^p_{best} - x_{i,g}) + F(x_{r1,g} - x_{r2,g}), \tag{34}$$

en donde x_{best}^{p} es seleccionado aleatoriamente de entre los $p \times NP$ mejores individuos de la población. De acuerdo con los autores de dicha estrategia de mutación, el rango de valores recomendado para p es entre el 5 % y el 20 %, es decir, $p \in [0.05, 0.2]$.

Actualización de memoria histórica

En cada generación del algoritmo, los valores de F_i y CR_i que generaron un vector *trial* $u_{i,g}$ mejor que el *target* $x_{i,g}$, se guardan temporalmente en S_F y S_{CR}, y al final de la generación la memoria se actualiza usando el Algoritmo 1 [15].

Algoritmo 1 Actualización de memoria

if $S_{CR} \neq \emptyset$ **and** $S_F \neq \emptyset$ **then**
  if $M_{CR,k,g} = \perp$ **or** $\max(S_{CR} = 0)$ **then**
    $M_{CR,k,g+1} = \perp$;
  else
    $M_{CR,k,g+1} = mean_{WL}(S_{CR})$;
  end if
  $M_{F,k,g+1} = mean_{WL}(S_F)$;
  $k++$;
  if $k > H$ **then**
    $k = 1$;
  end if
end if

En el Algoritmo 1, el índice k ($1 \leq k \leq H$) determina la posición de la memoria que será actualizada. Al inicio del algoritmo, k se inicializa en 1 y se incrementa cada vez que se añade un nuevo valor a la memoria. Si $k > H$ entonces nuevamente se le asigna el valor 1. Cuando todos los vectores *trial* de la generación g son peores que los vectores *target*, S_F y S_{CR} se encontrarán vacíos y la memoria no se actualizará. El promedio ponderado de Lehmer $mean_{WL}(S)$, se calcula con la Ecuación 35. S se refiere tanto a S_F como a S_{CR}:

$$mean_{WL}(S) = \frac{\sum_{k=1}^{|S|} w_k \cdot S_k^2}{\sum_{k=1}^{|S|} w_k \cdot S_k}, \tag{35}$$

en donde w_k es calculado usando la diferencia de aptitud (Δf_k) entre el *trial* y el *target*, con el fin de influenciar la adaptación de parámetros (Ecuaciones 36 y 37):

$$w_k = \frac{\Delta f_k}{\sum_{m=1}^{|S|} \Delta f_m}, \tag{36}$$

$$\Delta f_k = |f(u_k, g) - f(x_k, g)|. \tag{37}$$

Durante la actualización de M_{CR}, si $M_{CR}, k, g = \perp$ o el máximo de los valores almacenados en S_{CR} es 0, se asignará a M_{CR}, k, g el valor terminal $\perp$.

Reducción lineal del tamaño de población

El mecanismo de reducción de población en L-SHADE, decrementa linealmente el tamaño de la población como una función del número de evaluaciones de la función objetivo (NFE). En donde el tamaño de la población en la generación 1 es N^{init}, y el tamaño al final del proceso es N^{min}. Después de cada generación g, la población de la generación siguiente (NP_{g+1}) se calcula de acuerdo a la Ecuación 38 [15]:

$$NP_{g+1} = round\left[\left(\frac{N^{min} - N^{init}}{MAX\_NFE}\right) \cdot NFE + N^{init}\right], \tag{38}$$

en donde a N^{init} se le asigna el número mínimo de individuos necesarios para aplicar los operadores evolutivos. En el caso de current-to-$p$best (Ecuación 34), el número mínimo de individuos es 4. NFE es el número de evaluaciones de la función objetivo realizadas hasta el momento y MAX_NFE se refiere al número máximo de evaluaciones. Cada vez que NP_{g+1} sea menor que NP, los peores $NP - NP_{g+1}$ individuos de la población serán eliminados.

3.2. Adaptación de L-SHADE para Resolver Problemas Restringidos

Como se mencionó anteriormente, L-SHADE fue diseñado para resolver problemas sin restricciones, sin embargo, en este trabajo ha sido adaptado para manejar las restricciones relacionadas con el mecanismo de cuatro barras. A este nuevo enfoque se le ha llamado C-LSHADE (*Constrained-LSHADE*). En el Algoritmo 2 se muestra el seudocódigo correspondiente a C-LSHADE y se marcan con una $\Longrightarrow$ las modificaciones realizadas al algoritmo original, las cuales fueron las siguientes:

- Añadir los criterios propuestos por Deb [10], para la selección del mejor individuo (línea 17). Dichos criterios contemplan tres escenarios, los cuales son: (1) cualquier individuo factible es preferible sobre cualquier individuo no factible, (2) entre dos individuos factibles, aquel que tenga el mejor valor de función objetivo será seleccionado y (3) entre dos individuos no factibles, aquel que tenga la menor suma de violación de restricciones (SVR) será seleccionado.
  Es importante mencionar que para los escenarios 2 y 3, se consideró mejor al *trial* $u_{i,g}$, sólo si el valor comparado (Función Objetivo o SVR) era menor ($<$) que el valor del *target* $x_{i,g}$:
- Almacenar los valores Δf en $S_{\Delta f}$ (líneas 7 y 18). Aunque los autores de L-SHADE no lo mencionan, se pueden guardar los valores Δf (Ecuación 37) para realizar de forma más cómoda el cálculo de w_k (Ecuación 36) al final de la generación. En este trabajo, dichos valores se almacenan en $S_{\Delta f}$, una estructura similar a S_F y S_{CR}. Debe notarse que sin importar el escenario de comparación de Deb, siempre que el *trial* sea mejor que el *target*, se almacenará la diferencia de sus funciones objetivo.

- Ordenar a los individuos de la población con base en su SVR y Función Objetivo (línea 25). Para eliminar a los peores individuos, éstos deben ordenarse dando prioridad a la SVR y posteriormente a la Función Objetivo.

Algoritmo 2 C-LSHADE

1: **Inicio**
2: Crear aleatoriamente la población inicial $x_{i,0}$, donde $i = 0, ..., NP - 1$
3: Evaluar la población $f(x_{i,0})$, donde $i = 1, ..., NP$
4: $g = 1, NP = N^{init}$
5: Establecer todos los valores de M_F y M_{CR} en 0.5
6: **while** *(El criterio de paro no se haya cumplido)* **do**
7: $\Longrightarrow$ $S_F = \emptyset$, $S_{CR} = \emptyset$, $S_{\Delta f} = \emptyset$
8: **for** $i = 1$ hasta NP **do**
9: Seleccionar aleatoriamente r_i de $[1, H]$
10: $F_{i,g} = randc_i(M_{F,r_i}, 0.1)$
11: **if** $M_{CR} = \perp$ **then**
12: $CR_{i,g} = 0$
13: **else**
14: $CR_{i,g} = randn_i(M_{CR,r_i}, 0.1)$
15: **end if**
16: Generar el vector *trial* $u_{i,g}$, de acuerdo a current-to-$p$best/1/bin
17: $\Longrightarrow$ **if** $f(u_{i,g})$ es mejor que $f(x_{i,g})$ según criterios de Deb
18: $\Longrightarrow$ $F_{i,g} \rightarrow S_F, CR_{i,g} \rightarrow S_{CR}, |f(u_{i,g}) - f(x_{i,g})| \rightarrow S_{\Delta f}$
19: $x_{i,g+1} = u_{i,g}$
20: **end if**
21: **end for**
22: Actualizar las memorias M_F y M_{CR} (Algoritmo 1)
23: Calcular NP_{g+1} con la Ecuación 38
24: **if** $NP_{g+1} < NP$ **then**
25: $\Longrightarrow$ Ordenar los inviduos de la población P, de forma ascendente con base en su SVR y Función Objetivo, y eliminar los peores $NP - NP_{g+1}$ miembros
26: **end if**
27: $g++$
28: **end while**
29: **Fin**

4. Resultados

Dos experimentos fueron realizados para comparar el desempeño del algoritmo C-LSHADE contra un algoritmo DE/rand/1/bin del estado del arte [1], que promueve el uso de valores aleatorios para F y CR en la solución de problemas mecánicos de optimización. El primer experimento tenía como objetivo comparar los resultados obtenidos por los dos algoritmos, mientras que el segundo pretendía analizar el comportamiento de convergencia en cada uno. Con el fin de conocer qué tan difícil es encontrar soluciones factibles en cada caso de estudio, se calculó la medida ρ, que estima la proporción de la zona factible con respecto al espacio de búsqueda, calculando cuántas soluciones factibles existen en un millón de soluciones generadas de forma aleatoria. Los valores obtenidos fueron $\rho = 0.0033\,\%$, $\rho = 2.0607\,\%$ y $\rho = 0.0\,\%$ para CE1, CE2 y CE3, respectivamente. Estos valores indican que la región factible es muy pequeña y por lo tanto la optimización numérica con restricciones se complica.

En el primer experimento, se realizaron 31 ejecuciones independientes en ambos algoritmos por cada caso de estudio descrito en la Sección 2, y se obtuvieron

Tabla 2: Configuración de parámetros para DE/rand/1/bin basado en enfoque de [1].

Parámetros	DE(CE1)	DE(CE2)	DE(CE3)
NP	138	50	138
MAX_GEN	5435	600	5435
F (aleatorio entre)	$[0.3, 0.9]$	$[0.3, 0.9]$	$[0.3, 0.9]$
CR (aleatorio entre)	$[0.8, 1)$	$[0.8, 1)$	$[0.8, 1)$

medidas estadísticas. Para validar los datos obtenidos, se utilizó la prueba de la suma de rangos de Wilcoxon con un nivel de significancia del 95 % y en todos los casos las diferencias fueron significativas.

La configuración de parámetros empleada para el algoritmo DE/rand/1/bin se puede apreciar en la Tabla 2. El tamaño de la población NP y el número máximo de generaciones MAX_GEN se ajustaron con base en una experimentación previa, tomando en cuenta que para el CE2 se realizaran 30,000 evaluaciones y para los otros dos casos de estudio se efectuaran 750,000 aproximadamente, y seleccionando aquellos valores que brindaron mejores resultados.

En contraste, los parámetros F, CR y NP no necesitan ser configurados en L-SHADE ya que se controlan de forma automática. Sin embargo, L-SHADE incorpora nuevos parámetros que a pesar de ser menos sensibles que los parámetros originales de Evolución Diferencial, también deben ajustarse. En la variante[4] de L-SHADE implementada en este trabajo deben configurarse los parámetros H, N^{init} y p, que corresponden al número de entradas de la memoria, la población inicial y el porcentaje de current-to-$p$best respectivamente. Para los tres casos de estudio, los valores que se utilizaron en los parámetros mencionados se mantuvieron iguales que los recomendados por los creadores del algoritmo, que son: $H = 6$, $N^{init} = (D \times 18)$ y $p = 0.11$ [15]. Cabe mencionar que N^{init} varía de acuerdo a la dimensión del problema. El "valor terminal" $\perp$ mencionado en la Ecuación 32 se estableció en -1.

Los resultados finales obtenidos por ambos algoritmos se muestran en la Tabla 3, en donde se puede observar que en el CE1, C-LSHADE llega a un mejor resultado que DE/rand/1/bin y emplea un 46.66 % menos evaluaciones para encontrarlo. Para el CE2 y el CE3, ambos algoritmos encuentran soluciones competitivas, sin embargo, C-LSHADE emplea un 33.33 % menos evaluaciones en el CE2 y un 40 % menos en el CE3. Con base en los resultados presentados del primer experimento, se puede afirmar lo siguiente:

- C-LSHADE es capaz de encontrar soluciones mejores o iguales que DE.
- El porcentaje de evaluaciones de la función objetivo que emplea C-LSHADE para encontrar la mejor solución, es considerablemente menor que en DE.
- C-LSHADE libera al usuario de la tarea de ajustar los parámetros F, CR y NP.

[4] El algoritmo original tiene dos variantes que se distinguen por el uso de un archivo externo. En tal archivo, se almacenan los vectores *targets* que no fueron seleccionados y posteriormente se ocupan para realizar la mutación.

Tabla 3: Resultados estadísticos obtenidos por C-LSHADE y DE/rand/1/bin.

	CE1		**CE2**		**CE3**	
	C-LSHADE	DE	C-LSHADE	DE	C-LSHADE	DE
Mínimo	**0**	1.76E-28	2.62E-03	2.62E-03	2.74E-01	2.74E-01
Máximo	**6.42E-04**	2.76E-02	**2.62E-03**	2.62E-03	1.35E+01	1.35E+01
Promedio	**2.40E-04**	2.98E-03	**2.62E-03**	2.62E-03	**2.43E-01**	2.91E+00
Mediana	**3.61E-08**	4.27E-06	**2.62E-03**	2.62E-03	**2.74E-01**	4.39E-01
Desv. Estándar	**2.76E-04**	8.19E-03	**2.46E-13**	1.85E-11	**4.93E+00**	5.27E+00
Varianza	**7.66E-08**	6.70E-05	**6.07E-26**	3.45E-22	**2.43E+01**	2.78E+01
Evaluaciones	**400,000**	750,030	**20,000**	30,000	**450,000**	750,030

En el segundo experimento se graficaron los puntos de convergencia correspondientes a la ejecución ubicada en el valor de la mediana de las 31 ejecuciones independientes. En la Figura 2 se pueden apreciar las gráficas correspondientes a cada caso de estudio. Como se puede notar, C-LSHADE es capaz de encontrar mejores soluciones de forma más rápida que DE/rand/1/bin en el CE1 y el CE3. En el CE2, que es el más sencillo según la medida ρ, el algoritmo DE presenta un mejor comportamiento, sin embargo, al final de la ejecución C-LSHADE se recupera y llega incluso a una mejor solución que DE/rand/1/bin (Tabla 3).

Finalmente se muestran las mejores soluciones encontradas por C-LSHADE en los tres casos de estudio:
CE1: $\boldsymbol{p} = [38.45761229,\ 8.538400025,\ 28.15726641,\ 38.40204633,\ 37.83976344,\ 16.61315669,\ 3.949817649,\ -9.473622057,\ 59.45027718,\ 1.753788656,\ 2.466800077,\ 2.97662654,\ 3.472145065,\ 4.019649903,\ 5.124622599] = 0$

CE2: $\boldsymbol{p} = [14.31454712,\ 2.211165827,\ 14.31454712,\ 14.31454712,\ 2.174361582,\ 0.02220978] = 2.62E-03$

CE3: $\boldsymbol{p} = [2.614591256,\ 1.034801411,\ 1.826885061,\ 2.207891397,\ 1.250924809,\ 0.447340736,\ 5.826803988,\ 0.099169548,\ 1.328797661,\ 0.410281169,\ 1.039364885,\ 1.6500082,\ 2.260034437,\ 2.865981272,\ 3.490250928,\ 4.163941448,\ 4.905546352,\ 5.416480201,\ 6.067608264] = 2.74E-01$

5. Conclusiones

En este trabajo se modificó L-SHADE, un algoritmo adaptativo basado en Evolución Diferencial que controla los parámetros F, CR y NP, para resolver tres casos de estudio de la optimización de un mecanismo de cuatro barras. La modificación principal consistió en agregar los criterios de factibilidad propuestos por Deb en la etapa correspondiente a la selección del mejor individuo. A este enfoque se le llamó C-LSHADE.

Los resultados finales obtenidos por C-LSHADE fueron comparados contra un algoritmo DE/rand/1/bin del estado del arte que promueve valores aleatorios en F y CR para resolver problemas mecánicos de optimización. Con base en los

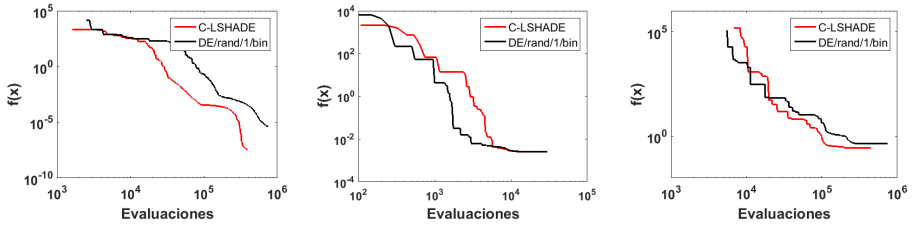


Fig. 2: Gráficas de convergencia de C-LSHADE y DE/rand/1/bin para el a) CE1, b) CE2 y c) CE3.

resultados presentados, se puede decir que aplicar C-LSHADE a problemas de diseño mecánico permite obtener soluciones mejores o iguales que DE/rand/1/bin, empleando un porcentaje considerablemente menor de evaluaciones de la función objetivo. Además, en los casos de estudio más complejos las gráficas de convergencia indicaron que el algoritmo propuesto es capaz de encontrar más rápido mejores soluciones. Finalmente, C-LSHADE libera al usuario de la tarea de realizar un ajuste manual de parámetros, que comúnmente se realiza con base en prueba y error. Como trabajo futuro se propone estudiar y adaptar la versión más reciente de L-SHADE presentada en [8] con el fin de comparar su desempeño contra C-LSHADE.

Agradecimientos. El primer autor agradece el apoyo de CONACyT a través de una beca para cursar estudios de posgrado en LANIA. El segundo autor agradece el apoyo por parte de CONACyT a través del proyecto No. 220522.

Referencias

1. Portilla-Flores, E. A., Mezura-Montes, E., Alvarez-Gallegos, J., Coello-Coello, C. A., Cruz-Villar, C. A., Villarreal-Cervantes, M. G.: Parametric reconfiguration improvement in non-iterative concurrent mechatronic design using an evolutionary-based approach. Engineering Applications of Artificial Intelligence, 24(5):757–771 (2011)
2. Sánchez-Márquez, A., Vega-Alvarado, E., Portilla-Flores, E. A., Mezura-Montes, E.: Synthesis of a planar four-bar mechanism for position control using the harmony search algorithm. In: Electrical Engineering, Computing Science and Automatic Control (CCE), 11th International Conference on, pp. 1–6, IEEE (2014)
3. Dasgupta, D., Michalewicz, Z.: Evolutionary algorithms in engineering applications. Springer Science & Business Media (2013)
4. Eiben A. E., Michalewicz Z., Schoenauer M., Smith J. E.: Parameter control in evolutionary algorithms. In: Parameter setting in evolutionary algorithms, pp. 19–46, Springer (2007)

5. Mezura-Montes, E., Portilla-Flores, E. A., Capistran-Gumersindo, E.: Dynamic parameter control in differential evolution with combined variants to optimize a three-finger end effector. In: Power, Electronics and Computing (ROPEC), IEEE International Autumn Meeting on, pp. 1–6, IEEE (2015)
6. Vega-Alvarado, E., Portilla-Flores, E. A., Mezura-Montes, E., Flores-Pulido, L., Calva-Yáñez, M. B.: A memetic algorithm applied to the optimal design of a planar mechanism for trajectory tracking. Polibits, 53:83–90 (2016)
7. Villarreal-Cervantes, M. G., Cruz-Villar, C. A., Alvarez-Gallegos, J., Portilla-Flores, E. A.: Differential evolution techniques for the structure-control design of a five-bar parallel robot. Engineering Optimization, 42(6):535–565 (2010)
8. Brest, J., Maučec, M. S. y Bošković, B.: il-shade: Improved l-shade algorithm for single objective real-parameter optimization. In: Evolutionary Computation (CEC), IEEE Congress on, pp. 1188–1195, IEEE (2016)
9. Zhang, J., Sanderson, A. C.: Jade: adaptive differential evolution with optional external archive. IEEE Transactions on evolutionary computation, 13(5):945–958 (2009)
10. Deb, K.: An efficient constraint handling method for genetic algorithms. Computer methods in applied mechanics and engineering, 186(2):311–338 (2000)
11. Price, K., Storn, R., Lampinen, J.: Differential evolution: a practical approach to global optimization. Springer Science & Business Media (2006)
12. Pant, M., Thangaraj, R., Singh, V.P.: Optimization of mechanical design problems using improved differential evolution algorithm. Int. Journal of Recent Trends in Engineering, 1(5) (2009)
13. Siarry, P., Michalewicz, Z.: Advances in metaheuristics for hard optimization. Springer Science & Business Media (2007)
14. Tanabe, R., Fukunaga, A. S.: Success-history based parameter adaptation for differential evolution. In: Evolutionary Computation (CEC), IEEE Congress on, pp. 71–78, IEEE (2013).
15. Tanabe, R., Fukunaga, A. S.: Improving the search performance of shade using linear population size reduction. In: Evolutionary Computation (CEC), IEEE Congress on, pp. 1658–1665, IEEE (2014)
16. Shiakolas, P. S., Koladiya, D., Kebrle, J.: On the optimum synthesis of six-bar linkages using differential evolution and the geometric centroid of precision positions technique. Mechanism and Machine Theory, 40(3):319–335 (2005)

ISSN 1870-4069

Gramática evolutiva con estrategia evolutiva como núcleo de una hiperheurística de generación aplicada al problema de empacado

Carlos Ávila[1], Marco Sotelo-Figueroa[2], Héctor Puga[1], Manuel Ornelas[1], Martín Carpio[1]

[1] Universidad de Guanajuato, División de Ciencias Económico Administrativas, Departamento de Estudios Organizacionales, Guanajuato, GTO, México

[2] Tecnologico Nacional de México, Instituto Tecnológico de León, Departamento de Estudios de Posgrado e Investigación, León, GTO, México

uribeabc@gmail.com, masotelo@ugto.mx, pugahector@yahoo.com, mornelas67@yahoo.com.mx, jmcarpio61@hotmail.com

Resumen. Las hiperheurísticas son herramientas que permiten la generación de metodologías para la solución de determinados problema. Existen dos tipos de hiperheurísticas, las de Generación y las de Selección; si bien el objetivo de ambas es generar metodologías los insumos de ambas cambian, las de Selección requieren de una serie de heurísticas de bajo nivel mientras que las de Generación requieren una serie de elementos heurísticos los cuales describen el problema. Dichas hiperheurísticas están compuestas por un núcleo el cual es un algoritmo de optimización, el cual por su naturaleza tiene una serie de parámetros. La optimización de dichos parámetros es una rama de estudio que permite buscar cuales son aquellos bajo los cuales el algoritmo tiene el mejor desempeño en una instancia de un problema. En el presente trabajo se propone el uso de una Gramática Evolutiva (GE) con un motor de búsqueda basado en una Estrategia Evolutiva (ES) como núcleo de una hiperheurística de generación debido a que cuentan con un proceso de auto adaptación que permite eliminar el proceso de optimización de parámetros. La propuesta es aplicada al Problema de Empacado y los resultados son comparados usando la prueba estadística no paramétrica de Friedman contra los obtenidos por una hiperheurística con motor de búsqueda basado en Gramática Evolutiva usando Optimización por Cumulo de Partículas (PSO) y Optimización Evolutiva por Cúmulo de Partículas (PESO).

Palabras clave: Hiperheurística, programación genética, gramáticas evolutivas, problema de empacado, estrategia evolutiva.

Carlos Ávila, Marco Sotelo-Figueroa, Héctor Puga, Manuel Ornelas, Martín Carpio

Evolutionary Grammar with an Evolution Strategy as a Generation Hiperheuristic Core Applied to the Bin Packing Problem

Abstract. Hyperheuristics are tools that allow the generation of methodologies for solving certain problems. There are two types of hyperheuristics, Generation and Selection; although the objective of both is to generate methodologies, their inputs change, the selection ones require a series of low level heuristics while those of Generation require a series of heuristic elements which describe the problem. These hyperheuristics are composed by a core which is usually an optimization algorithm, which by nature has a series of parameters. The optimization of these parameters is a branch of study that aims to find those values under which the algorithm has the best performance in an instance of a problem. In the present work we propose the use of an Evolutionary Grammar (GE) with a search engine based on an Evolution Strategy (ES) as the core of a generation hyperheuristic because of it's self-adaptive nature that allows to eliminate the process of parameter optimization. The proposal is applied to the Bin Packing Problem and the results are compared using Friedman's non-parametric statistical test against those obtained by a search engine hyperheuristic based on Evolutionary Grammar using Particle Swarm Optimization (PSO) and Particle Evolutionary Swarm Optimization (PESO).

Keywords: Hiperheuristic, genetic programming, evolutionary grammar, bin packing problem, evolution strategies.

1. Introducción

Las Hiperheurísticas (HH) [5,4] son herramientas que permiten la selección o generación de metodologías de búsqueda de soluciones de un determinado problema. Existen muchos trabajos que se han dedicado al estudio de las características y aplicaciones de las hiperheurísticas [11,30,13,29,17]. Las hiperheurísticas de generación son utilizadas cuando es necesario encontrar una metodología que permita obtener un resultado de un problema[5,4,11]; para esto es necesario tener caracterizado el problema de manera que se puedan obtener los elementos representativos del mismo, conocidos como componentes heurísticos. Dentro de los componentes que conforman las hiperheurísticas de generación esta el núcleo que requiere del uso de alguna metodología de construcción que permita hacer combinaciones de componentes heurísticos de bajo nivel.

Las Gramáticas Evolutivas (GE) han sido utilizadas en varias ocasiones como núcleo de las hiperheurísticas de generación obteniendo buenos resultados [3,**?**,24,26,23,2,27,25]. Una GE necesita de un algoritmo o motor de búsqueda que le permita evolucionar los individuos y así poder encontrar soluciones [16,15], algunos de los algoritmos usados como motor de búsqueda de GE son

Algoritmo Genético (GA) [3,18,23,2], Optimización por Cúmulo de Partículas (PSO) [26,27], Optimización Evolutiva por Cumulo de Partículas (PESO) [26], Optimización por Colonia de Abejas (BSO) [27] y Evolución Diferencial (DE) [24,25] entre otros. Uno de los mayores retos a la hora de implementar Algoritmos Evolutivos (AE) como los antes mencionados es el de la optimización de parámetros que en sí mismo es un problema difícil [21,14,22]. En el presente artículo se propone el uso de una Estrategia Evolutiva (ES) como motor de búsqueda dado que es un algoritmo auto-adaptativo [7,12] ya que adapta los parámetros del mismo en el transcurso de su ejecución.

La hiperheurística implementada es aplicada al problema de empacado (BPP) que es un problema de optimización combinatoria de tipo NP-difícil que busca empacar un conjunto de piezas en la menor cantidad de contenedores, donde todos los contenedores tienen la misma capacidad [20,19]. Los resultados son comparados usando la prueba no parámetrica de Friedman con los obtenidos en [25].

2. Hiperheurísticas de generación

Las Hiperheurísticas (HH) [5,4] son técnicas que trabajan en un espacio de búsqueda de heurísticas o metodologías en lugar de buscar soluciones directamente. Dentro del marco de las hiperheurísticas existen dos enfoques, el de selección que selecciona heurísticas y el de generación que se dedica a la construcción de nuevas metodologías para la solución de problemas.

Las hiperheurísticas requieren un conjunto de heurísticas o componentes heurísticos de bajo nivel que trabajan en un espacio de búsqueda de soluciones para el problema que se está trabajando, y un algoritmo que trabaje fuera del dominio del problema como núcleo de búsqueda de heurísticas.

Una parte fundamental de las hiperheurísticas de generación es la necesidad de contar con componentes heurísticos para poder construir heurísticas funcionales para los problemas con los que se está trabajando. Sin estos componentes el núcleo hiperheurístico no puede funcionar. Comúnmente un núcleo hiperheurístico de generación utiliza alguna variante de Programación Genética (GP) como elemento generador de heurísticas que han sido aplicadas en distintas ocasiones con resultados satisfactorios [3] [18,24] [26] [23] [2] [27] [25].

Uno de los algoritmos usados por las hiperheurísticas como núcleo de búsqueda son las Gramáticas Evolutivas (GE).

Las GE son una forma de programación genética basada en gramáticas que permite construir programas de forma automática usando un proceso de mapeo basado en una gramática [16] [15] [26]. Las GE trabajan con individuos formados por una cadena binaria llamada genotipo en la cual los codones son representados por un valor entero. En GE el genotipo es transformado en fenotipo que puede ser una función o programa por medio de un proceso de mapeo, como se puede ver en la Figura 1. La gramática está formada por la tupla $\{N, T, P, S\}$ donde N representa los caracteres no terminales, T los caracteres terminales, P las reglas de producción y S el carácter no terminal con el que se inicia el mapeo.

El proceso de mapeo utiliza la ecuación (1) para realizar el mapeo:

$$Regla = c\%r, \tag{1}$$

donde c es el codón actual y r el número de reglas de producción para el no terminal que está siendo mapeado.

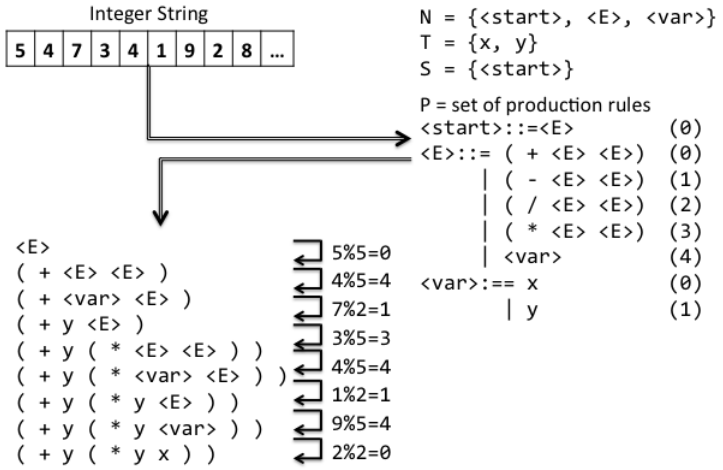


Fig. 1. Ejemplo de una gramática evolutiva BNF y su proceso de mapeo [26].

2.1. Estrategias evolutivas

Las Estrategias Evolutivas (ES) [7] son un tipo de Algoritmo Evolutivo (EA) que esta basado en el concepto de la evolución de la evolución. Las ES son algoritmos de naturaleza auto-adaptativa lo que les permite ajustar los parámetros de dicha estrategia de forma automática lo que implica ajustar el tamaño y la dirección de los pasos de muta de cada individuo [7,12] [1]. De acuerdo con [12] la auto-adaptación consiste en la búsqueda del espacio de parámetros de forma implícita en la ejecución del algoritmo.

En las ES los individuos o soluciones son representados por tuplas de la siguiente forma:

$$X(t) = (x(t), \sigma(t)), \tag{2}$$

donde $x(t)$ es el genotipo y $\sigma(t)$ es el vector de parámetros de la estrategia.

En [7,1] se indican los pasos a seguir para implementar una estrategia evolutiva:

- **Inicializar:** se inicializa el genotipo y los parámetros de la estrategia de cada individuo y se calcula el fitness.

- **Cruzar:** se genera una nueva generación de hijos a partir de la generación de padres haciendo uso de un operador de cruza.
- **Mutar:** a partir de los parámetros de la estrategia y de un operador de muta se muta a los individuos de la nueva generación.
- **Evaluar:** se calcula el fitness de cada individuo de la nueva generación.
- **Seleccionar:** los individuos con mejor fitness son seleccionados como la nueva generación de padres.

Una ES $(\mu+\lambda)$ genera λ hijos a partir de μ padres, donde $1 \leq \mu \leq \lambda < \infty$. La siguiente generación está conformada por los mejores individuos de μ y λ. La ES $(\mu + \lambda)$ implementa elitismo para conservar a los mejores padres. El Algoritmo 1 presenta la estructura de una ES $(\mu + \lambda)$ como se explica en [7].

Algorithm 1 Algoritmo de una ES $(\mu + \lambda)$.

```
1: Inicializar población de padres μ
2: while Criterio de paro no alcanzado do
3:    while hijos < λ do
4:       Cruzar individuos de población μ para obtener un hijo h
5:       Mutar h
6:       Evaluar h
7:    end while
8:    Seleccionar mejores individuos de μ y λ para nueva población μ
9: end while
```

3. Problema de empacado

El Problema de Empacado (BPP por su nombre en ingles Bin Packing Problem) [20,19] es un problema clásico de investigación operacional de tipo NP-difícil que consiste en empacar una cantidad x de piezas en la menor cantidad posible de contenedores, cada uno con la misma capacidad. El BPP unidimensional se define de la siguiente forma [19,6]. Se toma un conjunto $J = \{1, ..., n\}$ que contiene n número de piezas, cada una con un tamaño o peso w_j que deben ser empacadas en la menor cantidad de contenedores m sin exceder la capacidad c del contenedor. Por esa razón se establece que $w_j \leq c$.

Se han propuesto diferentes funciones objetivo para el problema de empacado [26], entre ellas se encuentran la ecuación (3), que mide la diferencia entre los contenedores usados y el límite superior teoríco de contenedores necesarios, y la (4), que calcula el espacio libre que queda en los contenedores.

$$Fitness = B - \frac{\sum_{i=1}^{n} w_i}{C}, \tag{3}$$

$$Fitness = 1 - \left(\frac{\sum_{i=1}^{n} ((\sum_{j=1}^{m} w_j x_{ij})/C)^2}{n} \right), \tag{4}$$

donde

- B es el número de contenedores usados,
- n el número de contenedores,
- m es el número de piezas,
- w_j es la j-ésima pieza, considerando

$$x_{ij} = \left\{ \begin{array}{c} 1 \text{ si la pieza j está en el contenedor i,} \\ 0 \text{ de otra forma,} \end{array} \right\}$$

- C es la capacidad del contenedor.

En [20] se propone agrupar las instancias de prueba en función a (5).

$$w \in [v_L X, v_U X], \tag{5}$$

donde w es la pieza, v_L es el límite inferior y v_U es el limite superior en relación a la capacidad del contenedor X.

A partir de la función (5) se obtienen tres diferentes grupos de instancias como se muestra en [25] y se propone en [9]. Los grupos se muestran en la Tabla 1.

Tabla 1. Condiciones de agrupamiento de las instancias.

Grupo	**Condición**
Triplets	$v_L = \frac{1}{4}$ y $v_U = \frac{1}{2}$
Hard	$\bar{w} \approx \frac{1}{3}$ o cercano a $\frac{1}{n}$ donde $n \geq 3$
Regular	cualquier otro caso

4. Diseño experimental

Los experimentos están configurados en base a los realizados en [26] donde se usa una GE con PSO y PESO como motor de búsqueda de un núcleo hiperheurístico de generación. En el presente trabajo se hace uso de un núcleo hiperheurístico de generación con una ES $(\mu + \lambda)$ como motor de búsqueda. La gramática 1.1 está basada en la heurística $FirstFit$ y fue propuesta en [28].

La función objetivo utilizada es la que recompensa a los contenedores con menos espacio libre propuesta en [8], Ecuación (4).

Las instancias de pruebas usadas son las siguientes:

- **Binpack:** utilizadas por E. Falkenauer [10] tienen dos tipos de instancias, uniformemente distribuidas y triplets que contienen grupos de 3 piezas que deben ser empacadas en un mismo contenedor para obtener una solución óptima.
- **BinData:** utilizadas por A. Scholl, R.Klein y C. Jürgens [19] contienen tres conjuntos de instancias uniformemente distribuidas.
- **Wäescher:** utilizadas por G. Wäescher y T. Gau [31] son 17 instancias consideradas de las mas difíciles.

$$
\begin{aligned}
\langle \text{N} \rangle &\models \langle \text{inicio} \rangle, \langle \text{expr} \rangle, \langle \text{expr2} \rangle, \langle \text{op} \rangle, \langle \text{var} \rangle \\
\langle \text{T} \rangle &\models \text{F}, \text{C}, \text{S}, +, \text{-}, /, *, \text{abs}, <= \\
\langle \text{inicio} \rangle &\models (\langle \text{expr} \rangle) <= (\langle \text{expr} \rangle) \\
\langle \text{expr} \rangle &\models (\langle \text{expr} \rangle \langle \text{op} \rangle \langle \text{expr} \rangle) \mid \langle \text{var} \rangle \mid abs(\langle \text{expr2} \rangle) \\
\langle \text{expr2} \rangle &\models (\langle \text{expr2} \rangle \langle \text{op} \rangle \langle \text{expr2} \rangle) \mid \langle \text{var} \rangle \\
\langle \text{var} \rangle &\models \text{F} \mid \text{C} \mid \text{S} \\
\langle \text{op} \rangle &\models + \mid * \mid \text{-} \mid /
\end{aligned}
$$

Gramática 1.1. Gramatica basada en la Heurística FirstFit [28].

- **Hard28:** utilizadas por J.E. Schoenfield son instancias que contienen entre 160 y 200 piezas con una capacidad del contenedor de 1000.

Se realizó un proceso de agrupamiento como se muestra en la sección 3. Se toma una instancia de cada uno de los grupos con la finalidad de usarla como parte del entrenamiento de la hiperheurística de generación, y los valores reportados son los que se obtienen al aplicar la heurística a todo el grupo. Los parámetros usados por la ES, PSO y PESO, el Cuadro 2, fueron obtenidos de [26].

Tabla 2. Parametros usados.

Parametros	PESO y PSO	ES
Tamaño de la Población	50	
Hijos	-	100
w	1.0	-
ϕ_1	0.8	-
ϕ_2	0.5	-
Llamadas a Función	1000	

Se realizaron 51 experimentos independientes sobre el conjunto de instancias de entrenamiento. Posteriormente se obtuvo la heurística media de los 51 experimentos para aplicarla a cada una de las instancias y poder calcular el rendimiento con la ecuación (4). Para finalizar se reagrupan las instancias en los conjuntos originales y se suman los rendimientos de todas las instancias de cada conjunto para calcular el rendimiento de la hiperheurística para cada conjunto de prueba.

5. Resultados

En el Cuadro 3 se muestran las heurísticas obtenidas para cada grupo de instancias y se puede observar que son heurísticas equivalentes a la heurística

$FirstFit$, dicha heurística es en la que se basa la gramática utilizada en el presente trabajo. El Cuadro 4 muestra los resultados obtenidos para cada conjunto de prueba después de calcular el rendimiento de cada instancia y reagruparlas en su conjunto original, los resultados de PSO y PESO se tomaron de los obtenidos en [26], dichos resultados corresponden a los obtenidos al aplicar la heurística $FirstFit$.

Tabla 3. Heurísticas generadas para cada grupo.

Grupo	**Heurística**
Triplets	$((abs(F) + S)) \leq (abs(C))$
Hard	$(abs((S + F))) \leq (C)$
Regular	$(F) \leq ((C - S))$

Tabla 4. Resultados obtenidos.

Instancia	**ES**	**PSO**	**PESO**
bin1data	316.10602	316.10602	316.10602
bin2data	93.38851	93.38851	93.38851
bin3data	1.885825	1.885825	1.885825
binpack1	2.604965	2.604965	2.604965
binpack2	2.396851	2.396851	2.396851
binpack3	2.133326	2.133326	2.133326
binpack4	1.935722	1.935722	1.935722
binpack5	0.0	0.0	0.0
binpack6	0.0	0.0	0.0
binpack7	0.0	0.0	0.0
binpack8	0.0	0.0	0.0
hard28	0.65535	0.65535	0.65535

Se aplicó la prueba estadística no paramétrica de Friedman para determinar si existen diferencias en el rendimiento de los algoritmos, el valor de la prueba es de 0.0 con un p-valor de 1.0 lo cual implica que no existe evidencia estadística para decir que existen diferencias en el rendimiento de los algoritmos.

6. Conclusiones

En base a los resultados obtenidos usando la Estrategia Evolutiva $(\mu + \lambda)$ como motor de búsqueda de una hiperheurística de generación podemos concluir que dicha metaheurística permite obtener resultados que compiten con los del estado del arte.

La prueba no paramétrica de Friedman no mostró evidencia de que la Optimización por Cumulo de Partícula, Optimización Evolutiva por Cúmulo de Partículas o la Estrategia Evolutiva tuvieran un mejor desempeño al ser usadas como motor de búsqueda en la hiperheurística de generación; sin embargo, los parámetros tanto de la Optimización por Cumulo de Partícula y Optimización Evolutiva por Cúmulo de Partículas fueron obtenidos mediante un proceso de optimización de parámetros.

Los resultados obtenidos muestran que es posible sustituir el proceso de optimización de parámetros, que conlleva a aumentar el tiempo de computo al buscar los paramétros de dichas metaheurísticas para que puedan ser usadas como núcleo de la hiperheurística, por un metaheurística que tiene un proceso auto-adaptativo.

Referencias

1. Bäck, T., Foussette, C., Krause, P.: Evolution Strategies, pp. 7–45. Springer Berlin Heidelberg, Berlin, Heidelberg (2013), http://dx.doi.org/10.1007/978-3-642-40137-4_2
2. Basgalupp, M.P., Barros, R.C., Barabasz, T.: A grammatical evolution based hyper-heuristic for the automatic design of split criteria. In: Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation. pp. 1311–1318. GECCO '14, ACM, New York, NY, USA (2014), http://doi.acm.org/10.1145/2576768.2598327
3. Burke, E.K., Hyde, M.R., Kendall, G.: Grammatical evolution of local search heuristics. IEEE Transactions on Evolutionary Computation 16(3), 406–417 (June 2012)
4. Burke, E.K., Gendreau, M., Hyde, M., Kendall, G., Ochoa, G., Özcan, E., Qu, R.: Hyper-heuristics: a survey of the state of the art. Journal of the Operational Research Society 64(12), 1695–1724 (2013), http://dx.doi.org/10.1057/jors.2013.71
5. Burke, E.K., Hyde, M., Kendall, G., Ochoa, G., Özcan, E., Woodward, J.R.: A Classification of Hyper-heuristic Approaches, pp. 449–468. Springer US, Boston, MA (2010), http://dx.doi.org/10.1007/978-1-4419-1665-5_15
6. Dokeroglu, T., Cosar, A.: Optimization of one-dimensional bin packing problem with island parallel grouping genetic algorithms. Computers & Industrial Engineering 75, 176 – 186 (2014), http://www.sciencedirect.com/science/article/pii/S036083521400182X
7. Engelbrecht, A.P.: Computational intelligence: an introduction. John Wiley & Sons (2007)
8. Falkenauer, E., Delchambre, A.: A genetic algorithm for bin packing and line balancing. In: Proceedings 1992 IEEE International Conference on Robotics and Automation. pp. 1186–1192 vol.2 (May 1992)
9. Falkenauer, E.: Tapping the Full Power of Genetic Algorithm through Suitable Representation and Local Optimization: Application to Bin Packing, pp. 167–182. Springer Berlin Heidelberg, Berlin, Heidelberg (1995), http://dx.doi.org/10.1007/978-3-642-61217-6_8
10. Falkenauer, E.: A hybrid grouping genetic algorithm for bin packing. Journal of Heuristics 2(1), 5–30 (1996), http://dx.doi.org/10.1007/BF00226291

11. Kheiri, A., Özcan, E.: An iterated multi-stage selection hyper-heuristic. European Journal of Operational Research 250(1), 77 – 90 (2016), http://www.sciencedirect.com/science/article/pii/S0377221715008255
12. Kramer, O.: Evolutionary self-adaptation: a survey of operators and strategy parameters. Evolutionary Intelligence 3(2), 51–65 (2010), http://dx.doi.org/10.1007/s12065-010-0035-y
13. Martin, M.A., Tauritz, D.R.: Hyper-heuristics: A study on increasing primitive-space. In: Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation. pp. 1051–1058. GECCO Companion '15, ACM, New York, NY, USA (2015), http://doi.acm.org/10.1145/2739482.2768457
14. Nannen, V., Smit, S.K., Eiben, A.E.: Costs and Benefits of Tuning Parameters of Evolutionary Algorithms, pp. 528–538. Springer Berlin Heidelberg, Berlin, Heidelberg (2008), http://dx.doi.org/10.1007/978-3-540-87700-4_53
15. O'Neil, M., Ryan, C.: Grammatical Evolution, pp. 33–47. Springer US, Boston, MA (2003), http://dx.doi.org/10.1007/978-1-4615-0447-4_4
16. O'Neill, M., Ryan, C.: Grammatical evolution. IEEE Transactions on Evolutionary Computation 5(4), 349–358 (Aug 2001)
17. Pillay, N.: A review of hyper-heuristics for educational timetabling. Annals of Operations Research 239(1), 3–38 (2016), http://dx.doi.org/10.1007/s10479-014-1688-1
18. Sabar, N.R., Ayob, M., Kendall, G., Qu, R.: Grammatical evolution hyper-heuristic for combinatorial optimization problems. IEEE Transactions on Evolutionary Computation 17(6), 840–861 (Dec 2013)
19. Scholl, A., Klein, R., Jürgens, C.: Bison: A fast hybrid procedure for exactly solving the one-dimensional bin packing problem. Computers & Operations Research 24(7), 627 – 645 (1997), http://www.sciencedirect.com/science/article/pii/S0305054896000822
20. Schwerin, P., Wäscher, G.: The bin-packing problem: A problem generator and some numerical experiments with ffd packing and mtp. International Transactions in Operational Research 4(5), 377 – 389 (1997), http://www.sciencedirect.com/science/article/pii/S0969601697000257
21. Smit, S.K., Eiben, A.E.: Comparing parameter tuning methods for evolutionary algorithms. In: 2009 IEEE Congress on Evolutionary Computation. pp. 399–406 (May 2009)
22. Smit, S.K., Eiben, A.E.: Parameter Tuning of Evolutionary Algorithms: Generalist vs. Specialist, pp. 542–551. Springer Berlin Heidelberg, Berlin, Heidelberg (2010), http://dx.doi.org/10.1007/978-3-642-12239-2_56
23. Sosa-Ascencio, A., Ochoa, G., Terashima-Marin, H., Conant-Pablos, S.E.: Grammar-based generation of variable-selection heuristics for constraint satisfaction problems. Genetic Programming and Evolvable Machines 17(2), 119–144 (2016), http://dx.doi.org/10.1007/s10710-015-9249-1
24. Sotelo-Figueroa, M.A., Soberanes, H.J.P., Carpio, J.M., Huacuja, H.J.F., Reyes, L.C., Soria-Alcaraz, J.A.: Evolving and reusing bin packing heuristic through grammatical differential evolution. In: 2013 World Congress on Nature and Biologically Inspired Computing. pp. 92–98 (Aug 2013)
25. Sotelo-Figueroa, M.A., Puga Soberanes, H.J., Carpio, J.M., Fraire Huacuja, H.J., Reyes, L.C., Soria Alcaraz, J.A.: Clustering Bin Packing Instances for Generating a Minimal Set of Heuristics by Using Grammatical Evolution, pp. 151–162. Springer International Publishing, Cham (2015), http://dx.doi.org/10.1007/978-3-319-10960-2_10

26. Sotelo-Figueroa, M.A., Soberanes, H.J.P., Carpio, J.M., Huacuja, H.J.F., Reyes, L.C., Soria-Alcaraz, J.A.: Improving the bin packing heuristic through grammatical evolution based on swarm intelligence. Mathematical Problems in Engineering 2014 (2014), http://dx.doi.org/10.1155/2014/545191
27. Sotelo-Figueroa, M.A., Soberanes, H.J.P., Carpio, J.M., Fraire Huacuja, H.J., Reyes, L.C., Alcaraz, J.A.S., Espinal, A.: Generating Bin Packing Heuristic Through Grammatical Evolution Based on Bee Swarm Optimization, pp. 655–671. Springer International Publishing, Cham (2017), http://dx.doi.org/10.1007/978-3-319-47054-2_43
28. Sotelo-Figueroa, M.A., Soberanes, H.J.P., Carpio, J.M., Fraire Huacuja, H.J., Reyes, L.C., Soria Alcaraz, J.A.: Evolving Bin Packing Heuristic Using Micro-Differential Evolution with Indirect Representation, pp. 349–359. Springer Berlin Heidelberg, Berlin, Heidelberg (2013), http://dx.doi.org/10.1007/978-3-642-33021-6_28
29. Swiercz, A., Burke, E.K., Cichenski, M., Pawlak, G., Petrovic, S., Zurkowski, T., Blazewicz, J.: Unified encoding for hyper-heuristics with application to bioinformatics. Central European Journal of Operations Research 22(3), 567–589 (2014), http://dx.doi.org/10.1007/s10100-013-0321-8
30. Urra, E., Cabrera-Paniagua, D., Cubillos, C.: Towards a distributed hyperheuristic deploy architecture. In: Proceedings of the 7th Euro American Conference on Telematics and Information Systems. pp. 31:1–31:4. EATIS '14, ACM, New York, NY, USA (2014), http://doi.acm.org/10.1145/2590651.2590682
31. Wäscher, G., Gau, T.: Heuristics for the integer one-dimensional cutting stock problem: A computational study. Operations-Research-Spektrum 18(3), 131–144 (1996), http://dx.doi.org/10.1007/BF01539705

ISSN 1870-4069

Modelo innovador para un aparador comercial usando un algoritmo competitivo imperialista

Yuliet Oliva-Romero[1], Alberto Ochoa-Zezatti[1], Ana Marcela-Herrera[1], Diego Alberto Oliva-Navarro[2]

[1] Universidad Autónoma de Querétaro, Campus Juriquilla, Querétaro, México

[2] Universidad de Guadalajara, CUCEI, Departamento de Ciencias Computacionales, Guadalajara, Jal., México

{yulietoliva, anaherreranavarro, diegoa.olivan}@gmail.com, alberto.ochoa@uacj.mx

Resumen. El artículo presenta la resolución de un problema a partir del modelado social asociado con la adecuada selección de ropa y su distribución en un aparador comercial utilizando conjunto de 87 prendas con diferentes características asociadas con una colección de moda. Se presenta un caso de estudio con respecto a la selección de 47 diferentes combinaciones posibles utilizando datos obtenidos de los patrones culturales de diversidad descritos en DuO Store y su repoblación de existencias de ropa. El objetivo de esta investigación es aplicar una solución computacional, en este caso el uso del Algoritmo Competitivo Imperialista, para resolver un problema específico. Combinado a esto, analizamos la selección de colores aplicados sobre el vestuario, el conjunto de estudios conformado por 85 prendas permitió analizar características individuales sin afectar la visualización en el aparador comercial propuesto y nos dio la oportunidad de entender la selección de colores y formas con respecto a la representación visual. Demostrando que la adecuación de características sociales y culturales especifica la correcta selección de colores. Esta investigación trata de explicar esta innovadora representación y localización de una muestra de prendas en un aparador comercial.

Palabras clave: Algoritmo competitivo imperialista, problema multiCombinatorio, reconocimiento de patrones.

Innovating Model to a Dresser using an Imperialist Competitive Algorithm

Abstract. The paper presents a problem resolved from social Modeling associated with the adequate selection of clothes and their distribution in a Dresser using a range of 87 clothes with different features associated with a fashion collection to represent the symbolic capital of it. A case of study is presented regarding to the selection of 47 different possible combinations using data obtained from the diversity cultural patterns described in DuO Store and its stock repository of clothes. The intention of this research is to apply the computational solution in this case the use of Imperialist Competitive Algorithm to resolve a specific problem, adapted from the modeled Literature about of

Societies. Combined to this, we analyzed the selection of colors to applied on the wearing, the set of studies conformed by 85 clothes allowed to analyze individual characteristics without affecting the visualization in the proposal Dresser, and gave the opportunity us to understand the selection of colors and forms with respect the visual representation. Demonstrating that the matching of characteristic social and cultural specify the correct selection of colors; this research tries to explain this innovative representation and location of a sample of clothes on a Dresser.

Keywords: Imperialist competitive algorithm, multicombinatorial problem, pattern recognition.

1. Introducción

El dominio de la computación bio-inspirada está ganando gradualmente impulso en los tiempos modernos. A medida que la sociedad avanza hacia una era digital, ha habido una explosión en la cantidad de datos generados. Esta explosión de datos hace que sea cada vez más difícil extraer información, así como recopilar conocimientos mediante algoritmos estándar, debido a la creciente complejidad del análisis [1]. Encontrar la mejor solución se vuelve cada vez más difícil de identificar, no es imposible debido al gran y dinámico alcance de las soluciones y la complejidad de los cálculos. La solución óptima para un problema *NP-hard* es un punto en el hiperespacio n-dimensional y la identificación de la solución es computacionalmente muy cara o incluso no factible. Por lo tanto, se necesitan enfoques inteligentes para identificar soluciones de trabajo adecuadas [1].

Dentro de las metaheurísticas, los algoritmos bio-inspirados están ganando prominencia gradualmente ya que estos son inteligentes, pueden aprender y adaptarse como organismos biológicos. Estos algoritmos están llamando la atención de la comunidad científica debido a la creciente complejidad de los problemas, a un creciente rango de soluciones potenciales en hiperplanos multidimensionales, a la naturaleza dinámica de los problemas y restricciones e incompletos, probabilísticos e imperfectos cambios en las decisiones de la información. Sin embargo, el rápido desarrollos en este dominio es cada vez más difícil de rastrear debido a los diferentes algoritmos introducidos.

Los métodos de optimización evolutiva, inspirados en procesos naturales, han demostrado un buen desempeño en la solución de problemas de optimización complejos. Evidencia de esto constituyen los algoritmos genéticos (inspirados en la evolución biológica de la especie humana y otros) [2], la optimización de colonias de hormigas (basada en el esfuerzo de las hormigas para encontrar la ruta óptima en la búsqueda de la fuente de alimento) [3] y optimización de enjambre de partículas (referencia al comportamiento de las partículas en la naturaleza), son ampliamente utilizados para resolver problemas de optimización de la ingeniería [4].

El algoritmo competitivo imperialista (ICA, por sus siglas en inglés) es un nuevo método evolutivo de optimización inspirado en la competencia imperialista. Comienza con una población inicial como otros algoritmos evolutivos denominada país, que son de tipo colonizado o imperialista [5]. La competencia imperialista es la parte principal

del algoritmo propuesto y permite que las colonias converjan al mínimo global de la función de coste. Este método es una nueva estrategia de búsqueda global, sociopolíticamente motivada, que se ha introducido recientemente para abordar diferentes tareas de optimización [6].

Algunos de los mejores países de la población son seleccionados para ser el imperialista y el resto forma las colonias de estos imperialistas. Cada país está determinado por:

$$\text{country}_i = [p1, p1, \ldots, pNvar], \tag{1}$$

donde N_{var} es la dimensión del problema a optimizar [6].

Los mejores países de la población inicial, considerando la función de costos de los mismos, son seleccionados como los imperialistas y otros países conocidos como las colonias de estos imperialistas. Los valores de variable en el país se representan como número de punto flotante. El costo de un país se encuentra evaluando la función de costo f en las variables var[p_1, p_2 ,..., p_N], por lo tanto, tenemos [5]:

$$\text{cost} = f(\text{country}) = [p_1, p_1, \ldots, p_{Nvar}]. \tag{2}$$

Seleccionamos el N_{imp} de los países más poderosos para formar los imperios. Las restantes Ncol de la población serán las colonias pertenecientes a los imperios. Para formar los imperios iniciales, dividimos las colonias entre los imperialistas basándonos en su poder. Para dividir las colonias entre los imperialistas proporcionalmente, se define el siguiente costo normalizado de un imperialista:

$$C_n = c_n - \max(c_i), \tag{3}$$

donde c_n es el costo del n-ésimo imperialista y C_n es el costo normalizado. Como resultado el poder normalizado de cada imperialista se define como:

$$p_{n=\frac{C_n}{\sum_{i=1}^{N_{imp}} C_i}}. \tag{4}$$

Por otra parte, el poder normalizado de un imperialista es evaluado por sus colonias. Entonces el número inicial de colonias de un imperio sería:

$$NC_n = \text{rand}\{p_n * N_{col}\}. \tag{5}$$

La figura 1 muestra el diagrama de flujo del ICA [5].

donde, NC_n es el número inicial de colonias del n-ésimo imperio y N_{col} es el número de total de colonias. Después de dividir las colonias entre los imperialistas, estas colonias comienzan a acercarse a su imperio. El poder total de un imperio puede ser determinado por el poder del país imperialista más el porcentaje de poder de sus colonias, como sigue [5]:

$$\text{power}_n = \text{cost}(\text{imperialists}_n) + \varepsilon\, \text{mean}\{(\text{cost}(\text{cost of empires}_n))\}, \tag{6}$$

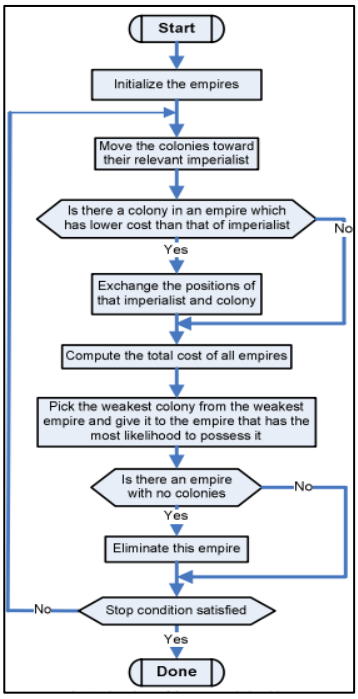


Fig. 1. Diagrama de flujo del ICA.

donde $power_n$ es el poder total (costo) del n-ésimo imperio ε es un número positivo menor que uno.

Los imperios más débiles colapsarán en la competencia imperialista y sus colonias se dividirán entre otros imperios. Después de un tiempo, todos los imperios, excepto el más poderoso, colapsarán y todas las colonias estarán bajo el control de este imperio [12], que es la respuesta del problema de optimización.

El desarrollo de un aparador comercial requiere, por un lado, del desarrollo conceptual y por otro, del desarrollo de diversas medidas de similitud que permitan establecer ubicaciones desde un concepto básico y en los datos. Pero es necesario priorizar el desarrollo conceptual y de las categorías del sistema del modelo visual, y paralelamente pensar en el modelo matemático.

Para este modelo, utilizamos la correlación para medir la similitud entre la ropa de una colección. Esta medida nos permite localizar cada prenda con respecto a otros y organizarla en grupos (diferentes temas en el aparador comercial). Un peso para cada atributo representa la importancia de un atributo. Para cada par de prenda se define la siguiente función de fitness:

$$\sum_{i}^{n} = \mathrm{abs}[\mathrm{corr}(W_i \mathrm{Attribute} A_i, W_i \mathrm{Attribute} B_i,)], \tag{7}$$

donde:

Fig. 2. Representación de colores a la ropa.

W_i= Vector de pesos que representan la importancia del atributo i,

A, B = Representan las prendas (hombre-mujer) que se desea medir la similitud y el $Attribute_i$ = Vector de los valores de los atributos de cada prenda.

Mientras que el valor de correlación disminuye, la relación entre las prendas es más fuerte. Se establece un umbral de 0.5, si el valor de correlación es menor que el umbral, asumimos que no hay relación entre las prendas.

2. Distribución de elementos basados en una adecuada representación visual de colores

Desde el punto de vista de un algoritmo bio-inspirado, este problema de optimización combinatoria es muy complejo, debido a que no se conoce la mejor ubicación de un individuo representativo de una prenda en específico con respecto a los otros individuos [7]. En el algoritmo propuesto, los individuos en el beliefscape (beliefscape) a través de su mejor paradigma (BestParadigm) se ponen a cero para representar el hecho de que la cultura aumenta la expectativa asociada con la ubicación de una prenda con respecto a las demás y usar un modelo de representación social, incentivando el comportamiento asociado con el mejor paradigma (BestParadigm). En la figura 2 se muestra la representación de colores de las prendas con la selección más adecuada de temas y representación visual para distribuir elementos en un aparador comercial.

Para ello se seleccionó un almacén de prendas en existencia y se caracterizó su capital simbólico con base en ocho atributos que tiene una representación visual: control emocional, habilidad de lucha, inteligencia, agilidad, fuerza, resistencia, liderazgo social y velocidad, estas características permiten describir tanto a una

Fig. 3. Modelo comparativo para asignar ubicaciones en un tocador representado por prendas de mujer y de hombre.

sociedad como a la ropa asociada. La investigación está basada en el deseo de compartir la comprensión intuitiva sobre el tratamiento de una nueva clase de sistemas, individuos capaces de tener empatía, una característica reservada a las personas vivas. Desde la perspectiva de un modelo matemático, esto tiene varias restricciones relacionadas con el espacio, la combinación de colores y precio.

3. Múltiples coincidencias

Se realiza un rango de diecisiete evaluaciones según diferentes combinaciones de colores y vestuario de más de 50 ejecuciones en diferentes escenarios. El proceso de licitación para la selección final del armario para cada emisión comenzará cuando se analizan diferentes características. En aras del desarrollo proporcional, todas las prendas deben estar representadas, en la medida de lo posible y depender del interés generado en cada una. El momento de evaluar y contratar cada oferta de prendas será muy importante, especialmente al comienzo dela serie. Se dará preferencia a las cuestiones con más similitudes socioculturales.

Para competir se seleccionarán 87 temas. Cada número aceptado y se propondrá para participar en exactamente diecisiete de estas evaluaciones. Los temas evaluarán su preferencia de torneos, una vez que la lista final de múltiples coincidencias se evalúa y el algoritmo evalúa estos. El algoritmo se reserva el derecho de asignar prendas a

evaluar de acuerdo a las necesidades de organización de su aparador comercial y las prendas para cada aparador comercial, así como asignar la lista de prendas antes de que comience el ciclo.

Cada evaluación tendrá hasta 88 prendas combinando sobre un programa de diecisiete evaluaciones con sus respectivas ejecuciones. Con el objetivo de armonización por parte del algoritmo, se programará para cumplir el programa de comparación de diferentes similitudes utilizando una ronda de análisis de concordancia múltiple y con base en el género asignado a un tema, como en la figura 3.

Las prendas que califiquen para la selección en un aparador comercial serán elegidas teniendo en cuenta la siguiente prioridad:

3.1. Coincidencias

Para el primer ciclo de similitud, toda la ropa en el almacén y coincidencias de similitud se invitará a participar en diferentes comparaciones, tratar de determinar las dos prendas más similares a una persona. Dada la organización de cada una de las prendas que coincidan en cada ronda en el algoritmo, todas estas prendas se les pedirá que comprometan su participación en la evaluación de cada serie. En caso de que alguna de estas prendas deje de participar en la serie, el algoritmo puede designar una prenda para reemplazar y esta prenda debe ser clasificada en parte superior de la reposición del almacén, teniendo en cuenta esta situación como relevante, principalmente cuando el almacén es limitado.

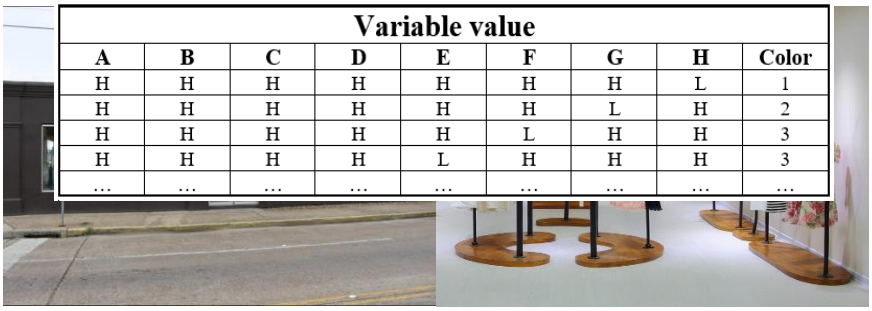
3.2. Clasificación

Basándose en un cálculo promedio a dos decimales, la lista de calificación en la serie de comparaciones antes del inicio del ciclo, se seleccionarán veinte calificadores (excluyendo las siete prendas que se van a comparar las coincidencias). Si la prenda tiene la misma calificación promedio, se usará el número de similitudes evaluadas (período de calificación) para determinar la clasificación.

Para asegurar una participación en el futuro, se recomienda un mínimo de 25 combinaciones en las cuatro listas de calificación, incluyendo el valor de inicio de la lista de calificación. Dada la breve explicación de su evaluación y notificación en la primera edición de evaluaciones completas, se propone el uso de una guía narrativa para ello. Si alguna ropa no acepta jugar en la serie de múltiples coincidencias, entonces se adoptará el proceso de selección usando la calificación promedio más el número de juegos jugados en el período de calificación hasta que se complete el número de calificadores requeridos para la serie.

4. Experimentación

Para poder organizar el arreglo más eficiente de las prendas en un tocador, desarrollamos una atmósfera capaz de almacenar los datos de cada uno de los que representan estas prendas, esto con el fin de distribuir de una forma óptima a cada una



Variable value								
A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**Color**
H	H	H	H	H	H	H	L	1
H	H	H	H	H	H	L	H	2
H	H	H	H	H	L	H	H	3
H	H	H	H	L	H	H	H	3
…	…	…	…	…	…	…	…	…

Fig. 4. Aparador final (interior-exterior) construido de acuerdo con hasta ocho atributos diversos basados en el Algoritmo Competitivo Imperialista (Sistema Multiagente).

de las prendas evaluadas. Una de las características más interesantes observadas en este experimento fue la diversidad de los patrones culturales establecidos por cada prenda con respecto a su capital simbólico.

Las escenas estructuradas asociadas con los agentes no se pueden reproducir en general, ya que sólo representan un poco de tiempo dados en el espacio y el tiempo de las diferentes prendas. Estos representan una forma única e innovadora de comportamiento adaptativo que resuelve un problema computacional que no intenta agrupar la prenda sólo con un factor asociado con su apariencia externa (atributos de cada atuendo), tratando de resolver un problema computacional que implica un cambio complejo entre las representaciones existentes.

Las configuraciones generadas pueden estar relacionadas con el conocimiento del comportamiento de un cliente potencial con respecto a un problema de optimización (para seleccionar culturalmente ropa similar, sin ser del mismo tipo [8]). El experimento principal consistió en detallar cada una de las prendas de una colección, con 500 agentes y una condición de 50 épocas, lo que nos permitió generar la mejor selección de cada tipo de prenda y su posible ubicación en un aparador comercial. Se obtuvo después de comparar las diferentes similitudes culturales y sociales de cada vestimenta, y evaluar con el modelo de múltiples coincidencias cada uno de ellos [9].

La investigación permitió clasificar cada una de las prendas pertenecientes a cada tipo, con diferente vestuario para prendas sólo con identidad cultural, lo que permite identificar cambios en el tiempo respecto a otras prendas. En la figura 4 se representa el aparador final (exterior-interior) construido según hasta ocho atributos diferentes, respectivamente (ver Figura 4).

El diseño del experimento consiste en una prueba de matriz ortogonal, con las interacciones entre las variables: control emocional, capacidad de lucha, inteligencia, agilidad, fuerza, resistencia, liderazgo social y velocidad. Estas variables se estudian en un rango de color (1 a 64). El arreglo ortogonal es $L_N(2^{**}8)$, en otras palabras, 8 factores en N ejecuciones, N se define por la combinación de los valores posibles de las 8 variables y el posible rango de color (Ver Figura 5). De acuerdo con los resultados obtenidos para el tiempo de alojamiento en el transporte debe realizarse para acomodar el tiempo de descarga en el siguiente letrero.

Sobre la base de los resultados obtenidos en el experimento, el promedio de similitud entre las prendas se encontró que el promedio es 4,0625; esto significa que debe tener

que mejorar en la gestión del uso de la combinación de colores y estándar según el clima. Otros factores que afectan serían el uso de accesorios tales como: pulseras, collares, alfileres y encantos, una implementación para resolver este problema es el uso de la visualización de la moda. Como se puede ver para mantener este tipo de materiales es necesario utilizar la mercancía directa e indirecta. La mercancía indirecta asegura el suelo, da resistencia al movimiento lateral y longitudinal de la ropa. Mientras que las corbatas directas ancladas a los maniquíes en la cómoda. Para realizar este proceso, es necesario contar con las dimensiones correctas y la distancia de la calle y la altura de estos, cables o cadenas, teniendo en cuenta el material de estos maniquíes.

5. Conclusiones

Mediante la utilización del ICA se mejora el entendimiento para obtener el cambio del "mejor paradigma", porque se clasifica apropiadamente las comunidades de agentes al basarse en un acercamiento a la relación que mantienen sus atributos, esto permitió entender que el concepto de "moda basada en el capital simbólico" existe con base en la determinación de la función de aceptación por parte del resto de la ropa al lugar propuesto para el resto de los mismos. ICA ofrece una poderosa alternativa a los problemas de optimización y redistribución de la técnica de clustering. Por esta razón, esta técnica ofrece un panorama bastante comprensible del fenómeno cultural representado [6]. Esta técnica permite incluir la posibilidad de generar conocimiento experimental creado por la comunidad de agentes para un nuevo dominio de aplicación.

El análisis del nivel y grado de conocimiento cognitivo de cada comunidad es un aspecto que se desea evaluar para el trabajo futuro. La respuesta puede residir entre la similitud que existe en la comunicación entre dos culturas diferentes y como estas son percibidas [10].

Por otro lado, comprender las verdaderas similitudes que tienen diferentes sociedades con base en las características que las hacen contribuyentes de cluster y le permite también mantener su propia identidad, demuestra que las pequeñas variaciones van más allá de las características fenotípicas y son principalmente asociadas a gustos y características similares desarrolladas a través del tiempo [11]. Una nueva Inteligencia Artificial puede tener cuidado de analizar al por menor estas complejidades que cada sociedad mantiene, sin olvidar que aún nos necesitan métodos para entender las características originales y particulares de cada sociedad.

Agradecimientos A la comunidad de Sociología Computacional que durante una década ha vuelto a buscar formas novedosas de mostrar representación visual y desarrollado diferentes Ateliers of Artificial Societies.

Referencias

1. Das, T. K., Venayagamoorthy, G. K., Aliyu. U. O.: Bio-inspired algorithms for the design of multiple optimal power system stabilizers: SPPSO and BFA. IEEE Trans. Ind. Appl., pp. 1445–1457 (2008)

2. Back, T.: Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms. (1996)
3. Colorni, A., Dorigo, M., Maniezzo, V.: Distributed Optimization by Ant Colonies. In: Actes de la première conférence européenne sur la vie artificielle, pp. 134–142 (1991)
4. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: Proceedings of IEEE International Conference on Neural Networks, Washington (1995)
5. Atashpaz-Gargari, E., Lucas, C.: Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In: IEEE Congress on evolutionary computation, Singapore (2007)
6. Zuckermann, D.: Culture and Organizations.
7. Ochoa, A. et al.: Baharastar – Simulador de Algoritmos Culturales para la Minería de Datos Social. In: Proceedings of COMCEV'2007 (2007)
8. Memory Alpha: Memory-alpha.org (Star Trek World) (2009)
9. Vukčević, I., Ochoa, A.: Similar cultural relationships in Montenegro. In: JASSS'2005 (2005)
10. Ponce, J. et al.: Data Mining and Knowledge Discovery in Real Life Applications. In: Book edited by: Julio Ponce and Adem Karahoca (2009)
11. Ustaoglu, Y.: Simulating the behavior of a minority in Turkish Society. In: ASNA'2009 (2009)
12. Bernal, E., Castillo, O., Soria, J., Valdez, F.: Imperialist Competitive Algorithm with Dynamic Parameter Adaptation Using Fuzzy Logic Applied to the Optimization of Mathematical Functions. pp. 1–19 (2017)

ISSN 1870-4069

Implementación del algoritmo MBS para resolver instancias de bin packing

Luis Alberto Silva Escamilla, Hilda Castillo Zacatelco, Rafael de la Rosa Flores

Benemérita Universidad Autónoma de Puebla, Puebla, México

siel_alb@me.com, hildacz@gmail.com, rafael@cs.buap.mx

Resumen. El Minimum Bin Slack es una heurística propuesta por Gupta y Ho [6]. Dicho algoritmo es definido por sus autores como un algoritmo orientado al contenedor. El MBS parte de la idea de que minimizar el número de contendores a utilizar es equivalente a reducir el espacio sobrante de los mismos: es decir, a menor cantidad de contenedores, menor espacio sobrante. Considerando la idea anterior, este algoritmo utiliza el concepto de búsqueda lexicográfica para ir llenando cada contenedor buscando optimizar el espacio del mismo y una vez lleno, proceder a llenar el siguiente contenedor, sin perder nunca el objetivo de reducir el espacio sobrante en los mismos. En este trabajo, se presenta una propuesta para la implementación de este algoritmo y se realiza una comparación de los resultados obtenidos con él contra los resultados generados por el algoritmo FFD. Finalmente se dan a conocer los resultados encontrados.

Palabras clave: Bin packing problem, herurística, minimum bin slack.

Implementation of the MBS Algorithm to Solve Instances of the Bin Packing Problem

Abstract. The Minimum Bin Slack is a heuristic proposed by Gupta and Ho [6]. Such algorithm is defined by its authors as an algorithm focused on the container. MBS starts from the idea that minimizing the number of containers is equivalent to reduce the left space in the containers: thus, less quantity of containers means less left space in the containers. Considering this idea, MBS algorithm uses the concept of lexicographic search in order to fill each container, searching to optimize space in container and once it is filled, proceed to fill the next one, without forgetting the goal of reducing the space behind. In this article, a proposal is presented to implement this algorithm and the results obtained from it are compared to results obtained from FFD. Finally, the results of this comparison are presented.

Keywords: Bin packing problem, heuristic, minimum bin slack.

1. Introducción

El *bin packing problem* (BPP) [4] es un problema NP-combinatoria [6] (es decir, NP y NP-hard) de optimización combinatoria en el cual dado un conjunto de ítems de tamaño variable, se busca acomodarlos dentro de contenedores de tamaño fijo, buscando optimizar el número de contenedores a utilizar, es decir, usando el menor número de contendores para colocar el mayor número de ítems posible. Para este problema existen tres variantes tales como el *bin packing* unidimensional (1D-BPP), de 2 dimensiones (2D-BPP) [9] o de 3 dimensiones (3D-BPP) [5]; en dichas variantes [3] solo cambia el número de dimensiones tanto de los ítems como de los contenedores no obstante, en el presente trabajo nos limitaremos al bin packing de una dimensión.

Aunque existe una variedad de trabajos relacionados a la búsqueda de la mejor solución a este problema, uno de los trabajos más conocidos y del cual parten muchas otras propuestas de soluciones es el *Minimum bin slack* presentado por Gupta y Ho en 1999. En dicho trabajo, se muestra una clara mejora respecto a otros algoritmos conocidos, tales como el *first fit decreasing* [10] o el *next fit decreasing* [2] sin embargo, las instancias presentadas por los autores, no son instancias reconocidas en la literatura, por lo cual no se puede tener la certeza del desempeño de dicho algoritmo con otros conjuntos de datos creados explícitamente para este problema.

Considerando lo anterior, el trabajo desarrollado a continuación presenta una implementación del algoritmo de MBS buscando si dicho algoritmo es igual de eficiente para conjuntos de datos particularmente diseñados para el *bin packing problem*. De esta forma, el documento se divide como sigue: en la sección 2 se presenta la idea básica sobre el funcionamiento del algoritmo, los dos conceptos fundamentales que son la base de dicho algoritmo y se describe de forma breve el funcionamiento del mismo. En la sección 3 se presentan las instancias utilizadas para probar el desempeño de esta implementación, de igual forma, se menciona de forma breve el algoritmo con el que se comparó MBS y el cual busca mejorar. Finalmente en las secciones 4 y 5 se muestran los resultados y las conclusiones respectivamente.

Respecto a la implementación, es importante mencionar que aunque los autores nos muestran una sugerencia para la implementación del mismo, hay ciertas funciones que no detallan, por lo cual la implementación y por ende los resultados pueden variar, ya que dependen de la interpretación que da el lector.

Durante las pruebas que se realizaron, se pudo observar que existen ciertos casos, los cuales los autores no mencionan que pueden afectar la ejecución del algoritmo, en particular aquellos casos en donde se repiten elementos, sin embargo, este aspecto se discute en los resultados.

2. Minimum Bin Slack

El algoritmo MBS (*Minimum Bin Slack*) es definido por sus autores como un algoritmo enfocado al contenedor [3]. Dicho algoritmo parte de la idea de que minimizar el número de contendores a utilizar es equivalente a reducir el espacio sobrante de los mismos: es decir, a menor cantidad de contenedores, menor espacio

sobrante. Considerando lo anterior, este algoritmo a diferencia del FFD o BFD [8] (los cuales se enfocan en colocar elementos en contenedores priorizando la velocidad y por ende, las soluciones ofrecidas por estos requieren más contenedores); se enfoca realmente en optimizar el espacio dentro del contenedor. El MBS utiliza el concepto de búsqueda lexicográfica [6] para ir llenando cada contenedor buscando optimizar el espacio del mismo y una vez lleno, proceder a llenar el siguiente contenedor, sin perder nunca el objetivo de reducir el espacio sobrante en los mismos.

Una vez entendiendo la idea anterior, existen 2 puntos que se consideran claves para el funcionamiento del algoritmo.

Para empezar, el concepto de backtracking. Recordemos que esta estrategia consiste en probar varias combinaciones de elementos hasta hallar una solución, sin embargo, cuando esta posible solución no cumple con una restricción previamente definida, se regresa a la combinación anterior y se exploran otras soluciones. El MBS utiliza esta estrategia de búsqueda para hallar y eliminar elementos en el contenedor.

El segundo punto es el orden. Considerando la estrategia anterior se intuye que ordenando los elementos el tiempo de búsqueda se reduce, por lo cual se asume que para el correcto funcionamiento del algoritmo los elementos están ordenados de forma descendente.

A continuación se describe una serie de pasos que detallan de manera breve el funcionamiento del algoritmo:

1. Se toma el primer elemento de la instancia y se agrega al contenedor.
2. Si la suma de los elementos contenidos hasta ese momento en el contenedor es exactamente igual al valor del contenedor, se termina la ejecución del algoritmo; en caso contrario se continúa con la ejecución.
3. Se agrega el siguiente elemento de la instancia al contenedor, en caso de que el elemento no pueda ingresarse debido a que no existe el espacio mínimo para ubicarse, se procede a buscar de entre los elementos contenidos en el contenedor, un elemento cuyo valor sea igual o mayor al valor del nuevo elemento que se pretende ingresar y se le reemplaza. Sin embargo, para esta búsqueda, se procederá por iniciar a partir del último elemento ingresado, es decir, en la cola de la lista. Se va a paso 4. Caso contrario regresamos a paso 2.
4. Mientras que la instancia no este vacía es decir, no se llegue al final de la lista o el contenedor quede exactamente lleno, se realiza el paso 2.
5. Al finalizar la ejecución del algoritmo debido a que el contenedor se llenó exactamente o se llegó al último elemento de la instancia, se procede a quitar los elementos que se agregaron al contenedor, de la instancia y se vuele a ejecutar el algoritmo ahora con los elementos sobrantes de la instancia. Todo esto hasta que ya no queden elementos en la misma.

La implementación de este algoritmo se puede realizar tanto recursiva como iterativamente [7], lo único importante a considerar es que la forma en que se acomodan los elementos en los contenedores ha de realizarse utilizando el concepto de búsqueda lexicográfica; y que esta búsqueda se repite hasta ya no existan más elementos en la instancia.

3. Instancias

Uno de los objetivos de este trabajo, es ver la eficiencia del MBS en conjuntos de datos que se crearon explícitamente para el BPP. De esta manera para la realización de las pruebas, se eligieron los siguientes conjuntos de datos: 53NIRUP, bin1data, bin2data y hard28. Se eligieron estos conjuntos de datos ya que son los que otros autores emplean en sus pruebas (disponibles en la literatura), de esta manera, se estará busca medir nuestra implementación contra benchmarks perfectamente conocidos e igualmente, se evita el sobreajuste al no utilizar conjuntos de datos propios.

Dichos conjuntos tienen instancias que son particularmente difíciles de resolver ni siquiera por métodos de reducción. Finalmente, tales conjuntos son también difíciles de resolver para el FFD.

El First Fit Decreasing (FFD) funciona como sigue: cada ítem es colocado dentro del contenedor siempre que haya espacio, en caso contrario, el ítem es colocado dentro del siguiente contenedor con espacio disponible [1]. Si hay espacio en un contenedor anterior, el ítem se coloca dentro, de este modo se elimina la restricción que tiene el NF, es decir, siempre se considerarán todos los contenedores siempre que exista espacio. Se eligió el FFD como algoritmo de comparación debido a que a pesar de que existen otros algoritmos derivados de éste, tales como los son el NFD o el BFD (y sus respectivas variantes), el FFD aún muestra un mejor desempeño por si mismo respecto a tiempo y cantidad de contenedores utilizados.

4. Resultados

El programa se corrió sobre una computadora portátil con procesador Intel Core i3 a 1.40 GHz con el sistema operativo Windows 7 de 64 bits. El algoritmo se implementó en lenguaje C++ debido principalmente a que su paradigma permite enfocarse en la solución del problema. Otra razón importante es que C++ tiene un buen manejo de los recursos computacionales, esto se ve reflejado a la hora de ejecutar el programa, donde los tiempos son relativamente cortos. El algoritmo fue puesto a prueba con las siguientes instancias: 53NIRUP, bin1data (Tabla 3, Figura 3), bin2data (Tabla 4. Figura 4) y hard28.

Nuestra implementación emplea arreglos de la clase vector para representar los contenedores, esto nos permitió enfocarnos en la implementación y no en la gestión de recursos, no obstante el mayor reto durante se dio a la hora de interpretar el algoritmo y codificarlo ya que de la forma de implementar podría haber modificado la eficiencia del algoritmo. Se encontró particularmente difícil interpretar la función de búsqueda lexicográfica y aunque como veremos más adelante los resultados obtenidos fueron aceptables, una mejor interpretación podría mejorar al menos ligeramente el desempeño del programa.

Como se puede observar, el algoritmo da buenos resultados al compararlo con el FFD y el BFD y cumple con la tesis propuesta por los autores, ya que el número de contendores disminuye al optimizar el espacio en cada uno [6]. Adicionalmente, con la implementación que se realizó del algoritmo, los tiempos también fueron mejores.

Por el contrario, los resultados varían para cada conjunto de instancias. Por ejemplo, MBS presenta un desempeño similar en cuanto al número de contenedores utilizados para las instancias del conjunto 53NIRUP y solo mostrando una ligera mejoría en cuanto al tiempo como se muestra en la Tabla 1 y Figura 1.

Tabla 1. Instancias de 53NIRUP. Se muestra el tiempo que hizo cada instancia utilizando el MBS y el FFD. Tambien se muestra el número de contenedores para cada instancia.

MBS			FFD	
Instancia	Tiempo	# Contenedores	Tiempo	# Contenedores
BPP02015007_0918.BPP	0.001	10	0.001	9
BPP04000107_0697.BPP	0.005	16	0	15
BPP04005007_0612.BPP	0.003	16	0	15
BPP04005009_0361.BPP	0.006	20	0.001	20
BPP18015010_0605.BPP	0.105	105	0.004	104
BPP20000108_0025.BPP	0.086	77	0.004	76
BPP20000108_0175.BPP	0.069	84	0.004	83
BPP08000107_0624.BPP	0.019	28	0.001	27
BPP06015009_0403.BPP	0.013	31	0.001	30
BPP06005008_0755.BPP	0.009	24	0.001	23

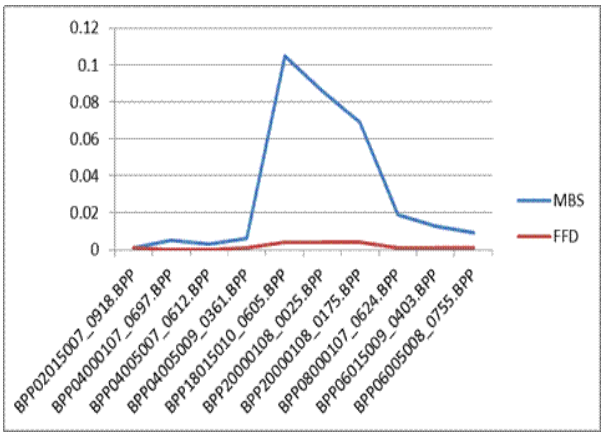


Fig. 1. Tiempos para la instancia 53nirup.

Por otra parte, para las instancias del conjunto de hard28 incluso llego a mostrar un peor desempeño respecto a la cantidad de contenedores utilizados como se observa en la tabla y figura 2, aunque como se puede apreciar, solo de manera sutil.

Esta variación en el desempeño puede tener dos causas principales. Por una parte, cada uno de los conjuntos presenta ciertas características como por ejemplo, el número de elementos repetidos en cada instancia. Igualmente, se podría considerar relevante el

número de dígitos de cada elemento de la instancia, ya que originalmente algunas de las instancias utilizadas por los autores para sus pruebas, se limitaba a utilizar conjuntos con datos de 1 a 100. Por otra parte, algunos de los conjuntos utilizados para nuestra prueba, utilizan una distribución de 1 a 1000

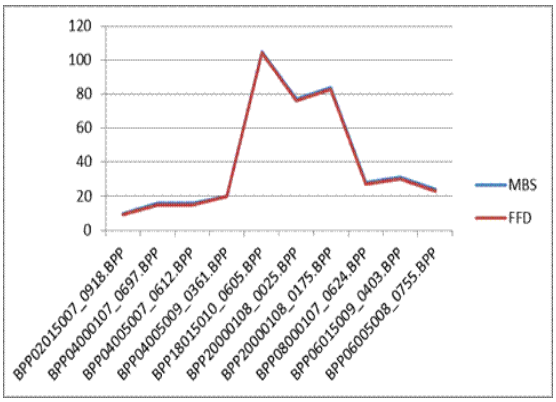


Fig. 2. Gráfica comparativa de contenedores utilizados para instancia 53 nirup.

Tabla 2. Instancias de 28hard. Se muestra el tiempo que hizo cada instancia utilizando el MBS y el FFD. También se muestra el número de contenedores para cada instancia.

MBS			FFD	
Instancia	Tiempo	# Contenedores	Tiempo	# Contenedores
BPP 13.BPP	0.065	68	0.003	67
BPP 40.BPP	0.048	60	0.002	59
BPP 60.BPP	0.065	64	0.002	63
BPP 175.BPP	0.068	84	0.004	83
BPP 359.BPP	0.073	76	0.004	76
BPP 781.BPP	0.076	72	0.004	71
BPP 814.BPP	0.08	82	0.003	81
BPP 419.BPP	0.084	81	0.004	80
BPP 645.BPP	0.055	59	0.002	58
BPP 531.BPP	0.081	84	0.004	83

Para la segunda causa se debe considerar a la siguiente premisa señalada por los autores: “el algoritmo propuesto es óptimo si la suma del requerimiento de los ítems, es menor o igual al doble de la capacidad del contenedor” [6]. Considerando que los conjuntos de instancias utilizadas para las pruebas, fueron diseñados explícitamente para “romper” cualquier propuesta de solución, no sería extraño que alguno de esos conjuntos tuviera alguna característica que haga que la premisa anteriormente mencionada no se cumpla.

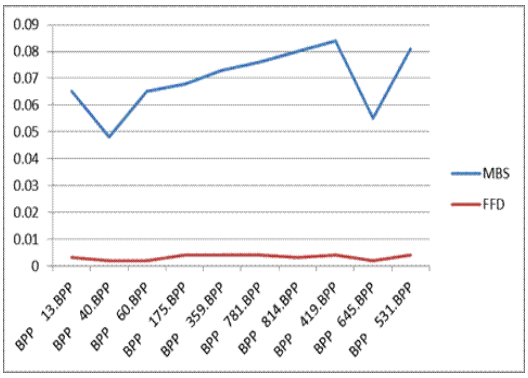


Fig. 3. Tiempos para la instancia *hard28.*

Tabla 3. Instancias de bin1data. Se muestra el tiempo que hizo cada instancia utilizando el MBS y el FFD. También se muestra el número de contenedores para cada instancia.

MBS			FFD	
Instancia	Tiempo	# Contenedores	Tiempo	# Contenedores
N1C1W1_A.BPP	0.007	25	0.004	25
N1C1W1_B.BPP	0.009	31	0.004	31
N1C1W1_C.BPP	0.006	20	0.001	21
N1C1W1_D.BPP	0.006	28	0.003	28
N1C1W1_M.BPP	0.006	30	0.003	30
N1C1W1_N.BPP	0.007	25	0.0004	25
N1C1W2_O.BPP	0.006	29	0.004	29
N3C1W2_F.BPP	0.128	123	0.001	123
N4C2W4_R.BPP	0.576	300	4.041	300
N4C3W1_I.BPP	0.222	168	0.2	168

Tabla 4. Instancias de bin2data. Se muestra el tiempo que hizo cada instancia utilizando el MBS y el FFD. Tambien se muestra el número de contenedores para cada instancia.

MBS			FFD	
Instancia	Tiempo	# Contenedores	Tiempo	# Contenedores
N1W1B1R0.BPP	0.009	20	0.004	18
N1W1B1R1.BPP	0.009	19	0.004	18
N1W1B1R2.BPP	0.009	21	0.003	19
N1W1B2R0.BPP	0.009	18	0.003	17
N1W1B2R3.BPP	0.009	17	0.003	16
N1W2B1R9.BPP	0.008	11	0.001	11

N4W4B3R9.BPP	0.123	56	0.17	56
N4W4B3R8.BPP	0.132	57	0.155	57
N4W4B3R7.BPP	0.011	57	0.13	57
N2W1B1R6.BPP	0.033	35	0.022	34

Finalmente, existen conjuntos de instancias que manejan números no enteros sin embargo, estos conjuntos no se contemplaron para la implementación. No obstante, su implementación es posible, siempre y cuando se considere la naturaleza de estos conjuntos de números en las estructuras de datos a utilizar.

5. Conclusiones

El algoritmo MBS mostró su mejor desempeño para el conjunto de instancias 53NIRUP donde se redujo ligeramente el tiempo de solución y el número de contenedores utilizados. Sin embargo, tales resultados se aprecian en un número reducido de instancias que podría aumentar al efectuar más pruebas con más instancias del mismo conjunto. Por otra parte, para el resto de los conjuntos de datos donde el desempeño en el mejor de los casos se mostró igual al compararlo con el FFD, se buscara mejorar el desempeño implementando el MBS' [7] y observando si esta implementación puede mejorar respecto al MBS y al FFD.

Como se puede observar, el algoritmo da buenos resultados al compararlo con el FFD y cumple con la tesis propuesta por los Gupta y Ho [6] sobre el MBS, ya que el número de contendores disminuye al optimizar el espacio en cada uno. Adicionalmente con la implementación que se realizó del algoritmo, los tiempos también fueron mejores.

Referencias

1. Coffman Jr., E. G., Garey, M.R., Johnson, D.S.: Approximation algorithms for bin packing: a survey. Approximation algorithms for NP-hard problems, pp. 46–93 (1996)
2. Garey, M.R., Johnson, D.S.: Approximation Algorithms for Bin Packing Problems: A Survey. In: Ausiello G., Lucertini M. (eds) Analysis and Design of Algorithms in Combinatorial Optimization. International Centre for Mechanical Sciences (Courses and Lectures), Vol 266, Springer (1981)
3. Bansal, N., Correa, J. R., Kenyon, C., Sviridenko, M.: Bin Packing in Multiple Dimensions: Inapproximability Results and Approximation Schemes. Mathematics of Operations Research, Vol. 31, No. 1, pp. 31–49 (2006)
4. Falkenauer, E., Delchambre, A.: A Genetic Algorithm for Bin Packing and Line Balancing. In: Proceedings IEEE International Conference on Robotics and Automation (1992)
5. Bortfeldt, A.: A genetic algorithm for the two-dimensional strip packing problem with rectangular pieces. European Journal of Operational Research 172, pp. 814–837 (2006)
6. Gupta, J. N.D., Ho, J.C.: A new heuristic algorithm for one-dimensional bin-packing problem. Production Planning & Control 10, pp. 598–603 (1999)

7. Flesznar, K., Hindi, K. S.: New heuristics for one dimensional bin-packing. Computers & Operations Research 29, pp. 821–839 (2002)
8. Maiza, M., Radjef, M.S.: Heuristics for Solving the Bin-Packing Problem with Conflicts. Applied Mathematical Sciences, Vol. 5(35), pp. 1739–1752 (2011)
9. Lodi, A., Martello, S., Vigo, D.: Heuristic algorithms for the three-dimensional bin packing problem. European Journal of Operational Research 141, pp. 410–420 (2002)
10. Dósa, G.: The Tight Bound of First Fit Decreasing Bin-Packing Algorithm Is $FFD(I) \leq 11/9OPT(I) + 6/9$. In: Chen, B., Paterson, M., Zhang, G. (eds), Combinatorics, Algorithms, Probabilistic and Experimental Methodologies, Lecture Notes in Computer Science, Vol. 4614, pp. 1–11 (2007)

ISSN 1870-4069

Algoritmos genéticos aplicados a la optimización de características en la clasificación de arritmias cardiacas utilizando los clasificadores KNN y naive Bayes

Christian Padilla-Navarro[1], Sheila González-Reyna[3], Gabriel Aguilera-González[1], Martín Ortega-Yepez[1], Sandra Bombela-Jiménez[1], Manuel Rangel-Huerta[1], Carlos Lino-Ramírez[2]

[1] Universidad Politécnica de Juventino Rosas, Gto., México

[2] Instituto Tecnológico de León, León, Gto., México

[3] Instituto Tecnológico Superior de Irapuato, Irapuato, Gto., México

{jpadilla_ptc,direccion_ite,
martin.ortega,sandra.bombela,manuel.rangel}@upjr.edu.mx
sheila.gonzalez@itesi.edu.mx, carlos.lino@itleon.edu.mx

Resumen. Muchas muertes en el mundo suceden a consecuencia de enfermedades cardiovasculares. El método propuesto combina metaheurístcas —Algoritmos Genéticos (AG)—, y los clasificadores KNN y Naive Bayes. Las pruebas se realizaron a través de una base de datos del Massachusetts Institute of Technology-Beth Israel Hospital (MIT-BIH) [1]. Las metaheurísticas se implementan para mejorar el rendimiento de los clasificadores. Los resultados experimentales demuestran que se logra hasta un 94 % de precisión en la clasificación.

Palabras clave: Electrocardiograma, arritmia, AG, KNN, naive Bayes.

Genetic Algorithms Applied to the Optimization of Characteristics in the Classification of Cardiac Arrhythmias Using the KNN and Naive Bayes Classifiers

Abstract. Many deaths in the world happen as a result of cardiovascular diseases. The proposed method combines metaheuristic – Genetic Algorithms (AG) – and the KNN and Naive Bayes classifiers. The tests were performed through a database of the Massachusetts Institute of Technology-Beth Israel Hospital (MIT-BIH) [1]. Metaheuristics are implemented to improve the performance of classifiers. Experimental results show that up to 94 % accuracy is achieved in the classification.

Keywords: Electrocardiogram, arrhythmia, GA, KNN, naive Bayes.

1. Introducción

Las arritmias son alteraciones de la frecuencia cardíaca o del ritmo cardíaco [2]. Hay en existencia dos grupos para las arritmias, el primero incluye las taquicardias y fibraciones ventriculares, las cuales provocan una grave amenaza para la vida del paciente, se requiere terapia inmediata con desfibrilador. El segundo grupo está compuesto por arritmias que no ponen en peligro la vida pero también requiere terapia [3].

Las Enfermedades Cardiovasculares (ECV) son la principal causa de muerte en todo el mundo. Cada año mueren más personas por ECV que por cualquier otra causa. Se calcula que en 2012 murieron por esta causa 17.5 millones de personas, lo cual representa un 31 % de todas las muertes registradas en el mundo [4]. Para las personas con ECV o con alto riesgo cardiovascular (debido a la presencia de uno o más factores de riesgo, como la hipertensión arterial, la diabetes, la hiperlipidemia o alguna ECV ya confirmada), son fundamentales la **detección precoz y el tratamiento temprano**, por medio de servicios de orientación o la administración de fármacos, según corresponda [4].

Ante una situación de emergencia lo más importante a considerar es el tiempo que se demora en atender dicha circunstancia [5]. Un paso importante hacia la identificación de una arritmia es la clasificación de los latidos del corazón, debido a la gran cantidad de información para analizar, los especialistas y aparatos tienen dificultades para detectar las irregularidades [6] y por lo tanto, cualquier sistema que ayude a realizar este proceso sería de gran apoyo en la detección de anomalías en las señales, siendo el objetivo principal de esta investigación. El problema a resolver es asegurar y determinar que la clasificación de los electrocardiogramas sea precisa, haciendo un sistema eficaz que en caso de salir alguna anomalía en el ECG notifique al personal encargado del paciente [3].

2. Estado del arte

Un electrocardiograma (ECG) es un procedimiento de diagnóstico con el que se obtiene un registro de la actividad eléctrica del corazón. Es la técnica más usada para el estudio electrofisiológico del corazón, debido a que es un método no invasivo y permite registrar la actividad eléctrica del corazón desde la superficie del cuerpo humano [7].

Se ha descubierto que se pueden detectar muchos de los padecimientos del corazón (arritmia, fibrilación auricular, auriculoventricular (AV) disfunciones, y la enfermedad arterial coronaria, etc.) en base a las anomalias en las señales de ECG [8].

Estas señales son clasificadas en base a sus características para determinar a que clase de arritmia pertenece.

Nait-Hamound M. y Moussaoui A. [9] presentaron dos métodos para la clasificación de arritmias multiclase aplicando Análisis de Componentes Principales (PCA), las Máquinas de Soporte de Vectores Difusos (FSVM, por sus siglas en inglés), y la Closterización Desbalanceada (UC, por sus siglas en inglés).

Thanapatay et al. [10] propusieron un nuevo método de clasificación de ECG usando PCA y SVM.

Los clasificadores tienen diferentes aplicaciones dentro de las señales de ECG, su función principal es separar a través de sus clases las distintos tipos de arritmias.

Nait-Hamoud et al. [11], muestran dos nuevos métodos para la clasificación de ECG para discriminar cinco tipos de ritmos cardíacos combinando Análisis de Componentes Principales (PCA) y Modified Fuzzy One-Against-One (MFOAO).

Molina F. y Benítez D. [12] describen las características más importantes de las señales de ECG mediante el filtrado con la transformada de Wavelet para detectar los puntos significativos en las señales y clasificarlas.

Gutiérrez et al. [13] desarrollaron e implementaron un filtro digital recursivo de señales de ECG para detectar complejos QRS basandose en Wavelet de Haar usando la base de datos de ECG del Massachusetts Institute of Technology-Beth Israel Hospital (MIT-BIT).

Fira et al. [14] investigaron los resultados de la clasificación de la comprensión de señales de ECG basandose en diferentes tipos de matrices de proyección. Se clasifican las señales comprimidas usando KNN, se analiza respecto a las matrices de proyección y a los resultádos obtenidos.

Mientras tanto en Martínez [15] se propone un modelo de clasificación de señales de ECG usando sistemas inteligentes para la detección de anomalías en el corazón usando Algoritmos Genéticos y Redes Neuronales (RN).

Algunos de los padecimientos requieren de un monitoreo constante de la actividad eléctrica del corazón para detectar alteraciones periodicas en las señales. Sannino et al. [16] muestran un sistema para monitorear aceleraciones cardíacas anormales en tiempo real, se basa en un método que analiza las señales por medio de un algoritmo basado en umbral para reducir el ruido.

Elena et al. presenta [17] un algoritmo eficiente para la compresión de ECG y monitoreo en tiempo real, actualizandose con cada nueva señal de entrada por medio de un umbral de ruido óptimo.

Hoffman et al. [25] presentan un algoritmo de procesamiento de ECG basado en Multiple Model Adaptive Estimator (MMAE) para un sistema de vigilancia fisiológica, utilizando veinte señales de la base de datos de ECG del MIT.

En arritmias cardíacas la cantidad de datos que se procesa para la detección suele ser muy grande, en consecuencia aumenta el tiempo de computación, así como los requerimientos de memoria y la cantidad de características usadas. En diferentes investigaciones se busca reducir el tiempo de ejecución y la cantidad de recursos computacionales, incrementar la efectividad y hacerla más precisa con menor cantidad de datos.

Kim en la investigación [18] propone Cuatro Niveles Vectoriales (Quad Level Vector-QLV) para el flujo de compresión y flujo de clasificación para mejorar el rendimiento con baja complejidad computacional. En el algoritmo de clasificación se emplean la segmentación de los latidos del corazón y los métodos de detección de R-peak.

Bilgin et al. en la investigación [19] aplican JPEG2000 para comprimir las señales de ECG usando la base de datos de MIT-BIH como ejemplo.

Cuesta-Frau et al. [20], muestran la comparación de distintos métodos que son usados para extraer la información principal de señales de ECG, también se busca reducir el costo computacional usando la base de datos del MIT.

Kallas et al. [21] mostraron la combinación de la clasificación de arritmias multiclase usando SVM con la extracción de características usando PCA en las señales de ECG.

Mientras tanto Abdeel-Badeeh M et al. [8] aplicó el aprendizaje automático de máquinas modernas en el diagnóstico de ECG.

Soman T., Bobbie P. [22] utilizaron sistemas de aprendizaje automático, Oner, J48 y Naive Bayes para clasificar el conjunto de datos de arritmias.

Gao et al. [23] describen un sistema para detectar arritmias cardiacas basandose en una red neuronal artificial bayesiana (ANN). Su desempeño en esta tarea se compara con otros clasificadores como Naive Bayes, árboles de decisión, regresión logística y redes RBF.

En Daamouche [24] se logró optimizar el nivel de clasificación usando Máquinas de Soporte Vectorial (Support Vector Machines-SVM, Optimización por Enjambre de Partículas (PSO) y se usaron Wavelets para reducir el ruido.

3. Metaheurísticas

Las metaheurísticas pueden concebirse como estrategias generales de diseño de procedimientos heurísticos (capacidad de un sistema para realizar de forma inmediata innovaciones positivas) para la resolución de problemas con un alto rendimiento o al diseño de alguno de los tipos fundamentales de procedimientos heurísticos de solución de un problema de optimización [26].

3.1. Algoritmos genéticos

Los Algoritmos Genéticos son métodos adaptativos, generalmente usados en problemas de búsqueda y optimización de parámetros, basados en la reproducción sexual y en el principio supervivencia del más apto.

Más formalmente, y siguiendo la definición dada por Goldberg, los Algoritmos Genéticos son algoritmos de búsqueda basados en la mecánica de selección natural y de la genética natural. Combinan la supervivencia del más apto entre estructuras de secuencias con un intercambio de información estructurado, aunque aleatorizado, para constituir así un algoritmo de búsqueda que tenga algo de las genialidades de las búsquedas humanas [27].

Para alcanzar la solución a un problema, se parte de un conjunto inicial de individuos, llamado población, generado de manera aleatoria. Cada uno de estos individuos representa una posible solución al problema. Estos individuos evolucionarán tomando como base los esquemas propuestos por Darwin [28] sobre la selección natural, y se adaptarán en mayor medida tras el paso de cada generación a la solución requerida.

3.2. Orígenes

El desarrollo de los Algoritmos Genéticos se debe en gran medida a John Holland, investigador de la Universidad de Michigan. A finales de la década de los 60 desarrolló una técnica que imitaba en su funcionamiento a la selección natural. Aunque originalmente esta técnica recibió el nombre de planes reproductivos, a raíz de la publicación en 1975 de su libro Adaptation in Natural and Artificial Systems [29] se conoce principalmente con el nombre de Algoritmos Genéticos.

La Programación Evolutiva surge principalmente a raíz del trabajo Artificial Intelligence Through Simulated Evolution de Fogel, Owens y Walsh, publicado en 1966 [30]. En este caso los individuos, conocidos aquí como organismos, son máquinas de estado finito. Los organismos que mejor resuelven alguna de las funciones objetivo obtienen la oportunidad de reproducirse. Antes de producirse los cruces para generar la descendencia se realiza una mutación sobre los padres.

3.3. Bases biológicas

En la naturaleza, los individuos de una población compiten constantemente con otros por recursos tales como comida, agua y refugio. Los individuos que tienen más éxito en la lucha por los recursos tienen mayores probabilidades de sobrevivir y generalmente una descendencia mayor. Al contrario, los individuos peor adaptados tienen un menor número de descendientes, o incluso ninguno. Esto implica que los genes de los individuos mejor adaptados se propagarán a un número cada vez mayor de individuos de las sucesivas generaciones.

La combinación de características buenas de diferentes ancestros puede originar en ocasiones que la descendencia esté incluso mejor adaptada al medio que los padres. De esta manera, las especies evolucionan adaptándose más y más al medio a medida que transcurren las generaciones [31].

3.4. Operadores genéticos

Selección La selección se realiza por medio del método de Vasconcelos. Para aplicar este método necesitamos ordenar aptitud de todos los individuos, ascendente o descendentemente, y tomar el individuo más apto y el menos apto.

Cruza Se realiza la cruza a partir de dos puntos aleatorios (P1 y P2). De 0 a la posición del primer corte se obtenían los datos del Padre (el mejor individuo), del primer al segundo corte se obtenían los datos de la Madre (el peor individuo), del segundo corte y hasta terminar se agregaban los datos del Padre como se muestra en la Figura 1.

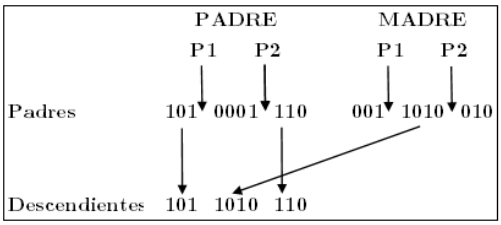


Fig. 1. Operador de cruce basado en un punto de corte.

Muta La muta de un individuo provoca que alguno de sus genes, generalmente uno sólo, varíe su valor de forma aleatoriase. La muta se muestra en la Figura 2.

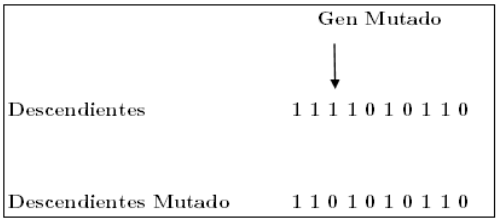


Fig. 2. Operador de mutación.

Elitismo Se clonan los mejores individuos de un porcentaje representativo de la población. Es decir, en caso de tener una población de 100 individuos, y un elitismo del 0.20, pasará a los 20 mejores individuos en espacios aleatorios.

3.5. Algoritmo genético simple

Los Algoritmos Genéticos trabajan sobre una población de individuos. Cada uno de ellos representa una posible solución al problema que se desea resolver.Todo individuo tiene asociado un valor de acuerdo a la aptitud con respecto al problema de la solución que representa (en la naturaleza el equivalente sería una medida de la eficiencia del individuo en la lucha por los recursos). El funcionamiento genérico de un Algoritmo Genético puede apreciarse en el siguiente pseudocódigo:

Algoritmo 1 Algoritmo genético simple.

```
Inicializar una población aleatoria
mientras No se cumpla el criterio de terminación hacer
    Crear población temporal vacía
    mientras población temporal no llena hacer
        Seleccionar padres
        Cruzar padres con probabilidad Pc
        si Se ha producido el cruce entonces
            Mutar uno de los descendientes con probabilidad Pm
            Evaluar descendientes
            Añadir descendientes a la población temporal
        si no
            Añadir padres a la población temporal
        fin si
    fin mientras
    Aumentar contador generaciones
    Establecer como nueva población actual la población temporal
fin mientras
```

3.6. Algoritmo genético con elitismo

El funcionamiento de un Algoritmo Genético con la inclusión de elitismo puede apreciarse en el siguiente pseudocódigo:

Algoritmo 2 Algoritmo genético con elitismo.

```
Inicializar una población aleatoria
mientras No se cumpla el criterio de terminación hacer
    Crear población temporal vacía
    mientras población temporal no llena hacer
        Realizar proceso de elitismo, según el porcentaje indicado
        Seleccionar padres
        Cruzar padres con probabilidad Pc
        si Se ha producido el cruce entonces
            Mutar uno de los descendientes con probabilidad Pm
            Evaluar descendientes
            Añadir descendientes a la población temporal
        si no
            Añadir padres a la población temporal
        fin si
        Agregar intensificador
    fin mientras
    Aumentar contador generaciones
    Establecer como nueva población actual la población temporal
fin mientras
```

4. Datos de prueba

El conjunto de datos experimentales proviene de la bases de datos del sitio Physionet. Los registros fueron obtenidos del conjunto MIT/BIH arrhytmia. Este contiene 48 registros de 30 minutos de duración [1]. Todos sus los registros fueron etiquetados.

El estándar de la Association for the Advancement of Medical Instrumentation (AAMI) [32] recomienda las siguientes consideraciones para etiquetar los tipos de arritmias existentes: ritmo normal (etiquetado como N), latido ectópico supraventricular (S), latido ectópico ventricular (V), latido de fusión (F), y latidos desconocidos (Q). Cualquier registro debería estar representado en alguno de estos tipos. Una descripción completa de todos los registros se incluyen en la Tabla 1, mostrando la equivalencia entre el AAMI y el MIT/BIH.

Extracción de características Tomamos como referencia el trabajo de [33] para realizar la extracción de características usadas en el conjunto de datos del MIT. El vector de características de entrada $x_j = \{x_{ij}, x_{2j}, \ldots, x_{pj}\}$, con $p = 100$ está compuesto por:
$x_{1j} = l_j - l_{j-1}$ (intervalo RR),
$x_{2j} = l_{j-1} - l_{j-2}$ (pre-intervalo RR),
$x_{3j} = l_{j+1} - l_j$ (post-intervalo RR),
$x_{4j} = x_{1j} - x_{2j}$,
$x_{5j} = x_{3j} - x_{1j}$,
$x_{6j} = (\frac{x_{3j}}{x_{1j}})^2 + (\frac{x_{2j}}{x_{1j}})^2 - (\frac{1}{3}\sum_{k=1}^{3} x_{kj}^2 log(x_{kj}^2))$.

Representación PQRST En x_{7j} se cuantifica la disimilitud entre el actual complejo QRS, y la linealidad acumulada de los últimos 10 complejos QRS [34] por medio de un enfoque de distorsión de tiempo dinámico (DTW).
$x_{8j} = \mid \frac{max\{QRS_j[t]\}}{min\{QRS_j[t]\}} \mid$,
$x_{9j} = \sum_{k=0}^{L_j} QRS_j[t]^2$.

Coeficientes de Hermite De x_{10j} a x_{19j} correspondiente a los primeros 10 polinomios de Hermite, usando el software de MATLAB.

Wavelet Daubechies Se utilizó el 4to. nivel de los coeficientes (A-Aproximación, D-Detalle) de la descomposición Wavelet Daubechies, (Db4)
De x_{20j} a $x_{25j} \Leftarrow A_4$,
De x_{26j} a $x_{31j} \Leftarrow D_4$,
De x_{32j} a $x_{42j} \Leftarrow D_3$,
De x_{43j} a $x_{58j} \Leftarrow D_2$,

De x_{59j} a $x_{90j} \Leftarrow D_1$,
De x_{91j} a $x_{95j} = var\{A_4, D_4, D_3, D_2, D_1\}$,
De x_{95j} a $x_{100j} = max\{A_4, D_4, D_3, D_2, D_1\}$.

5. Definición de la función objetivo

La función objetivo se definió utilizando diversos clasificadores, se ingresaron los vectores característicos hasta obtener el porcentaje de clasificación mayor (siendo 100 el máximo) y el menor número de carácterísticas del vector.

5.1. Naive Bayes

Se decidió utilizar la función del clasificador Naive Bayes que viene contenida en el software *Weka*, de esta forma tendríamos datos reales y probados.

5.2. KNN

Se aplicó la función del clasificador KNN que viene contenida en el software *Weka*, fue utilizada para 1 y 3 vecinos.

6. Implementación de las metaheurísticas

6.1. Algoritmos genéticos

Los operadores genéticos fueron utilizados de la siguiente manera:

Selección. La selección se realizó a través del método Vasconcelos. Para aplicar este método necesitamos ordenar el fitness de todos los individuos, ascendente o descendentemente, y tomar el mejor y el peor individuo.

Cruza. Realizamos la cruza a partir de dos puntos aleatorios. Se toma de la cadena del Padre (mejor individuo) desde la posición 0 hasta el primer punto aleatorio, desde el primer punto aleatorio hasta el segundo punto aleatorio se toma de la cadena de datos de la Madre (peor individuo), y finalmente desde el segundo punto hasta terminar la cadena se obtiene del Padre.

Muta. Se muta un porcentaje de la población, se toma un dato de la cadena de manera aleatoria y se cambia su valor de 0 a 1 o de 1 a 0.

Elitismo. Se toma el mejor individuo y se clona un porcentaje de veces.

Algoritmo genético simple para la clasificación de ECG Para realizar el proceso del Algoritmo Genético se utilizó el Algoritmo 3.5.

Algoritmo genético con elitismo para la clasificación de ECG Para realizar el proceso del Algoritmo Genético con elitismo se implementó el Algoritmo 3.6.

7. Pruebas y resultados

Las primeras pruebas se realizaron con los clasificadores 1NN, 3NN y Naive Bayes, sin metaheurísticas. Consideramos todas las características para tener un parámetro inicial y poder comparar. Después, probamos con los mismos clasificadores, pero utlizando los algoritmos genéticos y algoritmos genéticos con elitismo.

Se puede ver la Media del Porcentaje de Clasificación obtenido (ver la Tabla 2) y la Media del Número de Características empleadas en la clasificación (ver la Tabla 3).

8. Conclusiones

Los resultados fueron satisfactorios al implementar las metaheurísticas como optimización en la clasificación de arritmias cardiacas en señales de ECG. Como consecuencia de la reducción de características permite eliminar información innecesaria, reduciendo el ruido y mejorando el desempeño de los clasificadores.

Aplicando los clasificadores KNN y Naive Bayes y las metaheurísticas Algoritmos Genéticos y Algoritmos Genéticos con Elitismo a la Base de Datos MIT-BIH como auxiliares en la clasificación de arritmias cardiacas, el porcentaje de clasificación aumentó hasta en un 79.76 %, en el caso de las características se logró reducir la cantidad hasta en un 67 %. La reducción de la información innecesaria, quita el ruido en la clasificación, y hace que el porcentaje de la clasificación sea más alto y por tanto la clasificación sea mucho más óptima.

Referencias

1. Goldberger, A.L., Amaral, L.A.N., Glass, L., Hausdorff, J.M., Ivanov, P.Ch., Mark, R.G., Mietus, J.E., Moody, G.B., Peng, C.-K., Stanley, H.E.: PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. Circulation 101(23):e215-e220 (June 2000)
2. MedlinePlus Biblioteca Nacional de Medicina de los EE.UU. (2014)
3. De Chazal, P., O'Dwyer, M., Reilly, R. B.: Automatic classification of heartbeats using ECG morphology and heartbeat interval features. IEEE Transactions on Biomedical Engineering 51, pp. 1196–1206 (2004)
4. Organización Mundial de la Salud: Enfermedades cardiovasculares (2015)
5. Fuentes, C. R., Ordoñez, H. F. J.: Capa de sensado de un nodo para monitoreo de signos vitales. Ingenium (2010)
6. Médica Sur: Mexicanos desarrollan dispositivo portátil que detecta arritmias cardiacas en tiempo real. Instituto Carlos Slim de la Salud (2016)
7. Arévalo, A. M.: Electrocardiografía ECG DALCAME. Grupo de Investigación Biomédica (2005)
8. Abdeel-Badeeh, M., Revett, K., et al.: Machine Learning in Electrocardiogram Diagnosis. In: International Multiconference on Computer Science and Information Technology, pp. 429–433 (2009)

9. Nait-Hamoud, M.: Two Novel Methods for Multiclass ECG Arrhythmias Classification Based on PCA. In: Fuzzy Support Vector Machine and Unbalanced Clustering IEEE, pp. 140–145 (2010)
10. Thanapatay, D., Suwansaroj, C.: ECG Beat Classification Method for ECG Printout with Principle Components Analysis and Support Vector Machines. In: International Conference on Electronics and Information Engineering (ICEIC), 1, pp. 72–75 (2010)
11. Nait-Hamoud, M. C., Moussaoui, A.: Two novel methods for multiclass ECG arrhythmias classification based on PCA, fuzzy support vector machine and unbalanced clustering. In: Machine and Web Intelligence (ICMWI), 2010 International Conference on, pp. 140–145 (2010)
12. Molina, F. et al.: Desarrollo e Implementación de un Algoritmo Para la Caracterización de los Límites de Forma de Onda de Un Electrocardiograma (ECG) Utilizando Ondillas (Wavelets). En: XIX Jornadas en Ingeniería Eléctrica y Electrónica, 19, pp. 95–105 (2005)
13. Gutiérrez, A., Lara, M.: Evaluación de un Detector de Complejo QRS Basado en la Wavelet de Haar, Usando las Bases de Datos MIT-BIH de Arritmias y Europea del Segmento ST y de la Onda T. Computación y Sistemas, 8, pp. 293–302 (2005)
14. Fira, M., Goras, L., et al.: On the Projection Matrices Influence in the Classification of Compressed Sensed ECG Signals. International Journal of Advanced Computer Science and Applications (2012)
15. Martínez, A., Rojas, I., et al.: Utilización de Sistemas Inteligentes para la Detección de Problemas de Corazón Mediante ECG. (2005)
16. Sannino, G., Pietro, G.: An Advanced Mobile System for Indoor Patients Monitoring In: Proc. 2nd International Conference on Networking and Information Technology (ICNIT) (2011)
17. Elena, M., Quero, J., Borrego, I.: An optimal technique for ECG noise reduction in real time applications. Computers in Cardiology, pp. 225–228 (2006)
18. Kim, H., Yazicioglu, R. F., Merken, P., Van Hoof, C., Yoo, H.-J.: ECG signal compression and classification algorithm with quad level vector for ECG holter system. IEEE Transactions on Information Technology in Biomedicine, IEEE, 14, pp. 93–100 (2010)
19. Bilgin, A., Marcellin, M. W., Altbach, M. I.: Compression of electrocardiogram signals using JPEG2000. IEEE Transactions on Consumer Electronics, 49, pp. 833–840 (2003)
20. Cuesta-Frau D., et al.: Feature Extraction Methods Applied to the Clustering of Electrocardiographic Signals. A Comparative Study. In: 16th International Conference on Pattern Recognition, pp. 961–964 (2002)
21. Kallas, M., Francis, C., et al.: Multi-Class SVM Classification Combined with Kernel PCA Feature Extraction of ECG Signals. In: International Conference on Telecomunications (2012)
22. Soman, T., et al.: Classification of Arrhythmia Using Machine Learning Techniques. In: ICOSSE (2005)
23. Gao, D., Madden M., et al.: Bayesian ANN Classifier for ECG Arrhythmia Diagnostic System: A Comparison Study. In: IJCNN (2005)
24. Daamouche, A., Hamami, L., et al.: A Wavelet Optimization Approach for ECG Signal Classification. Biomedical Signal Processing and Control, 7, pp. 342–349 (2012)
25. Hoffman, G. S., Miller, M., Kabrisky, M., Maybeck, P., Raquet, J.: Novel electrocardiogram segmentation algorithm using a multiple model adaptive estimator.

In: Decision and Control, Proceedings of the 41st IEEE Conference on, 3, pp. 2524–2529 (2002)
26. Vega, J. M. M., Batista, M. B. M., Pérez, J. A. M.: Metaheuristicas: Una visión global. Inteligencia artificial: Revista Iberoamericana de Inteligencia Artificial, Asociación Española para la Inteligencia Artificial (AEPIA), 7, pp. 7–28 (2002)
27. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Longman Publishing Co. (1989)
28. Darwin, Ch.: On the Origin of Species by Means of Natural Selection. John Murray (1859)
29. Holland, J.: Adaptation in Natural and Artificial Systems. (1975)
30. Fogel, L., Owens, A., et al.: Artificial Intelligence through Simulated Evolution. (1966)
31. Beasley, D., Bull, D. R., et al.: An Overview of Genetic Algorithms: Part 1, Fundamentals University computing. pp. 58–69 (1993)
32. De Chazal, P., O'Dwyer, M., et al.: Automatic Classification of Heartbeats Using ECG Morphology and Heartbeat Interval Features. IEEE Transactions on Biomedical Engineering (2004)
33. Rodríguez-Sotelo, J., Peluffo, D., et al.: Unsupervised Feature Relevance Analysis Applied to Improved ECG Heartbeat Clustering. (2012)
34. Sornmo, L. P.: Bioelectrical Signal Processing In Cardiac And Neurological Applications. Academic Press Series in Biomedical Engineering (2005)

Tabla 1. Conjunto de registros de la base de datos MIT/BIH utilizados en los experimentos.

AAMI	N					S				V		F		Q	
Código	N	L	R	e	j	A	a	J	S	V	E	F	f	P	Q
MIT	1	2	3	34	11	8	4	7	9	5	10	6	38	12	13
Clase	0000	1010	0111	1011	0101	0010	1000	0110	1100	0001	1101	1001	0100	0011	1110
100	2237					33				1					
101	1858					3									2
102	99									4			56	2026	
103	2080					2									
104	163									2			666	1378	18
105	2524									41					5
106	1505									520					
107										59				2076	
108	1738				1	4				16		2			
109		2490								38		2			
111		2121								1					
112	2535					2									
113	1787						6								
114	1818					10		2		43		4			
115	1951														
116	2300					1				109					
117	1532					1									
118			2164			96				16					
119	1541									444					
121	1859					1				1					
122	2474														
123	1513									3					
124			1529		5	2		29		47		5			
200	1742					30				825		2			
201	1623				10	30	97	1		198		2			
202	2059					36	19			19		1			
203	2527						2			444		1			4
205	2569					3				71		11			
207		1457	85			106				105	105				
208	1585								2	992		372			2
209	2619					383				1					
210	2421						22			194	1	10			
212	922		1824												
213	2639					25	3			220		362			
214		2001								256		1			2
215	3194					2				164		1			
217	244									162			260	1540	
219	2080					7				64		1			
220	1952					94									
221	2029									396					
222	2060				212	208		1							
223	2027			16		72	1			473		14			
228	1686					3				362					
230	2253									1					
231	314		1252			1				2					
232			396		1	1381									
233	2229					7				830		11			
234	2698							50		3					
TOTAL	74986	8069	7250	16	229	2543	150	83	2	7127	106	802	982	7020	33

La primer fila corresponde a las etiquetas utilizadas de acuerdo al estándar AAMI.
La segunda fila corresponde a las etiquetas utilizadas en la base de datos MIT/BIH.
La tercer fila corresponde al código numérico de las etiquetas anteriores..
La primer columna es el nombre de los registros, mientras que los otros contienen el número de latidos del corazón de cada tipo.

Tabla 2. Media del porcentaje de clasificación.

Clasificador	Población	Media de clasificación	Capas	Llamadas a función
Linea de base				
1NN	-	94.11 %	2	-
3NN	-	94.71 %	2	-
Naive Bayes	-	9.72 %	2	-
Naive Bayes	-	9.93 %	5	-
Naive Bayes	-	9.93 %	10	-
Algoritmos Genéticos				
1NN	5	93.38 %	2	20
3NN	5	93.78 %	2	20
Naive Bayes	5	14.42 %	2	20
Naive Bayes	5	15.59 %	5	20
Naive Bayes	10	15.75 %	10	50
Algoritmos Genéticos con Elitismo				
1NN	5	93.81 %	2	20
3NN	5	94.32 %	2	20
Naive Bayes	5	16.01 %	2	20
Naive Bayes	5	17.46 %	5	20
Naive Bayes	10	17.85 %	10	50

Tabla 3. Media de las características en la clasificación (la línea base implica 100 características en todos los casos).

Clasificador	Población	Media de las características	Capas	Llamadas a función
Algoritmos Genéticos				
1NN	5	49	2	20
3NN	5	52	2	20
Naive Bayes	5	41	2	20
Naive Bayes	5	34	5	20
Naive Bayes	10	45	10	50
Algoritmos Genéticos con Elitismo				
1NN	5	46	2	20
3NN	5	51	2	20
Naive Bayes	5	40	2	20
Naive Bayes	5	33	5	20
Naive Bayes	10	43	10	50

ISSN 1870-4069

Comparativo empírico de medidas de diversidad en problemas de optimización evolutiva con restricciones

Luis Enrique Contreras-Varela, Efrén Mezura-Montes

Universidad Veracruzana, Centro de Investigación en Inteligencia Artificial, Xalapa, Veracruz, México

luisenrique.contreras.v@gmail.com, emezura@uv.mx

Resumen. La convergencia prematura es uno de los problemas centrales de los algoritmos evolutivos y su principal origen reside en la falta de diversidad en la población. Para monitorerar la diversidad es necesario aplicar medidas especiales, las cuales describen la dispersión en el espacio de las soluciones o de la función objetivo. Se han propuesto diferentes medidas de diversidad para optimización sin restricciones y su habilidad para medir diversidad al resolver problemas restringidos no ha sido estudiada exhaustivamente. En este trabajo se presenta un comparativo empírico del desempeño de seis medidas de diversidad aplicadas a dos algoritmos del estado del arte al resolver problemas de optimización con restricciones de la literatura especializada. Los resultados preliminares indican que las medidas de diversidad basadas en distancia entre pares de individuos tienen mejores resultados que el resto de medidas evaluadas.

Palabras clave: Algoritmos evolutivos, medidas de diversidad, optimización con restricciones.

Empirical Comparison of Diversity Measures in Constrained Optimization Problems

Abstract. Premature convergence is one of the central problems in evolutionary algorithms, and its main reason consists in the lack of diversity. In order to monitor diversity, special measures must be employed, which describe the population's spread over the search space or according to the fitness values. Some diversity measures have been proposed for unconstrained optimization and their capability to measure diversity when dealing with constrained problems has not been comprehensively studied. This paper presents an empiric comparison of six diversity measures applied to two state-of-the-art algorithms to solve constrained optimization problems from the literature. Preliminary results show that distance-based measures have the best results among the evaluated measures.

Keywords: Evolutionary algorithms, diversity measures, constrained optimization.

1. Introducción

Los algoritmos bio-inspirados son ampliamente utilizados para la optimización, al ser aplicables en una extensa gama de problemas y otorgar buenos resultados en un tiempo aceptable [6]. Estas técnicas suelen concluir su ejecución cuando ocurre el fenómeno llamado convergencia, donde los operadores de variación pierden la capacidad de generar nuevos individuos haciendo que todos los descendientes se encuentren en un subconjunto muy pequeño del espacio de búsqueda.

Si la convergencia se presenta antes de haber encontrado el óptimo global de la función se cataloga como convergencia prematura [12]. La convergencia prematura es uno de los principales problemas de los algoritmos evolutivos [5] y su principal origen reside en la falta de diversidad en la población, pues una vez que la población se ha tornado redundante el algoritmo toma una única trayectoria hacia el mejor individuo sin que éste sea necesariamente el óptimo [8].

La diversidad puede ser vista como un indicador de la disimilitud entre los individuos [8], una medida que aproxima el número de diferentes soluciones en la población en determinado momento [6].

Es evidente que la diversidad responde a la pregunta ¿Qué tan diferentes son los individuos de la población? Pero esta diferencia entre soluciones puede expresarse en distintas granularidades. De tal suerte, existen medidas de diversidad para los dos espacios en los que trabajan los algoritmos bio-inspirados, el espacio genotípico GDMs y el fenotípico PDMs [9]. Las GDMs están relacionadas con la ubicación de las soluciones en el espacio de búsqueda, mientras que las PDMs trabajan con los valores de aptitud de las soluciones [9].

Existen diversos trabajos relacionados con la diversidad, ya sea propiciándola en la población; o bien, aplicando medidas de diversidad para guiar la búsqueda del algoritmo o para el control de parámetros. Estos trabajos emplean algún método para medir diversidad. Para asegurarse que se está incrementando la diversidad en la población, mecanismos de promoción de la diversidad como el nitching no son suficientes debido a que se requiere conocer el paisaje de aptitud *a priori*. De modo que las medidas de diversidad son preferibles por ser métodos directos de sensar la diversidad [3]. A pesar de ello la habilidad de las distintas medidas reportadas en la literatura especializada para describir fielmente la diversidad no ha sido investigada exhaustivamente [3].

Para paliar esta situación, en los últimos años se han realizado nuevos estudios que comparan el desempeño de las diferentes medidas de diversidad, sean GDMs o PDMs, en diversos escenarios. Por ejemplo, en [14] Olorunda y Engelbrecht analizaron 6 diferentes GDMs aplicadas al algoritmo de cúmulo de partículas (PSO por sus siglas en inglés), en su trabajo encontraron que las medidas que responden mejor son la distancia promedio alrededor del centro del

cúmulo y la distancia alrededor de todas las partículas. Otro estudio muy interesante es el de Corriveau et al. [2], ellos hicieron una recopilación de las GDMs propuestas en el área y las emplearon en dos experimentos: 1) Un comparativo con algoritmos del estado del arte resolviendo el benchmark CEC 2005. 2) La evaluación del desempeño de las medidas en un simulador de convergencia de su autoría.

Un tipo de problema en el que no se ha profundizado el estudio de diversidad es la optimización con restricciones. La diversidad es importante, pues como menciona Deb, es conveniente mantener la pluralidad entre las soluciones y así permitir a los operadores de cruza encontrar constantemente mejores soluciones factibles [4].

Hasta nuestro conocimiento, no existen medidas de diversidad ni estudios del desempeño de las medidas existentes enfocados específicamente en problemas de optimización con restricciones. Por este motivo, sería deseable evaluar el comportamiento de las medidas de diversidad existentes en la literatura en algoritmos de distinta naturaleza al resolver problemas restringidos, en busca de obtener un acercamiento a la diversidad en optimización con restricciones.

La contribución de este trabajo responde a tal necesidad, y consiste en un comparativo empírico del desempeño de medidas de diversidad en algoritmos del estado del arte para optimización con restricciones.

El documento se divide de la manera siguiente: en la Sección 2 expone el problema, optimización con restricciones y la aplicación de medidas de diversidad en este tipo de problemas. La Sección 3 describe la contribución, un comparativo empírico del desempeño de medidas de diversidad en algoritmos del estado del arte, en la Sección 4 se plantea el diseño experimental y se presentan los resultados. Finalmente la Sección 5 resume los hallazgos y esboza el trabajo futuro.

2. Definición del problema

La optimización numérica con restricciones se define como:

$$\begin{aligned} &\text{encontrar } \boldsymbol{x} \text{ que optimice } f(x) \text{ sujeto a} \\ &g_i(\boldsymbol{x}) \leq 0, i = 1, ..., n, \\ &h_j(\boldsymbol{x}) = 0, j = 1, ..., p, \end{aligned} \tag{1}$$

donde $\boldsymbol{x}$ es el vector de las soluciones $\boldsymbol{x} = [x_1, x_2, ..., X_\tau]^\tau$, n el número de restricciones de desigualdad y p el de restricciones de igualdad. La zona factible F es aquella que donde todas las restricciones se satisfacen y S es el espacio total de búsqueda, por lo que $F \subseteq S$ [1]. Para resolver este tipo de problemas es necesario un algoritmo que soporte restricciones. Existen múltiples manejadores de restricciones reportados en la literatura especializada, como la superioridad de puntos, las funciones de castigo, etc.

Es importante resaltar que no existen medidas de diversidad específicas para problemas restringidos, hasta el momento si se desea conocer la diversidad al

resolver un problema de este tipo es necesario utilizar alguna de las medidas existentes, las cuales fueron pensadas para optimización sin restricciones.

Las GDMs son aplicables ya que solo requieren de la posición de cada individuo en la población, pero las PDMs tendrían un comportamiento poco informativo pues éstas, al haber sido diseñadas para problemas sin restricciones, asumen que el valor de la función objetivo de la mejor solución solo puede mejorar o permanecer en cada iteración del proceso de optimización. Lo anterior no sucede necesariamente en la optimización de problemas restringidos, donde el valor de la función objetivo de la mejor solución puede incrementarse o decrementarse si aún no se han satisfecho las restricciones (la dinámica dependerá de la técnica de manejo de restricciones seleccionada).

Debido a los problemas que presentan las medidas del espacio fenotípico este trabajo únicamente contempla GDMs para el comparativo.

La idea central tras las GDMs es representar la diversidad en términos de distancia Euclidiana entre individuos. Para representación real la mayoría de GDMs están basadas en distancia entre pares de individuos o la distancia máxima entre dos individuos de la población. Otro elemento importante de las medidas de diversidad es la normalización, la cual puede realizarse calculando la distancia de la diagonal del paisaje LD (distancia entre los extremos del espacio de búsqueda) o usando el valor de diversidad de la primera iteración $NMDF$ [2].

3. Contribución

Este trabajo presenta un comparativo empírico del desempeño de distintas medidas de diversidad aplicadas en dos algoritmos del estado del arte de distinta naturaleza al resolver problemas de optimización con restricciones. El objetivo de comparar la diversidad en algoritmos de diferentes familias es no sesgar el estudio a causa de las particularidades de los mismos, pues algunos comportamientos pueden darse gracias a los mecanismos específicos de un tipo de algoritmo.

Los algoritmos seleccionados para el estudio son CMODE [16] y SAM-PSO [7], los cuales se describen en el siguiente apartado.

3.1. Algoritmos del estado del arte empleados

CMODE [16] (Combining Multiobjective Optimization with Differential Evolution): Es un algoritmo basado en Evolución Diferencial cuya manera de lidiar con las restricciones es transformar el problema original en uno multi-objetivo. El nuevo problema tendrá dos objetivos a optimizar: la suma de violación de restricciones y la función objetivo.

En cada iteración se generan γ descendientes R, si el descendiente $\boldsymbol{x_i} \in R$ domina $n > 1$ del conjunto Q de padres $\boldsymbol{x_i}$ sustituirá a uno de éstos seleccionado aleatoriamente. En caso de no producirse ninguna solución que domine al menos un individuo del conjunto Q de padres el mejor descendiente no factible $\boldsymbol{x'}$ se incluye en el archivo de soluciones no factibles A. Cada k generaciones los individuos del archivo sustituyen al mismo número de soluciones en la población,

excluyendo del proceso a la mejor solución para que el porcentaje de factibilidad no se pierda.

Es importante resaltar que CMODE es un algoritmo de estado uniforme, gracias a la necesidad de tener soluciones no dominadas para que se dé el reemplazo de individuos y de que se cumplan las k generaciones necesarias para incluir soluciones no factibles, por lo que no necesariamente existirán cambios en la población en cada iteración.

SAM-PSO [7] (Self-adaptive mix of particle swarm methodologies):

Este algoritmo es una extensión a la optimización por cúmulo de partículas (PSO por sus siglas en inglés) que aplica diferentes estrategias simultáneamente durante el proceso de optimización y auto-adapta los parámetros de cada individuo. SAM-PSO divide la población de acuerdo al número de estrategias seleccionadas, progresivamente las proporciones de cada sub-cúmulo cambiarán en favor de la estrategia con mejores resultados. La implementación de los autores, al igual que la de este trabajo, incluye las siguientes variantes de PSO:

- *PSO local.* Trabaja de la misma manera que la variante de PSO original (PSO global) excepto que cada partícula es atraída por la mejor posición obtenida por los miembros de su vecindario local y no por la mejor posición de toda la población.
- *PSO con archivo.* La idea detrás de esta variante es conservar un archivo de soluciones pasadas para que exista un balance entre intensificación y diversificación de la búsqueda. Debido a que el archivo está formado por buenas soluciones e individuos cercanos al vector promedio, éste apoya tanto a la explotación de zonas prometedoras como a la variación de la población.
- *PSO sub-cúmulo* [10]. Esta estrategia divide la población en sub-cúmulos, y en cada uno de ellos la mejor partícula es identificada como el líder ese grupo. Los individuos no líderes de cada sub-cúmulo son actualizados escogiendo como g_{best} la mejor posición de una partícula aleatoria de su sub-cúmulo. Para el caso de los líderes, éstos evolucionarán usando la información de la mejor posición de otro líder elegido de manera aleatoria.

Adicionalmente, SAM-PSO añade mutación para prevenir convergencia prematura; problema importante en PSO, donde gracias a su rápida convergencia se incrementan las posibilidades de caer en óptimos locales.

3.2. Medidas de diversidad

En este documento se muestra un comparativo de algunas de las medidas de diversidad publicadas, en concreto se aplicaron seis medidas de diversidad en el espacio de las variables basadas en distancia: D^N_{DP}, D^{N2}_{TAP}, D^N_{TD}, D^N_{MI}, D^N_{PW} y D^N_{ED}. En la tabla 1 se listan las medidas y sus correspondientes formulaciones.

La selección de dichas medidas de diversidad tiene una intención similar a la del trabajo de Corriveau et al. [2], contrastar las medidas de diversidad más robustas contra las que mayores dificultades presentan, aunque en este caso aplicadas a optimización con restricciones. En su trabajo catalogan a D^N_{DP} y

D^N_{ED} como malos descriptores de diversidad mientras que D^{N2}_{TAP}, D^N_{TD}, D^N_{MI} y D^N_{PW} son algunas de las medidas con mejores resultados en sus experimentos sobre problemas sin restricciones.

Tabla 1. Medidas de diversidad utilizadas en el comparativo.

Número	Medida	Fórmula
1	Diámetro de la población	$D^N_{DP} = \frac{1}{LD} \max_{i \neq j \in (1,2,\ldots,N)} \sqrt{\sum_{k=1}^{n} (x_{i,k} - x_{j,k})^2}$
2	Distancia al punto promedio	$D^{N2}_{TAP} = \frac{\frac{1}{N} \sum_{i=1}^{N} \sqrt{\sum_{k=1}^{n} (x_{i,k} - \bar{x}_k)^2}}{NMDF}$
3	Momento de inercia	$D^N_{MI} = \frac{\sum_{k=1}^{n} \sum_{i=1}^{N} (x_{i,k} - \bar{x}_k)^2}{NMDF}$
4	Diversidad verdadera	$D^N_{TD} = \frac{\frac{1}{n} \sqrt{\sum_{k=1}^{n} (\bar{x^2}_k - \bar{x}_k)^2}}{NMDF}$ $\bar{x^2} = \frac{1}{N} \sum_{i=1}^{N} x^2_{i,k}$
5	Media de la distancia entre pares	$D^N_{PW} = \frac{\frac{2}{N(N-1)} \sum_{i=2}^{N} \sum_{j=1}^{i-1} \sqrt{\sum_{k=1}^{n} (x_{i,k} - x_{j,k})^2}}{NMDF}$
6	Distancia Euclidiana al mejor individuo	$D^N_{ED} = \frac{\bar{d} - d_{min}}{d_{max} - d_{min}}$ $\bar{d} = \frac{1}{N} \sum_{i=1}^{N} \sqrt{\sum_{k=1}^{n} (x_{i,k} - x_{best,k})^2}$ $d_{max} = \max_{i \in (1,2,\ldots,N)} \sqrt{\sum_{k=1}^{n} (x_{i,k} - x_{best,k})^2}$ $d_{min} = \min_{i \in (1,2,\ldots,N)} \sqrt{\sum_{k=1}^{n} (x_{i,k} - x_{best,k})^2}$

La primera medida, D^N_{DP}, lleva por nombre diámetro de la población [14], ella interpreta la diversidad usando la distancia entre los dos individuos más alejados de la población y normalizando los valores con la diagonal del espacio de búsqueda (o LD, acrónimo de *landscape diagonal*, la cual se obtiene computando la distancia entre los puntos formados por los límites inferiores y superiores del espacio de búsqueda). El no contemplar la dispersión de la población y unicamente ofrecer un valor con base en los extremos puede causar que no se describa correctamente la diversidad. Por otro lado utilizar LD para normalizar hace que la medida sea sensible a la dimensionalidad, pues las distancias se incrementan conforme aumenta el número de dimensiones [2].

En seguida se encuentra la distancia al punto promedio D^{N2}_{TAP} [15], la cual consiste en la media de las distancias de cada individuo de la población al vector promedio $\bar{\boldsymbol{x}}$. En esta medida la normalización se realiza mediante el valor máximo de diversidad hasta el momento (abreviado como $NMDF$), el cual comúnmente es el valor de diversidad obtenido en la primera generación; pues supone ser el momento en que la población se encuentra más dispersa.

Momento de inercia D^N_{MI} [13], la medida número 3, tiene el mismo principio que la distancia al punto promedio D^{N2}_{TAP}, aunque D^N_{MI} eleva al cuadrado las distancias haciendo más influyentes los valores atípicos.

Por otro lado, diversidad verdadera D^N_{TD}, interpreta la diversidad como la desviación estándar en cada dimensión de la población.

La quinta GDM es la media de la distancia entre pares D^N_{PW}. Esta intuitiva medida requiere que se calcule la distancia entre todas las combinaciones de pares de individuos, gracias a esto D^N_{PW} es la medida más completa aunque como consecuencia el cómputo se vuelve muy pesado.

La normalización de D_{MI}^{N}, D_{TD}^{N} y D_{PW}^{N}, al igual que en D_{TAP}^{N} se efectúa dividiendo el valor de diversidad obtenido en la iteración entre $NMDF$.

La última medida de diversidad considerada en el comparativo es D_{ED}^{N}. Como es evidente al observar su fórmula, D_{ED}^{N} difiere bastante en relación con las demás medidas presentadas ya que requiere conocer de antemano la mejor solución de la población en la generación. Nótese que la normalización se está llevando a cabo usando la diferencia de la distancias del individuo más alejado y más cercano al mejor, conforme avancen las generaciones dichas distancias decrecerán haciendo incapaz a la función de rastrear la convergencia de la población [2].

4. Experimentos y resultados

4.1. Diseño experimental

Se realizaron 25 ejecuciones de los algoritmos CMODE y SAM-PSO solucionando los problemas de prueba del benchmark CEC 2006 [11]. En este documento se presentan los resultados correspondientes a los primeros 13 problemas. En ambos algoritmos se conservaron los parámetros reportados en la experimentación de su correspondiente publicación, a excepción del tamaño poblacional y del número de evaluaciones permitidas; en ambos algoritmos se asignó un tamaño de población $NP = 180$ y número máximo de evaluaciones $FES = 500000$ (evaluaciones que propone el benchmark CEC 2006).

En ambos algoritmos por cada problema se tomó la ejecución con el resultado final ubicado en la mediana para graficar la diversidad con cada una de las medidas de la tabla 1, por cuestiones de espacio en el documento se muestran únicamente las gráficas de diversidad de las funciones más significativas.

Se computó el valor de todas las GDMs cada $NP = 180$ evaluaciones realizadas por los algoritmos. En el caso de SAM-PSO las medidas de diversidad emplearon la posiciones actuales de las partículas, y no los mejores valores de cada individuo. La tabla 4.1 lista los parámetros de CMODE y SAM-PSO.

También se obtuvieron valores estadísticos (véase la tabla 3) para no evaluar únicamente los comportamientos de convergencia descritos por las medidas de diversidad, si no también establecer el contexto de los resultados de cada algoritmo y así relacionar las estadísticas de los resultados finales con las gráficas de diversidad para cada problema resuelto de acuerdo a los mecanismos de cada algoritmo.

4.2. Desempeño de los algoritmos

Respecto a los resultados finales de las ejecuciones de ambos algoritmos (la tabla 3 muestran las estadísticas) se puede observar que CMODE encuentra mejores soluciones que SAM-PSO. En cuanto a factibilidad, el primero consigue que las 25 ejecuciones lleguen a la zona factible en casi todos los problemas; la excepción son los problemas $g03$ y $g07$, donde sólo el 16 % y 80 % alcanzaron el área factible, respectivamente. En cambio, SAM-PSO únicamente logra que el

Tabla 2. Parámetros de CMODE y SAM-PSO en el experimento.

CMODE		SAM-PSO	
NP	180	NP	180
FES	500000	FES	500000
CP	rand(0.90,0.95)	τ	[0.4,0.729]
F	rand(0.50,0.60)	c_1	[1.4,2]
γ	8	c_2	[1.4,2]
k	22	Variantes	*PSO local, PSO con archivo, PSO sub-cúmulo*
Estrategia ED	$rand$	NP_{arch}	$\frac{NP}{2}$
		N (archivo)	$\frac{NP}{5}$
		sub-cúmulos	5

100 % de ejecuciones satisfaga las restricciones en las funciones en los problemas $g06$, $g08$ y $g12$. Aunque para problemas con un espacio de búsqueda pequeño SAM-PSO puede no tener individuos en la zona factible al finalizar la ejecución debido a la mutación, lo cual no significaría que las partículas no guardan una posición factible en su memoria.

Otro aspecto notorio en el desempeño de los algoritmos es la estabilidad que demuestra CMODE en relación a SAM-PSO, en la mayoría de las funciones del benchmark la desviación estándar ($\sigma 2$) en SAM-PSO es mucho mayor que en CMODE. Particularmente existen diferencias considerables de desviación estándar en las funciones $g01$, $g04$, $g05$, $g07$, $g08$, $g10$ y $g11$; en cambio, sólo en la función $g03$ la desviación estándar es mayor en CMODE que SAM-PSO.

En general se aprecia que CMODE es el que mejor desempeño tiene entre los dos algoritmos. El mejor resultado es superior al de SAM-PSO en los problemas $g02$, $g03$, $g05$, $g07$, $g09$ y $g10$, mientras que en el resto de problemas ambos descubren la misma solución.

4.3. Resultados del comparativo de diversidad

En las figuras 1, 2, 3 y 4 se presentan los resultados de diversidad de la mediana de las ejecuciones de los algoritmos CMODE y SAM-PSO. Se tienen dos gráficos por función, correspondientes a cada uno de los algoritmos, exhibiendo las curvas de diversidad producidas por las seis GDMs de la tabla 1 en los dos algoritmos. A continuación se discuten los resultados, resumiendo los hallazgos por medida de diversidad evaluada.

Diámetro de la población. D_{DP}^{N}, representada en las gráficas con el color azul oscuro, utiliza la distancia entre los dos individuos más separados de la población y usa como factor de normalización la diagonal del espacio de búsqueda. Debido a lo anterior se puede prever en cierto sentido el comportamiento, pues mientras el algoritmo se encuentre en periodo de exploración es probable que la distancia más larga entre dos soluciones oscile fuertemente y por lo tanto su curva de diversidad se vea escarpada. Tal es el caso de ambos algoritmos al resolver las funciones $g02$ y de $g03$ con CMODE o $g02$, $g03$, $g11$ y $g13$ para SAM-PSO. Este problema no se ve reflejado en las gráficas donde la convergencia se da tan rápido que los vectores más alejados están prácticamente en el foco de convergencia desde las primeras iteraciones.

g2 (500000 FES)

	CMODE	SAM-PSO
Best	-0.803623789327	-0.803365020017
Mean	-0.800686985414	-0.792155641674
Median	-0.803623667644	-0.788432576597
Worst	-0.785269658317	-
Std.	0.00539501789782	0.0088797657819
FP	1.0	0.68

g2 (500000 FES)

	CMODE	SAM-PSO
Best	-0.803623789327	-0.803365020017
Mean	-0.800686985414	-0.792155641674
Median	-0.803623667644	-0.788432576597
Worst	-0.785269658317	-
Std.	0.00539501789782	0.0088797657819
FP	1.0	0.68

Fig. 1. Resultados para el problema $g02$ con CMODE y SAM-PSO.

Existe una situación particular con cada uno de los algoritmos al aplicar esta medida de diversidad: SAM-PSO incluye mutación, lo que propicia la exploración pero también podría hacer que la distancia entre los individuos más separados crezca. En CMODE sucede algo similar gracias a la introducción de soluciones no factibles del archivo cuando el algoritmo no es capaz de encontrar soluciones no dominadas, fenómeno visible en las funciones $g08$, $g09$ y $g11$; cada vez que individuos del archivo se incluyen en la población se observa un salto muy grande de diversidad.

Distancia al punto promedio, momento de inercia, diversidad verdadera y media de la distancia entre pares. Este conjunto de GDMs se muestra en las gráficas con los colores verde, rojo, azul claro y morado. Como se comentó en secciones anteriores, las cuatro medidas toman en cuenta a todos los individuos para describir la diversidad de la población y todas normalizan los resultados con la diversidad calculada en la primera generación, la mejor forma de normalizar de acuerdo al estudio de Corriveau et al. [2].

A pesar de tener diferencias en sus formulaciones, D_{TAP}^{N2}, D_{MI}^{N}, D_{TD}^{N} y D_{PW}^{N} trazan curvas de diversidad muy similares e incluso en algunos problemas el comportamiento es casi idéntico; tal es el caso en casi todas las funciones evaluadas. Existen algunos casos donde, aunque el comportamiento es similar, las gráficas difieren un poco: para CMODE esto sucede con las funciones $g02$ y $g08$, donde claramente se aprecia que D_{MI}^{N} y D_{TD}^{N} se encuentran alineadas pero separadas de D_{TAP}^{N2} y D_{PW}^{N}, que a su vez delinean los mismos valores de diversidad. Una situación distinta a las anteriores ocurre en la función $g03$, problema con un espacio de búsqueda pequeño ($[0, 1]^{10}$); todas las medidas evaluadas indican estancamiento en la población aunque cada una sugiere una dispersión diferente en la población. D_{TD}^{N} y D_{MI}^{N} tienen los valores más bajos, lo cual es ocasionado por el tamaño del espacio de búsqueda, ya que D_{MI}^{N} eleva al cuadrado las distancias mientras que D_{TD}^{N} lo hace con la diferencia de la media de cada dimensión al alelo correspondiente del punto promedio. Debido a que la distancia más larga que puede existir en la población es LD, cuyo valor es de 3.1622, al elevar los valores a la segunda potencia estos se homogenizan,

g3 (500000 FES)

	CMODE	SAM-PSO
Best	-1.00100040008	-1.00085118529
Mean	-0.61500684236	-0.999079806126
Median	-0.646760402447	-0.989127804122
Worst	-	-
Std.	0.317001761452	0.00297753985669
FP	0.16	0.52

g3 (500000 FES)

	CMODE	SAM-PSO
Best	-1.00100040008	-1.00085118529
Mean	-0.61500684236	-0.999079806126
Median	-0.646760402447	-0.989127804122
Worst	-	-
Std.	0.317001761452	0.00297753985669
FP	0.16	0.52

g6 (500000 FES)

	CMODE	SAM-PSO
Best	-6961.93682853	-6961.93682853
Mean	-6961.93682853	-6961.93682853
Median	-6961.93682853	-6961.93682853
Worst	-	-6961.93682853
Std.	0.0	1.04492985821e-12
FP	0.68	1.0

g6 (500000 FES)

	CMODE	SAM-PSO
Best	-6961.93682853	-6961.93682853
Mean	-6961.93682853	-6961.93682853
Median	-6961.93682853	-6961.93682853
Worst	-	-6961.93682853
Std.	0.0	1.04492985821e-12
FP	0.68	1.0

g7 (500000 FES)

	CMODE	SAM-PSO
Best	24.3060374155	24.8912591955
Mean	24.3060374155	26.1580483849
Median	24.3060374155	26.0773741064
Worst	-	-
Std.	8.50225527548e-15	1.31049956133
FP	0.88	0.84

g7 (500000 FES)

	CMODE	SAM-PSO
Best	24.3060374155	24.8912591955
Mean	24.3060374155	26.1580483849
Median	24.3060374155	26.0773741064
Worst	-	-
Std.	8.50225527548e-15	1.31049956133
FP	0.88	0.84

g8 (500000 FES)

	CMODE	SAM-PSO
Best	-0.095825041418	-0.095825041418
Mean	-0.095825041418	-0.095825041418
Median	-0.095825041418	-0.095825041418
Worst	-	-0.095825041418
Std.	1.11632878849e-17	2.92423134397e-17
FP	0.68	1.0

g8 (500000 FES)

	CMODE	SAM-PSO
Best	-0.095825041418	-0.095825041418
Mean	-0.095825041418	-0.095825041418
Median	-0.095825041418	-0.095825041418
Worst	-	-0.095825041418
Std.	1.11632878849e-17	2.92423134397e-17
FP	0.68	1.0

Fig. 2. Resultados para problemas $g03$, $g06$, $g07$ y $g08$ con CMODE y SAM-PSO.

g9 (500000 FES)

	CMODE	SAM-PSO
Best	680.629943402	680.688628672
Mean	680.629943402	680.841567111
Median	680.629943402	680.825108765
Worst	680.629943402	-
Std.	9.3748568033 7e-14	0.230279784477
FP	1.0	0.8

g9 (500000 FES)

	CMODE	SAM-PSO
Best	680.629943402	680.688628672
Mean	680.629943402	680.841567111
Median	680.629943402	680.825108765
Worst	680.629943402	-
Std.	9.3748568033 7e-14	0.230279784477
FP	1.0	0.8

g11 (500000 FES)

	CMODE	SAM-PSO
Best	0.7498	0.749830118447
Mean	0.7498	0.749880399459
Median	0.7498	0.956942145085*
Worst	-	-
Std.	1.110223024 63e-16	5.028101246 79e-05
FP	0.8	0.08

g11 (500000 FES)

	CMODE	SAM-PSO
Best	0.7498	0.749830118447
Mean	0.7498	0.749880399459
Median	0.7498	0.956942145085*
Worst	-	-
Std.	1.110223024 63e-16	5.028101246 79e-05
FP	0.8	0.08

g12 (500000 FES)

	CMODE	SAM-PSO
Best	-1.0	-1.0
Mean	-1.0	-1.0
Median	-1.0	-1.0
Worst	-1.0	-1.0
Std.	0.0	1.153777611 83e-16
FP	1.0	1.0

g12 (500000 FES)

	CMODE	SAM-PSO
Best	-1.0	-1.0
Mean	-1.0	-1.0
Median	-1.0	-1.0
Worst	-1.0	-1.0
Std.	0.0	1.153777611 83e-16
FP	1.0	1.0

g13 (500000 FES)

	CMODE	SAM-PSO
Best	0.0539374983859	0.4816311865 66
Mean	0.361804709732	0.853082070219
Median	0.438771512569	0.894796108884*
Worst	0.438771512569	-
Std.	0.153933605673	0.167574637 22
FP	1.0	0.4

g13 (500000 FES)

	CMODE	SAM-PSO
Best	0.0539374983859	0.4816311865 66
Mean	0.361804709732	0.853082070219
Median	0.438771512569	0.894796108884*
Worst	0.438771512569	-
Std.	0.153933605673	0.167574637 22
FP	1.0	0.4

Fig. 3. Resultados para problemas $g09$, $g11$, $g12$ y $g13$ con CMODE y SAM-PSO.

resultando en valores de diversidad más bajos.

Para SAM-PSO también se obtuvieron gráficas de diversidad que exhiben los comportamientos antes mencionados: en la gran mayoría de los problemas las cuatro medidas dibujan sus curvas de diversidad con casi la misma trayectoria, $g02$, $g03$ y $g13$ guardan el mismo comportamiento con una separación mínima entre las curvas. Una situación especial es el resultado del problema $g11$, cuyo espacio de búsqueda $[-1, 1]^2$ resulta tan pequeño que la mutación de SAM-PSO hace que la población nunca converja, sin que esto signifique que no se optimizó la función pues las partículas pueden tener almacenada en la memoria una buena solución. En cuanto a D_{TAP}^{N2} y D_{PW}^{N}, permanecen cercanas en todas las funciones de prueba de ambos algoritmos. Teóricamente D_{PW}^{N} tendría que ofrecer la mejor descripción de diversidad entre las cuatro ya que contempla la distancia entre todos los individuos, aunque el costo computacional es mucho más elevado que el resto de medidas.

Distancia Euclidiana al mejor individuo. La última medida considerada, trazada en color café , tiene muchos problemas. Al observar las curvas de diversidad que D_{ED}^{N} dibuja por cada una de las funciones éstas resultan poco informativas. Únicamente si la convergencia se da muy pronto es posible que D_{ED}^{N} describa la diversidad correctamente, por ejemplo en las funciones $g06$, $g08$, $g09$ y $g11$ con CMODE.

El principal problema reside en que la normalización es incorrecta; si el algoritmo converge, la distancia promedio al mejor vector y el factor de normalización $(d_{max} - d_{min})$ decrecerán, causando una curva de diversidad que se mantiene en el mismo intervalo de diversidad sin demostrar la concentración de individuos que presenta la población. En caso de estancamiento o exploración sin convergencia, las distancias al mejor vector cambiarán muy poco, también se obtendrá una trayectoria escarpada y atrapada en valores cercanos de diversidad.

5. Conclusiones

En este trabajo se presentó un comparativo empírico del desempeño de seis medidas de diversidad aplicadas a dos algoritmos del estado del arte, CMODE [16] y SAM-PSO [7], al resolver las funciones de prueba del benchmark CEC2006, estudio necesario ya que únicamente se han publicado comparativos de medidas de diversidad para problemas sin restricciones.

Los resultados indican que D_{TAP}^{N2} y D_{PW}^{N} describen mejor la diversidad que el resto de medidas, al ser consistentes en ambos algoritmos y todas las funciones de prueba evaluadas. En algunos escenarios el costo computacional es un problema, cuando éste es el caso D_{TAP}^{N} podría ser la mejor opción pues sólo calcula NP distancias, a diferencia de D_{PW}^{N}, la cual computa todas los pares de distancias posibles.

D_{ED}^{N} y D_{DP}^{N} son malos descriptores, pero a pesar de que D_{DP}^{N} no calcula correctamente la diversidad podría ser útil para rastrear mecanismos de promoción de diversidad, como sucede en CMODE cuando se introducen soluciones no factibles y el gráfico de diversidad lo refleja con cambios abruptos.

Tabla 3. Estadísticas de 25 ejecuciones de $g01$ a $g13$ en CMODE y SAM-PSO.

$g01$	CMODE	SAM-PSO	$g02$	CMODE	SAM-PSO	$g03$	CMODE	SAM-PSO
Mejor	-15.00005	-15.00005	Mejor	-0.803623	-0.803365	Mejor	-1.0010	-1.0008511
Media	-15.00005	-14.94535	Media	-0.800686	-0.792155	Media	-0.6115	-0.999879
Mediana	-15.00005	-14.88988	Mediana	-0.803623	-0.788432	Mediana	-0.6467	-0.989127
Peor	-15.00005	-	Peor	-0.785269	-	Peor	-	-
σ^2	0.0	0.0794	σ^2	0.00539	0.0088	σ^2	0.31788	0.00297
%EF	100 %	56 %	%EF	100 %	68 %	%EF	16 %	52 %

$g04$	CMODE	SAM-PSO	$g05$	CMODE	SAM-PSO	$g06$	CMODE	SAM-PSO
Mejor	-30665.61	-30665.61	Mejor	5126.4961	5127.39	Mejor	-6961.93	-6961.93
Media	-30665.61	-30659.78	Media	5126.4961	5173.81	Media	-6961.93	-6961.93
Mediana	-30665.61	-30659.35	Mediana	5126.4961	5211	Mediana	-6961.93	-6961.93
Peor	-30665.61	-	Peor	5126.4961	-	Peor	-6961.93	-6961.93
σ^2	0.0	13.72	σ^2	$6.289e^{-13}$	42.02	σ^2	0.0	0.0
%EF	100 %	64 %	%EF	100 %	32 %	%EF	100 %	100 %

$g07$	CMODE	SAM-PSO	$g08$	CMODE	SAM-PSO	$g09$	CMODE	SAM-PSO
Mejor	24.306037	24.89125	Mejor	-0.095825	-0.095825	Mejor	680.62994	680.68862
Media	24.306037	26.15804	Media	-0.095825	-0.095825	Media	680.62994	680.84156
Mediana	24.306037	26.07737	Mediana	-0.095825	-0.095825	Mediana	680.62994	680.82510
Peor	-	-	Peor	-0.095825	-0.095825	Peor	680.62994	-
σ^2	$8.50e^{-15}$	1.3104	σ^2	$1.11e^{-}17$	0.0	σ^2	$9.37e^{-14}$	0.2302
%EF	88 %	84 %	%EF	100 %	100 %	%EF	100 %	80 %

$g10$	CMODE	SAM-PSO	$g11$	CMODE	SAM-PSO	$g12$	CMODE	SAM-PSO
Mejor	7048.7269	7160.465	Mejor	0.7498	0.74983	Mejor	-1.0	-1.0
Media	7048.7269	7656.016	Media	0.7498	0.74988	Media	-1.0	-1.0
Mediana	7048.7269	7888.102	Mediana	0.7498	0.74993	Mediana	-1.0	-1.0
Peor	7048.7269	-	Peor	0.7498	-	Peor	-1.0	-1.0
σ^2	$9.67e^{-8}$	539.10	σ^2	$1.11e^{-16}$	$5.02e^{-5}$	σ^2	0.0	$1.1537e^{-16}$
%EF	100 %	60 %	%EF	100 %	80 %	%EF	100 %	100 %

$g13$	CMODE	SAM-PSO
Mejor	0.05393	0.48163
Media	0.36180	0.85308
Mediana	0.43877	0.99823
Peor	0.43877	-
σ^2	0.1539	0.16757
% E. F.	100 %	40 %

Como trabajo futuro resta evaluar las medidas del espacio fenotípico, pues ellas interpretan la diversidad con base en el valor de aptitud. Para ser aplicadas en optimización con restricciones debe considerarse que el valor de aptitud no decrecerá de manera directa como sucede en la optimización sin restricciones. Sería interesante probar el desempeño de las medidas de diversidad del espacio fenotípico usando el valor de aptitud y las violaciones de restricciones. Otro aspecto a investigar es la influencia del manejador de restricciones en el comportamiento de las medidas de diversidad, pues el manejador es un elemento central en los algoritmos para optimización con restricciones.

Finalmente se debe reconocer que un comparativo como el presentado en este documento no puede reflejar si las medidas de diversidad son aplicables en todos los algoritmos de optimización con restricciones, ya que es necesario medir diversidad en escenarios donde se tenga un comportamiento esperado antes de evaluar el desempeño de las medidas [2]. Al medir diversidad en algoritmos el comportamiento está sesgado por las estrategias y mecanismos que ellos implementan. En este sentido, desarrollar un simulador de optimización con restricciones donde sea posible generar escenarios con un comportamiento esperado puede ayudar a evaluar mejor las medidas y posteriormente diseñar una nueva medida de diversidad específica para entornos restringidos.

Agradecimientos. El primer autor agradece el apoyo de CONACyT para realizar estudios de posgrado en la Universidad Veracruzana. El segundo autor agradece el apoyo de CONACyT mediante el proyecto No. 220522.

Referencias

1. Coello, C.A.C.: Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. Computer methods in applied mechanics and engineering 191(11), 1245–1287 (2002)
2. Corriveau, G., Guilbault, R., Tahan, A., Sabourin, R.: Review and study of genotypic diversity measures for real-coded representations. IEEE transactions on evolutionary computation 16(5), 695–710 (2012)
3. Corriveau, G., Guilbault, R., Tahan, A., Sabourin, R.: Evaluation of genotypic diversity measurements exploited in real-coded representation. arXiv preprint arXiv:1507.00088 (2015)
4. Deb, K.: An efficient constraint handling method for genetic algorithms. Computer methods in applied mechanics and engineering 186(2), 311–338 (2000)
5. Diaz-Gomez, P.A., Hougen, D.F.: Empirical study: Initial population diversity and genetic algorithm performance. Artificial Intelligence and Pattern Recognition 2007, 334–341 (2007)
6. Eiben, A.E., Smith, J.E.: Introduction to evolutionary computing, vol. 53. Springer (2003)
7. Elsayed, S.M., Sarker, R.A., Mezura-Montes, E.: Self-adaptive mix of particle swarm methodologies for constrained optimization. Information sciences 277, 216–233 (2014)
8. Friedrich, T., Oliveto, P.S., Sudholt, D., Witt, C.: Analysis of diversity-preserving mechanisms for global exploration. Evolutionary Computation 17(4), 455–476 (2009)
9. Herrera, F., Lozano, M.: Adaptation of genetic algorithm parameters based on fuzzy logic controllers. Genetic Algorithms and Soft Computing 8, 95–125 (1996)
10. Liang, J.J., Suganthan, P.N.: Dynamic multi-swarm particle swarm optimizer with a novel constraint-handling mechanism. In: Evolutionary Computation, 2006. CEC 2006. IEEE Congress on. pp. 9–16. IEEE (2006)
11. Liang, J., Runarsson, T.P., Mezura-Montes, E., Clerc, M., Suganthan, P., Coello, C.C., Deb, K.: Problem definitions and evaluation criteria for the cec 2006 special session on constrained real-parameter optimization. Journal of Applied Mechanics 41(8) (2006)
12. Mauldin, M.L.: Maintaining diversity in genetic search. In: AAAI. pp. 247–250 (1984)
13. Morrison, R.W., De Jong, K.A.: Measurement of population diversity. In: International Conference on Artificial Evolution (Evolution Artificielle). pp. 31–41. Springer (2001)
14. Olorunda, O., Engelbrecht, A.P.: Measuring exploration/exploitation in particle swarms using swarm diversity. In: 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence). pp. 1128–1134. IEEE (2008)
15. Ursem, R.K.: Diversity-guided evolutionary algorithms. In: International Conference on Parallel Problem Solving from Nature. pp. 462–471. Springer (2002)

16. Wang, Y., Cai, Z.: Combining multiobjective optimization with differential evolution to solve constrained optimization problems. IEEE Transactions on Evolutionary Computation 16(1), 117–134 (2012)

ISSN 1870-4069

Análisis y diseño de operadores de cruce basados en el cruce binario simulado

J. Chacón, C. Segura, A. Hernández Aguirre

Centro de Investigación en Matemáticas,
Guanajuato, México

{joel.chacon, carlos.segura, artha}@cimat.mx

Resumen. El operador de Cruce Binario Simulado (SBX) es uno de los operadores más utilizados para codificaciones con números reales. En este artículo se presenta un análisis novedoso del operador SBX, que involucra aspectos relacionados con la diversidad, similitud y reflexiones que se dan entre los padres y los hijos. En base a este análisis, se propone un nuevo operador denominado el operador Dinámico sin Reflexiones basado en el Cruce Binario Simulado (NRD-SBX). NRD-SBX aporta un mayor grado de intensificación y es capaz de adaptar el grado de exploración en el proceso de optimización. Esto se consigue modificando el operador SBX mediante la implementación de un modelo dinámico para el control de diversidad y las reflexiones. La validación experimental, la cual considera los tests WFG y varios Algoritmos Evolutivos Multi-Objetivo muy populares, muestra de forma clara los beneficios aportados por el nuevo operador.

Palabras clave: Optimización multi-objetivo, algoritmos evolutivos, operadores de cruce.

Analysis and Design of Crossover Operators based on the Simulated Binary Crossover

Abstract. The Simulated Binary Crossover (SBX) is a popular operator used in continuous domains. In this work, an analysis of the SBX is presented, which involves the study of several diversity related issues, similarities among solutions and reflections that appear in the reproduction phase. Taking the analysis into account, the operator Non-Reflection Dynamic Simulated Binary Crossover (NRD-SBX) is proposed. This new operator extends SBX by implementing a dynamic model to control the diversity and reflections. NRD-SBX provides a higher intensification level than SBX and is able to adapt the exploration degree through the optimization process. The experimental validation is carried out with the WFG benchmark and several multi-objective EAs. The NRD-SBX operator shows a better performance than the original SBX.

Keywords: Optimization multi-objective, evolutionary algorithms, crossover operators.

1. Introducción

En las últimas décadas, la implementación de metaheurísticas para resolver problemas de optimización complejos ha ganado una gran popularidad [18]. Por ello, se han propuesto diversos tipos de metaheurísticas, destacando entre las más utilizadas los Algoritmos Genéticos (Genetic Algorithm - GA), Evolución Diferencial (Differential Evolution - DE), Estrategias Evolutivas (Evolution Strategies - ES), y Optimización por Enjambres de Partículas (Particle Swarm Optimization - PSO). Dadas las similitudes que existen entre algunas de las metaheurísticas poblacionales más populares, se propuso el termino Algoritmo Evolutivo (Evolutionary Algorithm - EA) para agrupar a varias de ellas [16].

Debido al comportamiento prometedor de los EAs, estos han sido ampliados en múltiples formas. Por ejemplo, han sido usados para resolver Problemas de Optimización Multi-objetivo (Multi-objective Optimization Problems - MOPs). Un MOP continuo basado en minimización puede ser definido como se indica en la ecuación (1):

$$\begin{aligned} Minimizar \quad & f_m(\mathbf{x}), \quad m = 1,2,...,M; \\ Sujeto \quad a \quad & x_i^{(L)} \leq x_i \leq x_i^{(U)}, \quad i = 1,2,...,n. \\ & \mathbf{x} \in \Omega \end{aligned} \tag{1}$$

donde $\mathbf{n}$ es la dimensión correspondiente al espacio de las variables, Ω es el espacio factible, $\mathbf{x}$ es un vector compuesto por $\mathbf{n}$ variables de decisión $x = (x_1, ..., x_n) \in R^n$, y $x_i^{(L)}$ y $x_i^{(U)}$ son los límites inferior y superior de cada variable. Cada solución es evaluada mediante el uso de la función $F : \Omega \rightarrow R^m$, la cual consiste de m funciones objetivo y R^m es conocido como el espacio objetivo. Dadas dos soluciones $\mathbf{x}$, $\mathbf{y} \in \Omega$, $\mathbf{x}$ domina a $\mathbf{y}$, denotado por $\mathbf{x} \prec \mathbf{y}$, si y solo si $\forall m \in 1,2,...,M : f_m(x_i) \leq f_m(y_i)$ y $\exists m \in 1,2,...,M : f_m(x_i) < f_m(y_i)$. Una solución $\mathbf{x}^* \in \Omega$ es conocida como solución óptima de Pareto si no existe otra solución $\mathbf{x} \in \Omega$ que domine a $\mathbf{x}^*$. El conjunto de Pareto es el conjunto de todas las soluciones óptimas de Pareto y el frente de Pareto está formado por las imágenes del conjunto de Pareto. El propósito de los optimizadores multi-objetivo es, esencialmente, obtener un conjunto de soluciones bien distribuidas y cercanas a las soluciones del frente de Pareto.

Este trabajo se centra en el análisis y diseño de Algoritmos Evolutivos Multi-objetivo (Multi-objective Evolutionary Algorithm - MOEAs) basados en codificación real. Se ha mostrado que una de las claves para el diseño apropiado de los EAs consiste en inducir un balance apropiado entre exploración e intensificación [5]. En el caso de optimización mono-objetivo, se puede inducir

este balance controlando la diversidad en la fase de reemplazo [3]. En el caso de los MOEAs, dado que se manejan varios objetivos de forma simultánea, se mantiene un cierto grado de diversidad en la población de forma intrínseca. Sin embargo, esto no siempre es suficiente para asegurar los resultados deseados, pues los operadores genéticos deberían utilizar dicha diversidad para generar nuevas soluciones prometedoras, siendo este el tema en el que se centra este artículo.

En los EAs, los operadores de cruce tienen un efecto importante en la exploración e intensificación.Usualmente, los operadores de cruce tienden a promover mayor diversidad cuando se consideran padres distantes que cuando se consideran padres cercanos. De esta forma, los operadores de cruce son métodos adaptativos que dependen de la diversidad mantenida dentro de la población. En el caso de la codificación real, se han propuesto varios operadores de cruce [11], que pueden ser clasificados como centrados en los padres o en la media, en base a la región donde tienden a crear a los hijos.

El operador de Cruce Binario Simulado (Simulated Binary Crossover - SBX) [8] es probablemente uno de los más utilizados. En la versión inicial del SBX, se definió la forma de intercambiar la información para una variable. Tomando en cuenta las características de las distribuciones involucradas, el operador SBX se clasifica como un operador de cruce centrado en los padres. Sin embargo, en su versión inicial no se discutió como extender el mismo a casos de múltiples variables. Como resultado, se han propuesto diferentes implementaciones para hacer frente a problemas de múltiples variables. Las principales diferencias entre las diferentes propuestas tienen que ver con el número de variables que se heredan sin modificarse.

Además, en los casos de múltiples variables se dan algunos efectos como las reflexiones, que no se producen en los casos de una variable. Este tipo de transformaciones incrementan considerablemente la distancia entre los padres y los hijos cuando son utilizados en problemas de alta dimensionalidad. Dado que en comparación con los EAs mono-objetivo, los MOEAs internamente mantienen un grado más elevado de diversidad, los MOEAs que utilizan el operador SBX podrían crear individuos muy alejados de las regiones promisorias identificadas en fases anteriores de la búsqueda. Basados en el análisis del operador SBX presentado en este artículo, se propone el operador Dinámico sin Reflexiones basado en el Cruce Binario Simulado "Non-Reflection Dynamic Simulated Binary Crossover Operator" (NRD-SBX). El operador NRD-SBX adapta la probabilidad de heredar las variables sin modificarlas con el objetivo de inducir un mayor grado de exploración en las fases iniciales del proceso de optimización. Adicionalmente, el operador propuesto es modificado para no realizar las reflexiones propias de varias de las implementaciones más populares del SBX, con el objetivo de proveer un grado de intensificación adicional.

El resto de este artículo está organizado de la siguiente forma. La sección 2 provee una breve descripción de operadores de cruce y MOEAs más populares. La sección 3 analiza algunas de las dificultades que podrían estar presentes en la aplicación del operador SBX. La sección 4 describe el operador NRD-SBX. La

validación experimental se describe en la sección 5. Finalmente, las conclusiones y trabajos futuros se detallan en la sección 6.

2. Estado del arte

En esta sección se revisan algunos de los trabajos más importantes que están relacionados con la investigación que se presenta en este artículo, disponiendo de una sección dedicada a operadores de cruce y de otra dedicada a MOEAs.

2.1. Operadores de cruce

Los operadores de cruce se diseñan con el objetivo de generar soluciones candidatas utilizando la información de dos o más soluciones padres. Dado que existen múltiples operadores, se han propuesto varias taxonomías para clasificarlos. Una de las clasificaciones más populares distingue entre los operadores de cruce "Centrados en los Padres" y los "Centrados en la media" [14]. En los operadores centrados en los padres, las soluciones hijas son creadas alrededor de cada solución padre, mientras que en los operadores centrados en la media, las soluciones hijas son creadas alrededor de la media de las soluciones padres. Entre los operadores de cruce, el operador SBX es probablemente el más utilizado.

Cruce Binario Simulado (SBX) Uno de los operadores de cruce que se ha utilizado más comúnmente en los MOEAs es el SBX. El SBX se clasifica como un operador centrado en los padres y se basa en una función de distribución [11] que usa el parámetro η (llamado índice de distribución y que debe ser fijado por el usuario) para determinar la apertura de la distribución, y por tanto, su capacidad de exploración. Específicamente, un índice de distribución pequeño induce una mayor probabilidad de construir soluciones hijas alejadas de los padres, mientras que con un índice de distribución elevado, la probabilidad de crear soluciones hijas más similares a los padres se incrementa. Esto es ilustrado en la Figura 1, donde se muestran las funciones de densidad asociadas a los hijos para dos índices de distribución distintos. En la misma los círculos representan a los padres, y se puede apreciar que con η igual a 5, la probabilidad de crear soluciones más cercanas a los padres es mayor que con $\eta = 2$.

Algunas características interesantes del operador SBX son las siguientes:

- Los valores de las soluciones hijas son equidistantes de los padres.
- Existe una probabilidad no nula de generar una solución hija en cualquier parte en el espacio independientemente de donde se localicen los padres.
- La probabilidad de crear un par de soluciones hijas dentro del rango de las soluciones padres es idéntica a la probabilidad de crear dos soluciones hijas fuera de dicho rango.

Debido a la popularidad y al buen rendimiento del SBX, se han propuesto varias extensiones de este operador. En las primeras implementaciones del operador SBX [1], la forma de la distribución no estaba afectada por los límites

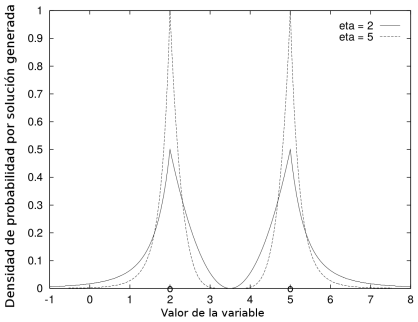


Fig. 1. Función de densidad del operador SBX con índices de distribución 2 y 5.

de cada variable, y todas las variables se modificaban en cada operación de cruce. Posteriormente, se propusieron variantes que sí consideran dichos límites así como una variante auto-adaptativa [8], que controla la diversidad mediante una variable conocida como el factor de dispersión. Dado que cambiar todas las variables de forma simultánea puede inducir un operador muy disruptivo, en las implementaciones actuales cada variable se cambia con una probabilidad igual a 0.5, mientras que en el resto de casos los valores se heredan sin alterar [7,15]. Además, por la forma en que se heredan las variables, pueden aparecer ciertas reflexiones que se analizan posteriormente. Por último, cabe destacar que no existen versiones del operador SBX en las que en lugar de usar dicha probabilidad fija en 0.5, esta sea cambiada a lo largo de la ejecución.

2.2. Algoritmos evolutivos multi-objetivo

Actualmente, existe una gran cantidad de MOEAs que se han diseñado siguiendo distintos principios. Para tener una mejor clasificación de los mismos, se han propuesto varias taxonomías [17]. Así, de acuerdo a los principios de diseño, los MOEAs pueden estar basados en la dominancia de Pareto, en indicadores y en descomposición [6]. Dado que todos ellos han reportado resultados de alta calidad, para la validación experimental del operador de cruce diseñado en este artículo, se ha seleccionado un método de cada tipo. Específicamente, la validación se ha llevado a cabo incluyendo el Algoritmo Genético de Ordenación de No Dominados II (Non-Dominated Sorting Genetic Algorithm - NSGA-II) [7], el MOEA Basado en Descomposición (MOEA Based on Decomposition -MOEA/D) [19], y Algoritmo de Optimización Multi-objetivo basado en la Selección con métrica S (S-Metric Selection Evolutionary Multi-objective Optimization Algorithm - SMS-EMOA) [2]. Los siguientes apartados describen brevemente cada uno de los paradigmas, así como los métodos seleccionados.

MOEAs basados en dominancia - NSGA-II Estos MOEAs se basan en considerar la noción de dominancia de Pareto para diseñar los distintos componentes

de los EAs. Dado que la relación de dominancia no promueve la preservación de diversidad en el espacio objetivo de forma intrínseca, es necesario utilizar técnicas auxiliares, como las basadas en nichos, muchedumbre o clústers, para promover la preservación de dicha diversidad. Probablemente, la técnica más popular de esta categoría es el NSGA-II [7]. Este algoritmo implementa un operador especial para la selección de padres e incorpora el uso de elitismo en la fase de reemplazo.

MOEAs basados en indicadores - SMS-EMOA Con el objetivo de comparar el rendimiento de los MOEAs, se han ideado varios indicadores de calidad que mapean las aproximaciones del conjunto de Pareto a valores reales. Dado que estos indicadores miden la calidad de las aproximaciones obtenidas por un determinado MOEA, se ha propuesto un paradigma basado en la aplicación de estos indicadores. En estos casos, en lugar de utilizar la dominancia de Pareto, se usan dichos indicadores en el MOEA para guiar el proceso de optimización. Entre los distintos indicadores, el hipervolumen es de lo más aceptados, habiendo sido usado en varios esquemas [13]. Una de las ventajas de los esquemas basados en indicadores es que estos usualmente toman en cuenta tanto la calidad de las soluciones como la diversidad en el espacio objetivo, por lo que no es necesario incluir mecanismos adicionales para preservar la diversidad. Entre los distintos MOEAs basados en indicadores, el SMS-EMOA [2] es de los más utilizados, probablemente debido a su simplicidad y superioridad sobre varios otros esquemas [10].

MOEAs basados en descomposición - MOEA/D Los MOEAs basados en descomposición transforman un problema multi-objetivo en un conjunto de problemas de optimización de un solo objetivo que son considerados de forma simultánea [4]. Esta transformación puede ser realizada de varias formas, por ejemplo, utilizando una suma lineal con pesos o con una función Tchebycheff basada en la agregación con pesos. Dado un conjunto de pesos que definen distintas funciones objetivo, el MOEA trata de encontrar buenas soluciones para cada una de ellas. Dentro de este grupo, el MOEA/D [19] es el algoritmo más popular. En este algoritmo se asocia un individuo de la población a cada subproblema. Entre sus principales características, además de que está basado en descomposición, cabe destacar que esto la hace con funciones basadas en peso, y que aplica restricciones de emparejamiento a través de la definición de un conjunto de vecindarios. En el MOEA/D se han propuesto distintos enfoques para la implementación de los pesos. Entre estos enfoques, el de Tchebycheff es uno de los más populares y es el que se usa en este artículo.

3. Análisis del operador SBX

La implementación más utilizada del operador SBX se encuentra integrada en el NSGA-II publicada por Deb et al.[7]. En esta implementación cabe destacar dos puntos clave. El primero está relacionado con la similitud. En dicha

implementación los valores de las soluciones hijas son heredados directamente de las soluciones padre con una probabilidad fija igual a 0.5, mientras que el resto de las variables son modificadas mediante la distribución de probabilidad propia del SBX. En consecuencia la similitud que existe entre los padres y los hijos depende altamente del número de variables que se consideran en el problema de optimización, pues el incremento de la dimensionalidad involucra la creación de soluciones más distantes.

El segundo punto clave, que nunca ha sido analizado en detalle, está relacionado con el conjunto de reflexiones que se realizan. Después de generar los dos valores que deben ser heredados en los dos hijos, dichos valores son intercambiados entre sí con una probabilidad fija del 0.5. En consecuencia, cada vez que las variables son intercambiadas se realiza una reflexión, que puede inducir grandes distancias entre los padres y los hijos, a pesar de que el SBX es considerado como un cruce centrado en los padres.

La parte izquierda de la Figura 2 ilustra este comportamiento en el operador SBX para dos y tres variables. En esta figura, los padres están identificados con dos puntos rojos, y se ejecutó el operador SBX diez mil veces. Cada uno de los puntos de color negro es una solución hija, con lo que esta figura ilustra las zonas en las que se tienden a generar a los hijos. Se puede ver que los valores de cada variable siempre están cercanos a uno de los valores asociados a las variables de los padres. Sin embargo, debido al proceso de intercambio de valores, las soluciones hijas no siempre se encuentran cercanas a los padres.

Particularmente, en el caso de dos dimensiones mostrado, se crean soluciones en la esquina superior izquierda y en la esquina inferior derecha, mientras que los padres están es las esquinas opuestas. A medida que aumenta el número de dimensiones d, la probabilidad de que siempre o nunca haya intercambios, y que por tanto, la nueva solución esté cercana a uno de los padres es $k^d+(1-k)^d$, con lo que se produce una disminución exponencial respecto al número de dimensiones. En consecuencia, las reflexiones provocan un alto grado de exploración. En algunos MOPs con alta dimensionalidad en el espacio de las variables, esto podría representar un inconveniente porque las reflexiones localizan soluciones hijas en cada esquina del hipercubo mínimo que contiene a las soluciones padre, lo que significa que el operador original podría inducir un nivel muy bajo de intensificación.

El inconveniente relacionado con las reflexiones puede ser manejado parcialmente implementando restricciones de emparejamiento que traten de emparejar exclusivamente a soluciones similares. Bajo esta condiciones, el hipercubo sería de menor tamaño, con lo que se induciría un mayor grado de intensificación. Esta puede ser una de las razones por las que MOEA/D, que incorpora restricciones de apareamiento, ha sido capaz de resolver muchos problemas de forma exitosa. En este artículo, se trata de resolver esta problemática eliminando el intercambio de variables, así como realizando otras modificaciones en el SBX que se describen en la siguiente sección. La ventaja de esta segunda alternativa es que se puede incorporar de forma sencilla en cualquier MOEA.

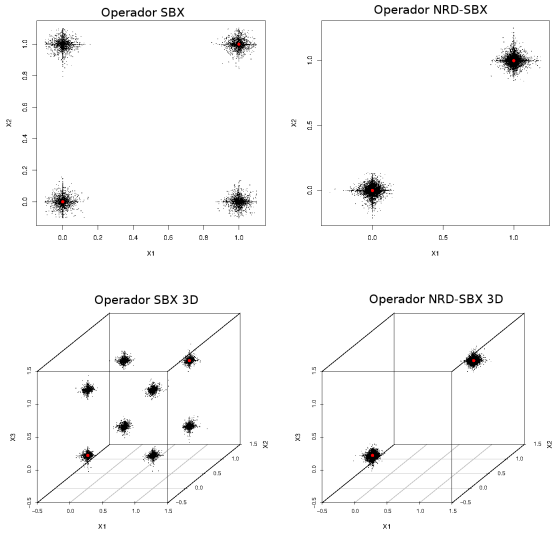


Fig. 2. En la columna izquierda se presenta la simulación del operador SBX y en la columna de la derecha se presenta el operador NRD-SBX, ambas distribuciones con un índice de distribución de 20.

4. Propuesta algorítmica

Tomando en cuenta el análisis previo, así como el deseo de adaptar el operador a los requerimientos de las diferentes fases de optimización, se diseñó e implementó un nuevo operador de cruce, que se denominada operador Dinámico sin Reflexiones basado en el Cruce Binario Simulado (NRD-SBX). NRD-SBX realiza dos modificaciones al operador SBX descrito anteriormente. En primer lugar, en el NRD-SBX, la probabilidad de intercambiar variables se fija a 0, con lo que nunca aparecen las reflexiones propias del SBX original. El efecto de este cambio se muestra en la parte derecha de la Figura 2, en la que podemos ver que ahora todos los hijos quedaron localizados en regiones cercanas a los padres. Esto implica un mayor grado de intensificación. Es importante hacer notar la diferencia entre el operador modificado SBX y un operador de mutación. La distancia entre padres e hijos en el NRD-SBX es proporcional a la distancia entre los padres, mientras que los operadores de mutación sólo consideran la información de una solución padre.

La segunda modificación del NRD-SBX está relacionada con la cantidad de variables que se heredan sin modificarse. El operador SBX hereda directamente los valores que corresponden a los padres con una probabilidad igual a 0.5. Esto podría limitar el grado de exploración, especialmente en las primeras generaciones. Para evitar esta problemática, el operador NRD-SBX altera la probabilidad de heredar sin modificar las variables de los padres durante la ejecución del algoritmo. Particularmente, se implementa un modelo dinámico lineal, en el que

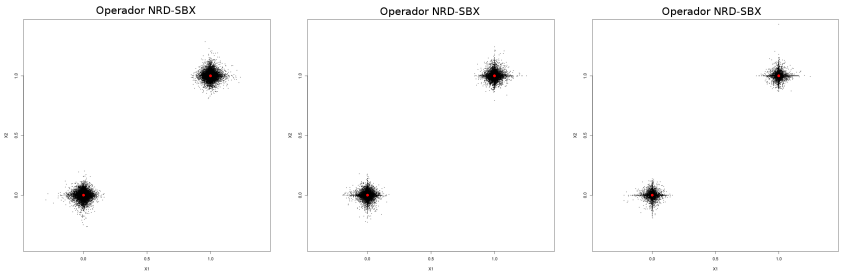


Fig. 3. Simulación del operador SBX con la probabilidad de heredar directamente las variables con 0 %, 25 % y 50 % respectivamente. Los puntos rojos representan a los individuos padres.

en la primera generación, esta probabilidad es asignada a cero y con el transcurso de las generaciones, la probabilidad se incrementa de manera lineal, de forma que cuando han transcurrido la mitad de las generaciones la probabilidad es igual a 0.5. A partir de este momento, y hasta el final de la ejecución, se mantiene fija la probabilidad a 0.5. Esta regla tiende a ir disminuyendo paulatinamente el número de variables que se heredan sin realizar modificaciones, con lo que en las primeras fases se induce un mayor grado de exploración y las últimas fases, se induce un mayor grado de intensificación. La Figura 3, muestra para el caso de dos dimensiones, una simulación heredando directamente las variables con una probabilidad de 0, 0.25 y 0.50. Se puede apreciar que con probabilidades bajas (fases iniciales de la optimización) se exploran más zonas, mientras que en las fases finales se tienden a crear muchas soluciones que comparten valores con los padres, siendo este un paso mucho más intensificador. Cabe destacar que el comportamiento final en el que se mantienen bastantes variables sin cambiar, es deseable con MOPs en los que las variables son independientes. Sin embargo, en los problemas que involucran dependencia entre las variables, utilizar modificaciones de este tipo puede llevar a que el algoritmo se quede estancado en óptimos locales de baja calidad.

5. Validación experimental

Esta sección está dedicada a mostrar el conjunto de experimentos que se realizaron para validar el rendimiento del operador NRD-SBX. Se muestran estudios con los tres algoritmos previamente descritos, considerando la implementación presente en jMetalcpp [15]. El único cambio que hubo que realizar en cada caso fue la inclusión del operador de cruce. La versión de SBX que viene incorporada en jMetalcpp difiere ligeramente de la incorporada en la versión oficial del NSGA-II, pues la variante de jMetalcpp incluye un paso de reflexión adicional. En nuestra validación se consideró la versión incluida en el NSGA-II oficial, así como el nuevo operador NRD-SBX. En lo referente a las funciones multi-objetivo, se consideran las 9 funciones WFG que se propusieron en [12].

Dado que todos los optimizadores son estocásticos, cada ejecución se repitió 35 veces con diferentes semillas en el generador de números aleatorios. Los siguientes aspectos fueron comunes para todos los algoritmos: el criterio de parada se fijó a 250,000 generaciones, el tamaño de población a 100, el operador de cruce SBX o NRD-SBX se configuró con un índice de distribución igual a 20 y la probabilidad de su uso fue de 0.9, se utilizó la mutación polinomial con un índice de distribución igual a 50 y su probabilidad de aplicación se fijó a 1/24. Además, los problemas WFG fueron configurados con 24 parámetros, siendo 20 de ellos parámetros de distancia, y 4 de ellos parámetros de posición. Por otro lado, (a excepción del NSGA-II) hay algunas configuraciones adicionales que son propias de cada uno de los algoritmos utilizados:

- **SMS-EMOA**: el desplazamiento se fijó a 100.
- **MOEA/D**: el tamaño de la vecindad se fijó a 20, el número máximo de actualización de subproblemas se fijó a 2 y la probabilidad de emparejar soluciones únicamente de la vecindad es de 0.9 .

El análisis experimental se realizó teniendo en cuenta las superficies de cubrimiento y el hipervolumen. En el caso de las comparativas en base al hipervolumen, y teniendo en cuenta el comportamiento estocástico de los optimizadores, se realizaron un conjunto de tests estadísticos siguiendo los criterios establecidos en [9]. Concretamente, en primer lugar se utilizó el test Shapiro-Wilk para comprobar si los resultados se ajustaban a una distribución Gaussiana. En los casos en que sí se ajustaban, se utilizó el test de Levene para comprobar la homogeneidad de las varianzas, procediendo con el test de ANOVA en caso positivo o con el de Welch en caso negativo. Por otro lado, para los casos que no se ajustaban a distribuciones Guassianas, se utilizó el test de Kruskal-Wallis. En todos los casos se fijó el nivel de confianza al 95 %. Se considera que un algoritmo X es superior a un algoritmo Y, si el procedimiento anterior reporta diferencias significativas y si la media y mediana del hipervolumen obtenido por el método X son superiores a las obtenidas por el método Y.

En primer lugar se van a analizar los resultados en base a las superficies de cubrimiento. Para ello, de los 9 problemas se seleccionaron los 4 casos en los que se detectaron mayores diferencias. La Figura 4 muestra las superficies de cubrimiento al 50 % del WFG1, WFG5, WFG8 y WFG9. Se puede apreciar que para cada algoritmo, el uso del operador NRD-SBX produce resultados que dominan o igualan a los alcanzados por SBX en la mayor parte de las casos. De hecho, la única excepción se da en el WFG5 con el NSGA-II, en el que la utilización de NRD-SBX produce resultados ligeramente inferiores. En cualquier caso, los beneficios ofrecidos por NRD-SBX en el resto de casos son mucho mayores que la penalización que aparece en el caso del WFG5. Particularmente, los beneficios ofrecidos en el caso del WFG1 y WFG8 son especialmente claros. En el caso del WFG1, se ha mostrado en otros artículos [3] que la pérdida rápida de la diversidad puede ser un grave problema, con lo que el mayor grado de exploración que induce NRD-SBX en las fases iniciales es de gran ayuda. Por otro lado, dado que es un problema unimodal y separable, evitar las reflexiones

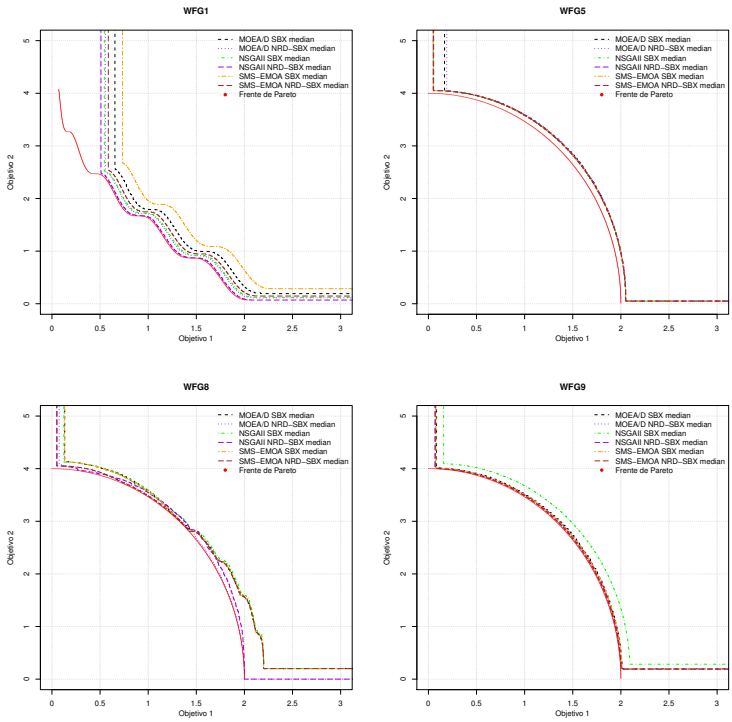


Fig. 4. Superficies de cubrimiento al 50 % obtenidas por los diferentes MOEAs con SBX y NRD-SBX.

y centrarse en crear soluciones cercanas a los padres también contribuye al buen rendimiento. En el caso del WFG8, hay un alto grado de dependencia entre las variables, por lo que ir cambiando de forma dinámica la cantidad de variables que se heredan sin realizar modificaciones y por tanto no centrarse tanto en soluciones que comparten muchos valores con los ya encontrados anteriormente es de gran ayuda. En el WFG9 se da una situación similar, aunque por ser este problema más sencillo, los beneficios sólo se aprecian de forma clara en el caso del NSGA-II.

Con el objetivo de confirmar y complementar los hallazgos anteriores, se realizaron análisis del hipervolumen alcanzado por los diferentes algoritmos, con cada uno de los operadores de cruce. La Tabla 1 muestra para cada algoritmo y operador de cruce el mínimo, máximo, media y desviación estándar de los valores del hipervolumen alcanzados. Tal y como es habitual [20], para el cálculo de este valor, se fijó el punto de referencia en (3.0, 5.0). Para cada algoritmo y problema, se realizaron comparativas estadísticas entre el operador SBX y el operador NRD-SBX. En los casos en los que se pudo confirmar la superioridad de alguno de los operadores, se resaltan los datos en la tabla. Para el caso del MOEA/D

Tabla 1. Valores de hipervolumen alcanzados por los diferentes MOEAs con los operadores de cruce SBX y NRD-SBX.

MOEA/D

	NRD-SBX				SBX			
	Min.	Max.	Media	Desv.	Min.	Max.	Media	Desv.
WFG1	**9.61**	**10.72**	**10.26**	**3.1E-01**	8.39	10.29	9.62	4.3E-01
WFG2	**10.62**	**11.45**	**10.66**	**1.9E-01**	9.28	10.62	10.50	3.7E-01
WFG3	10.95	10.95	10.95	2.5E-04	10.95	10.95	10.95	3.0E-04
WFG4	8.68	8.68	8.68	1.0E-04	8.67	8.68	8.68	1.0E-04
WFG5	8.13	8.34	8.16	4.7E-02	**8.13**	**8.24**	**8.19**	**3.0E-02**
WFG6	**8.39**	**8.57**	**8.49**	**4.8E-02**	7.76	8.54	8.30	1.2E-01
WFG7	8.67	8.68	8.67	1.5E-04	8.67	8.68	8.67	2.3E-04
WFG8	**8.20**	**8.60**	**8.54**	**8.5E-02**	7.80	7.87	7.83	1.8E-02
WFG9	**7.70**	**8.56**	**8.41**	**1.9E-01**	7.70	8.54	8.20	3.1E-01

NSGA-II

	NRD-SBX				SBX			
	Min.	Max.	Media	Desv.	Min.	Max.	Media	Desv.
WFG1	**9.91**	**10.70**	**10.52**	**1.8E-01**	9.51	10.76	10.21	3.1E-01
WFG2	**10.59**	**11.44**	**10.66**	**1.9E-01**	9.28	10.62	10.58	2.2E-01
WFG3	10.91	10.93	10.92	4.0E-03	**10.92**	**10.93**	**10.93**	**2.7E-03**
WFG4	8.65	8.66	8.66	2.3E-03	**8.66**	**8.67**	**8.67**	**2.2E-03**
WFG5	8.26	8.27	8.27	2.3E-03	8.27	8.28	8.27	1.6E-03
WFG6	**8.33**	**8.59**	**8.47**	**5.6E-02**	8.12	8.46	8.34	6.5E-02
WFG7	8.64	8.66	8.65	3.9E-03	**8.66**	**8.67**	**8.66**	**2.6E-03**
WFG8	**7.88**	**8.50**	**8.26**	**2.7E-01**	7.71	7.83	7.78	3.2E-02
WFG9	**8.34**	**8.61**	**8.43**	**7.9E-02**	7.69	8.51	7.85	2.8E-01

SMS-EMOA

	NRD-SBX				SBX			
	Min.	Max.	Media	Desv.	Min.	Max.	Media	Desv.
WFG1	**8.60**	**10.70**	**10.03**	**4.6E-01**	8.00	9.89	8.92	5.0E-01
WFG2	**10.63**	**11.46**	**10.65**	**1.4E-01**	9.29	11.46	10.61	2.7E-01
WFG3	10.96	10.96	10.96	2.8E-04	10.96	10.96	10.96	3.6E-04
WFG4	8.69	8.69	8.69	7.7E-06	8.69	8.69	8.69	6.8E-06
WFG5	8.22	8.29	8.27	2.1E-02	**8.25**	**8.29**	**8.29**	**8.6E-03**
WFG6	**8.36**	**8.58**	**8.50**	**4.7E-02**	8.22	8.47	8.34	5.8E-02
WFG7	8.69	8.69	8.69	4.6E-06	8.69	8.69	8.69	6.5E-06
WFG8	**7.98**	**8.61**	**8.12**	**2.2E-01**	7.81	7.89	7.85	2.2E-02
WFG9	**8.33**	**8.66**	**8.47**	**8.7E-02**	7.71	8.56	8.20	3.1E-01

la superioridad del NRD-SBX se pudo confirmar en cinco casos, mientras que sólo fue inferior en uno; en el NSGA-II, NRD-SBX fue superior en 5 casos e inferior en 3; finalmente en el SMS-EMOA, NRD-SBX fue superior en 5 casos e inferior sólo en 1. Los resultados anteriores confirman la superioridad de NRD-SBX, pero además es importante recalcar que en los casos en que el operador SBX fue superior, la diferencia entre los valores de hipervolumen alcanzados fue muy pequeña. Por ejemplo, en los 3 problemas en los que NSGA-II con SBX fue superior a NSGA-II con NRD-SBX la diferencia de la media de los valores alcanzados fue sólo 0.01, mientras que en los casos en que NRD-SBX fue superior la diferencia es mayor en más de un orden de magnitud.

Finalmente, con el objetivo de recalcar la superioridad del NRD-SBX, se realizó un test adicional que cuantifica la mejora o empeoramiento aportado por cada algoritmo. En concreto, en todos los problemas, cada pareja de algoritmo y operador de cruce fue comparado estadísticamente frente al resto para obtener una puntuación global. En los casos en los que se pudo comprobar

Tabla 2. Puntuación obtenida por los diferentes algoritmos en base a los test estadísticos.

	NRD-SBX									SBX								
	MOEA/D			NSGA-II			SMS-EMOA			MOEA/D			NSGA-II			SMS-EMOA		
	↑	↓	Punt	↑	↓	Punt	↑	↓	Punt	↑	↓	Punt	↑	↓	Punt	↑	↓	Punt
WFG1	2.2	0.3	1.9	3.6	0.0	3.6	1.5	0.7	0.8	0.7	2.5	-1.8	1.9	0.3	1.5	0.0	6.0	-6.0
WFG2	0.2	0.0	0.2	0.1	0.0	0.1	0.3	0.0	0.2	0.0	0.4	-0.4	0.0	0.1	-0.1	0.1	0.1	0.0
WFG3	0.1	0.0	0.0	0.0	0.2	-0.2	0.1	0.0	0.1	0.1	0.0	0.0	0.0	0.1	-0.1	0.1	0.0	0.1
WFG4	0.0	0.0	0.0	0.0	0.1	-0.1	0.1	0.0	0.1	0.0	0.0	0.0	0.0	0.1	-0.1	0.1	0.0	0.1
WFG5	0.0	0.5	-0.5	0.2	0.0	0.2	0.2	0.0	0.2	0.0	0.4	-0.3	0.2	0.0	0.2	0.3	0.0	0.3
WFG6	0.5	0.0	0.5	0.4	0.0	0.4	0.5	0.0	0.5	0.0	0.5	-0.5	0.0	0.5	-0.5	0.0	0.4	-0.4
WFG7	0.0	0.0	0.0	0.0	0.1	-0.1	0.1	0.0	0.1	0.0	0.0	0.0	0.0	0.1	-0.1	0.1	0.0	0.1
WFG8	2.9	0.0	2.9	1.3	0.3	1.0	0.9	0.4	0.5	0.1	1.4	-1.4	0.0	1.7	-1.7	0.1	1.4	-1.3
WFG9	1.0	0.1	0.9	1.0	0.0	1.0	1.3	0.0	1.3	0.4	0.7	-0.4	0.0	2.5	-2.5	0.4	0.7	-0.4
Total	**6.8**	0.9	6.0	**6.6**	0.8	5.9	**4.9**	1.2	3.8	**1.2**	6.1	-4.8	**2.1**	5.3	-3.2	**1.1**	8.7	-7.6

su superioridad, se sumó a su puntuación el valor absoluto de la diferencia entre la media de los valores de hipervolumen alcanzados, mientras que en los casos en que se pudo comprobar la inferioridad se restó dicho valor. La Tabla 2 muestra la puntuación obtenida en cada problema para cada par de algoritmo y operador de cruce. En las columnas etiquetadas con ↑ se muestra el acumulado de las puntuaciones positivas, mientras que en las columnas etiquetadas con la columna ↓ se muestra el acumulado de puntuaciones negativas. Finalmente, en la columna *Punt* se muestra la puntuación final. Se puede apreciar que los algoritmos que consideraron el operador NRD-SBX fueron los tres superiores. Entre ellos, el NSGA-II fue el superior, aunque con un rendimiento muy similar al MOEA/D. Por su parte, el rendimiento de los algoritmos con el operador SBX fue claramente inferior independientemente del MOEA aplicado.

6. Conclusiones y trabajos futuros

Los MOEAs son uno de los esquemas más populares para lidiar con problemas de optimización multi-objetivo complejos. En el caso de los entornos continuos, el operador de cruce SBX es uno de los más populares, habiendo mostrado un gran rendimiento tanto en optimización mono-objetivo como multi-objetivo. En este artículo se realiza un análisis del operador SBX y en base a este análisis se propone un nuevo operador, el NRD-SBX. El análisis realizado, muestra que las implementaciones actuales realizan un conjunto de reflexiones que provocan que el operador pueda crear individuos muy alejados de los padres, reduciendo así su capacidad de intensificación. Además, se destaca que el SBX cambia cada variable con una probabilidad igual a 0.5 durante toda la optimización. El NRD-SBX introduce dos cambios principales. Por un lado, evita la utilización de reflexiones con el fin de aumentar el poder de intensificación. Por otro lado, adapta la probabilidad de alterar las variables a lo largo de la ejecución, induciendo un mayor grado de exploración en las fases iniciales que en las fases finales de la optimización. Con el objetivo de validar el NRD-SBX, se realizó un análisis experimental con tres MOEAs muy populares, considerándose el conjunto de

tests WFG. El análisis de las superficies de cubrimiento e hipervolumen muestran las ventajas del NRD-SBX frente al SBX.

Actualmente se están abordando varias líneas de trabajo futuro. Por un lado se está trabajando en operadores de cruce que adapten la probabilidad de realizar reflexiones, así como la apertura de las distribuciones utilizadas. Por otro lado, se están diseñando algoritmos que controlan la diversidad de forma explícita para complementar los cambios realizados en el operador de cruce.

Referencias

1. Agrawal, R.B., Deb, K., Deb, K., Agrawal, R.B.: Simulated binary crossover for continuous search space. Tech. rep. (1994)
2. Beume, N., Naujoks, B., Emmerich, M.: SMS-EMOA: Multiobjective selection based on dominated hypervolume. European Journal of Operational Research 181(3), 1653 – 1669 (2007)
3. Chacón, J., Segura, C., Hernández-Aguirre, A., Miranda, G., León, C.: In: Proceedings of the 2017 on Genetic and Evolutionary Computation Conference Companion. p. In Press. GECCO '17 Companion, ACM (2017)
4. Chiang, T.C., Lai, Y.P.: MOEA/D-AMS: Improving MOEA/D by an adaptive mating selection mechanism. In: 2011 IEEE Congress of Evolutionary Computation (CEC). pp. 1473–1480 (June 2011)
5. Crepinšek, M., Liu, S.H., Mernik, M.: Exploration and Exploitation in Evolutionary Algorithms: A Survey. ACM Computing Surveys 45(3), 35:1–35:33 (Jul 2013)
6. Das, S., Suganthan, P.N.: Differential Evolution: A Survey of the State-of-the-Art. IEEE Transactions on Evolutionary Computation 15(1), 4–31 (Feb 2011)
7. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (Apr 2002)
8. Deb, K., Beyer, H.g.: Self-adaptive genetic algorithms with simulated binary crossover. Evol. Comput. 9(2), 197–221 (Jun 2001), http://dx.doi.org/10.1162/106365601750190406
9. Durillo, J.J., Nebro, A.J., Coello, C.A.C., Garcia-Nieto, J., Luna, F., Alba, E.: A Study of Multiobjective Metaheuristics When Solving Parameter Scalable Problems. IEEE Transactions on Evolutionary Computation 14(4), 618–635 (Aug 2010)
10. Hernández Gómez, R., Coello Coello, C.A., Alba, E.: A Parallel Version of SMS-EMOA for Many-Objective Optimization Problems, pp. 568–577. Springer International Publishing, Cham (2016)
11. Herrera, F., Lozano, M., Sánchez, A.M.: A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study. International Journal of Intelligent Systems 18(3), 309–338 (2003), http://dx.doi.org/10.1002/int.10091
12. Huband, S., Barone, L., While, L., Hingston, P.: A Scalable Multi-objective Test Problem Toolkit, pp. 280–295. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
13. Ishibuchi, H., Masuda, H., Tanigaki, Y., Nojima, Y.: Modified Distance Calculation in Generational Distance and Inverted Generational Distance, pp. 110–125. Springer International Publishing, Cham (2015), http://dx.doi.org/10.1007/978-3-319-15892-1_8

14. Jain, H., Deb, K.: Parent to mean-centric self-adaptation in sbx operator for real-parameter optimization. In: Proceedings of the Second International Conference on Swarm, Evolutionary, and Memetic Computing - Volume Part I. pp. 299–306. SEMCCO'11, Springer-Verlag, Berlin, Heidelberg (2011), http://dx.doi.org/10.1007/978-3-642-27172-4_37
15. jMetalcpp: http://jmetalcpp.sourceforge.net/
16. Jong, K.A.D.: Evolutionary computation - a unified approach. MIT Press (2006)
17. Slim Bechikh, Rituparna Datta, A.G.: Recent Advances in Evolutionary Multi-objective Optimization. springer
18. Talbi, E.G.: Metaheuristics - From Design to Implementation. Wiley (2009)
19. Zhang, Q., Li, H.: MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. IEEE Transactions on Evolutionary Computation 11(6), 712–731 (Dec 2007)
20. Zhu, Q., Lin, Q., Du, Z., Liang, Z., Wang, W., Zhu, Z., Chen, J., Huang, P., Ming, Z.: A novel adaptive hybrid crossover operator for multiobjective evolutionary algorithm. Information Sciences 345, 177 – 198 (2016)

ISSN 1870-4069

Transport Routes Optimization in Martinez de la Torre, Veracruz City, Using Dijkstra's Algorithm with an Additional Parameter

Hugo Lucas-Alvarado, Eddy Sánchez-DelaCruz, R. R. Biswal

Technological Institute of Misantla, Veracruz, Mexico

esanchezd@itsm.edu.mx**

Abstract. A solution to the shortest path problem in Martinez de la Torre, a city in the state of Veracruz, Mexico, is presented. Dijkstra's algorithm was applied to find the shortest route and additional parameter that is used to choose the optimal route was also added. The shortest path problem is a specific case of optimization problems and can be addressed using graph theory. From a computational point of view, graphs are data structures that can be treated as arrays, tables or lists. On the other hand, several organizations in Martinez de la Torre face the problem of finding the optimal route to reduce time and cost, and to offer a better service to its customers. This article addresses the problem of finding an optimal route and implementing the solution to a small portion (area of interest) of Martinez de la Torre City, to test the algorithm and analyze the results, and ultimately leading to a detailed analysis that covers the whole city.

Keywords: Route optimization, Dijkstra's algorithm, shortest route problem, graph, adjacency matrix.

1 Introduction

Optimization problems are presented in diverse industries such as banks, education, freight transport, among others. Many important optimization problems are best analyzed using graphical (graphs) or network representation. Some specific network models are: shortest path problem, maximum-flow problems, minimum spanning tree problems, etc. [6].

Nowadays, systems that provide the best route in a specific geographic area are useful. In many places there are various needs that require these types of systems, to mention a few: distribution networks, road programs, telecommunications networks, electricity networks, among others. Martinez de la Torre, Veracruz is no exception, therefore, in this research we intend to construct a graph representing that city and implementing the Dijkstra's algorithm by adding another parameter to determine the shortest path. For purposes of testing,

** Corresponding author is Eddy Sánchez-DelaCruz.

a zone of interest has been segmented. The graph is constructed taking into consideration, Melchor Ocampo (a district belonging to Martinez de la Torre). It is important to mention that the implementation of Dijkstra's algorithm in this project is useful to find the shortest route.

However, not only is distance important, there are also other parameters to consider when choosing the best route in order to reduce cost and time. It is thus necessary to add them to manipulate them by Dijkstra's algorithm. These parameters are: the sense of streets, severe flooding during the rainy season, restricted access to heavy transport, vehicular flow, breadth of streets, sections in repair, traffic lights, local traffic regulations, insecurity level, among others. The optimization of the routes will allow, various sectors in the Martinez de la Torre City, to be benefited, since they would have at their disposal a useful tool to find the best route.

The present article is structured as follows: section 1 deals with introduction, the problem is mentioned in section 2, Dijkstra's algorithm is described in section 3, section 4 mentions previous work related to Dijkstra' algorithm and optimization of routes, section 5 describes the tools used as well as the methodology to be followed, section 6 shows the experiments performed and the results obtained thereof. Finally, the conclusion and future work are presented in section 7.

2 Problem Formulation

Formally, the route optimization problem can be represented by graphs. Here we mention some important concepts related to the problem that is approached in this investigation.

- **Simple graph.** A simple graph $G = (V, E)$ consists of a set V of vertex (or nodes) and a set E of edges, each edge is a set of two vertices. A simple graph edge has the form $\{v_i, v_j\}$ and for a edge such as this, $\{v_i, v_j\} = \{v_j, v_i\}$. $|V|$ denotes the number of vertices and $|E|$ denotes the number of edges.
- **Directed graphs o digraph.** It consists of a non-empty set V of vertices and a set E of edges (also called arcs). In a digraph, each edge is of the form $\{v_i, v_j\}$ and in this case, $\{v_i, v_j\} \neq \{v_j, v_i\}$.
  It is said that an edge e in a graph that is associated with the pair of vertices u and w is incident on u and w, and it is said that u and w are incidents over e and are adjacent vertices.
- **Weighted graph.** A graph with numbers in the edges is called a weighted graph. If the edge e is labeled k, it is said that the weight of the edge e is k. In a weighted graph, the length of the route is the sum of the weights of the edges on the route.
- **Route (or path).** Let v_0 and v_n vetices in a graph. A path from v_0 to v_n is an alternating sequence of $n + 1$ vertices and n edges that begins at vertex v_0 and ends at vertex v_n,
  $$(v_0, e_1, v_1, e_2, v_2, ..., v_{n-1}, e_n, v_n),$$
  where the edge e_1 is incident on the vertices v_{i-1} and v_i for $i = 1, ..., n$.

– **Graph representation.** Computationally, graphs can be represented by adjacency lists and adjacency and incidence matrices.

In Martinez de la Torre, municipality of Veracruz, there are often situations such as: excessive expenses in the distribution of products by citrus packers, inability to meet the demand for LP Gas distribution, inconsistencies in fuel consumption of distribution vehicles, inefficient taxi service. Among other road problems, these cases have a common denominator: the problem of finding the best route. By solving this problem, it is possible to have more precise control of both time and cost. On the other hand, there are advanced cartographic tools such as Google Maps, which provide relevant information that can be used to build a system to determine the best route. In this research we intend to construct a graph representing the Martinez de la Torre City and implement the Dijkstra's algorithm by adding other parameters to determine the shortest route.

3 Dijkstra's Algorithm

In weighted graphs it is often desired to find the shortest path between two given vertices, that is, the sequence of vertices that must be visited to arrive from a source node to a destination node, whose length is minimal. Dijkstra's algorithm solves this problem efficiently. For the application of Dijkstra's algorithm it is assumed that the weights are positive numbers and that we want to find the shortest path from vertex a to vertex z. Let $L(v)$ be the vertex label v. At any point, some vertices have temporary tags and the rest are permanent tags. Let T be the set of vertices that have temporary tags. If $L(v)$ is the permanent label of vertex v, then $L(v)$ is the length of a shortest path from a to v. At the beginning, all vertices have temporary tags. Each iteration of the algorithm changes the state of a label from temporary to permanent; the algorithm ends when z receives a permanent label [2].

Algorithm 1 finds the length of a shortest path from vertex a to vertex z in a weighted connected graph. The weight of the edge (i, j) is $w(i, j) > 0$, and the vertex label is $L(x)$. When finished, $L(z)$ is the length of the shortest path from a to z.

The above algorithm finds the length of the shortest path from a to z. In most applications, you also want to identify the shortest route. A slight modification to achieve this is to add $L(v)$, the name of the previous vertex (in the path), that is, the label of a vertex is "$L(v), v_a$", where $L(v)$ is the minimum length from a to v, and v_a is the previous vertex in the path. This ensures that on completion, z (destination node) has a label indicating the minimum length from a and it is also possible to move backwards on the labels to find the shortest route. Subsequently, we refer to this modification as the *extended version of the Dijkstra's algorithm*.

In the algorithm, line 5 runs $O(n)$ times. Within the "while" cycle, line 9 takes a time $O(n)$. The body of the cycle "for" (line 12) takes a time $O(n)$.

Algorithm 1 Dijkstra's algorithm

Require: A weighted connected graph in which all weights are positive; vertices a to z

Ensure: $L(z)$, the length of the shortest route from to a to z

```
1: dijkstra(w,a,z,L)
2: {
3:    L(a) = 0
4:    for all vertices x ≠ a do
5:       L(x) = ∞
6:    end for
7:    T = Set of all vertices  //T is the set of all vertices whose shortest distances
      from a have not been found
8:    while z ∈ T do
9:       select v ∈ T with L(v) minimum
10:      T = T − v
11:      for  each x ∈ T adjacent to v do
12:         L(x) = min{L(x), L(v), +w(v, x)}
13:      end for
14:   end while
15: }
```

Since lines 9 and 12 are nested in the "while" cycle that takes a time $O(n)$, the total time for lines 9 and 12 is $O(n^2)$. Then Dijkstra's algorithm runs at a time $O(n^2)$.

For manipulation of an additional parameter (slow traffic), a condition indicating the availability of the edge according to the value of said parameter has been introduced, prior to line 9 of the algorithm. If the edge is not available, then the nature of the algorithm is to continue looking for a vertex with $L(v)$ minimum. In the worst case, if that vertex is not found, that is, if there is no alternate path, the algorithm ignores that parameter, so that it returns a shorter path.

4 Previous Works

The following information is about research that addresses the routes optimization problem, as well as the applications and improvements made to Dijkstra's algorithm:

Edsger W. Dijkstra, solved two problems related to graphs, one of them is to find the total minimum length path between two given nodes P and Q. The solution provided is widely used and is known as Dijkstra's algorithm [1].

Wang Shu-Xi, analyzed the Dijkstra's algorithm and identified some aspects that were not considered in the original problem. Based on this analysis, he proposed improvements to the algorithm and validated it by experiments. The algorithm improvements allow: to control the output mechanism and avoid entering an infinite cycle in case of vertices without connection, save information

about all adjacent vertices and not only the previous vertex in the route, and two (or more) vertices are labeled simultaneously to find the shortest path [5].

Strehler et al., identified, among other things, the need to determine the shortest route and travel planning for use in electric and hybrid vehicles, since it is necessary to regulate speed and range, as well as choose stops for recharging the vehicle battery in a strategic way, the above because the time required for the recharge is greater than the vehicles that use fossil fuels. In the study mentioned, they developed an appropriate model to find the shortest route for such vehicles based on Dijkstra's algorithm and considering that a route may contain cycles leading to a recharge station. The study also considered cases where a charging station could be occupied by another vehicle [4].

Galán-García et al. state that although it is possible to calculate the shortest route using the Dijkstra's algorithm. However, in real life situations this is not always the chosen one. In order to obtain more realistic simulations of traffic flow in intelligent cities, the authors present PEDA (Probabilistic Extension of Dijkstra's algorithm) as an extension to the Dijkstra's algorithm in which they introduce probabilistic changes in the weight of the edges and also in decisions while choosing the shortest path. As an application to the example, they present ATISMART*, which is a model that provides simulations of traffic flow in smart cities using PEDA. When PEDA is applied to the traffic flow, more realistic simulations are obtained, in contrast to the ATISMART model, which uses Dijkstra's algorithm (without probabilistic extension) and therefore saturates those tracks that belong to the shortest route, leaving some map areas almost empty [3].

5 Materials and Methods

5.1 Materials

To obtain the cartographic information, Google Maps, a web application server belonging to Alphabet Inc was used. It offers scrollable map images as well as satellite photographs of the world. In addition, it allows to obtain information about a specific place, to measure the distance between two points, to consult information on vehicular traffic, public transport, routes, among others. For the present investigation, this tool was used to obtain information about Martinez de la Torre, a Veracruz City, from which the graph was constructed.

Through Google Maps, we obtained the Map of Martinez de la Torre City. According to the INEGI Intercensal Survey of 2015, the municipality of Martinez de la Torre has a total population of 110 415 habitants (1.4 % of the state population), a population density of 277.0 hab/km^2 and an area of 815.13 km^2 (0.6 % of the state territory). These data help us to understand that there is a considerable population in the city, which leads to road congestion. Figure 1 a) shows the location of Martinez de la Torre in the state of Veracruz and Fig. 1 b) shows the map of the city. To perform the initial tests, the area of interest shown in Fig. 2 has been selected, in the same image the graph generated is shown.

In the graph, the nodes represent the intersections or ends of streets (numbered from 0 to 37) and the weights of the edges represent the distance (in meters) between the nodes.

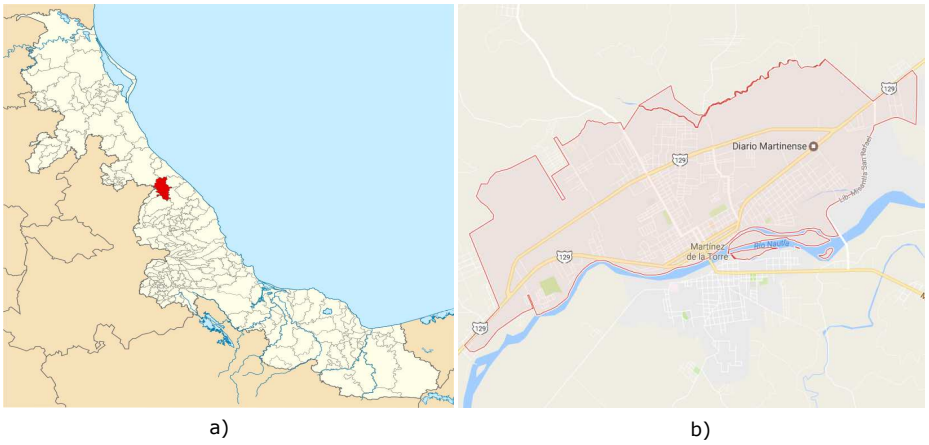


Fig. 1. (a) Location of Martinez de la Torre in the state of Veracruz and (b) Map of the Martinez de la Torre City.

The algorithm is implemented on the information of the graph of the selected region, in addition the algorithm was extended, to consider an additional parameter, as described in the last paragraph of section 3: *Dijkstra's algorithm*. For the implementation, NetBeans IDE 8.2 and Java language were used. The program reads a text file containing the adjacency matrix of the graph, implements the algorithm and finds the shortest path between two points. Object-oriented programming is used, so nodes (or vertices) are objects capable of storing multiple parameters.

5.2 Methodology

Figure 3 shows the flowchart of the methodology to be followed. The process consists of four phases.

As mentioned earlier, the model is intended to be developed for Martinez de la Torre City, and with the aim of having a visual panorama of the problem. In phase one the map of the city was acquired. In the second phase, an area of interest was chosen within Martinez de la Torre City. The basic data were obtained and as there were intersections (nodes) and distance between them (weights of the edges), a graphic representation was created (graph) and finally represented by an adjacency matrix. In phase three the programming of the algorithm in Java was carried out, the *extended version of the Dijkstra's algorithm* mentioned in section 3 is used, which allows to find not only the distance

Fig. 2. Area of interest: Melchor Ocampo District.

of the shorter route (sum of weights) but also the shortest route (sequence of vertices that form the route); and secondly, the adaptations needed to handle some parameters. In this phase the tests were also performed with the original algorithm and adding parameters. In phase four, the results are obtained and analyzed, to determine the correct operation. The data is checked manually and the results are compared with the results of the Google Maps tool.

6 Experiments and Results

For the experiments carried out, a computer with the following specifications was used: Lenovo laptop, YOGA 510-14IKB model, Windows 10 Home Single Language operating system, Intel(R) Core(TM) i7-7500U CPU 2.70 GHz 2.90 GHz, 8.00 GB of RAM, 1.00 TB hard disk capacity, 64-bit operating system and x64 processor.

Strategically, routes have been chosen that could compromise the algorithm or show the management of an additional parameter, specifically those that could have more than one "shorter" route, one-way streets and streets that present road chaos in hours peak. Table 1 shows the results obtained in 8 tests. The columns *Src* and *Dst* indicate the source and destination nodes respectively, the *Dist* column contains the total distance (sum of the weights) of the shortest route and the *Path (without parameters)* column shows the vertices of that route, for tests that consider only the distance (see Fig. 2).

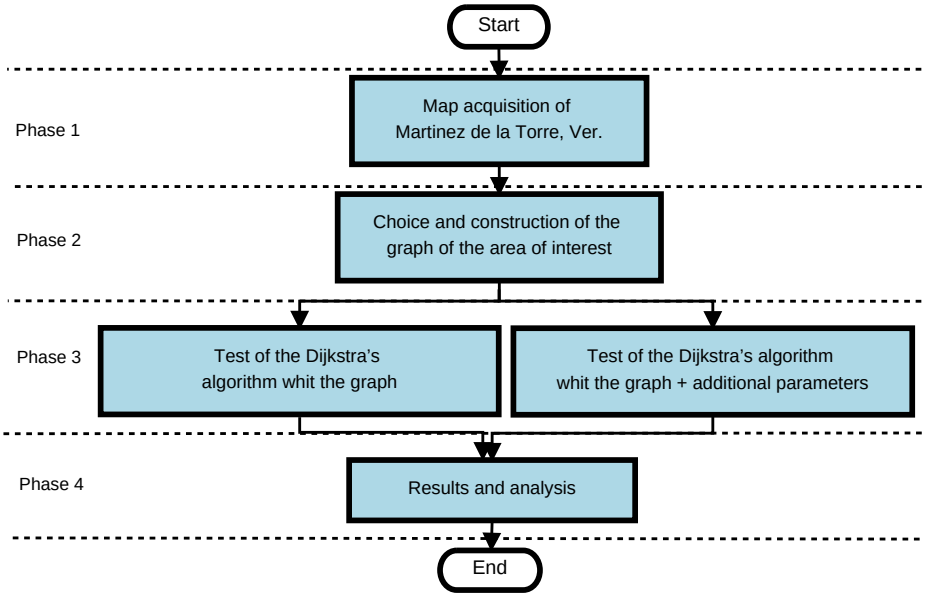


Fig. 3. Proposed methodology.

Table 1. Inputs and outputs of the algorithm in the tests performed.

No.	Src	Dst	Dist	Path (without parameters)	$Dist_p$	Path (with parameters)
1	31	37	289	31-30-33-36-37	289	31-30-33-36-37
2	0	37	665	0-4-10-16-22-26-29-32-35-36-37	850	0-1-2-6-12-25-31-30-33-36-37
3	0	4	65	0-4	311	0-1-5-4
4	0	35	448	0-4-10-16-22-26-29-32-35	879	0-1-2-6-12-25-31-30-33-36-35
5	35	0	448	35-32-29-26-22-16-10-4-0	877	35-36-37-31-25-12-6-2-1-0
6	3	35	760	3-8-14-20-25-31-30-29-32-35	761	3-8-14-20-25-31-30-33-36-35
7	27	30	312	27-26-29-30	563	27-23-17-11-12-25-31-30
8	12	25	415	12-25-20-14-13	415	12-25-20-14-13

Returning to the example of our area of interest, some edges have been marked that show road congestion in peak hours, namely: $(0, 4)$, $(4, 10)$, $(10, 16)$, $(16, 22)$, $(22, 26)$, $(26, 29)$, $(29, 32)$, $(32, 35)$. The $Dist_p$ column contains the total distance of the shortest route and the *Path (with parameters)* column shows the vertices of that route, for the tests that consider a parameter apart from the distance (see Fig. 2).

Note that the routes 2, 3, 4, 5, 6 and 7 are those that, pass through streets that have traffic congestion in the test without parameters. As can be seen, the shorter routes that evade these streets have been chosen, therefore, the expected

results have been achieved, i.e. the shortest route in a section of the city is found considering an additional parameter besides the distance.

It is important to note that additional parameters may belong to both nodes and edges, eg. road congestion and sections in repair are exclusive parameters of the edges (streets), while the presence of traffic lights and the step 1x1 belong to the nodes (intersections). The edge e can be treated as an object with n parameters (one of which is the weight), that is, $e = (p_1, p_2, ..., p_n)$ and the vertex v as an object with m parameters, that is, $v = (p_1, p_2, ..., p_m)$.

7 Conclusion and Future Work

It has been possible to construct the graph of an area of interest of Martinez de la Torre, Veracruz City. The Dijkstra's algorithm has been used to find the shortest route and also, by manipulating other parameter, it has been possible to determine the optimal route. It was possible to create the complete city graph, to determine routes considering various parameters, including travel time. In addition to a real application of the Dijkstra's algorithm, this work is a useful tool for the different organizations that face the problem of finding the optimal route.

It is intended, in a later stage, to construct the complete city graph, to implement the algorithm and to add all the parameters that are of interest for the choice of the optimal route. In addition, it will be necessary to add information from the intersections and streets, which are represented by nodes and edges respectively.

To take advantage of this research, implementing the current model to find the best route in a real application, is not ruled out, to mention a few, in local distribution networks and in the taxi service. To provide greater functionality and to facilitate the conditions to operate dynamically, it is also possible to integrate GPS technology, so that a user (a driver, for example) has a route orientation to follow and reorganize routes from the current location of the vehicle when new destinations are required.

References

1. Dijkstra, E.W.: A note on two problems in connexion with graphs. Mathematisch Centrum (1959) 4
2. Johnsonbaughc, R.: Matemáticas discretas. Pearson Educación (2005) 3
3. José L. Galán-García, Gabriel Aguilera-Venegas, M.A.G.G.: A new probabilistic extension of dijkstra's algorithm to simulate more realistic traffic flow in a smart city. ELSEVIER (2014) 5
4. Martin Strehler, Soren Merting, C.S.: Energy-efficient shortest routes for electric and hybrid vehicles. ELSEVIER (2017) 5
5. Shu-Xi, W.: The improved dijkstra's shortest path algorithm and its application. IWIEE (2012) 5
6. Winston, W.L.: Investigación de operaciones Aplicaciones y algoritmos. CENGAGE Learning (2005) 1

ISSN 1870-4069

Análisis multiobjetivo de la selección de líderes en redes inalámbricas de sensores

Karen Miranda, Antonio López-Jaimes, Abel García-Nájera

Universidad Autónoma Metropolitana (UAM),
Depto. de Matemáticas Aplicadas y Sistemas,
Cuajimalpa, Ciudad de México, México

{kmiranda,alopez,agarcian}@correo.cua.uam.mx

Resumen. Una red sensora consiste en un conjunto de sensores para monitorizar las condiciones ambientales. Los sensores transmiten cooperativamente la información recolectada a través de la red a una estación base que se encarga de recolectar la información de todos los sensores. Para ahorrar energía, los sensores se agrupan y eligen a un líder para que se encargue de reenviar los datos recolectados a la estación base. La selección del líder puede hacerse de manera aleatoria o con criterios específicos como energía residual, la distancia o la conectividad; así, el problema de agrupamiento y selección de líder es un problema NP-difícil. En este trabajo se analiza el problema de la selección de líderes de grupo desde una perspectiva multiobjetivo para determinar la pertinencia de algunos los objetivos utilizados que se pueden encontrar en la literatura.

Palabras clave: Optimización multi-objetivo, algoritmos genéticos, redes inalámbricas de sensores, cluster head.

A Multi-objective Analysis of the Cluster-head Selection Problem in WSN

Abstract. A Wireless Sensor Network (WSN) is composed of a set of energy and processing-constrained devices that gather data about a set of phenomena. An efficient way to enlarge the lifetime of a wireless sensor network is clustering organization, which structures hierarchically the sensors in groups and assigns one of them as a cluster-head. Such a head is in charge of specific tasks as data gathering from other cluster sensors and resending that data through the network to the base station. The cluster-head selection may be random or based on well defined criteria such as residual energy, node distance, signal strength or connectivity; hence, this problem is NP-hard for WSN. In the literature we may find proposals based on heuristics algorithms that consider at the same time different objectives and purposes. Therefore, in this paper we perform an analysis from the many-objective perspective about the pertinence of the most commonly used objectives.

Keywords: Multi-objective optimization, genetic algorithms, wireless sensor networks, cluster head.

1. Introducción

Las redes inalámbricas de sensores (Wireless Sensor Networks) están compuestas por un gran número de pequeños dispositivos electrónicos que son capaces de monitorizar condiciones ambientales, tales como, la temperatura, la humedad o la velocidad del viento. Los sensores recolectan datos sobre un fenómeno dado y reenvían esos datos a una estación base a través de la red que los mismos sensores forman. Estos dispositivos son de bajo costo y fáciles de desplegar; sin embargo, también tienen capacidad de memoria y de energía limitada. Así, se busca que la red sensora esté activa y trabajando el mayor tiempo posible por lo que es necesario que cada sensor administre sus recursos de forma autónoma y que permita el ahorro de energía [6].

Una manera de optimizar el tiempo de vida una red sensora es organizar jerárquicamente a los sensores, formando grupos o *clusters* y seleccionando un líder por cada cluster (o *cluster head* del inglés) [14]. Los líderes están a cargo de concentrar la información que los otros sensores han recolectado y a su vez la reenvía a la estación base. El agrupamiento es una técnica eficiente para ahorrar energía; paradójicamente, al llevar a cabo estas tareas adicionales, los líderes gastan más rápidamente su propia energía, por lo que cada vez que un líder se queda sin batería, es necesario elegir un nuevo líder de entre los sensores restantes.

La selección de los líderes puede seguir diferentes criterios como la energía residual, la distancia entre sensores, la potencia de la señal o la conectividad e incluso una combinación de criterios. Estos criterios se utilizan en función de los objetivos que se desean optimizar, por ejemplo, maximizar el tiempo de vida de la red o minimizar el consumo de energía. Este problema de selección se puede ver como un proceso de toma de decisiones multicriterio. Dependiendo del propósito particular de la red sensora, i.e., en la agricultura, monitorización de la vida silvestre o condiciones climatológicas podemos encontrar una gran variedad objetivos que se desean optimizar [9]. Por lo tanto, es importante determinar cuáles de esos objetivos son relevantes para el problema, cuáles son deseables y cuáles se satisfacen al optimizar otros objetivos ya que se sabe que no todos los objetivos son necesarios [4]. En general, el objetivo primordial es maximizar el tiempo de vida de la red.

En este trabajo se presenta un análisis de tres de los objetivos que se encuentran comúnmente en la literatura [2]: *i)* minimizar la distancia entre miembros del grupo; *ii)* minimizar la distancia de los líderes a la base y; *iii)* maximizar la energía residual de los líderes. Para ello, realizamos un estudio experimental utilizando el bien conocido algoritmo NSGA-II y donde analizamos la formación de grupos de sensores y la relación de conflicto entre los objetivos.

El resto del documento está estructurado de la siguiente manera. En la Sección 2 presentamos el problema de selección de líder en el contexto de las

redes inalámbricas de sensores así como el modelo del sistema y de energía. En la Sección 3, describimos las bases de problemas multiobjetivo, conflicto entre objetivos y presentamos el problema de selección de líder como un problema multiobjetivo. En la Sección 4, recordamos los conceptos básicos del NSGA-II, así como la representación de las soluciones como cadenas binarias. Presentamos los resultados del estudio experimental en la Sección 5 y finalmente, concluimos este trabajo en la Sección 6.

2. Problema de selección de líder

Las redes inalámbricas de sensores son ampliamente usadas en la actualidad. Entre algunas de las aplicaciones se encuentran la agricultura de alta precisión, la motorización de volcanes, mapeo de la biodiversidad, la domótica (automatización de casas y edificios) y en la concepción de ciudades inteligentes [11]. Las redes de sensores se pueden clasificar por su propósito y sus capacidades físicas específicas. En particular, en este trabajo nos enfocamos a las redes donde los sensores se despliegan una vez y permanecen en el mismo lugar hasta que termina su ciclo de vida.

Dado que el objetivo de una red sensora es proveer información sobre diferentes fenómenos y muchas veces se despliegan en zonas de difícil acceso, se busca que dicha red cumpla su objetivo durante el mayor tiempo posible. Existen diferentes técnicas de ahorro energético para redes con estas características, por ejemplo, reducción de paquetes de datos o control de topología [3]. Particularmente, el agrupamiento (del inglés *clustering*) es un método para reducir la cantidad de datos transmitidos a través de la red hacia una estación base donde se caracterizan los datos y se obtiene información.

Este método consiste en formar grupos de sensores y entre ellos elegir uno que funja como líder de cada grupo [14]. El líder es el encargado recolectar los datos en el grupo y determina cuales de ellos son relevantes para ser reenviados a la estación base, reduciendo así la cantidad de datos intercambiados entre sensores y por consecuencia ahorrando energía [12].

Existen diferentes propuestas para el agrupamiento y la selección de líderes. Por ejemplo, los más sencillos únicamente consideran la energía residual de los nodos y eligen como líder a aquel con la mayor energía. Si bien es una regla simple que no consume recursos suplementarios para su ejecución, no siempre es útil para las aplicaciones específicas de las redes sensoras. De hecho, los objetivos a considerar para seleccionar a los líderes depende, precisamente, de la propia aplicación. Algunos ejemplos de objetivos, no restrictivos, son: escalabilidad, tolerancia a fallas, balance de carga, tiempo de vida de la red o incrementar la zona de cobertura de los sensores [2].

En general, se puede considerar un problema genérico de optimización multiobjetivo para asignación de recursos con entradas, salidas requeridas, objetivos y restricciones [9]. Así, se pueden ver reflejados objetivos específicos como maximizar el tiempo de vida de la red, maximizar la cobertura, minimizar el costo, minimizar el consumo de energía o maximizar la utilización del espectro.

2.1. Modelo de la red de sensores

Una red de sensores se representa como un gráfo $G = V, E$, donde V es un conjunto finito de n sensores V_i más una estación base V_B y E es el conjunto finito de m conexiones entre sensores en la red. Por conveniencia, se asume que: *i)* los sensores son distribuidos aleatoriamente sobre un espacio cuadrado de dos dimensiones; *ii)* los sensores son estáticos una vez que son desplegados; *iii)* la energía de los sensores no es renovable; *iv)* inicialmente todos los sensores tienen la misma cantidad de energía; *v)* los enlaces de comunicación son bidireccionales; *vi)* las capacidades computacionales son las mismas para todos los sensores; *vii)* los sensores no conocen su ubicación precisa.

2.2. Modelo de energía

Dado que la mayoría de los objetivos en el análisis del consumo de energía de los líderes con base en la distancia entre los miembros del grupo y la distancia entre el líder y la estación base, es necesario establecer el modelo de energía utilizado en los casos de estudio enfocado a los métodos de agrupamiento. Para ello, usamos la notación y valores de los parámetros usados para nuestras evaluaciones como se describe en la Tabla 1 y el modelo de energía como sigue [13]:

Tabla 1. Notación.

Significado	Símbolo	Valor
Energía Inicial	E_{I}	9 J
Nodos	N	20–500
Número de Miembros en el *cluster*	N_{M}	
Miembro	M	100μJ
Líder	LG	
Estación Base	EB	Posición (0,0)
Tamaño del mensaje	k	2000 bits
Energía usada para transmitir	E_{tx}	
Consumo electrónico	E_{elec}	50 nJ/bit
Energía usada por líder de grupo por miembro por mensaje	E_{LGM}	
Distancia al líder	D_{LG}	
Distancia a la EB	D_{EB}	
Energía por transmisión múltiple	ϵ_{amp}	
Área	(m × m)	400 × 400

3. Problemas de optimización multiobjetivo

Muchos problemas reales tienen no sólo una función objetivo, sino m de ellas. En este caso, tenemos el vector $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$ de m funciones objetivo, en donde $\mathbf{f} : \mathcal{X} \rightarrow \mathbb{R}^m$ define el espacio objetivo. Entonces, tenemos que

minimizar: $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$ sujeto a $g_i(\mathbf{x}) = 0$, $i = 1, \ldots, p$, $h_j(\mathbf{x}) \leq 0$, $j = 1, \ldots, q$.

Los principales problemas de optimización pertenecen a la clase de complejidad NP-difícil, esto quiere decir que no se conoce un método exacto que pueda resolver todos los casos del problema de forma eficiente. No obstante, los métodos heurísticos, a pesar de no ofrecer una garantía de desempeño, usualmente encuentran soluciones con ciertos criterios de calidad en un tiempo razonable. Se sabe que para los problemas que involucra agrupar miembros, específicamente el agrupamiento de sensores, son problemas de clase NP-difícil [8].

3.1. Conflicto entre objetivos

Una propiedad importante de un problema multiobjetivo es el conflicto entre sus objetivos. Si los objetivos no tienen conflicto entre ellos, entonces podríamos resolver el problema optimizando cada objetivo de manera independiente. Sin embargo, en algunos problemas, aunque existe cierto conflicto, entre algunos objetivos no hay conflicto. Aunque diferentes autores han propuesto una definición para conflicto (véase por ejemplo, [4, 5]), en este documento utilizamos la definición propuesta por Carlsson y Fullér [5], que es la más intuitiva.

Definición 1. *Sea $S_{\mathcal{X}}$ un subconjunto de $\mathcal{X}$, entonces, dos objetivos pueden estar relacionados de las siguientes maneras (suponiendo minimización):*

1. *f_i está en conflicto con f_j en $S_{\mathcal{X}}$ si $f_i(\mathbf{x}^1) \leq f_i(\mathbf{x}^2)$ implica que $f_j(\mathbf{x}^1) \geq f_j(\mathbf{x}^2)$ para toda $\mathbf{x}^1, \mathbf{x}^2 \in S_{\mathcal{X}}$.*
2. *f_i apoya a f_j en $S_{\mathcal{X}}$ si $f_i(\mathbf{x}^1) \geq f_i(\mathbf{x}^2)$ implica $f_j(\mathbf{x}^1) \geq f_j(\mathbf{x}^2)$ para toda $\mathbf{x}^1, \mathbf{x}^2 \in S_{\mathcal{X}}$.*
3. *f_i y f_j son independientes en $S_{\mathcal{X}}$, en otro caso.*

En los casos 2 y 3, estos objetivos son llamados no conflictivos o redundantes.

3.2. Selección del líder como problema multiobjetivo

En una red de sensores, el objetivo que se busca mejorar en la mayoría de los casos es el tiempo de vida de la red. Sin embargo, este objetivo no se puede evaluar directamente ya que solamente hasta que se pone a funcionar la red sabemos el tiempo de vida real de la red.

Por esta razón, en varias propuestas [1, 10] se han utilizado otros objetivos que al parecer contribuyen a maximizar el tiempo de vida de la red. Entre estos objetivos podemos encontrar:

1. Minimizar la distancia de los líderes a la base, ya que estos sensores son los que envían más mensajes.
2. Minimizar la distancia de cada sensor a su líder, ya que evita que líderes potenciales consuman rápidamente su energía.
3. Maximizar la energía residual de los líderes, ya que de esta manera será menos frecuente tener que reemplazar a un líder.

4. Optimizar el balance de carga, lo cual ayuda a que los líderes necesiten ser reemplazados en tiempos similares.

En varios trabajos se utilizan estos objetivos de manera individual [9], mientras que en otros se utilizan simultáneamente [2]. Sin embargo, aún no se conoce si todos estos objetivos son necesarios para encontrar la solución óptima. Como se mencionó en la sección anterior, en ciertos problemas algunos objetivos no son necesarios o esenciales [4, 5].

Por este motivo, tomaremos los tres objetivos más utilizados para mejorar el tiempo de vida de una red de sensores con el fin de analizar si todos los objetivos son esenciales o hay alguno redundante en un planteamiento multiobjetivo. Los tres objetivos que adoptamos en el estudio son los tres primeros de la lista anterior.

4. Algoritmo para resolver el problema de selección

4.1. Algoritmo de optimización

Puesto que este estudio está enfocado al análisis de las características del problema, como algoritmo de optimización utilizamos uno de los algoritmos evolutivos más usados y conocidos, es decir, el *Non-dominated Sorting Genetic Algorithm II* (NSGA-II) [7]. Si bien este algoritmo ya no se considera del estado del arte, utilizando un número adecuado de individuos y generaciones podemos obtener soluciones representativas del frente de Pareto óptimo.

Este algoritmo utiliza una clasificación por capas de los individuos de acuerdo a su no dominancia. Para esto se obtienen los individuos que dominan al resto de la población. Este grupo de individuos conforma el primer frente no dominado y se les asigna un valor de aptitud de uno. Posteriormente, estos individuos son ignorados momentáneamente y se genera el segundo frente no dominado de individuos, a los cuales se les asigna una aptitud aumentada en uno. Este proceso continúa hasta que todos los individuos están clasificados.

4.2. Representación de las soluciones

Un parte importante para aplicar NSGA-II al problema de selección del líder es la representación de las soluciones y las suposiciones acerca del problema.

Para representar una solución utilizamos una cadena binaria $\mathbf{x} \in \{0,1\}^N$, donde N es el número de sensores. De esta manera, tenemos que $x_i = 1$ si el i-ésimo sensor es líder, y $x_i = 0$ si el sensor es parte de un grupo (sensor miembro).

Para facilitar la notación consideremos que los índices de los líderes de grupo conforman el conjunto $L = \{i \,|\, x_i = 1\}$ y que la posición del i-ésimo sensor se denota mediante $\mathbf{s}^{(i)} \in \mathbb{R}^2$. Por otra parte, la posición de la base está dada por $\mathbf{s}^{(\text{base})}$.

Así, para determinar a qué grupo pertenece un sensor miembro, buscamos el líder más cercano. Es decir, el grupo $G(\mathbf{s}^{(i)})$ al que pertenece el sensor $\mathbf{s}^{(i)}$ $(i = 1, \ldots, N)$ está dado por

$$G(\mathbf{s}^{(i)}) = \arg\min_{j \in L} \|\mathbf{s}^{(i)} - \mathbf{s}^{(j)}\|,$$

donde $\|\cdot\|$ es la distancia Euclideana entre dos puntos. El conjunto de índices de los sensores que son miembros del grupo de un líder j se denota mediante $M_j = \{i \,|\, G(\mathbf{s}^{(i)}) = j, i = 1, \ldots, N\}$. Nótese que con esta definición cada grupo tendrá al menos como único miembro a un líder. Así, nuestros objetivos se definen de la siguiente manera:

1. Minimizar distancia de los líderes L a la base:

$$\text{Min} \quad f_1(\mathbf{x}) = \frac{1}{|L|} \sum_{j \in L} \|\mathbf{s}^{(j)} - \mathbf{s}^{(\text{base})}\|.$$

2. Minimizar distancia de los sensores a su líder de grupo:

$$\text{Min} \quad f_2(\mathbf{x}) = \frac{1}{|L|} \sum_{j \in L} \left(\frac{1}{|M_j|} \sum_{i \in M_j} \|\mathbf{s}^{(i)} - \mathbf{s}^{(j)}\| \right).$$

3. Maximizar energía residual de los líderes de cada grupo:

$$f_3(\mathbf{x}) = \frac{1}{|L|} \sum_{j \in L} E_I - LG_j,$$

donde E_I y LG son los valores definidos en la Tabla 1.

Con esta representación utilizamos la cruza binaria de dos puntos y la mutación uniforme. Solamente tuvimos que hacer una ligera modificación para evitar soluciones con cero líderes. Al final de la cruza y mutación, si el resultado es la cadena binaria con ceros solamente, entonces en una posición aleatoria de la cadena se escribe un uno.

5. Estudio experimental

Para realizar los experimentos consideramos instancias del problema donde los sensores están ubicados de manera aleatoria en una cuadrícula usando los parámetros descritos en la Tabla 1. Para todas las instancias, excepto para 20 sensores, se realizaron simulaciones usando el NSGA-II con 2000 individuos evolucionados durante $20,000$ generaciones. El porcentaje de cruza utilizado fue 0.9, mientras que el porcentaje de mutación fue de $1/N$, donde N es el número de sensores. En el caso de 20 sensores fue posible obtener la solución óptima utilizando búsqueda exhaustiva.

Es importante mencionar que para facilitar el planteamiento del problema en NGSGA-II, el objetivo para maximizar f_3 (la energía residual) fue transformado

a un objetivo de minimización. Es decir, en la discusión de las secciones siguientes el objetivo f_3 es minimizar la energía utilizada por los líderes de los grupos.

Para mostrar la forma y tipo de frente de Pareto que se produce al minimizar los tres objetivos se presenta la Fig. 1. La figura muestra el frente de Pareto obtenido por NSGA-II para una instancia con 300 sensores. Note que el tercer objetivo (eje z) se refiere a la energía utilizada por los líderes al recibir y enviar mensajes (la cual fácilmente se puede interpretar como energía residual). A diferencia de los frentes de Pareto típicos de problemas con 3 objetivos, en este caso el frente no es una superficie sino que parece ser una línea (objeto en una dimensión).

En las siguientes secciones se analizará con detalle el conflicto entre los objetivos del problema.

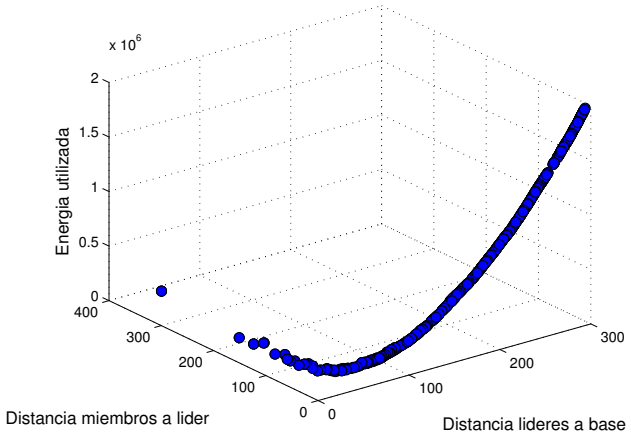


Fig. 1. Vista en 3D del frente de Pareto obtenido para una instancia con 300 sensores.

5.1. Formación de grupos de sensores

En esta sección se muestra la forma de los grupos de sensores y sus líderes según el compromiso entre los 3 objetivos del problema. En la Fig. 2 se ilustra el frente de Pareto óptimo compuesto por 82 soluciones para una instancia con solamente 20 sensores. Dado que que para 20 sensores solamente hay $2^{20} \approx 10^6$ soluciones posibles, obtuvimos el conjunto de óptimos completo realizando una búsqueda exhaustiva al evaluar todas las soluciones posibles.

Para tener una idea del tipo de configuraciones de grupos de sensores y sus líderes, en las Fig. 3-5 se muestran 3 configuraciones de agrupaciones posibles que representan soluciones óptimas de Pareto de la Fig. 2. En primer lugar tenemos la solución que obtiene el mejor valor para la distancia del líder a la base (véase

Fig. 3). En esta figura se muestra la posición de los 20 veinte sensores utilizando círculos, mientras que el sensor líder es el círculo con un punto en su interior.

Recordemos que la base en todos los casos está localizada en la posición $(0, 0)$. Puesto que para esta pequeña instancia no hay conflicto entre este objetivo y la energía utilizada, la solución que minimiza la energía es la misma configuración de grupos. Es decir, la solución la solución óptima en términos de distancia a la base y energía es aquella que forma un único grupo donde el líder es el sensor más cercano al origen (la base).

En el otro extremo, tenemos a la solución que minimiza la distancia de los miembros del grupo a su líder. La Fig. 4 muestra los grupos conformados para esta solución óptima de Pareto. Como era de esperarse, para minimizar la distancia a su líder se forman grupos con un solo sensor, el cual es su propio líder. Como se muestra en la figura, hay 20 grupos denotados con un símbolo diferente y su líder es indicado mediante un punto en el centro. En otras palabras, esta solución es aquella donde no se utiliza el esquema de líderes ya que cada sensor transmite directamente a la base. Si bien esta configuración minimiza la distancia a los líderes también es la que más energía consume ya que la base está muy alejada para la mayoría de los líderes.

En la práctica, una solución que se podría elegir para implementar es alguna ubicada en la zona media del frente Pareto (llamada "rodilla" del frente en optimización multiobjetivo). La distribución de los grupos de esta solución de la rodilla del frente se muestra en la Figura 5. En este caso, se formaron 5 grupos, de los cuales 3 tienen solamente un sensor. Los otros 2 grupos tiene 9 y 8 sensores, respectivamente. Como se puede observar en la figura, los líderes de esta solución son aquellos sensores más cercanos a la base. Aunque no se muestra en este documento, para instancias con muchos más sensores (300 o 400) el sensor más cercano a la base también fue seleccionado como líder del grupo.

5.2. Relación de conflicto entres los objetivos

En esta sección se presenta un análisis para conocer la relación de conflicto entre cada par de los objetivos del problema de selección del líder. Para estimar el conflicto entre los objetivos utilizamos el coeficiente de correlación de Spearman. Este coeficiente mide la dependencia entre dos variables representadas por dos muestras de puntos (en nuestro caso, el conjunto de soluciones en el espacio objetivo). A diferencia del coeficiente de correlación de Pearson, la dependencia no necesariamente debe ser lineal, ya que solamente necesita describir una función monótona (creciente o decreciente). Los valores del coeficiente de Spearman caen en el intervalo $[-1, 1]$, donde los valores límite -1 y 1 significan que el par de variables siguen un orden monótono perfecto decreciente o creciente, respectivamente. En términos de conflicto, dos objetivos están en conflicto si el coeficiente de correlación es -1, mientras que los objetivos se apoyan cuando el valor de correlación es 1. Si tenemos un valor cercano a cero significa que los objetivos son independientes.

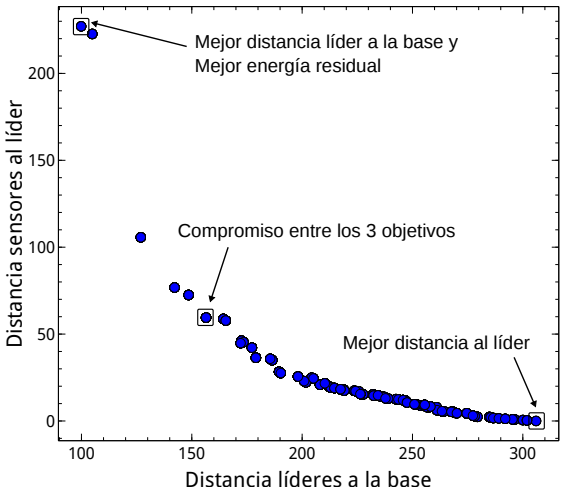


Fig. 2. Vista de 2 objetivos del frente de Pareto de una instancia con 20 sensores.

Para aplicar este coeficiente de correlación al problema de los líderes, utilizamos el NSGA-II para generar aproximaciones del frente de Pareto. Para obtener una buena aproximación del frente, para cada instancia, utilizamos una población de 2000 individuos durante 20000 generaciones. Es decir, un total de 4×10^7 evaluaciones de las tres funciones objetivo. Este número de evaluaciones significa que la porción estimada de espacio explorado por NSGA-II es de $4 \times 10^7/2^{100} = 3.1 \times 10^{-23}, 2.4 \times 10^{-53}, 1.9 \times 10^{-83}, 1.5 \times 10^{-113}$, para instancias de 100, 200, 300 y 400 sensores.

Para determinar de manera objetiva si los objetivos están correlacionados o son independientes utilizamos una prueba estadística (nivel de significación de 1 %) en la cual la hipótesis alternativa es que el coeficiente de Spearman sea diferente de cero para cada par de valores objetivo.

En la Tabla 2 se muestran los coeficientes de correlación de Spearman para cada par de objetivos del problema usando 5 instancias variando el número de sensores: 20, 100, 200, 300 y 400 sensores. En todos los casos que aparecen en dicha tabla se aceptó la hipótesis de que el coeficiente es diferente de cero. En términos generales, podemos decir que según el coeficiente de Spearman, hay claro conflicto entre f_1 y f_2 sin importar el número de sensores, es decir, la distancia de los líderes a la base y la distancia de los miembros de grupo a su líder. Esto quiere decir que si queremos reducir la distancia de los líderes a la base, esto se hará a expensas de empeorar la distancia de los miembros de un grupo a su líder.

De igual manera, hay conflicto entre f_2 y f_3, la distancia de los miembros de grupo a su líder y la energía utilizada. Esto quiere decir que si hacemos más compactos los grupos, entonces la energía utilizada por los líderes aumentará en la mayoría de los casos. Por otra parte, entre los objetivos f_1 y f_3 (distancia de

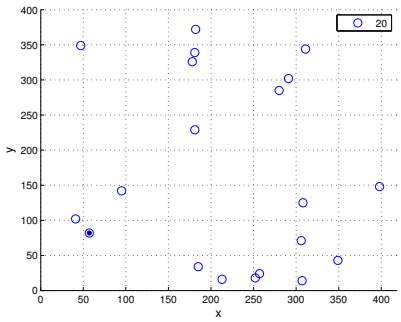


Fig. 3. Grupo único formado con 20 sensores para la solución con la mejor distancia de líderes a la base y la mejor energía residual. El círculo con un punto interior representa el único líder.

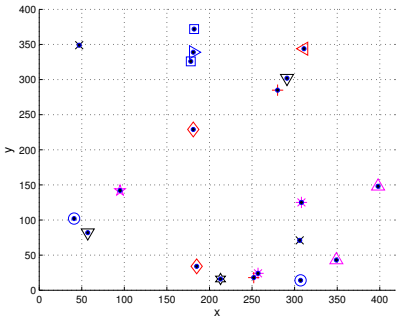


Fig. 4. Grupos formados con 20 sensores para la solución con la mejor distancia de los miembros a su líder. Los símbolos con un punto interior representan un líder.

los líderes a la base y la energía que utilizan) no hay conflicto en la mayoría de los casos.

Observando los resultados de la Tabla 2 para los casos de 300 y, principalmente, para 400 sensores, aunque aún la tendencia para f_1 y f_3 es de apoyo, y para f_2 y f_3 de conflicto, el valor absoluto de ambos valores no es tan grande como para los otros casos. La explicación de estos valores es clara al observar a detalle el frente de Pareto de f_1 y f_3 que se muestra en la Fig. 6. En la mayoría de las configuraciones de grupos formados, al disminuir la distancia de los líderes a la base también se reduce la energía que utilizan. No obstante, al acercarse a una distancia aproximada a 35m o menor, entonces la relación de dependencia cambia, es decir, se establece una relación de conflicto entre estos 2 objetivos. Esto explica por qué el coeficiente de Spearman aunque es positivo está más cercano a cero.

En términos de la red sensores este compromiso ocurre por la siguiente razón. La energía utilizada está determinada principalmente por el consumo necesario para enviar mensajes a la base (dependiente de la distancia). Sin embargo, también entra en juego la energía utilizada para recibir mensajes de los miembros del grupo (dependiente solamente del número de miembros). Así, la distancia a la base es prioridad para ahorrar energía, pero en cierto punto este ahorro ya no es suficiente para contrarrestar el gasto al recibir mensajes. Para nuestra instancia de 400 sensores distribuidos en un área de 400m $\times$ 400m, este umbral de de 35m aproximadamente. Para otras instancias conocer el frente de Pareto sería útil para saber la distancia mínima para elegir los líderes sin aumentar la energía utilizada.

En la Fig. 7 se muestran la proyección del frente de Pareto con 400 sensores en los planos f_1-f_3 y f_2-f_3. Nuevamente se resalta la zona de conflicto para f_1-f_3, pero es importante notar que también para la distancia de los miembros

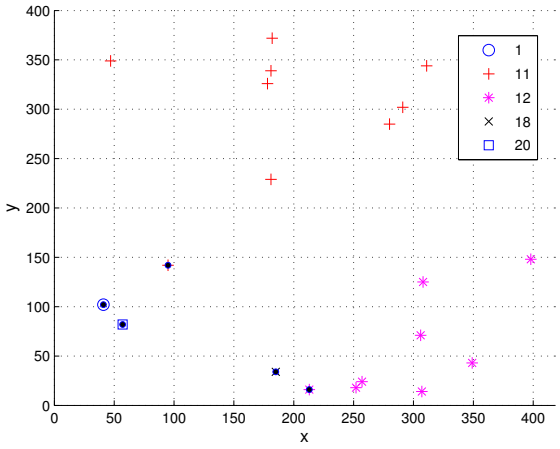


Fig. 5. Cinco grupos formados con 20 sensores para una solución que representa un compromiso intermedio entre los 3 objetivos. Los símbolos con un punto interior representan un líder.

de grupo a su líder y la energía utilizada, en la mayoría de los casos hay una relación de conflicto (de 0 a 50 metros apróx.). Sin embargo, más allá de 50m hay una relación de apoyo entre los objetivos, lo cual explica el valor del coeficiente correlación negativo pero cercano a cero para 300 y 400 sensores. Por cuestiones de espacio no se muestra el frente de Pareto para 20 y 100 sensores, pero para estos 2 casos, los frentes no tienen regiones mixtas de apoyo y conflicto. Es decir, para f_1-f_3 toda la curva es de apoyo, mientras que para f_2-f_3 solamente hay relación de conflicto.

Tabla 2. Coeficientes de correlación de Spearman para estimar el conflicto entre objetivos.

# sensores	f_1 vs f_2	f_1 vs f_3	f_2 vs f_3
20	-0.9993	0.9990	-0.9997
100	-0.9998	0.9685	-0.9687
200	-1.0000	0.9996	-0.9996
300	-1.0000	0.9975	-0.9976
400	-0.9203	**0.2907**	**-0.4281**

6. Conclusiones

En este documento se presentó un análisis de tres de los objetivos que se pueden encontrar comúnmente en la literatura minimizar la distancia entre

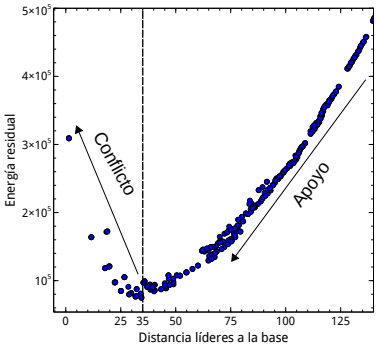
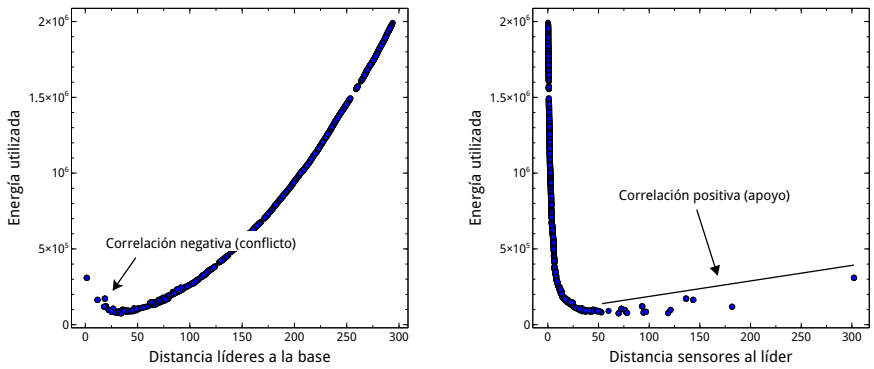


Fig. 6. Ilustración del conflicto entre distancia de los líderes a la base y la energía utilizada a partir de cierta distancia.



f_1 vs f_3 f_2 vs f_3

Fig. 7. Frente de Pareto obtenido para una instancia con 400 sensores.

miembros del grupo, minimizar la distancia de los líderes a la base y maximizar la energía residual de los líderes. Para ello, realizamos un estudio experimental utilizando el bien conocido algoritmo NSGA-II y donde analizamos la formación de grupos de sensores y la relación de conflicto entre los objetivos.

Respecto al análisis de conflicto entre los objetivos encontramos que bajo ciertas condiciones hay conflicto entre cada par de objetivos del problema de selección del líder. En particular, hay claro conflicto entres los dos pares de objetivos siguientes: distancia de los líderes a la base y distancia de los sensores a su líder; y distancia de los sensores a su líder y energía residual. Por otra parte, hay una la relación mixta entre los objetivos distancia de los líderes a la base y energía residual. De una distancia entre 0 y 35 m (esta distancia depende del número de sensores y área en la que se distribuyen los sensores) hay una relación de conflicto, mientras que a una distancia mayor hay relación de apoyo. Esta distancia de umbral donde cambia la relación entre objetivos puede conocer hasta encontrar el frente de Pareto de cada instancia. En la práctica, los resultados

en cuanto a conflicto significan que los tres objetivos necesitan ser optimizados simultáneamente para optimizarlos ya que ningún objetivo es redundante.

En cuanto al tipo de grupos de sensores de la soluciones óptimos se observó que el líder de cada grupo siempre es el sensor más cercano a la base. De igual manera, la solución para minimizar la energía residual es una configuración con un solo grupo. Sin embargo, es importante notar que esta solución solamente toma en cuenta una ronda de comunicación donde no hay cambio de líderes.

Finalmente, como trabajo futuro extenderemos este estudio con más objetivos representativos de otras características de las redes; además consideraremos el escenario donde hay cambio de líderes y donde deberían reconfigurarse los *clusters*. Asímismo, se buscará la incorporación de los algoritmos evolutivos en un simulador de redes como NS-2 o WSNet.

Referencias

1. Abo-Zahhad, M., Ahmed, S.M., Sabor, N., Sasaki, S.: A New Energy-Efficient Adaptive Clustering Protocol Based on Genetic Algorithm for Improving the Lifetime and the Stable Period of Wireless Sensor Networks. International Journal of Energy, Information and Communications (2014)
2. Afsar, M.M., Tayarani-N, M.H.: Clustering in sensor networks: A literature survey. Journal of Network and Computer Applications 46, 198–226 (Nov 2014)
3. Anastasi, G., Conti, M., Di Francesco, M., Passarella, A.: Energy conservation in wireless sensor networks: A survey. Ad Hoc Networks 7(3), 537–568 (2009)
4. Brockhoff, D., Zitzler, E.: Are All Objectives Necessary? On Dimensionality Reduction in Evolutionary Multiobjective Optimization. In: Parallel Problem Solving from Nature - PPSN IX, pp. 533–542 (2006)
5. Carlsson, C., Fullér, R.: Interdependence in fuzzy multiple objective programming. Fuzzy Sets and Systems 65(1), 19–30 (1994)
6. da Cunha, A.B., da Silva, D.C.: Behavioral Model of Alkaline Batteries for Wireless Sensor Networks. IEEE Latin America Transactions 10(1), 1295–1304 (Jan 2012)
7. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (April 2002)
8. Elhabyan, R., Yagoub, M.C.: PSO-HC: Particle Swarm Optimization Protocol for Hierarchical Clustering in Wireless Sensor Networks. In: Proceedings of CollaborateCom. pp. 417–424. Miami, FL, USA (Oct 2014)
9. Iqbal, M., Naeem, M., Anpalagan, A., Qadri, N., Imran, M.: Multi-objective optimization in sensor networks: Optimization classification, applications and solution approaches. Computer Networks 99, 134–161 (Apr 2016)
10. Kheireddine, M., Abdellatif, R., Ferrari, G.: Genetic Centralized Dynamic Clustering in Wireless Sensor Networks, pp. 503–511 (2015)
11. Miranda, K.: Conexión total: Internet de las Cosas. ¿Cómo ves? Revista de Divulgación de la Ciencia de la UNAM 15(174), 22–25 (May 2013)
12. Miranda, K., Ramos, V.: Improving data aggregation in Wireless Sensor Networks with time series estimation. IEEE Lat. Am. Trans. 14(5), 2425–2432 (May 2016)
13. Slavik, M., Mahgoub, I., Badi, A., Ilyas, M.: Analytical model of energy consumption in hierarchical wireless sensor networks. In: 7th International Symposium on High-capacity Optical Networks and Enabling Technologies. pp. 84–90 (Dec 2010)

14. Tubaishat, M., Madria, S.: Sensor networks: an overview. IEEE Potentials 22(2), 20–23 (Apr 2003)

ISSN 1870-4069

Resúmenes de múltiples documentos guiados por consulta empleando representaciones distribucionales

Leticia Luna-Tlatelpa[1], Esaú Villatoro-Tello[2], Gabriela Ramírez-de-la-Rosa[2], Carlos J. Rivero-Moreno[2]

[1] Maestría en Diseño, Información y Comunicación (MADIC),
División de Ciencias de la Comunicación y Diseño,
Universidad Autónoma Metropolitana (UAM) Unidad Cuajimalpa, México

[2] Departamento de Tecnologías de la Información,
Universidad Autónoma Metropolitana (UAM) Unidad Cuajimalpa, México

letyludigital@gmail.com, {evillatoro,gramirez,crivero}@correo.cua.uam.mx

Resumen. Actualmente, la generación automática de resúmenes de múltiples documentos guiados por consulta ha cobrado mucha importancia dentro de la comunidad del Procesamiento de Lenguaje Natural, principalmente debido a la necesidad de información cada vez más específica por parte de usuarios especializados. En este contexto, el presente trabajo evalúa la pertinencia de emplear técnicas de representación distribucional contra técnicas tradicionales al momento de construir un resumen, el cual debe satisfacer una necesidad particular de información, *i.e.,* la consulta. Nuestra hipótesis plantea que la representación distribucional incide de manera positiva en la selección de oraciones relevantes a la consulta, ya que esta representación se aproxima mejor al contenido semántico de las palabras. Los resultados obtenidos durante la fase experimental son alentadores, pues muestran que las representaciones distribucionales pueden ser aplicadas de manera eficiente en el problema planteado.

Palabras clave: Representación de textos, representación distribucional, aproximación semántica, agrupamiento de textos, resumen automático de múltiples documentos.

Query-based Multi-document Summarization through Distributional Representations

Abstract. Recently, research in query-based multi-document summarization has gain the attention of the natural language processing community. This is mainly due to the growing need of specialized users to obtain very specific information. In this context, this paper describes a method for automatically generate a summary from multiple documents considering the user's profile. Particularly, we evaluate the pertinence

of using a distributional representation against traditional document representation techniques. Our main hypothesis suggests that through the information captured by a distributional representation, is possible to build better summaries in response to a short query. Our experimental results are encouraging and indicate that distributional representations can be efficiently applied to the posed problem.

Keywords: Document representation, distributional term representation, semantic information, text clustering, multi-document summarization.

1. Introducción

Es un hecho que la cantidad de información existente ha superado la capacidad, en tiempo y almacenamiento, que tenemos los humanos para poder procesarla; por lo que, ante tal situación, ha sido necesaria la creación de herramientas que puedan aligerar el peso de la información, y una de ellas es la generación automática de resúmenes.

Un resumen es la síntesis concisa y coherente de la información más importante contenida en uno o más documentos, por lo que un sistema generador de resúmenes tiene como objetivo presentar al usuario las ideas principales de los documentos de referencia en un texto pequeño [9,3]. Existen varias categorías para definir a este tipo de sistemas, por ejemplo, sistemas que realizan el resumen de forma general, *i.e.,* simplemente tratan de inferir qué información podría ser relevante para pertenecer al resumen. Por otro lado, sistemas más complejos toman en cuenta el perfil del usuario para la construcción del resumen, usualmente el perfil va indicado por medio de una consulta específica, a estos últimos se les denomina sistemas de generación de resúmenes guiados por consulta. En general, los sistemas de generación de resúmenes transforman los documentos de entrada en oraciones y después, extraen aquellas más importantes. El problema fundamental que enfrentan es cómo calcular la relevancia de las oraciones. Así entonces, el sistema requiere de una representación de las oraciones que permita a los distintos algoritmos identificar aquellas oraciones más relevantes, ya sea por su contenido semántico o porque satisfacen de manera adecuada una necesidad de información [7].

Dentro de este trabajo nos enfocaremos específicamente en el problema de generación de resúmenes de múltiples documentos guiados por consulta. Esto quiere decir que el usuario conoce a priori el conjunto de documentos que versan sobre un tópico de su interés. Así entonces, el usuario desea construir un resumen de estos documentos a partir de una consulta que él define. Idealmente, el resumen generado, debería contener toda la información relevante compartida entre los documentos de la colección (una sola vez), y además toda la información única y relevante a la consulta proporcionada por el usuario.

Es sabido en el área de Procesamiento de Lenguaje Natural (PLN), que el cálculo de la relevancia depende de manera importante de la representación que

se tenga de los documentos. Tradicionalmente, la representación de bolsa de palabras (BoW[3]) ha sido ampliamente utilizada debido a que es relativamente fácil de implementar y ha mostrado ser eficiente en distintas tareas de PLN [4,13,10]. Este modelo, no obstante, tiene algunas desventajas: no distingue la polisemia ni la sinonimia, se pierde el orden de las palabras y se omite información semántica. Con el fin de resolver algunos de estos problemas, surge la representación denominada "bolsa de conceptos" (BoC por sus siglas en inglés), específicamente, nos concentraremos en la representación distribucional TCOR[4] [7]. En este tipo de representación, un término se representa en función de otros términos que co-ocurren con él en el documento. De esta manera se tiene un acercamiento al significado de un término en función de otros términos que tengan una distribución o patrones de uso similar dentro del documento.

El objetivo principal de este trabajo es evaluar si la representación distribucional TCOR mejora la generación automática de resúmenes de múltiples documentos guiados por una consulta. La hipótesis es que la representación TCOR incide positivamente en la selección de las oraciones más importantes ya que ésta representación considera, hasta cierto punto, la semántica contenida en los términos de los documentos y la consulta. Para evaluar la pertinencia de nuestra hipótesis se realizaron experimentos con una colección estándar proporcionada por los organizadores del DUC (Document Understandig Conference) del 2005. Los resultados obtenidos son alentadores y muestran que el uso de representaciones distribucionales permite obtener resúmenes relevantes a una necesidad de información específica.

El resto de este documento se encuentra organizado de la siguiente manera: en la sección 2 se describen algunos trabajos relacionados a la utilización de diferentes tipos de representación y de agrupamiento. En la sección 3 se describe en detalle el método propuesto. En la sección 4 se describen la metodología experimental y la sección 5 muestra los resultados obtenidos. Finalmente en la sección 6 se plantean las conclusiones y se discuten posibles líneas a futuro.

2. Trabajo relacionado

Para la generación automática de resúmenes, existen dos enfoques principales: el método extractivo y el abstractivo. El primero, consiste en extraer (literalmente) la información que se considere importante de los documentos a resumir, y construir con éstas el resumen final. El segundo enfoque, consiste en generar nuevas oraciones a partir de la información identificada como importante; este tipo de técnicas poseen la capacidad de general lenguaje natural [9,3]. Dentro de este trabajo nos enfocaremos en resúmenes extractivos de múltiples documentos.

El proceso de generar un resumen de múltiples documentos consiste en la creación de un resumen simple de un conjunto de documentos relacionados

[3] *Bag-of-Words* por sus siglas en Inglés.
[4] *Term co-occurrence.*

temáticamente; además, dicho resumen debe satisfacer la necesidad de información del usuario, es decir, responder a la consulta proporcionada. Existen tres grandes problemas que surgen en este escenario: (i) reconocer y resolver redundancias, (ii) identificar diferencias importantes entre los documentos, y (iii) asegurar la relevancia del resumen final, tomando en cuenta que diferentes porciones de texto podrían ser relevantes a la consulta proporcionada. En general, para enfrentar estos problemas se ha seguido dos grandes pasos [6,3]: la fase de pre-procesamiento y la fase de procesamiento de la información.

En la etapa de pre-procesamiento de información se consideran acciones como el segmentado de la información (dividir documentos en oraciones), eliminación de palabras vacías, identificación de la raíz léxica de las palabras, etc. Por otro lado, en la etapa de procesamiento de la información se decide, por un lado, los atributos con los que se representará la información, y por otro, la técnica de identificación de la información más relevante. Respecto a las técnicas para la identificación de la información más relevantes es conveniente mencionar que existen una gran variedad de estrategias, por ejemplo, métodos estadísticos, métodos basados en tópicos, técnicas basadas en aprendizaje automático, basados en grafos, y basados en discurso. Para tener un referente más amplio de las distintas estrategias refiérase a [3].

Dentro de este trabajo nos enfocaremos únicamente en los atributos, *i.e.*, la representación empleada. Muchos trabajos existentes recurren a la tradicional *Bolsa-de-Términos* debido a su simplicidad [9,3], donde los *términos* pueden ser palabras simples o secuencias de palabras [4,10,2,5] los cuales son ponderados de acuerdo a su valor TF-IDF. La desventaja principal de este tipo de representación es que carece de información semántica, y para que la relevancia de las oraciones con respecto a la consulta sea determinada, requiere de que los términos de la consulta existan de manera idéntica en los documentos de la colección, de otra forma la relevancia no puede ser determinada o es prácticamente nula. Recientemente, el uso de representaciones más semánticas ha permitido enfrentar estas limitantes, por ejemplo en [12,11] se propone el uso de LSA[5] para la generación de resúmenes. Estos trabajos han mostrado la pertinencia de éste tipo de formas de representación, sin embargo tienen la principal desventaja de que requieren de un corpus de entrenamiento para poder generar la representación.

Contrario a los trabajos descritos anteriormente, dentro de este artículo proponemos utilizar una forma de representación distribucional para la generación de resúmenes de múltiples documentos a partir de consultas. A diferencia de trabajos previos, este tipo de representación captura la semántica de los términos sin la necesidad de un corpus de entrenamiento, permitiendo esto la independencia de dominio y de lenguaje.

3. Método propuesto

En la figura 1 se muestra el esquema general de nuestro método para la generación de resúmenes de múltiples documentos guiado por consulta. Básí-

[5] Latent Semantic Analysis

camente, dada una colección de documentos relacionados temáticamente y una consulta proporcionada por el usuario, nuestro método aplica algunas operaciones de pre-procesamiento previo al proceso de generación de resúmenes. Una vez dentro de la etapa de procesamiento de la información, la parte fundamental de nuestro método recae en la construcción de la representación, la cual sirve posteriormente tanto para agrupar como para extraer información relevante a la consulta, misma que es empleada para la construcción del resumen final. A continuación describimos detalladamente cada uno de los módulos involucrados en nuestro método propuesto.

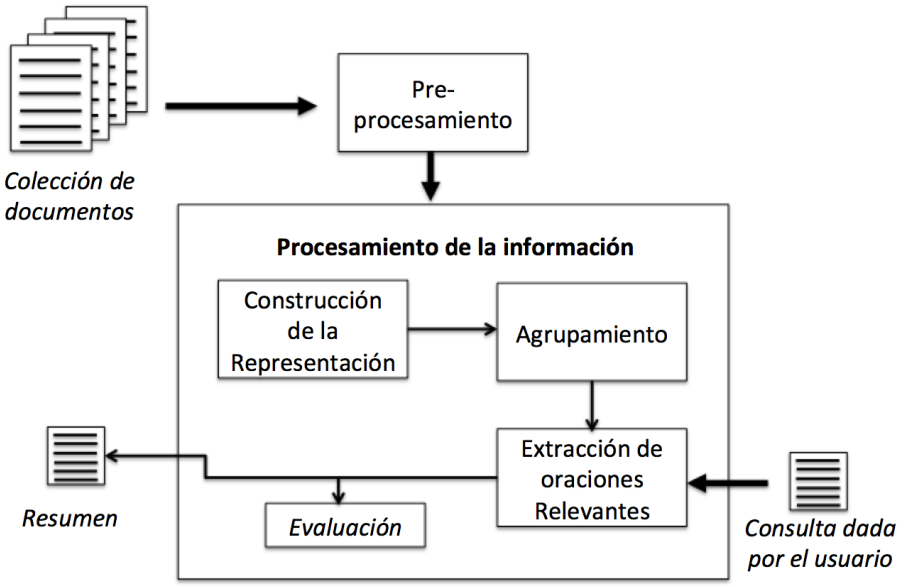


Figura 1. Arquitectura general del método propuesto.

3.1. Pre-procesamiento

En la etapa del pre-procesamiento, se eliminaron todas las palabras que no contribuyen a la semántica de los textos de entrada, como son: etiquetas html y xml, signos de puntuación y palabras vacías. Todos los textos son convertidas a minúsculas y se realizó truncamiento de palabras[6], el cual es un proceso heurístico para aproximar las palabras a su raíz léxica. En esta etapa se hace el segmentado de los documentos en oraciones, para lo cual se empleó el programa proporcionado por los organizadores de la conferencia DUC.[7]

[6] Para este proceso se utilizó el método de *Porter*
[7] `http://duc.nist.gov/`

3.2. Construcción de la representación

El primer paso obligado es el *indexado* de las oraciones (S), actividad que denota hacer el mapeo de una oración s_i en una forma compacta de su contenido. La representación más comúnmente utilizada para representar textos es un vector con términos ponderados como entradas, concepto tomado del modelo de espacio vectorial usado en recuperación de información. Esto es, cada texto s_i es representado como el vector $\overrightarrow{s}_i = \langle w_{ki}, \ldots, w_{|\tau|i} \rangle$, donde τ es el *diccionario*, *i.e.*, el conjunto de términos que ocurren al menos una vez en algún elemento de S, mientras que w_{ki} representa la importancia del término t_k dentro del contenido del documento s_i. Este método de representación, también conocido como bolsa de palabras (BoW), propone varios esquemas para definir w_{ki}, en particular, para nuestro método base se utilizó el esquema de pesado TF-IDF.

Nuestra propuesta como alternativa a las limitantes de la representación BoW es utilizar una forma de representación distribucional llamada TCOR. La representación de co-ocurrencia de términos (TCOR) se basa en las estadísticas de co-ocurrencia de los mismos términos. La idea intuitiva es que la semántica del termino t_j puede ser revelada por medio de los términos con los que co-ocurre dentro de la colección de documentos. Aquí, cada término $t_j \in \tau$ es representado por un vector de pesos $\overrightarrow{w}_j = \langle w_{1,j}, \ldots, w_{|\tau|,j} \rangle$, donde $0 \leq w_{k,j} \leq 1$ representa la contribución del término t_k a la descripción semántica de t_j:

$$w_{k,j} = tf(t_k, t_j) \cdot log \frac{|\tau|}{\tau_k}, \tag{1}$$

donde τ_k es el número de diferentes términos en el diccionario τ que co-ocurren con t_j en almenos un documento y

$$tf(t_k, t_j) = \begin{cases} 1 + \log(\#(t_k, t_j)) & si(\#(t_k, t_j) > 0), \\ 0 & \text{en otro caso}, \end{cases} \tag{2}$$

donde $\#(t_k, t_j)$ denota el número de documentos en los cuales el término t_j co-ocurre con el término t_k. La intuición detrás de este esquema de pesado es que entre más veces t_k y t_j co-ocurran más importante será t_k para describir la semántica de t_j; mientras más términos co-ocurran con t_k menos importante será éste en la definición de la semántica de t_j.

Una vez que se tiene el vector $\overrightarrow{w}_j$ de pesos de cada término, la forma de representar cada oración s_i se obtiene por medio de:

$$\overrightarrow{s}_i^{tcor} = \sum_{t_j \in s_i} \alpha_{t_j} \cdot \overrightarrow{w}_{t_j}, \tag{3}$$

donde α_j es un escalar que pondera la contribución del término $t_j \in s_i$ dentro de la representación de la oración, normalmente el valor TF-IDF del término t_j. De esta forma, la representación de una oración está dada por la suma de los vectores contextuales de sus términos.

Observe que bajo la representación TCOR cada oración s_i es representada por $\overrightarrow{s}_i^{tcor} \in \mathbb{R}^{|\tau|}$, un vector de la misma dimensionalidad que el vocabulario. Los

valores de $\overrightarrow{s}_i^{tcor}$ indican el grado de asociación entre los términos del vocabulario y los términos que ocurren dentro de s_i.

3.3. Agrupamiento

El proceso de agrupamiento tiene como principal objetivo permitir al sistema de generación de resúmenes dividir la colección inicial en sus diferentes subtemas. Idealmente, esto permitirá identificar similitudes (redundancias) entre documentos y además detectar la información única (relevante) dentro de cada uno de ellos.

Para los experimentos realizados en este trabajo se utilizó el algoritmo de agrupamiento estrella. Este es un algoritmo que ha mostrado evidencia positiva sobre la verdad de la hipótesis del grupo[8] [1]. Este es un algoritmo que induce de manera natural el número de grupos y la estructura de los temas dentro del espacio de textos. Es un algoritmo de tipo particional, y que se basa en Teoría de grafos. La salida de este algoritmo son grupos de documentos en forma de "estrella", donde se garantiza que el elemento central de cada una de las estrellas es el más representativo del grupo. Esto se logra a través de un parámetro de similitud σ que es pasado al algoritmo, el cual permite incorporar elementos al grupo cuando estos son semejantes al centro por un factor mayor o igual a σ.

Un paso fundamental del algoritmo estrella es el cálculo de un grafo umbralizado G_σ [1]. Para esto, es necesario aplicar técnicas de medición de similitud entre documentos por medio de las cuales es posible definir el umbral σ. Para nuestros experimentos se utilizó la medida cosenoidal. La idea básica de ésta es medir el ángulo entre el vector de dos oraciones cualesquiera s_i y de s_j, para hacerlo calculamos:

$$SC(s_i, s_j) = \frac{\sum_{k=1}^{|\tau|} w_{ik} w_{jk}}{\sqrt{\sum_{k=1}^{|\tau|} (w_{jk})^2 \sum_{k=1}^{|\tau|} .(w_{ik})^2}} \tag{4}$$

Note que dependiendo de la representación que estemos utilizando, BoW o TCOR, w_{ik} puede tomar diferentes significados. Por un lado, puede referir al valor TF-IDF del termino k bajo una representación BoW; mientras que bajo una representación TCOR representa el vector contextual del término k calculado por medio de la expresión 1.

Finalmente, para hacer la selección del mejor umbral σ, se hace uso de la información estadística de la matriz de similitud que emplea el algoritmo estrella. Para esto, se calcula la media ($\overline{x}$) y la desviación estándar (δ) de los datos. Así entonces, se definen dos posibles valores para σ, un valor ALTO ($\overline{x} + \delta$) y un valor BAJO ($\overline{x} - \delta$).

[8] El uso del agrupamiento dentro del área de Recuperación de Información surge debido a una hipótesis *(hipótesis del grupo)*, que dice que los documentos relevantes a una petición tienden a ser más cercanos entre ellos que aquellos que no son relevantes a una petición en particular.

3.4. Extracción de oraciones relevantes

Una vez que se ha hecho el agrupamiento de la información contenida en los documentos de la colección, el siguiente paso es la extracción de la información más relevante para construir el resumen final. Para la realización de este proceso se definieron dos métodos, los cuales describimos a continuación.

- CENTRO_ESTRELLA (**CE**). Bajo este esquema, se aprovecha la estructura de salida generada por el algoritmo de agrupamiento estrella. Dado que este algoritmo garantiza que el centro de cada estrella es el elemento más representativo, se ordenan los centros de las estrellas de acuerdo a su similitud con la consulta. Una vez hecho esto, se toman los centros más similares para la construcción del resumen final hasta que se alcanza el tamaño de resumen requerido por el usuario.
- SATÉLITE_CERCANO (**SC**). En esta configuración no se toman como elementos más representativos a los centros de las estrellas. La hipótesis aquí es que, a pesar de que el elemento más representativo de cada grupo es el centro de la estrella, este agrupamiento no se hizo contemplando información de la consulta y por ende, podría haber un elemento más relevante a la consulta entre los satélites de la estrella. Así entonces, para cada estrella formada en la etapa de agrupamiento se identifica al elemento (satélite o centro) más similar con la consulta y éste es incorporado a una lista ordenada. Finalmente, se van tomando las oraciones más similares de esta lista hasta construir un resumen del tamaño requerido.

4. Configuración experimental

En esta sección se describen el conjunto de datos con el que se realizaron los experimentos. Agregado a esto se describen los métodos base con los que se compara la propuesta hecha en este trabajo, se explican las métricas de evaluación, y se definen los experimentos realizados para comprobar las hipótesis planteadas en este artículo.

4.1. Conjunto de datos

Los datos con los que se trabajó corresponden a los proporcionados por DUC 2005[9], los cuales consisten en noticias de *Los Angeles Times* y *Financial Times of London*, divididos en 50 tópicos (colecciones de documentos) independientes entre sí. Cada tópico tiene una consulta asociada, la cual consiste de un título, una pequeña narrativa y un valor de granularidad, el cual indica la especificidad con que se sugiere realizar el resumen. Para nuestros experimentos no se consideró en valor de granularidad.

[9] `http://www-nlpir.nist.gov/projects/duc/data/2005_data.html`

4.2. Métodos base

A continuación se describen los tres métodos base que se definieron para la tarea. Es conveniente mencionar que en las especificaciones del DUC 2005, el tamaño del resumen final no debería de ser superior a 250 palabras .

- Método Base 1 (**MB1**). Este método corresponde al método propuesto por los organizadores del DUC 2005. Consiste en extraer las primeras 250 palabras del documento más reciente de la colección. Aunque simple, este método base es considerado un método fuerte y difícil de superar.
- Método Base 2 (**MB2**). Este método consiste en segmentar todos los documentos de cada tópico en oraciones. Posteriormente, las oraciones son ordenadas de acuerdo a su grado de similitud con la consulta, y para la construcción del resumen se toman las oraciones más similares hasta completar 250 palabras. Para esto se empleó una representación tipo BoW con pesado TF-IDF y empleando como medida de similitud el coseno.
- Método Base 3 (**MB3**). Muy similar al MB2, con la diferencia de que en lugar de emplear palabras simples como atributos de la BoW, se empleó la combinación de uni-gramas, bi-gramas y tri-gramas de palabras como elementos del vocabulario τ. El objetivo fue evaluar si por medio de incorporar información contextual a través del uso de n-gramas se podía obtener mejores resultados.

4.3. Experimentos

Con el objetivo de comprobar la pertinencia del método propuesto descrito en la sección 3, se definieron dos grandes conjuntos de experimentos: *i)* aplicando la metodología propuesta en la figura 1 empleando una representación tradicional tipo BoW considerando palabras simples como atributos del vector de la representación; y *ii)* aplicando la metodología propuesta en la figura 1 empleando una representación TCOR.

Recuerde que el método descrito en la sección 3 tiene algunos parámetros como lo son el valor del umbral de similitud (ALTO/BAJO) y la forma en que se extraen las oraciones para conformar el resumen final (CE/SC). Así entonces, con el objetivo de cubrir todos estos aspectos, se realizaron un total de ocho experimentos. La nomenclatura para nombrar a los experimentos es como sigue: **REP-STAR-UMB-EXT**. Donde REP puede ser BOW o TCOR en referencia al tipo de representación empleado, STAR indica que se utilizó el método descrito en la figura 1, UMB indica si el experimento consideró un umbral ALTO o BAJO en su configuración, y finalmente EXT indica el método empleado (CE o SC) para la extracción y construcción final del resumen.

4.4. Evaluación

Para la evaluación de los resultados se empleó ROUGE, un sistema automático para la evaluación de resúmenes propuesto por Lin y Hovy [8]. Este

sistema está basado en el método propuesto para la evaluación de traducciones automáticas BLEU, *i.e.*, en la co-ocurrencia de n-gramas de palabras. Lin y Hovy demuestran en [8] como este tipo de métricas pueden ser aplicados para evaluar la calidad de los resúmenes generados automáticamente.

ROUGE incluye diferentes métricas para evaluar ésta correlación, en particular nos enfocaremos en ROUGE-N, la cual fue utilizada para reportar nuestros resultados. ROUGE-N es un método de evaluación basado en el recuerdo entre un resumen "candidato" (resumen generado automáticamente) y un resumen "referencia" (generado por un experto humano). ROUGE-N es calculado por medio de:

$$ROUGE{-}N = \frac{\sum_{S_i \in \{ResumenReferencia\}} \sum_{gram_n \in S_i} Count_{match}(gram_n)}{\sum_{S_i \in \{ResumenReferencia\}} \sum_{gram_n \in S_i} Count(gram_n)}, \quad (5)$$

donde S_i se refiere a la oración i dentro del resumen de referencia, n es la longitud del n-grama, $gram_n$ y $Count_{match}(gram_n)$ es el máximo número de n-gramas que co-ocurren en el resumen candidato y el conjunto de resúmenes de referencia. La idea intuitiva detrás de esta medida de evaluación radica en comparar el resumen realizado por un sistema automático contra uno o varios resúmenes generados por un humano. Entre mayor recuerdo (elementos similares) haya entre el generado por el sistema y el generado por el humano, mejor se considera su desempeño. Para reportar nuestros resultados obtenidos se utilizó tanto ROUGE-1 como ROUGE-2.

5. Resultados

Las tablas 1 y 2 muestran los resultados obtenidos por los diferentes experimentos definidos en la sección 4. Las tablas muestran los valores de *Recuerdo*, *Precisión* y *F-score* obtenidos al aplicar ROUGE-N a los resúmenes generados de forma automática. Para evaluar nuestros experimentos se consideraron todos resúmenes de referencia proporcionados por los organizadores del DUC 2005 (*i.e.,* resúmenes generados por humanos, en promedio 4 por tópico).

Una primer observación de los experimentos realizados es que los métodos base propuestos (MB2 y MB3) superan de manera importante al método base propuesto por los organizadores del DUC (F-score de 0.33 contra 0.29 en ROUGE-1, tabla 1; F-score de 0.06 contra un 0.04 en ROUGE-2, tabla 2). Esto indica, hasta cierto punto, que es posible construir resúmenes relevantes a la consulta por medio de simplemente recuperar las oraciones más similares a la consulta. No obstante, el método propuesto resulta ser mejor, en particular la configuración que emplea una representación TCOR con un umbral de similitud alto y un esquema de extracción que no depende en los centros de las estrellas formadas por el algoritmo de agrupamiento, *i.e.*, TCOR-STAR-ALTO-SC. La ventaja de ésta técnica con respecto a los métodos base es que garantiza una mayor heterogeneidad en la información contenida en el resumen final gracias a la etapa de agrupamiento de información, etapa que ayuda en la eliminación de redundancias e identificación de información relevante.

Tabla 1. Resumen de los resultados experimentales empleando como métrica de evaluación ROUGE-1. Los datos reportados representan el desempeño promedio obtenido a través de los 50 tópicos; entre paréntesis se muestra el valor de la desviación estándar.

Nombre del experimento	**Recuerdo**	**Precisión**	**F-score**
Método Base 1 (MB1)	0.3014 (0.05)	0.2921 (0.05)	0.2966 (0.05)
Método Base 2 (MB2)	0.3396 (0.06)	0.3292 (0.05)	0.3342 (0.06)
Método Base 3 (MB3)	0.3374 (0.06)	0.3271 (0.06)	0.3320 (0.06)
BOW-STAR-ALTO-CE	0.3054 (0.08)	0.3252 (0.05)	0.3115 (0.06)
BOW-STAR-ALTO-SC	0.3365 (0.06)	0.3261 (0.05)	0.3311 (0.06)
BOW-STAR-BAJO-CE	0.1718 (0.05)	**0.3674** (0.06)	0.2278 (0.05)
BOW-STAR-BAJO-SC	0.3396 (0.06)	0.3289 (0.05)	0.3341 (0.05)
TCOR-STAR-ALTO-CE	0.3021 (0.04)	0.2925 (0.04)	0.2971 (0.04)
TCOR-STAR-ALTO-SC	**0.3599** (0.06)	0.3476 (0.06)	**0.3535** (0.06)
TCOR-STAR-BAJO-CE	0.1787 (0.07)	0.2643 (0.05)	0.2025 (0.06)
TCOR-STAR-BAJO-SC	0.3099 (0.09)	0.3613 (0.06)	0.3245 (0.07)

Note que el desempeño de la configuraciones que emplean un esquema de extracción tipo SC son en general mejores que su contra parte, es decir, superan al esquema de extracción basado 100 % en los centros de las estrellas. Este comportamiento respalda nuestra intuición respecto a la relevancia del centro de las estrellas. Como se mencionó en la sección 3.4, a pesar de que el algoritmo de agrupamiento garantiza que el centro de la estrella es el elemento más representativo, este elemento no es necesariamente el más relevante a la consulta, y la razón es simple, el algoritmo de agrupamiento no considera a la consulta en el proceso de agrupamiento.

Tabla 2. Resumen de los resultados experimentales empleando como métrica de evaluación ROUGE-2. Los datos reportados representan el desempeño promedio obtenido a través de los 50 tópicos; entre paréntesis se muestra el valor de la desviación estándar.

Nombre del experimento	**Precisión**	**Recuerdo**	**F-score**
Método Base 1 (MB1)	0.0466 (0.02)	0.0445 (0.02)	0.0455 (0.02)
Método Base 2 (MB2)	0.0668 (0.03)	0.0642 (0.03)	0.0655 (0.03)
Método Base 3 (MB3)	0.0667 (0.03)	0.0641 (0.03)	0.0654 (0.03)
BOW-STAR-ALTO-CE	0.0484 (0.03)	0.0499 (0.02)	0.0487 (0.02)
BOW-STAR-ALTO-SC	0.0666 (0.03)	0.0639 (0.03)	0.0652 (0.03)
BOW-STAR-BAJO-CE	0.0236 (0.01)	0.0503 (0.02)	0.0314 (0.01)
BOW-STAR-BAJO-SC	0.0682 (0.03)	0.0652 (0.03)	0.0666 (0.03)
TCOR-STAR-ALTO-CE	0.0416 (0.02)	0.0400 (0.02)	0.0408 (0.02)
TCOR-STAR-ALTO-SC	**0.0755** (0.04)	**0.0723** (0.03)	**0.0739** (0.04)
TCOR-STAR-BAJO-CE	0.0160 (0.01)	0.0241 (0.01)	0.0182 (0.01)
TCOR-STAR-BAJO-SC	0.0598 (0.03)	0.0687 (0.03)	0.0623 (0.03)

6. Conclusiones

En este trabajo hemos propuesto un método para la generación de resúmenes de múltiples documentos guiados por consulta, el cual emplea una técnica de representación distribucional para hacer la caracterización de la información. Contrario a las técnicas tradicionales de representación de textos, las representación utilizada permite obtener una aproximación semántica de los términos contenidos en una colección de documentos y, en consecuencia, es posible satisfacer de manera más efectiva las necesidades de información del usuario al momento de construir un resumen.

Los experimentos realizados sobre un conjunto estándar de evaluación demuestran que la metodología propuesta es pertinente para la tarea en mano. Específicamente se mostró que la representación TCOR permite obtener resultados que superan a los métodos base al mismo tiempo que a técnicas de representación tradicionales.

A pesar de los buenos resultados obtenidos, hace falta mucho trabajo por delante. Por ejemplo, nos interesa incorporar en la fase de agrupamiento información de la consulta, de manera que los grupos formados puedan ser más adecuados a la necesidad de información establecida por el usuario. Además, nos interesa realizar experimentos empleando otro tipo de técnicas de agrupamiento de forma que nos sea posible determinar la pertinencia del algoritmo seleccionado para los experimentos reportados en este trabajo.

Agradecimientos. El trabajo del primer autor fue parcialmente financiado por el CONACyT a través de la beca 615483. También se agradece el apoyo otorgado a través de la Coordinación de la Maestría en Diseño, Información y Comunicación (MADIC) de la UAM Cuajimalpa, así como al Departamento de Tecnologías de la Información de la UAM Cuajimalpa.

Referencias

1. Aslam, J., Pelekhov, K., Rus, D.: A practical clustering algorithm for static and dynamic information organization. In: Proceedings of the 1999 Symposium on Discrete Algorithms. pp. 208–217 (1999)
2. Baralis, E., Cagliero, L., Jabeen, S., Fiori, A.: Multi-document summarization exploiting frequent itemsets. In: Proceedings of the 27th Annual ACM Symposium on Applied Computing. pp. 782–786. ACM (2012)
3. Gambhir, M., Gupta, V.: Recent automatic text summarization techniques: a survey. Artificial Intelligence Review 47(1), 1–66 (2017), http://dx.doi.org/10.1007/s10462-016-9475-9
4. García-Hernández, R.A., Ledeneva, Y.: Word sequence models for single text summarization. In: 2009 Second International Conferences on Advances in Computer-Human Interactions. pp. 44–48 (Feb 2009)
5. Glavaš, G., Šnajder, J.: Event graphs for information retrieval and multi-document summarization. Expert Systems with Applications 41(15), 6904 – 6916 (2014), http://www.sciencedirect.com/science/article/pii/S0957417414001985

6. Gupta, V., Lehal, G.S.: A survey of text summarization extractive techniques. Journal of emerging technologies in web intelligence 2(3), 258–268 (2010)
7. Lavelli, A., Sebastiani, F., Zanoli, R.: Distributional term representations: an experimental comparison. In: Proceedings of the thirteenth ACM international conference on Information and knowledge management. pp. 615–624. ACM (2004)
8. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Proceedings of Workshop on Text Summarization Branches Out, Post-Conference Workshop of ACL. Barcelona, Spain (2004)
9. Torres-Moreno, J.M.: Automatic text summarization. John Wiley & Sons (2014)
10. Villatoro-Tello, E., Villaseñor-Pineda, L., Montes-y Gómez, M.: Using word sequences for text summarization. In: International Conference on Text, Speech and Dialogue. pp. 293–300. Springer (2006)
11. Yao, J.g., Wan, X., Xiao, J.: Compressive document summarization via sparse optimization. In: IJCAI. pp. 1376–1382 (2015)
12. Yeh, J.Y., Ke, H.R., Yang, W.P., Meng, I.H.: Text summarization using a trainable summarizer and latent semantic analysis. Information Processing & Management 41(1), 75 – 95 (2005), http://www.sciencedirect.com/science/article/pii/S0306457304000329, an Asian Digital Libraries Perspective
13. Zhang, Y., Zincir-Heywood, N., Milios, E.: Narrative text classification for automatic key phrase extraction in web document corpora. In: Proceedings of the 7th Annual ACM International Workshop on Web Information and Data Management. pp. 51–58. WIDM '05, ACM, New York, NY, USA (2005), http://doi.acm.org/10.1145/1097047.1097059

ISSN 1870-4069

Detección del engaño en notas de opinión a través de técnicas tradicionales de clasificación automática de textos

Javier Sánchez-Junquera, Luis Villaseñor-Pineda, Hugo J. Escalante, Manuel Montes-y-Gómez

Instituto Nacional de Astrofísica, Óptica y Electrónica, Laboratorio de Tecnologías del Lenguaje, Puebla, México

jjsjunquera@gmail.com, {villasen,hugojair,mmontesg}@inaoep.mx

Resumen. En este trabajo se realiza un estudio del alcance de técnicas tradicionales usadas en clasificación automática de textos (*v. gr.* bolsa de palabras) para la detección de engaño. Comúnmente las técnicas tradicionales funcionan adecuadamente en clasificación temática. Sin embargo, se desea conocer el rendimiento de dichas técnicas en una tarea intuitivamente no-temática. La colección empleada es un conjunto de notas en inglés de opiniones sobre hoteles, incluyendo notas verdaderas y falsas. Se realizaron experimentos utilizando la representación de bolsa de palabras con esquemas de pesado binario, *tf* y *tf-idf* y entrenando un clasificador probabilista. Los resultados muestran que el engaño puede ser detectado con el enfoque tradicional. Un primer análisis de estos resultados identifica aquellos elementos sobre los que recayó la discriminación.

Palabras clave: Detección de engaño, marcadores sintácticos, clasificación de texto.

Deceptive Detection in Opinion Notes by Traditional Techniques of Automatic Text Classification

Abstract. This work studies the scope of traditional techniques used in automatic text classification (*v. gr.* bag of words) for the deceptive detection. Commonly, traditional techniques work well in thematic classification. However, it is desired to know the performance of these techniques in an intuitively non-thematic task. The collection used is a set of English notes of hotel reviews, including truthful and deceptive notes. Experiments were performed using bag of words with binary weighing schemes, *tf* and *tf-idf* and training a probabilistic classifier. The results show that deception can be detected with the traditional approach. A first analysis of these results identifies those elements on which discrimination fell.

Keywords: Deceptive detection, POS tagging, text classification.

1. Introducción

En la literatura se puede encontrar estudios sobre el engaño como aquella acción en la que una persona intenta convencer de la veracidad de algo que en realidad sabe que es falso. Dichos estudios abarcan desde las respuestas fisiológicas, el lenguaje corporal y el lenguaje natural tanto escrito como hablado. El presente trabajo se centra en la detección de engaño en opiniones que fueron escritas con la intención de resumir o valorar una supuesta experiencia.

El engaño está presente en múltiples actividades en las que el ser humano tiene la posibilidad, necesidad u obligación de expresarse verbalmente. Ejemplo de ello se puede apreciar en la web, donde los usuarios dejan plasmada una reseña, valoración u opinión sobre prácticamente todo. A menudo puede ser útil leer sobre la experiencia de otros para tomar una elección propia, y sabiendo esto a muchos les conviene falsear opiniones en busca de algún beneficio.

Existe gran motivación en detectar automáticamente las opiniones falsas[1], por ejemplo, TripAdvisor[2] posee millones de opiniones de viajeros acerca de alojamientos y tiene particular interés en la detección de opiniones falsas cuyo fin generalmente es aumentar o disminuir la reputación de un establecimiento por parte de propietarios o competidores, respectivamente.

El presente trabajo muestra un análisis de técnicas de procesamiento de lenguaje natural empleando técnicas probabilistas -las cuales han mostrado muy buen desempeño estableciendo un balance entre la calidad de los resultados y la complejidad computacional- en la clasificación de opiniones falsas. En este análisis se utiliza el modelo *Naïve Bayes Multinomial Updatable*, implementado en la plataforma WEKA, el cual se ha usado satisfactoriamente en problemas de clasificación, fundamentalmente en clasificación temática de textos. Con este modelo se analizan algunas representaciones simples de los documentos con el objetivo de estudiar el alcance de técnicas relativamente sencillas para la detección de engaño.

A continuación, en la Sec. 2, se comentará algunos trabajos relacionados a la detección de engaño. En la Sec. 3, se discutirá la metodología llevada a cabo, el corpus con el que se trabajó y la evaluación del sistema. En la Sec. 3.3 se mostrarán los resultados obtenidos y una breve discusión de los mismos. Finalmente se darán las conclusiones y el trabajo futuro en la Sec. 5.

[1] Si bien las opiniones falsas no implican necesariamente la existencia de engaño, o sea, la intención de engañar, en este documento se mencionará "opiniones falsas" como expresión alternativa para referirse a "opiniones engañosas".

[2] TripAdvisor es un sitio web con facilidades a los viajeros. Cuenta con 435 millones de opiniones y comentarios sobre 6,8 millones de alojamientos, restaurantes y atracciones. Para más detalles sobre la moderación de opiniones y detección de fraude en TripAdvisor visitar https://www.tripadvisor.es/vpages/review_mod_fraud_detect.html

2. Trabajos relacionados

Entre los trabajos relacionados más citados se encuentra [7], en el que intentan detectar el engaño desde la psicología y la lingüística computacional. Los autores proponen una colección balanceada de opiniones positivas en inglés, tomada como *gold standard*. Con esta colección entrenan los clasificadores *Naïve Bayes* (NB) y *Support Vector Machine* (SVM), bajo la consideración de trigramas, bigramas y unigramas basados en análisis psicolinguísticos y mediante el recurso externo *Linguistic Inquiry and Word Count* (LIWC). Estos autores concluyen que en las opiniones engañosas predominan las características de la escritura imaginativa y que se carece de información espacial.

Más allá del análisis léxico, en [3] proponen investigar las estructuras sintácticas notando que en las opiniones engañosas las frases verbales, las cláusulas subordinadas y las frases adverbiales con *wh-words* son más frecuentes que en las opiniones verdaderas. En [6] incrementan la colección mencionada con opiniones negativas, igualmente balanceadas entre verdaderas y falsas, también en inglés; detectando que en las opiniones engañosas se usa más la primera persona del singular, así como un lenguaje con tendencia a la exageración.

Mediante aprendizaje semisupervisado, en [4] intentan abordar el problema de la detección a partir de la escasez de ejemplos engañosos. Estos investigadores sugieren la existencia de aspectos comunes en la manera de escribir opiniones positivas y negativas cuando son falsas, ya que sus resultados muestran que un clasificador entrenado con opiniones de ambas polaridades es más efectivo que un clasificador para cada una por separado, confirmando resultados de [6].

Por otra parte, el clasificador *Bayes Multinomial Updatable* ha sido usado en la detección de lenguaje ofensivo [8]; la clasificación de conductas colaborativas en texto [1] y la detección de *spam* a partir del reconocimiento de la personalidad en mensajes cortos [2]. Entre los clasificadores bayesianos disponibles en WEKA 3.8, este alcanzó mejores resultados. Por esta razón, la metodología seguida en este trabajo y los resultados que se reportan son empleando este clasificador.

3. Metodología

3.1. Configuración experimental

Para el entrenamiento y validación del clasificador que se utiliza en este trabajo se emplea el corpus[3] de opiniones sobre 20 hoteles de Chicago construido por [7,6]. Por cada uno de estos hoteles, se cuenta con 20 opiniones negativas falsas, 20 negativas verdaderas, 20 positivas falsas y 20 positivas verdaderas. En total son 1600 opiniones. En este trabajo solamente se tienen en cuenta la información sobre la falsedad o veracidad de las opiniones, proponiéndose como trabajo futuro la inclusión de la polaridad.

El pre-procesamiento de los textos se llevó a cabo con el uso de *NLTK 3.0*. Con los siguientes pasos:

[3] El corpus está disponible gratuitamente en http://myleott.com/op_spam/

- *lowercase*,
- *stemming* (Porter Stemmer),
- eliminación de palabras vacías (según las incluidas en *NLTK* para el inglés),

e ignorándose todos los caracteres que no fueran alfabéticos. Después del preprocesamiento se observaron 6637 *tokens*, encontrándose por documento un promedio de aproximadamente 60 *tokens*, con un número máximo y mínimo de 286 y 12 respectivamente.

3.2. Sobre la selección de términos

Las palabras vacías son aquellas que *carecen* de significado y no aportan sentido distintivo a los textos, o sea, no son útiles para diferenciar unos de otros (por ejemplo: artículos, pronombres y conjunciones). Sin embargo, según [6], en las opiniones positivas, los pronombres en primera persona del singular son más frecuente cuando se trata de opiniones falsas. Por tanto, excluir las palabras vacías del vocabulario de las opiniones podría influir negativamente en la clasificación.

Además de las palabras vacías, existen otras que no son precisamente carentes de significado, pero que por su alta o baja frecuencia tampoco ayudan a separar los textos en clases. Mientras que las palabras más frecuentes tienden a estar indistintamente en todas las clases, las que son poco frecuentes no llegan a ser comunes dentro de una sola clase; estos dos tipos de palabras dificultan la clasificación de una opinión, y son consideradas ruido.

En un intento de observar los términos más representativos para un posterior estudio, se ordenaron las palabras según la diferencia en las que aparecían en ambas clases, así como la ganancia de información (GI) de cada una de ellas. En la Sec 3.3 se discutirá algunas observaciones al respecto. Es importante notar que tanto el ordenamiento como la GI no se emplearon para la selección de términos en el esquema de clasificación propuesto, para evitar un posible sobre-ajuste a la colección empleada.

3.3. Pesado de términos

Tradicionalmente, en la representación de documentos, hay tres tipos de pesado para los términos.

En el pesado binario solo se tiene en cuenta la presencia de los términos en los documentos. De esta forma los pesos son 1 o 0, dependiendo de si el término está en el texto o no, respectivamente.

Por otra parte, el pesado *tf* consiste en la frecuencia del término en el documento. De esta forma, un término que sea muy frecuente tendrá un peso mayor que otro que sea poco frecuente.

El *tf* es que no considera el hecho de que un término sea muy frecuente precisamente en todos los documentos, lo cual indicaría que el término no es útil para representar los documentos. Sin embargo, el *tf-idf* establece un compromiso

entre la frecuencia del término en el documento y la cantidad de documentos que lo contienen. O sea,

$$tf\text{-}idf = tf \times \log \frac{|D|}{|\{d \in D : t \in d\}|},$$

donde $|D|$ es la cantidad total de documentos y $|\{d \in D : t \in d\}|$ es la cantidad de documentos en los que aparece el término t.

4. Resultados

En esta sección se presenta los resultados obtenidos con el clasificador *Bayes Multinomial Updatable* utilizando diferentes esquemas de pesado tradicionales. También se muestran algunas observaciones sobre los términos más relevantes dada la colección de opiniones usada.

El clasificador fue entrenado antes y después del pre-procesamiento indicado en la Sec. 3.1, considerándose unigramas de tokens como rasgos y tres esquemas de pesado (binario, *tf* y *tf-idf*). Los seis modelos obtenidos fueron evaluados mediante una validación cruzada de 10 iteraciones. Para ambos, el entrenamiento y la validación, se usó la implementación de WEKA 3.8.

Los resultados mostrados en la Tabla 1 incluyen precisión, recuerdo y f_1-*measure* por clase, así como la exactitud global y son calculados como sigue:

$$P(c) = \frac{\#\ de\ predicciones\ correctas\ de\ la\ clase\ c}{\#\ de\ \boldsymbol{predicciones}\ para\ la\ clase\ c}, \tag{1}$$

$$R(c) = \frac{\#\ de\ predicciones\ correctas\ de\ la\ clase\ c}{\#\ de\ \boldsymbol{opiniones}\ de\ la\ clase\ c}, \tag{2}$$

$$F_1(c) = \frac{2 \times P(c) \times R(c)}{P(c) + R(c)}, \tag{3}$$

$$E = \frac{\#\ predicciones\ correctas}{\#\ total\ de\ opiniones\ clasificadas}, \tag{4}$$

donde $P(c)$, $R(c)$ y $F_1(c)$ son precisión, recuerdo y f_1-*measure* de la clase c respectivamente, y E es la exactitud global de la clasificación.

Como se puede notar en la Tabla 1, cada esquema muestra una exactitud entre el 70 % y el 88 %, aún cuando no se está usando ningún recurso externo como en los trabajos de [5] y [7], ni desarrollándose un análisis psicolingüístico [7,6] o sintáctico [3]. Sin embargo, los autores de este trabajo consideran más relevante para trabajos futuros tener en cuenta los esquemas que consigan mejor resultado en el recuerdo de las opiniones falsas o en la precisión de las opiniones verdaderas, como es el caso del esquema con pesado *tf* sin pre-procesamiento o con pre-procesamiento, respectivamente.

Tabla 1. Se incluye Precisión, Recuerdo y F_1-*measure* por cada categoría y la Exactitud para cada pesado En cada caso se indica el tipo de pesado y si se hizo pre-procesamiento. Resaltado en negrita los mejores resultados entre los esquema.

			P		R		F_1	
Pesado	**Pre-proc.**	***E***	**V**	**F**	**V**	**F**	**V**	**F**
binario		**87.94 %**	0.914	**0.850**	0.838	0.921	**0.874**	**0.884**
tf		70.38 %	0.931	0.633	0.440	**0.968**	0.598	0.766
tf-idf		82.38 %	0.832	0.816	0.811	0.836	0.822	0.826
binario	√	87.31 %	0.900	**0.850**	**0.840**	0.906	0.869	0.877
tf	√	80.81 %	**0.951**	0.734	0.650	0.966	0.772	0.834
tf-idf	√	82.5 %	0.826	0.824	0.824	0.826	0.825	0.825

4.1. Análisis de los resultados

Palabras relevantes por clase En la colección de opiniones que se procesó en este trabajo, se buscó eliminar primeramente aquellos términos considerados palabras vacías. Sin embargo, un análisis de aquellos términos más útiles para la clasificación confirmaron la conclusión de [7]: la primera persona del singular es más frecuente en las opiniones engañosas que en las verdaderas. Por este motivo se optó por no eliminar las *stopwords* y contemplarlas como posibles términos relevantes para la clasificación.

Para analizar las palabras más relevantes se halló la frecuencia de estas en cada clase. Ordenadas según la diferencia entre estas frecuencias, la Tabla 2 muestra las primeras y algunas de las últimas palabras de esta relación. Como se puede notar, palabras vacías como *i, my, a, to, on, the* y *at* pueden servir para determinar si las opiniones de esta colección son verdaderas o falsas, al igual que palabras con demasiada frecuencia como *chicago* y *hotel*. Por otro lado, hay palabras como *magnificent* y *complimentary* que no son del todo atípicas pero están distribuidas en las clases casi por igual, siendo así irrelevantes para la clasificación.

Sin embargo, del total de 9527 palabras -solamente *lowercase* como pre-procesamiento-, solo 792 tuvieron ganancia de información mayor que cero, algunas coinciden con las primeras listadas en la Tabla 2, y se observan otras como:

- *priceline*: Ocurre 50 veces en las verdaderas y nunca en las falsas.
- *reviews*: En singular y plural, suman por encima de 100 ocurrencias en las verdaderas, mientras que en las falsas solo 24.
- *ave* y *avenue*: Entre ambas ocurren 114 veces en las verdaderas, y solo 23 en las falsas.
- *experience*: Entre plural y singular, ocurre más del doble de veces en las falsas que en las verdaderas.

Tabla 2. Frec. V: Frecuencia en la clase de opiniones verdaderas, **Frec. F**: Frecuencia en la clase de opiniones falsas, **Dif**: Diferencia entre las frecuencias de ambas clases. En la tabla de la izquierda las 20 palabras con mayor diferencia entre las dos clases; en negrita la frecuencia en la clase en la que es mucho más común. En la tabla de la derecha las últimas 20 palabras con menor diferencia de frecuencia entre las clases.

Palabras	Frec. V	Frec. F	Dif
i	2432	**3799**	1367
my	783	**1526**	743
chicago	484	**1020**	536
a	**3463**	3003	460
is	**1257**	875	382
was	2737	**3110**	373
to	3201	**3559**	358
hotel	1485	**1808**	323
we	**1724**	1422	302
on	**951**	686	265
the	**8133**	7882	251
great	**549**	316	233
location	**349**	134	215
for	**1554**	1342	212
very	**752**	582	170
be	436	**600**	164
at	1094	**1257**	163
no	**382**	234	148
floor	**202**	58	144
from	**559**	431	128

Palabras	Frec. V	Frec. F	Dif
magnificent	41	40	1
complimentary	39	38	1
end	38	37	1
fine	33	32	1
arrival	32	31	1
poor	30	31	1
immediately	29	30	1
provided	27	26	1
paying	23	24	1
fresh	23	24	1
seem	22	23	1
loud	22	23	1
standard	20	21	1
system	20	21	1
sink	19	20	1
uncomfortable	20	19	1
makes	19	20	1
surprised	20	19	1
thank	17	18	1
lady	18	17	1

Utilizar estas palabras cuya frecuencia en cada clase, o su ganancia de información, indican que son útiles en la clasificación, podría hacer al modelo sensible de un cambio de colección o dominio. Una cuestión que podría ser menos sensible a este tipo de cambios es el trabajo con patrones o marcadores sintácticos. La siguiente sección tiene algunas observaciones sobre las etiquetas sintácticas de las palabras así como los bigramas de etiquetas.

Marcadores sintácticos por clase El proceso de etiquetar las palabras según la función sintáctica que tienen en la oración recibe el nombre de *POS Tagging*. Un modelo de clasificación que se base en las etiquetas es menos vulnerable a un cambio o incremento de vocabulario respecto a la colección de entrenamiento.

La Tabla 3 muestra la frecuencia de algunas etiquetas dada la clase. Según la colección de opiniones sobre hoteles [7,6] se observa que en la clase de las opiniones verdaderas hay más sustantivos, artículos definidos, adjetivos, verbos en presente y singular, y palabras extranjeras que en la clase de opiniones falsas. Sin embargo, en esta última son más frecuentes los verbos en pasado y los pronombres posesivos.

Tabla 3. Frecuencia de algunas etiquetas según la clase.

POS Tag	**Verdaderas**	**Falsas**
Sustantivo (NN y NNS)	**30,228**	28,802
Artículo Definido (DT)	**14,332**	13,404
Adjetivos (JJ)	**10,843**	9,866
Verbo en presente y singular (VBZ y VBP)	**4,921**	4,041
Palabras Extranjeras (FW)	**28**	5
Verbo en pasado (VBD)	8,263	**9,076**
Pronombre Posesivo (PRP$)	1,911	**2,762**

Al analizar los bigramas de etiquetas (ver Tabla 4 y Tabla 5), o sea, secuencias consecutivas de dos etiquetas, es visible que en algunos casos la diferencia es por encima del doble. A pesar de los errores de cualquier etiquetador automático, se muestra que sí existen —al menos en esta colección— patrones o marcadores sintácticos interesantes como potenciales rasgos para la representación de opiniones.

Tabla 4. Frecuencia de algunos bigramas de etiquetas según la clase.

Bigramas de *Tags*	**Verdaderas**	**Falsas**
NN+FW	**14**	0
EX+VBZ	**74**	34
EX+VBP	**73**	33
WRB+NN	134	**269**
TO+PRP$	106	**214**

Tabla 5. Ejemplos de los bigramas representados en la Tabla 4.

	EX VBP		
NN FW	EX VBZ	WRB NN	TO PRP$
tv etc	there is	why business	to our
policy etc	there are	when dresser	to my
room etc	there isnt	why i	to its
someone knew	there weren't	how rude	to their
las vegas	there wasn't	where they'd	

Las etiquetas *PRP$*, *FW*, *WRB*, *VBZ* y *VB*[4] alcanzan ganancia de información mayor que cero, y si se usan únicamente estas etiquetas como rasgos

[4] Significado de las etiquetas usadas en el documento: artículo definido (DT), *There* (EX), palabra extranjera (FW), adjetivo (JJ), sustantivo en singular o no contable (NN), sustantivo en plural (NNS), pronombre posesivo (PRP$), preposición *to* (TO), verbo en pasado (VBD), verbo en su forma base (VB), verbo

para la clasificación de las opiniones -con pesado *tf* sin normalizar- se logra un 61.31 % de exactitud. Lo cual indica que pueden ser de ayuda para representar las opiniones y distinguir las notas engañosas. Además, usando unigramas y bigramas de etiquetas se obtiene un 66.44 % de exactitud.

5. Conclusiones y trabajos futuros

En este trabajo preliminar se han explorado técnicas simples de representación de las opiniones escritas, sin el empleo de recursos externos, reglas sintácticas ni análisis psicolingüístico, para comprobar su eficacia en la detección del engaño en opiniones. También se observaron algunas características de ambas clases según la colección de opiniones descrita en la Sec. 3.1.

Con los resultados mostrados en la sección anterior podemos afirmar que con técnicas más simples de representación de textos (bolsa de palabras y esquemas de pesados tradicionales) es posible alcanzar un 87.94 % de exactitud; y dependiendo del pesado y el pre-procesamiento, aproximadamente 0.90 en precisión. Seleccionar el pesado y pre-procesamiento adecuado dependerá de cuál es la prioridad: la detección de opiniones engañosas o de opiniones verdaderas.

Además de los pronombres en primera persona del singular [7], se hallaron otras palabras vacías útiles para caracterizar las opiniones falsas, así como otras palabras que por su alta frecuencia suelen ser removidas.

Finalmente se observó que la información sintáctica es importante. Incluso con solo cinco etiquetas sintácticas se puede alcanzar un desempeño arriba del azar.

En nuestro trabajo futuro consideramos pertinente comprobar cada una de estas observaciones sobre otras colecciones de opiniones e incluir la polaridad de las opiniones en el entrenamiento y la clasificación. También se recomienda una fase que consista en clasificar con el pesado *tf* y sin pre-procesamiento (ver Tabla 1), de esta forma se garantiza un buen recuerdo de las opiniones falsas, las cuales pueden ser re-analizadas para detectar las estructuras o características que las hacen ser engañosas.

Agradecimientos. Este trabajo ha sido realizado con el apoyo del Consejo Nacional de Ciencia y Tecnología (CONACyT) a través de la beca No. 613411, y del proyecto CONACYT PDCPN-2014-247870.

Referencias

1. Cincunegui, M., Berdun, F., Armentano, M.G., Amandi, A.: Clasificación de conductas colaborativas a partir de interacciones textuales. In: Argentine Symposium on Artificial Intelligence (ASAI 2015)-JAIIO 44 (Rosario, 2015) (2015)

en presente de la primera o segunda persona del singular (VBP), verbo en presente de la tercera persona del singular (VBZ), adverbio interrogativo (WRB). https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html.

2. Ezpeleta, E., Zurutuza, U., Hidalgo Gómez, J.M.: Los spammers no piensan: usando reconocimiento de personalidad para el filtrado de spam en mensajes cortos
3. Feng, S., Banerjee, R., Choi, Y.: Syntactic stylometry for deception detection. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2. pp. 171–175. Association for Computational Linguistics (2012)
4. Fusilier, D.H., Montes-y Gómez, M., Rosso, P., Cabrera, R.G.: Detecting positive and negative deceptive opinions using pu-learning. Information processing & management 51(4), 433–443 (2015)
5. Mihalcea, R., Strapparava, C.: The lie detector: Explorations in the automatic recognition of deceptive language. In: Proceedings of the ACL-IJCNLP 2009 Conference Short Papers. pp. 309–312. Association for Computational Linguistics (2009)
6. Ott, M., Cardie, C., Hancock, J.T.: Negative deceptive opinion spam. In: HLT-NAACL. pp. 497–501 (2013)
7. Ott, M., Choi, Y., Cardie, C., Hancock, J.T.: Finding deceptive opinion spam by any stretch of the imagination. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. pp. 309–319. Association for Computational Linguistics (2011)
8. Razavi, A.H., Inkpen, D., Uritsky, S., Matwin, S.: Offensive language detection using multi-level classification. In: Canadian Conference on Artificial Intelligence. pp. 16–27. Springer (2010)

ISSN 1870-4069

Minería de opiniones centrada en tópicos usando textos cortos en español

José A. Reyes-Ortiz, Fabián Paniagua-Reyes, Leonardo Sánchez

Universidad Autónoma Metropolitana, Departamento de Sistemas, Ciudad de México, México

{jaro, ldsm}@correo.azc.uam.mx, al2112002241@alumnos.azc.uam.mx

Resumen. Los usuarios expresan sus sentimientos sobre una entidad de un tema específico de manera libre utilizando textos cortos en las redes sociales. El análisis de sentimientos, también conocido como minería de opiniones, se enfoca en examinar estos textos para determinar su polaridad. Este artículo presenta un enfoque para la minería de opiniones basada en tópicos a partir textos de Twitter en español. El objetivo principal es decidir la polaridad de un texto, determinando si el contenido tiene implicaciones positivas, negativas o neutras en la reputación de entidades para un tópico. En este trabajo se utiliza un enfoque supervisado para la clasificación de textos utilizando el modelo bolsa de palabras para la representación de características. La experimentación muestra resultados prometedores, aportando un recurso para el análisis de textos en español.

Palabras clave: Minería de opiniones, identificación de polaridad, análisis de sentimientos, recursos lingüísticos para el español.

Mining of Opinions Centered on Topics Using Short Texts in Spanish

Abstract. Users express their feelings about an entity of a specific topic in a free way using short texts on social networks. Sentiment analysis, also known as opinion mining, focuses on examining these texts to determine their polarity. This article presents an approach to the mining of opinions based on topics from Twitter texts in Spanish. The main objective is to decide the polarity of a text, determining if the content has positive, negative or neutral implications in the reputation of entities for a topic. In this paper a supervised approach is used for the classification of texts using the word bag model for the representation of characteristics. The experimentation shows promising results, providing a resource for the analysis of texts in Spanish.

Keywords: Mining of opinions, identification of polarity, analysis of feelings, linguistic resources for Spanish.

1. Introducción

La red social denominada Twitter es utilizada por diversos usuarios para expresar sus sentimientos sobre un tema, producto, servicio o entidad. Diariamente, miles de textos cortos, no mayores a 140 caracteres, son generados en esta red social. Esto representa una gran cantidad de información que no pueden ser procesadas de manera manual ya que involucraría una tarea costosa y que consume mucho tiempo. Es posible tener acceso a esta información y aplicar técnicas de minería de datos, aprendizaje automático y procesamiento de lenguaje natural para descubrir conocimiento útil. Este conocimiento puede ser acerca de la reputación de una marca, el nivel de aceptación sobre un producto, el sentimiento generado por un acto o evento donde participa un personaje famoso.

Existe una necesidad de contar con herramientas para el análisis de textos cortos, específicamente, para la minería de opiniones. La necesidad se incrementa en los recursos para el análisis de textos en español ya que existe muy poca investigación en este rubro. Con estas herramientas se pueden tomar decisiones oportunas y rápidas, por ejemplo, conocer hacia dónde se debe enfocar un nuevo producto o servicio, mejorar una marca de automóviles o corregir algún aspecto sobre un personaje famoso.

Se obtendrían resultados en tiempo real y las decisiones serían tomadas con datos frescos, es decir, producidos en ese momento o instante. Este es una ventaja de utilizar datos generados en las redes sociales.

Es por ello que, este artículo se centra en aportar un recurso de minería de opiniones utilizando datos de la red social Twitter, para disminuir la carencia de recursos para el idioma español. Estos datos son textos cortos, conocidos como *tweets.* En este trabajo se utiliza un léxico o diccionario obtenido a partir de un corpus mediante aprendizaje automático y utilizando modelos de clasificación. Los textos son clasificados por el tipo de polaridad (positiva, negativa o neutra) que contiene en mensaje en un tópico determinado, a saber: automóviles, bancario y artistas/músicos. Por lo tanto, el enfoque presentado en este artículo es de gran utilidad para la minería de opiniones sobre un tópico ya que identifica la polaridad de un mensaje utilizando una clasificación de textos. Por ejemplo, es posible determinar la polaridad general del tema de automóviles a partir del análisis de los textos generados por usuarios en la red social Twitter.

El resto de este artículo está organizado como sigue. En la Sección 2 se presentan los trabajos relacionados con la minería de opiniones con textos de redes sociales. La Sección 3 presenta el enfoque utilizado para la minería de opiniones que determina la polaridad de un mensaje centrado en un tópico. El conjunto de datos utilizado para el aprendizaje del lexicón y para la experimentación se describe en la Sección 4. Por su parte, la Sección 5, muestra los resultados de la experimentación con cuatro algoritmos (Naïve Bayes, *k*-vecino más cercano, máquinas de soporte vectorial y algoritmo basado en árboles de decisión). Finalmente, las conclusiones y el trabajo a futuro son presentados en la Sección 6.

2. Trabajo relacionado

El análisis de sentimientos y la minería de opiniones con datos de redes sociales han sido temas con gran interés en los últimos años. En estas tareas se encuentra la clasificación de opiniones en Twitter, la cual consiste en determinar la polaridad expresada por un texto corto, es decir, determinar la carga positiva o negativa que contiene dicho texto. En este contexto, la minería de opiniones con datos de redes sociales ha sido abordado desde diversas perspectivas en años recientes, como las que se describen a continuación.

Enfoques supervisados basados en análisis estadísticos han sido utilizados en [1], donde los autores utilizan los medios de comunicación social, específicamente, Twitter y Facebook para el análisis de polaridad sobre personas, entidades y marcas. Ellos presentan un marco de trabajo organizado por módulos, en el cual se puede experimentar con diversos clasificadores como Naïve Bayes, máquinas de soporte vectorial, árboles de decisión y *k*-vecino más cercano; donde el análisis de polaridad se centra en clasificar textos en tres categorías: positivo, negativo y neutro. En [2] se presenta un enfoque híbrido para la clasificación de textos extraídos de Twitter. El enfoque presentado considera la minería de opiniones utilizando una lista de símbolos y un análisis con *SentiWordNet*, además usan la frecuencia de palabras negativas y positivas; por último, la clasificación de los textos se centra en tres categorías: positivos, negativos y neutros. También, en [3] presentan un algoritmo de aprendizaje automático para clasificar la polaridad de los mensajes en español; los autores realizan un estudio de diferentes características como mensajes reenviados, menciones, ligas y etiquetas con el símbolo '#', los autores generan un corpus en español llamado COST y experimentaron con algoritmos de aprendizaje supervisado como Naïve Bayes (NB), máquinas de soporte vectorial y el algoritmo de Regresión Logística (LR).

Otro mecanismo de apoyo o soporte que se ha aplicado en la minería de opiniones son las ontologías. Como en [4] que utilizan una técnica basada en ontologías para la minería de mensajes generados en Twitter. La novedad del enfoque propuesto es que los mensajes se caracterizan con grados de sentimiento distintos por cada tema existente en el mensaje. Esto genera un análisis más detallado de las opiniones de los mensajes sobre un tema específico.

El uso de *n-gramas* y el modelo denominado bolsa de palabras se expone en los siguientes trabajos. En [5] se presenta un enfoque basado en una clasificación supervisada de textos utilizando unigramas, bigramas y trigramas con un análisis de frecuencias; el análisis estadístico genera un léxico específico y reducido para Twitter, el cual es utilizado para el análisis de sentimientos y está compuesto por 187 características que reduce la complejidad del modelo, mientras mantiene un alto grado de cobertura del corpus y, además, produce una mejor precisión para la clasificación de sentimientos.

En análisis de sentimientos para textos entre diferentes lenguajes ha sido propuesto por [6], quienes proponen un enfoque simple para múltiples lenguajes, basado en el modelo espacio vectorial, empleando características que pueden ser utilizadas entre tres lenguajes (español, inglés e italiano) y características independientes del lenguaje, entre las que destacan: operadores de negación, si es una palabra derivada o no, entre otros.

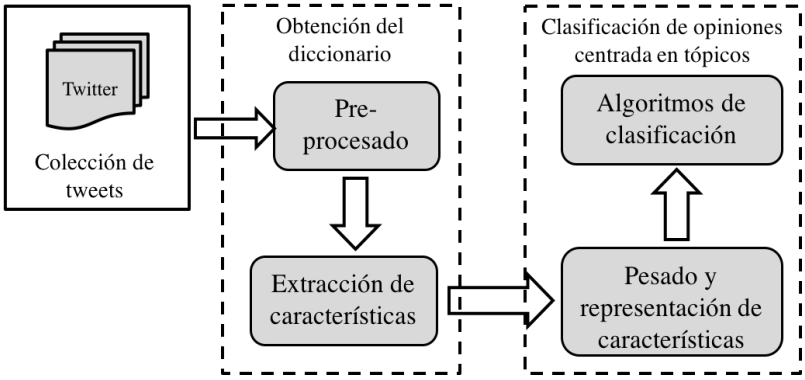


Fig. 1. Arquitectura del enfoque para la minería de opiniones.

Finalmente, en algunos casos, lexicones o diccionarios son combinados en un enfoque basado en aprendizaje supervisado. Estos enfoques aportan una solución al problema de la clasificación de sentimientos u opiniones expresadas en mensajes de Twitter o Facebook; como se presenta en [7], donde se propone un método que adopta un lexicón para llevar a cabo el análisis de sentimientos a nivel de entidades, logrando una alta precisión pero con una baja sensibilidad; en [8] se propone un método basado en un lexicón para el análisis de sentimientos utilizando mensajes de Facebook en vietnamita, los autores construyen un diccionario de emociones para el vietnamita incluyendo cinco sub-diccionarios: sustantivo, verbo, adjetivo, adverbio y un diccionario adaptado del inglés; en [9] presentan los pasos detallados para la construcción de los componentes que componen un enfoque basado en lexicones para el análisis de sentimientos.

La mayoría de trabajos del estado del arte se centran en el idioma inglés. Sin embargo, existe una necesidad de recursos para la minería de opiniones en español. Se ha detectado una carencia de enfoques para el español con las perspectivas descritas anteriormente. Con respecto a esto, este trabajo centra su aportación en reducir esta falta de recursos para la minería de opiniones, presentando un enfoque supervisado para la identificación de polaridad centrada en tópicos utilizando textos generadas en Twitter en idioma español.

La minería de opiniones consiste en determinar si un mensaje expresa una polaridad positiva, negativa o neutra sobre un tópico específico, tales como: automóviles, bancos y artistas/músicos.

3. Minería de opiniones centrada en tópicos

En esta sección se presentan los componentes del enfoque para la minería de opiniones, es especifico, la identificación de una polaridad mediante una clasificación de textos cortos centrada en tópicos para el español. Estos componentes se observan en la Figura 1, los cuales incluyen diversas tareas como el pre-procesado de los textos, la

Tabla 1. Normalización de la risa.

Patrón	Frase	Risa normalizada
(ja)+	ja	jaja
(je)+	jeje	jaja
(jo)+	jojojo	jaja
(ji)+	jijijiji	jaja
lol	lol	jaja

Tabla 2. Forma enraizada de palabras para el español.

Palabra	Raíz
buenos	buen
rebajarán	rebaj
fueron	ir
llegaría	lleg

extracción de características o la obtención del lexicón/diccionario, la ponderación de las características considerando la frecuencia de aparición de los términos y finalmente, los algoritmos de clasificación supervisada que se centran en tres tópicos: automóviles, bancos y música.

3.1. Pre-procesado de textos

La primera tarea para la obtención de un diccionario o lexicón de palabras, es la limpieza de los textos, para ello se realiza una segmentación por palabras (*tokens*) y la eliminación de caracteres especiales. Después, las unidades léxicas son filtradas eliminando las ligas (*url*) a sitios web externos, menciones de usuarios en Twitter (@), entidades nombradas para cada dominio, ya sea directamente o mediante *hashtag* (#). Por ejemplo, para el dominio automóviles fueron eliminadas menciones a entidades como: #bmw, #volvo, #ferrari, entre otras.

También, en esta fase se lleva a cabo una normalización de las unidades léxicas resultantes a minúsculas y se eliminan las *stopwords*, palabras que no aportan significado y por lo tanto, no son funcionales para la identificación de polaridad. Esta lista de palabras contiene artículos (un, la, los), preposiciones (a, con, de, para), verbos no funcionales (ser, estar), entre otros. Sin embargo, se descartan de esta lista palabras de negación (no, ni) o afirmación (si), al ser consideradas como funcionales para la identificación de la polaridad manifestada por un mensaje (*tweet*).

Para evitar redundancia en la forma de expresar una risa en los textos, en esta etapa del pre-procesado, se considera un paso de normalización. Para el cual, se aplican reglas y se sustituyen las diversas formas de expresar risa por un término en común, como se muestra en la Tabla 1, donde el símbolo (+) significa una o más ocurrencias, es decir que una secuencia, por ejemplo "*ja*", se puede repetir una o más veces en los textos.

Finalmente, en esta fase, para cada palabra del léxico obtenido hasta este punto, se lleva a cabo un proceso *Stemming*, el cual consiste en reducir una palabra a su raíz, es

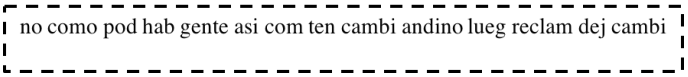


Fig. 2. Algunas palabras del diccionario (bolsa de palabra) como características.

decir, eliminar los sufijos o flexiones de las palabras. Esto permite agrupar todas las palabras con la misma raíz en una sola representación en el lexicón. Para esta tarea se utiliza el, bien conocido, algoritmo de *Porter – Snowball Stemmer* [10, 11], el cual tiene soporte para el español. En la Tabla 2, se muestra un ejemplo de palabras en español como aparecen en los textos y con su forma derivada (raíz) que es generada por el algoritmo.

3.2. Extracción de características

El diccionario o lexicón es obtenido como las unidades léxicas de todos los textos sin repeticiones. Este lexicón constituye las características para la etapa de clasificación y se utiliza la representación mediante bolsa de palabras (*bag-of-words*).

Con esta tarea de extracción se obtiene un diccionario de palabras normalizado y reducido, el cual será utilizado para representar cada instancia (*tweet*) mediante una ponderación, para su posterior clasificación en positivo, negativo o neutro. Un total de 6037 palabras fueron obtenidas a partir del conjunto de datos (corpus), las cuales conforman el vocabulario o diccionario.

La Figura 2 muestra una lista de algunas palabras del diccionario extraído, el cual funciona como el conjunto de características para la tarea denominada identificación de polaridad mediante clasificación textual. Las características mostradas en la Figura 2 son ponderadas con la métrica denominada "Frecuencia de los términos- Frecuencia inversa en los documentos (TF-IDF)", la cual se describe en la Sección 3.3.

3.3. Ponderación de las características

En este trabajo, confiamos en una ponderación de las características basada en la importancia de los términos en un mensaje (*tweet*) enfocada en tópicos. De manera específica, la ponderación de los términos con la finalidad de clasificar los textos está centrada en tres tópicos: automóviles, bancos, y artistas/música.

Existen diferentes enfoques para obtener la importancia o ponderación de los términos del vocabulario sobre un texto corto. Este vocabulario es representado mediante el modelo espacio vectorial con la representación llamada bolsa de palabra (*BoW* por sus siglas en inglés) [12], el cual consiste en una colección de textos y su vocabulario de términos (características). Cada *tweet* es representado como un vector $S_j = (w_{1j}, w_{2j} \dots w_{nj})$, donde cada componente w_{ij} expresa la importancia que produce la característica i, palabra del vocabulario, en el mensaje j.

Para el pesado de los términos o palabras, es decir, determinar su importancia en un *tweet*, se utiliza el pesado basado en la frecuencia de aparición del término dentro de una colección de textos para un tópico (TF-IDF).

Esta ponderación utiliza la frecuencia de aparición para los términos del vocabulario en un texto, la cual consiste en el número de veces que un término (*t*) del vocabulario aparece en un *tweet* (*S*), ver Ecuación 1, y la frecuencia inversa que determina si el término es común en la colección (Ecuación 2). Esta información se utiliza, entonces, para calcular el valor final de *TF-IDF* (Ecuación 3):

$$TF(t_i, S_j) = \mathrm{f}\,(t_i, S_j), \tag{1}$$

$$IDF(t_i, S_j) = \log \frac{|S|}{1 + |s \in S \,:\, t_i \in s|}, \tag{2}$$

$$w_{ij} = TF(t_i, S_j) \times IDF(t_i, S_j). \tag{3}$$

3.4. Identificación de polaridad centrada en tópicos

La identificación de una polaridad, se centra en una tarea típica de clasificación supervisada, la cual se basa en el vector ponderado de los términos del vocabulario con respecto a cada mensaje. La ponderación de los términos se centra en la métrica *TF-IDF* presentada previamente y la clasificación en un enfoque supervisado. Estos vectores son la entrada para la etapa de clasificación, la cual ha sido desempeñada con diversos algoritmos para decidir la polaridad de los textos.

El objetivo de esta fase es construir un clasificador textual capaz de predecir la polaridad de un mensaje en tres posibles categorías: positivo, negativo y neutro. Para ello, es necesario dividir los textos en dos subconjuntos: entrenamiento y prueba.

Como ya se ha mencionado anteriormente, el clasificador está centrado en tópicos, es decir, se obtiene un mismo vocabulario para el todos los mensajes. Sin embargo, la ponderación y clasificación está focalizada en tres tópicos, a saber: automóviles, bancos, y artistas/música.

La tarea de clasificación centrada en los tópicos se lleva a cabo mediante cuatro algoritmos: el clasificador Naïve-Bayes (NB) que se basa en el teorema de Bayes y su función es encontrar la hipótesis más probable que describa los vectores que representan a los textos de prueba, con esto obtiene la probabilidad para que dado los valores que describen a un *tweet*, éste pertenezca a una clase dada [13]; las máquinas de soporte vectorial (SVM) [14] que construyen un conjunto de hiperplanos en un espacio n-dimensional con los textos de entrenamiento, estos hiperplanos son utilizados para predecir la clase de los nuevos textos; algoritmo basado en árboles de decisión (C4.5) es un algoritmo que realiza la inducción a partir de ejemplos preclasificados generando un árbol de decisión con los datos, mediante particiones realizadas recursivamente [15]; y, el algoritmo del k-vecino más cercano (kNN) [16] que estima la función de densidad para los pares a predecir por cada clase basándose en el conjunto de entrenamientos y prototipos.

La idea es evaluar la tarea de clasificación, combinando los cuatro algoritmos (NB, SVM, C4.5 y kNN) con la ponderación de los términos (TF-IDF), para encontrar la mejor solución en cuanto a precisión y cobertura. La implementación de los algoritmos de clasificación se ha llevado a cabo mediante la herramienta WEKA [17].

Tabla 3. Distribución de los textos por tópicos.

Tópico	Textos de entrenamiento	Textos de prueba	Total
Automóviles	1388	716	**2104**
Bancario	1215	626	**1841**
Artistas/Músicos	1993	1027	**3020**
Total	**4596**	**2369**	**6965**

Tabla 4. Resultados para la identificación de polaridad centrada en tópicos utilizando el algoritmo Naïve Bayes.

Tópico	Automóviles			Bancos			Artistas/Músicos		
	P	R	F	P	R	F	P	R	F
Positiva	0.51	0.59	0.55	0.69	0.74	0.72	0.68	0.60	0.64
Negativa	0.58	0.56	0.57	0.68	0.66	0.67	0.67	0.63	0.65
Neutra	0.51	0.54	0.52	0.61	0.70	0.66	0.52	0.63	0.57

Tabla 5. Resultados para la identificación de polaridad utilizando el algoritmo C4.5 (basado en árboles de decisión).

Tópico	Automóviles			Bancos			Artistas/Músicos		
	P	R	F	P	R	F	P	R	F
Positiva	0.69	0.80	0.74	0.77	0.79	0.78	0.77	0.70	0.74
Negativa	0.78	0.72	0.75	0.83	0.82	0.82	0.79	0.83	0.81
Neutra	0.69	0.63	0.66	0.69	0.77	0.73	0.73	0.74	0.73

4. Conjunto de datos

La evaluación del enfoque para la identificación de polaridad centrada en tópicos fue realizada con el conjunto de datos proporcionado por la competencia RepLab [18] para la tarea específica denominada "polaridad de la reputación", cuyo objetivo es decidir si el contenido de un mensaje (*tweet*) en español tiene implicaciones positivas o negativas para la reputación de una entidad, tal como: marca automotriz, entidad financiera, institución educativa o persona famosa en la música.

El conjunto original de datos consta de cuatro tópicos: automóviles, bancario, universidades y artistas/músicos. Sin embargo, en este artículo solo se toma el conjunto de datos de tres tópicos, debido a que el conjunto para el tópico universidades no está balanceado con respecto a los otros tópicos, ya que contiene únicamente 223 mensajes efectivos a diferencia de 3020 para el tópico artistas.

A partir del conjunto de datos seleccionado, se obtuvieron 6965 textos efectivos, para los cuales fue posible obtener su contenido de Twitter y que, además, estaban clasificados manualmente con su etiqueta o categoría para la polaridad (Positivo, Negativo, Neutro). Este conjunto de datos representa un excelente marco de referencia

Tabla 6. Resultados de la experimentación utilizando el algoritmo kNN (*k*-vecino más cercano).

Tópico	Automóviles			Bancos			Artistas/Músicos		
	P	R	F	P	R	F	P	R	F
Positiva	0.54	0.66	0.60	0.75	0.87	0.81	0.61	0.73	0.67
Negativa	0.47	0.51	0.49	0.85	0.73	0.79	0.63	0.67	0.65
Neutra	0.39	0.49	0.44	0.74	0.62	0.68	0.47	0.55	0.51

para la evaluación de algoritmos de minería de opiniones. El conjunto de datos finales se divide en tres tópicos, que a su vez son divididos en dos conjuntos, 66% para el entrenamiento y 34 % para la evaluación, quedando distribuido de la forma como se muestra en la Tabla 3.

5. Experimentación y resultados

La experimentación consiste en utilizar cada uno de los algoritmos con la ponderación TF-IDF, utilizando el mismo conjunto de datos centrado en cada tópico para el entrenamiento y pruebas.

La evaluación de todos los experimentos se realiza utilizando las métricas *Precisión (P)*, *Cobertura (R)* y *medida F*, las cuales han sido ampliamente utilizadas en cualquier tarea de clasificación textual. Estas métricas comparan los resultados del clasificador a evaluar con los valores externos de confianza (texto preclasificado), utilizando los siguientes valores: a) *Verdadero Positivo (VP)* es el número de predicciones correctas del clasificador que corresponden al juicio externo de confianza (texto preclasificado); *Verdadero Negativo (VN)* es el número de predicciones correctas del clasificador de opiniones que no corresponden al juicio externo de confianza; *Falso Positivo (FP)* corresponde al número predicciones incorrectas del clasificador que corresponden al juicio externo de confianza; y, finalmente *Falso Negativo (FN)* es el número de predicciones incorrectas del clasificador que no corresponden al juicio externo de confianza.

Bajo estos criterios, se emplea la *Precisión (P)* para evaluar los algoritmos en términos de los valores de predicciones positivas, la cual se define, en la Ecuación 4, como:

$$P = \frac{VP}{VP + FP}. \tag{4}$$

También, se utiliza el *Cobertura (R)* para expresar la tasa de correspondencias correctas con las opiniones de textos preclasificados de manera externa con una alta confianza (Ecuación 5):

$$R = \frac{VP}{VP + FN}. \tag{5}$$

Tabla 7. Resultados para la identificación de polaridad utilizando SVM (máquinas de soporte vectorial).

Tópico	**Automóviles**			**Bancos**			**Artistas/Músicos**		
	P	R	F	P	R	F	P	R	F
Positiva	0.79	0.83	0.81	0.88	0.86	0.87	0.84	0.86	0.85
Negativa	0.81	0.83	0.82	0.87	0.87	0.87	0.87	0.85	0.86
Neutra	0.79	0.80	0.79	0.81	0.89	0.85	0.84	0.83	0.83

Tabla 8. Resumen de resultados de los algoritmos por tópico

Tópico	Automóviles		Bancos		Artistas/Música		Promedio	
Algoritmo	C	I	C	I	C	I	C	I
NB	54.9	45.1	68.3	31.7	63.2	36.8	62.1	37.9
C4.5	71.5	28.5	77.8	22.2	76.1	23.9	75.2	4.8
kNN	55.3	44.7	74.2	25.8	65.4	34.6	64.9	35.1
SVM	82.3	17.7	87.3	12.7	84.6	15.4	84.7	15.3

Finalmente, la *medida F* que representa la media armónica entre *Precisión* y *Cobertura*, la cual tiene como fundamento el obtener un valor único ponderado entre ellas (Ecuación 6):

$$medida\ F = 2 * \frac{P * R}{P + R}. \tag{6}$$

Todos los experimentos utilizan la ponderación de palabras mediante la métrica TF-IDF. Por su parte, los resultados de esta experimentación están centrados en los tres tópicos mencionados y se analizan por cada categoría (Positiva, Negativa, Neutra) para cada algoritmo de clasificación.

La Tabla 4 muestra los resultados, por tópico, de los experimentos utilizando el algoritmo Naïve Bayes.

La Tabla 5 muestra los resultados de los experimentos utilizando el algoritmo C4.5 (basado en árboles de decisión) por tópico.

La Tabla 6 muestra los resultados de los experimentos utilizando el algoritmo kNN (*k*-vecino más cercano) por tópico.

La Tabla 7 muestra los resultados de los experimentos utilizando el algoritmo SVM (máquinas de soporte vectorial) por tópico.

Finalmente, la Tabla 8 hace un resumen de los resultados para cada algoritmo por tópico y en términos prácticos para el análisis: porcentaje de instancias clasificadas correctamente (C) para las tres categorías y el porcentaje de instancias clasificadas incorrectamente (I).

La Tabla 8, que representa un resumen de los resultados, hace notar que el algoritmo de clasificación llamado Máquinas de Soporte Vectorial clasifica el conjunto de datos con el mejor porcentaje (84.7 en promedio), mostrando, los mejores resultados para el

tópico bancario con un 87.3 de instancias clasificadas correctamente, comportamiento que se presente en todos los algoritmos.

6. Conclusiones y trabajo futuro

Este artículo ha presentado un enfoque para la minería de opiniones centrada en tópicos, la cual consiste en identificar la polaridad de un mensaje (*tweet*). Para ello, se ha presentado un esquema para la clasificación de mensajes en tres categorías: positivo, negativo y neutro. La clasificación se lleva a cabo mediante cuatro algoritmos (NB, SVM, C4.5 y kNN). Para la clasificación se ha obtenido un vocabulario a partir de un conjunto de textos en español y posteriormente, este vocabulario ha sido ponderado mediante la métrica TF-IDF que se basa en la frecuencia de aparición del término en una colección de textos de un tópico.

A partir de los experimentos, se ha notado un mejor resultado en el algoritmo SVM en cuanto a precisión y cobertura para la identificación de polaridad utilizando los mensajes de Twitter, alcanzando un porcentaje promedio cercano a 85% de instancias clasificadas correctamente.

Adicionalmente, debido a que el objetivo de este trabajo es centrar la clasificación en tópicos, se puede aportar que la mejor identificación de polaridad se presenta en el tópico "Bancario" obteniendo una los promedios de cobertura más altos, para las tres categorías, en todos los algoritmos.

Las principales contribuciones de este trabajo son: a) un enfoque para la minería de opiniones que representa un recurso valioso para el análisis en textos extraídos de las redes sociales para el idioma español; b) la comparativa de cuatro algoritmos bajo el mismo escenario y configuración de los experimentos para la predicción de polaridad en los textos.

Como trabajo futuro, es posible realizar una clasificación basada en entidades ya que el conjunto de datos o corpus original incluye mensajes de texto para 61 entidades divididas en los cuatro tópicos mencionados. Un análisis de sentimientos combinando pares con la configuración *tópico: entidad* resultaría de gran utilidad para determinar la reputación de una entidad en un tópico específico.

Agradecimientos. Los autores agradecen al PRODEP-SEP, SNI-CONACyT y a la Universidad Autónoma Metropolitana Azcapotzalco por las facilidades proporcionadas para la realización de este artículo.

Referencias

1. Lima, A. C. E., de Castro, L. N., Corchado, J. M. A.: Polarity analysis framework for Twitter messages. Applied Mathematics and Computation, Vol. 270, pp. 756−767 (2015)
2. Khan, F. H., Bashir, S., Qamar, U.: TOM: Twitter opinion mining framework using hybrid classification scheme. Decision Support Systems, Vol. 57, pp. 245−257 (2014)

3. Martínez-Cámara, E., Martín-Valdivia, M. T., Ureña-López, L. A., Mitkov, R.: Polarity classification for Spanish tweets using the COST corpus. Journal of Information Science (2015)
4. Kontopoulos, E., Berberidis, C., Dergiades, T., Bassiliades, N.: Ontology-based sentiment analysis of twitter posts. Expert systems with applications, Vol. 40, No. 10, pp. 4065−4074 (2013)
5. Ghiassi, M., Skinner, J., Zimbra, D.: Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network. Expert Systems with applications, Vol. 40, No. 16, pp. 6266−6282 (2013)
6. Tellez, E. S., Jiménez, S. M., Graff, M., Moctezuma, D., Suárez, R. R., Siordia, O. S.: A Simple Approach to Multilingual Polarity Classification in Twitter. CoRR abs/1612.05270 (2016)
7. Khan, A. Z., Atique, M., Thakare, V. M.: Combining lexicon-based and learning-based methods for Twitter sentiment analysis. International Journal of Electronics, Communication and Soft Computing Science & Engineering (IJECSCSE), 89 (2015)
8. Trinh, S., Nguyen, L., Vo, M., Do, P.: Lexicon-based sentiment analysis of Facebook comments in Vietnamese language. In: Recent Developments in Intelligent Information and Database Systems, pp. 263−276 (2016)
9. Abdulla, N. A., Ahmed, N. A., Shehab, M. A., Al-Ayyoub, M., Al-Kabi, M. N., Al-rifai, S.: Towards improving the lexicon-based approach for Arabic sentiment analysis. In Big Data: Concepts, Methodologies, Tools, and Applications, IGI Global, pp. 1970−1986 (2016)
10. Porter, M. F.: An algorithm for suffix stripping. Program, Vol. 14, No. 3, pp.130−137 (1980)
11. Karen, S. J., Peter, W.: Readings in Information Retrieval. San Francisco: Morgan Kaufmann (1997)
12. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys, 34 (1), pp. 1−47 (2002)
13. Aha, D. W., Kibler, D., Albert, M. K.: Instance-based learning algorithms. Machine Learning, Vol. 6, No. 1, pp. 37–66 (1991)
14. Chang, Ch., Lin, Ch.: LIBSVM - A Library for Support Vector Machines. ACM Transactions on Intelligent Systems and Technology (TIST), Vol. 2, No. 3, pp. 27−28 (2001)
15. John, G. H., Langley, P.: Estimating continuous distributions in Bayesian classifiers. In: Eleventh Conference on Uncertainty in Artificial Intelligence (UAI'95), Montreal, Canada, pp. 338−345 (1995)
16. Quinlan, J. R.: C4.5: Programs for Machine Learning. San Mateo, Calif.: Morgan Kaufmann Publishers (1993)
17. Garner, S.R.: Weka: The Waikato environment for knowledge analysis. In: Proc. of the New Zealand Computer Science Research Students Conference, pp. 57−64 (1995)
18. Amigó, E., De Albornoz, J. C., Chugur, I., Corujo, A., Gonzalo, J., Martín, T., Spina, D.: Overview of replay 2013: Evaluating online reputation monitoring systems. In: International Conference of the Cross-Language Evaluation Forum for European Languages, pp. 333−352, Springer Berlin Heidelberg (2013)

ISSN 1870-4069

Generación y enriquecimiento automático de recursos léxicos para el análisis de sentimientos

Gerardo Real-Flores[1], Betzabet García-Mendoza[1], Ester Calderón-Casanova[1], Gabriela Ramírez-de-la-Rosa[2], Esaú Villatoro-Tello[2]

[1] Maestría en Diseño, Información y Comunicación (MADIC),
División de Ciencias de la Comunicación y Diseño,
Universidad Autónoma Metropolitana (UAM) Unidad Cuajimalpa, México

[2] Departamento de Tecnologías de la Información,
Universidad Autónoma Metropolitana (UAM) Unidad Cuajimalpa, México

gereflo@aol.com,{betza07gm,estercalderon7}@gmail.com,
{gramirez,evillatoro}@correo.cua.uam.mx

Resumen. La generación de contenidos por parte de los usuarios de internet en diversas plataformas, genera oportunidades y retos a los interesados en el análisis de sentimientos ya que se pueden clasificar grandes cantidades de información y conocer la opinión de los usuarios respecto a diversos temas. Una de las desventajas del análisis de sentimientos es que los dominios y por lo tanto las entidades a las que se les presenta una opinión cambian constantemente y de forma muy rápida. Por ello, en el enfoque de clasificación basada en recursos léxicos, la obtención de diccionarios adaptables a diferentes dominio, idiomas, entidades, es algo muy costoso. En este artículo proponemos un esquema de generación de recursos léxicos a la medida del dominio que se desea analizar, esto reduce el costo en tiempo y esfuerzo de la creación de un recurso léxico genérico. Adicionalmente, se presenta un método de enriquecimiento de recursos léxicos existentes usando información del dominio en análisis. Los resultados obtenidos son alentadores, donde se obtienen mejoras de hasta el doble en F-score que al usar un recurso genérico como ANEW para la clasificación de opiniones de revisiones de películas.

Palabras clave: Representación distribucional, generación de recurso léxico, análisis de sentimientos, clasificación de sentimientos.

Automatically Generating and Enriching Lexical Resources for Sentiment Analysis

Abstract. The big amount of content generated from Internet users creates opportunities and challenges to researchers interested in the problem of sentiment analysis. All the information available regarding user's opinions on different topics represent a great opportunity for automatic sentiment classification systems. However, user's interest change very

rapidly over time, making automatic classification systems obsolete to new domains. In addition, tunnig lexical resources and training data for creating new classification models is a very expensive and complicated task. In order to overcome such limitations, we propose a method for automatically generate a lexical resource from scratch, which in turn is useful in the training process of new classification models. Additionally, our proposed method allows to enrich an existing lexical resource. In general, our proposed method reduces time and effort costs associated to the creation of lexical resources. Basically, our proposed method analyses the semantic information contained in the data to be classified for building the lexical resource. We evaluate our proposal in two well known datasets for sentiment analysis, obtaining encouraging results in contrast to the use of generic lexical resources such as ANEW.

Keywords: Distributional terms representation, lexical resources generation, sentiment analysis, sentiment classification.

1. Introducción

En los últimos años la enorme cantidad de contenidos generados por los usuarios de sitios Web, *blogs*, *wikis*, redes sociales, entre otros, ha dado oportunidad a empresas, gobierno e incluso personajes públicos de conocer la opinión de los usuarios respecto a diversos temas y desde distintos enfoques. Una estrategia para aprovechar estas opiniones expuestas en dichos medios digitales es el análisis de sentimientos, una tarea particular del Procesamiento del Lenguaje Natural (PLN). El análisis de sentimientos se enfoca en analizar las opiniones, sentimientos, actitudes y emociones del lenguaje escrito [10]. El resultado de este análisis deriva en la clasificación de opiniones positiva, negativa o en algunos casos neutra.

Usualmente el análisis de sentimientos (AS) se aborda como una tarea de clasificación de textos, donde existen clases predefinidas cuyo objetivo principal es determinar a cuál de estas clases pertenece un documento nuevo. En el caso de AS, las clases comúnmente usadas son positiva y negativa. Una particularidad del análisis de sentimientos es que es una tarea de clasificación no temática, pues usualmente una opinión se genera sobre un mismo tema, producto, persona y/o entidad.

Históricamente, el AS se ha abordado desde tres diferentes enfoques: i) basados en estrategias de aprendizaje supervisado, ii) basados en recursos léxicos, e iii) híbridos, una combinación de los dos enfoques previos. Por un lado, los métodos de aprendizaje supervisado dependen de la existencia de una gran cantidad de documentos etiquetados además de la definición de una representación para estos documentos. Específicamente, para el AS, al ser una tarea de clasificación no temática, la representación utilizada para describir a estas opiniones tiene gran relevancia. Uno de los modelo de representación de documento más usado, debido a su simplicidad, es la bolsa de palabras (BoW) [11].

Por otro lado, en el segundo enfoque, aquellos basados en recursos léxicos, si bien no requieren un conjunto de documentos etiquetados y la definición de una representación particular, sí necesitan la compilación del recurso léxico que contenga un conjunto de términos o palabras asociadas a un tipo de sentimiento. La construcción de estos recursos son realizados por especialistas, por ejemplo lingüístas, lo que hace la generación de estos recursos una tarea costosa [12]. Un ejemplo de un recurso léxico de este tipo es ANEW (*Affective Norms for English Words*), el cual proporciona un conjunto de valoraciones emocionales normativas para 1030 palabras en el idioma inglés [1], cabe mencionar que también existe una adaptación de este recurso para idioma español con 1034 palabras [13].

Los recursos léxicos de este tipo son genéricos y se usan igual para diferentes dominios, sin embargo esto no garantiza que una palabra que sea positiva para un dominio, dígase revisiones de películas, lo sea también para un dominio distinto, dígase revisiones de libros. En este sentido, en este trabajo se exploran dos ideas para generar recursos léxicos adaptables al dominio en análisis: i) generar de manera automática una lista de palabras positivas y negativas a partir de un conjunto de documentos etiquetados y, ii) extender un recurso léxico inicial agregando información con carga positiva o negativa semánticamente similar a las palabras del recurso léxico inicial obtenidas de un conjunto de documentos del dominio de estudio.

El resto de este artículo está organizado como sigue: en la Sección 2 se hace un repaso de los métodos empleados en el análisis de sentimientos, luego en la Sección 3 se describe el esquema de generación automática de recursos léxicos. En la Sección 4 brevemente se mencionan dos de los enfoques de clasificación usados en el análisis de sentimientos. La configuración experimental y detalles de la evaluación realizada se muestra en la Sección 5. Finalmente, las conclusiones y trabajo a futuro se presenta en la Sección 6.

2. Trabajo relacionado

La principal tarea en el análisis de sentimientos es clasificar un texto dado un documento, esto significa determinar si la opinión expresada es positiva, negativa o neutral. Las técnicas usuales para clasificar sentimientos pueden dividirse en tres enfoques: basado en técnicas de aprendizaje supervisado, basado en recursos léxicos y un enfoque combinado de las dos anteriores. En este trabajo nos enfocaremos los métodos que usan recursos léxicos ya que son los que se vinculan con el enfoque de nuestra propuesta.

El enfoque basado en recursos léxicos se basa en una colección de palabras previamente recopiladas y etiquetadas en términos de sentimientos. Un ejemplo es ANEW que se encuentra disponible en inglés [1] y una adaptación al español [13]. La fortaleza de este tipo de recurso es que ha sido desarrollado y cuidadosamente evaluado por especialistas en el área de lingüística, prueba de su eficiencia se presenta en [4] donde reportan resultados de precisión del 90 % con la versión adaptada al idioma español. Adicionalmente, en este trabajo se creó un recurso

léxico propio para clasificar opiniones de servicios turísticos en español, con este nuevo recurso obtienen como máximo 80 % en términos de precisión.

En este mismo sentido, existen otros esfuerzos que han planteado la generación semi o automática de recursos léxicos. Un ejemplo de esto es el trabajo realizado por [8], donde mediante de recolección de datos de Internet, un filtrado cuidadoso de los datos y etiquetado POS logra extraer un conjunto de palabras para clasificar opiniones de películas en inglés. Este recurso llamado Senti-CS plantea ser una mejora al recurso SentiWordNet [3]. Sus resultados están reportados en Exactitud y llegan a un 86.1 %. A pesar de los resultados prometedores de este conjunto de métodos su construcción requiere procesos en su mayoría costosos. Más aún, la mayoría de estos esfuerzos van enfocados a la construcción del recursos en inglés.

Un enfoque similar al que se propone en este artículo es descrito en [7]. Los autores proponen la construcción de un recurso léxico a partir de una colección masiva de documentos HTML de donde se extraen oraciones con polaridad (positiva o negativa) usando pistas estructurales. Al resultado lo denominan como corpus de oraciones polares, de ahí se determinan las oraciones polares por el mayor número de veces que aparece en oraciones positivas o negativas para finalmente seleccionar las oraciones con (mayor) polaridad y agregarlas al recurso léxico. Reportan resultados 92 % de exactitud sustentando la solidez de su propuesta.

En sentido opuesto a los trabajos mencionados, nuestra propuesta es la construcción de recursos léxicos a bajo costo. Usando solo información obtenida del dominio al cual pertenecen las opiniones que se desean clasificar. Más aún, nuestro esfuerzo también va enfocado al mejoramiento de recursos léxicos existentes usando información del dominio a analizar.

3. Método propuesto

Como se ha establecido previamente, el objetivo de este trabajo es generar un recurso léxico particular a un dominio para el análisis de sentimientos. Este recurso léxico consiste en una lista de palabras asociada a una polaridad, i.e, positiva y negativa.

El método propuesto cuenta con dos etapas, en la primera se obtiene una lista inicial de palabras positivas y negativas, i.e, un recurso léxico inicial, el objetivo aquí es explorar la generación de un recurso léxico nuevo usando información de documentos etiquetados como positivos o negativos. En la segunda etapa se busca enriquecer un recurso léxico inicial utilizando información semánticamente relacionada al conjunto de palabras positivas y negativas iniciales. En la figura 1 se presenta la arquitectura general del método propuesto y en las siguientes secciones se explican cada una de las etapas y subetapas que lo conforman.

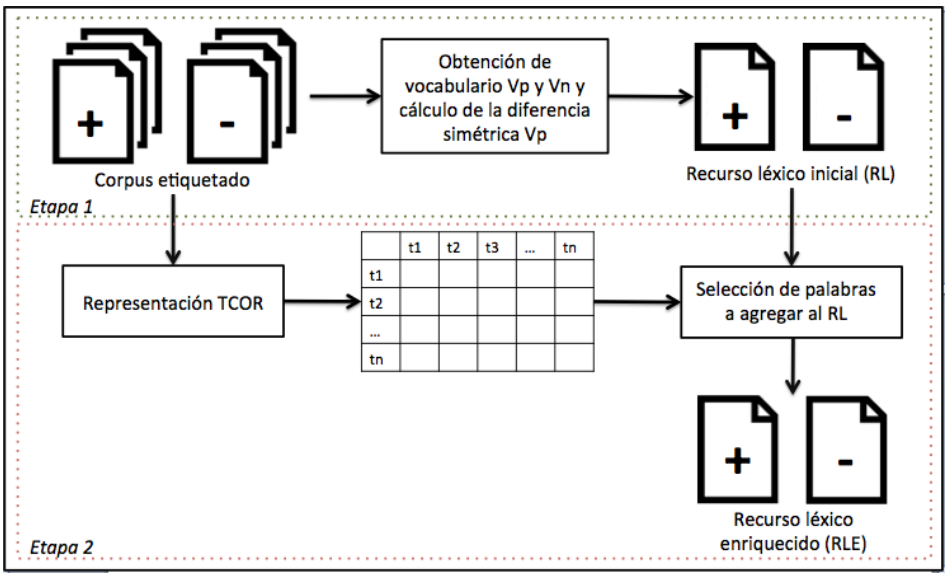


Fig. 1. Método propuesto para la generación automática de un recurso léxico a partir de un conjunto de datos etiquetados.

3.1. Etapa 1. Generación del recurso léxico inicial

La propuesta de este artículo es la generación automática de un recurso léxico a partir de un corpus. Para generar el recurso léxico inicial (RL) se realizaron los siguientes pasos:

1. Considerando que se cuenta con un conjunto de documentos etiquetados, se obtuvo el vocabulario de cada clase: positiva y negativa. Siendo V_p el vocabulario positivo y V_n el vocabulario negativo.
2. Se eliminan aquellas palabras que comparten ambos vocabularios, pues se considera que estas palabras no son discriminatorias de la polaridad, de modo que se conserva únicamente la diferencia simétrica de V_p y V_n ($V_p \triangle V_n$). Esto se define formalmente como: dados dos conjuntos A y B, su diferencia simétrica $A \triangle B$ es un conjunto que contiene los elementos de A y los de B excepto los que son comunes a ambos.
3. Las palabras en la diferencia simétrica será el recurso léxico inicial RL. Debido a que en esta propuesta no se consideró la obtención de orientación semántica de las palabras, como en el trabajo de Turney [14], o algún proceso que permita dar un valor numérico de que tan positiva o negativa es una palabra, se decidió asignar un peso igual a uno para las palabras positivas y un peso de menos uno para las palabras negativas.

3.2. Etapa 2. Enriquecimiento del léxico inicial mediante una representación distribucional

Como puede verse, la etapa inicial de la construcción del léxico sólo conserva las palabras que aparecen en los documentos de una única clase. Esta forma simple de obtener una primer RL puede ocasionar que los términos que ahí aparecen no sean suficientes para la tarea o que debido a que se eligen del conjunto previamente etiquetado, pueda generar sobre-ajuste en la etapa de la clasificación. Por ello, nuestro enfoque considera una segunda etapa donde el objetivo es desambigüar palabras que son comunes a ambas clases, pero que aún así tienen una carga hacia alguno de los polos. En otras palabras, la idea detrás de esta fase es enriquecer un RL inicial con palabras semánticamente relacionadas a las palabras positivas y negativas previamente encontradas.

Para generar un recurso léxico expandido (RLE), primero se necesita un RL inicial y un conjunto de documentos del dominio de análisis. El primer proceso, como puede verse en la Figura 1 usa el conjunto de documentos del dominio para que éstos sean representados mediante el modelo de espacio vectorial usando una representación TCOR.

La idea de esta representación distribucional es que la semántica de un término t_i puede revelarse a partir de otros términos con los que co-ocurre dentro de un documento de la colección [2,9]. De esta forma, cada término en el conjunto de términos únicos (vocabulario total) de la colección de documentos T, $t_i \in T$, es representado con un vector de pesos $t_i =< w_1, w_2, ..., w_n >$ donde w_j representa la contribución del término j a la descripción semántica de t_i como lo indica la siguiente fórmula:

$$w_k = tff(t_k, t_i) \cdot \log \frac{|T|}{T_k}, \tag{1}$$

donde T_k es el número de diferentes términos en el diccionario T que co-ocurren con t_i en al menos un documento y

$$tff(t_k, t_i) = \begin{cases} 1 + \log(\#(t_k, t_i)) & \text{if } (\#(t_k, t_i)) > 0, \\ 0 & \text{en otro caso,} \end{cases} \tag{2}$$

donde $(\#(t_k, t_i))$ denota el número de documentos en los que el término t_i co-ocurre con el término t_k. El vector de pesos luego es normalizado.

Luego, en el segundo proceso de la Etapa 2 se obtienen las palabras o términos t_k con mayor similitud semántica a cada término $t_j \in RL^{pos}$ y a $t_l \in RL^{neg}$. Para ello se crea una matriz de similitud de términos, dado que cada término en T está representado con un vector de pesos. De esta matriz de similitud se ordenan todos los términos de mayor a menor similitud con respecto a cada clase. Es decir, para la clase positiva se obtienen todos los t_k más similares a cada $t_j \in RL^{pos}$. En un paso posterior se conjuntan las listas de cada t_j, se ordenan y de esta lista general se obtiene el top-3 % de términos.

Al final se obtiene un conjunto enriquecido $RLE^{pos} = RL^{pos} \cup E^{pos}$ donde E^{pos} es el conjunto de términos que resultaron ser más similares a la clase

positiva. De forma equivalente se realiza el proceso para la clase negativa. Siendo el recurso léxico extendido representado por $RLE = RLE^{pos} \cup RLE^{neg}$.

4. Métodos de clasificación de sentimientos

El objetivo de este apartado es presentar los métodos de clasificación que se usarán en los experimentos. Los enfoques que abordaremos para evaluar los recursos léxicos generados son: clasificación basada en recursos léxicos y clasificación basada en aprendizaje supervisado.

4.1. Enfoque de clasificación basada en recursos léxicos

La clasificación basada en recursos léxicos parte de la premisa que existe un conjunto de términos asociados a una clase, es decir, diccionarios de pares $< termino, v >$ donde si $v > 0$ el término tienen carga positiva, y si $v < 0$ tiene carga negativa. Por lo tanto cuando se quiere clasificar un nuevo documento primero se determina la polaridad de ese documento de acuerdo con la siguiente fórmula:

$$pol = \sum_{w \in T \wedge w \in L} valor(w, L), \tag{3}$$

donde T es el conjunto de palabras en el documento a analizar, L es el conjunto de palabras en el recurso léxico usado y $valor(w, L)$ es el valor numérico asociado a w en el recurso léxico L. Luego la clase es asignada de acuerdo a la expresión 4.

$$clase = \begin{cases} positiva & \text{if } (pol > 0), \\ negativa & \text{en otro caso.} \end{cases} \tag{4}$$

4.2. Enfoques de clasificación híbrida

Los métodos de clasificación híbridos combinan conocimiento previo mediante el uso de los recursos léxicos y algoritmos de aprendizaje para generar un clasificador capaz de determinar la clase de un nuevo documento.

En este sentido, una forma de aprovechar la información del recurso léxico es a través de la representación de los documentos. Es decir, se usan como atributos solo aquellos términos que aparecen en el recurso léxico (L). De tal manera que se genera una vector por cada documento en la colección de la forma: $d_i =< w_1, w_2, ..., w_m >$, donde w_1 determina la relevancia del termino j en el documento i, y $m = |L|$ para L el conjunto de términos en el recurso léxico usado.

El pesado que se utilizó para cada vector fue TF-IDF, el cual se define como:

$$w_{ki} = TF(t_k) \times IDF(t_k), \tag{5}$$

donde $TF(t_k)$ es la frecuencia del término t_k en el documento d_i. IDF es la frecuencia inversa del término t_k dentro del documento d_i y es calculado a partir de la siguiente ecuación:

$$IDF(t_k) = \log \frac{|D|}{|\{d_i \in D : t_k \in d_i\}|}, \tag{6}$$

donde D representa la colección de documentos que está siendo indexada.

5. Evaluación experimental

5.1. Colecciones de datos

Para la evaluación del método propuesto se usaron dos colecciones de datos de un solo dominio: revisiones de películas. Con el objetivo de comprobar la independencia del idioma en el enfoque propuesto una de las colección usadas está en español mientras que la otra en inglés. En la Tabla 1 se muestran algunas estadísticas del corpora usado.

Tabla 1. Estadísticas de corpus empleados.

	Español	Inglés
Num. documentos	2,622	2,000
Vocabulario total	29,160	24,992
Promedio de palabras por docs.	255.6 ±15.7	351.4 ±15.2
Similitud promedio entre docs.	0.05 ±0.02	0.05 ±0.02

El corpus en español consiste en críticas de películas extraídas del sitio web *www.muchocine.net* [6], tiene originalmente un total de 3,878 críticas y aproximadamente 2 millones de palabras, con un promedio de 546 palabras por crítica. Cada crítica tiene un atributo de calificación de la película en la escala del 1 al 5. En nuestro procesamiento del corpus se conserva el mismo número de críticas pero el total de palabras descendió a 670,225 igual que el promedio de palabras por crítica que descendió a 255 palabras en promedio, como se puede ver en la Tabla 1. Otro aspecto a notar es que es un corpus relativamente balanceado, en la clase de los positivos tiene 1,349 documentos y en la negativa 1,273. Finalmente el vocabulario global consta de 29,160 palabras.

En relación al corpus en inglés [10] se caracteriza por estar totalmente balanceado: 1000 documentos por la clase negativa y 1000 de la positiva. Después del pre-procesamiento del corpus se cuenta con 702,836 palabras en total y un promedio de 351 palabras por documento. Este corpus es uno ampliamente usado para validar métodos de análisis de sentimientos.

5.2. Pre-procesamiento del corpora

Antes de comenzar los experimentos se realizaron algunos procesos que nos permitieron estandarizar los documentos y la organización de los mismos, los cuales se describen a continuación. Primero se organizaron los documentos de acuerdo a las dos clases para cada corpus: críticas positivas y críticas negativas. Para esta tarea fue requerido dividir el corpus en español en las dos clases mencionadas, de acuerdo a la puntuación que contenía cada crítica (valores de 1 a 5). Se decidió que los valores de 1 y 2 se tomaran como una crítica negativa, el valor de 3 como neutra, y los valores de 4 y 5 como positiva. Debido a que el corpus en inglés no contaba con críticas neutras, aquellas críticas en español que fueron evaluadas con una puntuación igual a 3, fueron excluidas de nuestro corpus.

Finalmente, el pre-procesamiento involucró las siguientes tareas:

- El texto fue convertido a minúsculas, para evitar repetición de términos, tal como el caso de *Excelente* y *excelente*.
- Se remplazaron las secuencias de espacios en blanco por un solo espacio en blanco.
- Se eliminaron signos de puntuación, tales como signos de admiración e interrogación, comas, puntos, comillas, paréntesis, etc.
- Se eliminaros los números.
- Se eliminaron palabras vacías por medio de una lista de palabras en español y una en inglés.

5.3. Diseño experimental

Para validar la pertinencia del recurso léxico generado automáticamente a partir de un conjunto de datos de un dominio específico se diseñaron 2 experimentos con el corpora previamente descrito. Todos los experimentos se realizaron con tres particiones distintas de 70 %-30 %. Se utilizaron 70 % de los documentos para entrenar el modelo de clasificación o para generar el recurso léxico, el 30 % restante fue usado para la evaluación. Cabe notar que cada partición contiene conjuntos de prueba disjuntos.

- Experimento 1: Clasificación de opiniones basada en recursos léxicos. Este experimento consiste en utilizar el recurso léxico (RL) generado en la etapa 1 del método propuesto en un enfoque de *clasificación basado en léxico* (ver Sección 4.1). La idea de este experimento es comparar el RL generado con un recurso léxico genérico, particularmente con ANEW. Además, se evaluará que el recurso léxico extendido RLE generado en la etapa 2 del método propuesto mejora la clasificación de cuando únicamente se utiliza el RL inicial. La idea de esta variación es determinar si al enriquecer el RL inicial usando información semántica mediante la representación TCOR, es posible mejorar la clasificación de opiniones.

Tabla 2. Número de palabras en los recursos léxicos utilizados, divididos por clases: positiva y negativa.

Recurso léxico	*Español*		*Inglés*	
	Positivas	*Negativas*	*Positivas*	*Negativas*
ANEW	511	523	581	449
ANEW-E	7052 ±199	8747 ±56	7140 ±95	7799 ±110
RL	16333 ±437	12115 ±117	9887 ±88	8188 ±125
RLE	19674 ±559	15922 ±291	12115 ±135	10839 ±152

- Experimento 2: Clasificación de opiniones basado en un enfoque híbrido. Este conjunto de experimentos consisten en utilizar un recurso léxico para representar los documentos y mediante un enfoque de clasificación supervisada, generar un clasificador que sea capaz de distinguir opiniones positiva de las negativas.

Experimento 1: Clasificación de opiniones basada en recursos léxicos. Como se mencionó antes, el objetivo de este conjunto de experimentos es evaluar, por un lado, que tanto mejora la clasificación de sentimientos usando un recurso léxico hecho a la medida en contraste con un recurso léxico genérico (por ejemplo, ANEW). Por otro lado, se desea evaluar en qué medida es posible mejorar la clasificación si se enriquece un recurso léxico existente, ya sea genérico o a la medida. El enriquecimiento de los dos recursos léxicos se realizó como se describe en la Sección 3 donde el porcentaje de términos agregados más similares al léxico inicial es de 3 %.

En la Tabla 2 se muestran algunas estadísticas tanto del recurso ANEW para español e inglés como de los generados por el método propuesto en ambos idiomas. Como puede verse en la tabla, el número de palabras del recurso generado RL es mucho mayor que las palabras contenidas en ANEW para ambos idiomas, esto sugiere que para revisiones de películas existen muchos más términos asociados a las clases positiva y negativa que un recurso genérico con ANEW.

Cabe mencionar que al extender RL en RLE, el incremento de términos por clase es de entre el 20 % y 30 %, un número relativamente pequeño si comparamos el incremento de los términos de ANEW en ANEW-E (extendido), pues incrementa entre 12 y 16 veces el número de términos.

En el caso del recurso ANEW, por cada término se tiene asociado un valor de felicidad entre 0 y 10. Para este experimento se considera que un puntaje de felicidad entre 6 y 10 es considerado positivo, mientras que los términos asociados a un puntaje de felicidad de 0 a 5 son considerados negativos. Por lo tanto, en los pares $< termino, valor >$ para este recurso, $valor$ es un número entero y $0 \leq valor \leq 10$. Para el caso de RL, $valor = 1$ o $valor = -1$. La asignación de clase en estos métodos es dada como se indicó en las Ecuaciones 3 y 4 descritas en la Sección 4.1.

Tabla 3. Resultados de clasificar opiniones usando recursos léxicos.

Recurso léxico	*Español*			*Inglés*		
	Precisión	*Recuerdo*	*F-score*	*Precisión*	*Recuerdo*	*F-score*
ANEW	0.544	0.521	0.456	0.475	0.497	0.352
ANEW-E	0.629	0.583	**0.523**	0.640	0.634	**0.629**
RL	0.616	0.590	0.571	0.625	0.621	0.618
RLE	0.654	0.612	**0.587**	0.662	0.659	**0.658**

La Tabla 3 muestra los resultados de clasificación de usar cada uno de los 4 recursos. En esta tabla es posible observar que al usar un recurso léxico a medida del conjunto de documentos a clasificar, el resultado de clasificación mejora. Particularmente, para el caso del corpus en español la mejora del valor de F-score va de 0.45 a 0.57; para el caso del corpus en inglés la mejora es aún mayor, de 0.35 a 0.61 (comparando los renglones correspondientes a los recursos ANEW y RL).

Por otro lado, el recurso léxico extendido también mejora los resultados de clasificación al compararlo con el recurso léxico inicial. Es decir, si se usa ANEW como recurso léxico inicial el valor de F-score mejora de 0.45 a 0.52 y de 0.35 a 0.62, para español e inglés respectivamente. Esta mejora se conserva también cuando el recurso léxico inicial es RL, aumentado el valor del F-score de 0.57 a 0.58 y de 0.61 a 0.65, para español e inglés respectivamente.

De estos resultados podemos inferir, por un lado, que RL al ser hecho a la medida, contiene términos relevantes para este dominio en particular y que al extenderlo ya no se aporta información relevante para mejorar en mayor medida la clasificación. Por otro lado, al ser ANEW un recurso léxico genérico, el desempeño en este dominio es pobre, pero al enriquecerlo con información del dominio la clasificación mejora, esto se puede deber a que el número de términos que se agregaron al recurso tienen relación directa con el dominio en análisis.

De manera global se puede decir que utilizar un recurso léxico a medida extendido (es decir, RLE) es mejor que usar un recurso genérico. Pero que si se desea usar un recurso genérico como ANEW, es mejor utilizar su versión enriquecida (es decir, ANEW-E). Estos resultados dan pauta para suponer que el enfoque propuesto puede funcionar ya sea para otros recursos léxico iniciales diferentes a ANEW como para otros dominios. Considerando que ANEW es un recurso léxico creado por expertos lingüístas, consideramos que nuestra propuesta es relevante para la clasificación de sentimientos, pues en base a intuiciones básicas se consigue mejores resultados a los que obtiene el recurso léxico ANEW.

Experimento 2: Clasificación de opiniones basado en un enfoque híbrido. Este conjunto de experimentos tiene como objetivo evaluar, en un enfoque de clasificación de opiniones híbrido, el desempeño de la clasificación al usar como vocabulario para la representación de los documentos el conjunto de términos

Tabla 4. Resultados de clasificar opiniones usando un enfoque hibrido.

Recurso léxico + SVM	*Español*			*Inglés*		
	Precisión	*Recuerdo*	*F-score*	*Precisión*	*Recuerdo*	*F-score*
ANEW	0.635	0.645	0.635	0.662	0.661	0.661
ANEW-E	0.679	0.679	**0.678**	0.693	0.692	**0.692**
RL	0.671	0.594	0.538	0.636	0.618	0.587
RLE	0.628	0.613	**0.600**	0.634	0.587	**0.604**

de un recurso léxico. Nuevamente, se utilizaron los recursos léxicos descritos en la Tabla 2, y como algoritmo de aprendizaje se usó SVM con los parámetros por defecto de la implementación de la herramienta Weka [5]. El resto de los detalles fueron descritos previamente en la Seccion 4.2.

En la Tabla 4 se muestran los resultados de clasificación para precisión, recuerdo y F-score. De esos resultados se puede decir que también en un enfoque híbrido, los recursos léxicos enriquecidos (ANEW-E y RLE) son mejores para representar a los documentos que los recursos iniciales (ANEW, RL).

En la Figura 2 se observa un resumen de los resultados obtenidos en ambos experimentos en términos del F-score. En esta figura es posible apreciar mejor que el desempeño de nuestra propuesta funciona mejor para enfoques basados en diccionarios que en enfoques híbridos. De forma que si se cuenta con medios para generar conjuntos de datos etiquetados es mejor usar ese enfoque que uno basado en diccionarios. Sin embargo, si no se cuenta con suficiente documentos etiquetados para generar un clasificador confiable, nuestros resultados sugieren que es mejor generar un recurso léxico a la medida y luego extenderlo, esto bajo un esquema de clasificación basado en recursos léxicos.

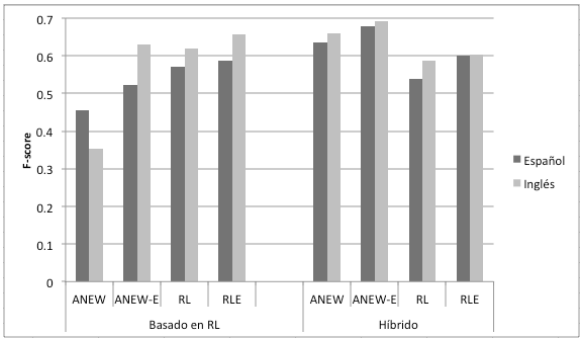


Fig. 2. Valores de F-score para la clasificación de críticas de películas usando dos enfoques: clasificación basada en recursos léxicos (Basada en RL), e híbrida.

6. Conclusiones y trabajo a futuro

En este artículo se presentó un esquema para la generación automática de recursos léxicos con polaridad positiva y negativa para el análisis de sentimientos. La ventaja de este esquema es que se generan diccionarios a medida del dominio de los textos analizados. Por otro lado, el esquema propuesto plantea un método de enriquecimiento de un léxico dado, usando un conjunto de documentos que no necesitan estar etiquetados. Nuestro método de enriquecimiento hace uso de una representación distribucional, i.e., TCOR para encontrar términos semánticamente relacionados a aquellos en el diccionario inicial.

Los resultados obtenidos de evaluar los recursos léxicos generados muestran que es mejor hacer un recurso a medida que usar un recurso genérico. También muestra que si ya se tiene un recurso genérico, al enriquecerlo con nuestro método, los resultados pueden mejorar. Adicionalmente, al usar los recursos generados a medida en un enfoque de clasificación híbrida, es decir representar los documentos con los términos en el recurso léxico usado, no mejora el desempeño de clasificación cuando los documentos se representan con un recurso léxico genérico.

Se requieren más experimentos para concluir de manera contundente la importancia y utilidad de generar recursos léxico a bajo costo. Una de las pruebas que consideramos realizar a corto plazo es el uso de otros recursos genéricos como SentiWordNet. También se planea probar la creación del recurso léxico con un corpus de dominio diferente como revisiones de libros, hoteles u otro tipo de entidades. Como trabajo futuro adicional se contempla utilizar la orientación semántica de los términos incluidos en el recurso léxico, es decir, incluir valores que nos indiquen que tan positiva o negativa es una palabra; creemos que incorporar esta característica puede mejorar el esquema propuesto.

Agradecimientos. El trabajo de los primeros tres autores fue realizado con el apoyo de CONACyT (número de becas 458842, 458835 y 616127). Agradecemos también el apoyo otorgado a través de la Coordinación de la Maestría en Diseño, Información y Comunicación (MADIC) de la UAM Cuajimalpa, así como al CONACyT-SNI.

Referencias

1. Bradley, M.M., Lang, P.J.: Affective norms for English words (ANEW): Stimuli, instruction manual, and affective ratings. Tech. rep., Center for Research in Psychophysiology, University of Florida, Gainesville, Florida (1999)
2. Cabrera, J.M., Escalante, H.J., Montes-y Gómez, M.: Distributional Term Representations for Short-Text Categorization, pp. 335–346. Springer Berlin Heidelberg, Berlin, Heidelberg (2013), http://dx.doi.org/10.1007/978-3-642-37256-8_28
3. Cruz, F.L., Troyano, J.A., Enriquez, F., Ortega, J.: Clasificación de documentos basada en la opinión: experimentos con un corpus de críticas de cine en español. Procesamiento del Lenguaje Natural 41(0) (2008), http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/2551

4. Esuli, A., Sebastiani, F.: Sentiwordnet: A publicly available lexical resource for opinion mining. In: In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC'06. pp. 417–422 (2006)
5. García, A., Gaines, S., Linaza, M.T.: A lexicon based sentiment analysis retrieval system for tourism domain. e-Review of Tourism Research (eRTR) pp. 35–38 (May 2012)
6. Garner, S.R.: Weka: The waikato environment for knowledge analysis. In: In Proc. of the New Zealand Computer Science Research Students Conference. pp. 57–64 (1995)
7. Kaji, N., Kitsuregawa, M.: Building lexicon for sentiment analysis from massive collection of html documents. In: EMNLP-CoNLL. pp. 1075–1083 (2007)
8. Khan, F.H., Qamar, U., Bashir, S.: Senti-cs: Building a lexical resource for sentiment analysis using subjective feature selection and normalized chi-square-based feature weight generation. Expert Systems 33(5), 489–500 (2016), http://dx.doi.org/10.1111/exsy.12161, eXSY-Nov-14-253.R1
9. Lavelli, A., Sebastiani, F., Zanoli, R.: Distributional term representations: An experimental comparison. In: Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management. pp. 615–624. CIKM '04, ACM, New York, NY, USA (2004), http://doi.acm.org/10.1145/1031171.1031284
10. Liu, B.: Sentiment analysis and opinion mining. Synthesis Lectures on Human Language Technologies 5(1), 1–167 (2012), http://dx.doi.org/10.2200/S00416ED1V01Y201204HLT016
11. Medhat, W., Hassan, A., Korashy, H.: Sentiment analysis algorithms and applications: A survey. Ain Shams Engineering Journal 5(4), 1093 – 1113 (2014), http://www.sciencedirect.com/science/article/pii/S2090447914000550
12. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: Sentiment classification using machine learning techniques. In: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10. pp. 79–86. EMNLP '02, Association for Computational Linguistics, Stroudsburg, PA, USA (2002), https://doi.org/10.3115/1118693.1118704
13. Redondo, J., Fraga, I., Padrón, I., Comesaña, M.: The spanish adaptation of anew (affective norms for english words). Behavior Research Methods 39(3), 600–605 (2007), http://dx.doi.org/10.3758/BF03193031
14. Turney, P.D.: Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. pp. 417–424. ACL '02, Association for Computational Linguistics, Stroudsburg, PA, USA (2002), http://dx.doi.org/10.3115/1073083.1073153

ISSN 1870-4069

Minería de textos para el análisis del subregistro de lesiones por causa externa en el servicio de urgencias de un hospital de tercer nivel

José Ramón Consuelo Estrada[1,2], Otniel Portillo Rodríguez[2], Laura Soraya Gaona Valle[1]

[1] Instituto de Salud del Estado de México, Centro Médico "Lic. Adolfo López Mateos", México

[2] Universidad Autónoma del Estado de México, Facultad de Ingeniería, Ciudad de México, México

jramon208@gmail.com, oportillor@uaemex.mx, gaonav_81@yahoo.com.mx

Resumen. Las Lesiones por Causa Externa (LCE) representan un serio problema de salud, sin embargo el diagnóstico mediante códigos inespecíficos puede subestimar su gravedad. Esto es común en muchos hospitales, afectando la asignación de recursos y prioridades en salud. Analizando datos de un servicio de urgencias de un hospital de tercer nivel, se implementaron cuatro modelos predictivos: regresión logística con texto (TF-IDF), regresión logística, árboles de decisión y boosting para determinar el porcentaje de diagnósticos de *dolor agudo* que pudieran ser LCE. El método más exacto fue el basado en texto y estima que, 12240 (82.56 % n=14826) de los *dolores agudos* son LCE. El porcentaje de LCE subestimado como resultado del uso códigos inespecíficos es alto y la minería de textos es una opción viable para su estimación.

Palabras clave: Minería de textos, machine learning, código inespecífico, lesiones por causa externa.

Text Mining to Analyze Subrecord of External Causes of Injuries in the Emergency Department of a Third Level Hospital

Abstract. External Cause of Injuries (ECI) represent a serious health problem, however diagnosis using nonspecific codes may underestimate their severity. This situation is common in many hospitals, affecting the resources allocation and health priorities. Analyzing data from the emergency department of a hospital, four predictive models were implemented: Text-Logistic regression (TF-IDF), logistic regression, decision

trees and boosting to determine the percentage of *acute pain* diagnoses that could be classified as ECI. The text-based method shows better accuracy and estimates that 12240 (82.56% n = 14826) of acute pain are ECI. The percentage of LCE underestimated by using nonspecific codes is high and text mining is a viable option for their estimation.

Keywords: Text mining, machine learning, non-specific code, external cause of injuries.

1. Introducción

Las Lesiones por Causa Externa (LCE) contribuyen aproximadamente con el 10 % de la mortalidad mundial y con el 12 % de la morbilidad [1]. Estas lesiones engloban situaciones entre las que podemos encontrar los accidentes de tránsito, las caídas, las agresiones interpersonales dentro y fuera del hogar o golpes de cualquier tipo. Cada una de ellas puede representar problemas serios a nivel mundial para los sistemas de salud y para la ciudadanía en general [2].

El Catálogo Internacional de Enfermedades (CIE-10) se ha desarrollado como herramienta para la clasificación de diagnósticos y de las causas que los generan [3]. Ciertos diagnósticos dentro del catálogo están asociados a lo que se denomina *Causa Externa* la cual se refiere a un evento que generó el padecimiento y que requiere el registro de los datos relacionados (lugar de ocurrencia, atención prehospitalaria, agente de la lesión, datos del agresor, etc). En la Figura 1 se muestra el proceso general de registro en los servicios de urgencias, donde se puede apreciar que, cuando el diagnóstico está relacionado con una LCE, entonces se requiere realizar el registro de los datos de la lesión.

Dentro del propio CIE-10 se ha identificado un grupo de diagnósticos que no proporcionan la información suficiente para su posterior análisis estadístico o epidemiológico. A estos diagnósticos se les ha denominado *inespecíficos* o, incluso *códigos basura* [4]. Esta situación también puede presentarse debido al diseño de los Sistemas de Información (SI) cuando los usuarios tienen la posibilidad de elegir opciones tales como *Otros*, *se desconoce*, *no disponible*, etc. dentro de una lista desplegable y no se solicita la aclaración pertinente; o cuando se permite que se guarde información con datos faltantes.

El problema de códigos inespecíficos se ha estudiado desde distintos puntos de vista, Pérez-Nuñez et al. [4] proponen el uso de modelos de imputación simple [5] para identificar muertes que pudieran ser atribuidas al tránsito. Sus resultados muestran un subregistro del 18.85 % y presentan una lista de diagnósticos CIE-10 que pueden ser considerados como inespecíficos. Bhalla et al. [1] realizan una evaluación de la calidad y disponibilidad de los datos sobre muertes causadas por lesiones, donde un registro menor al 20 % de causas inespecíficas se considera bueno. Se encontró que solo 20 de los 83 países analizados presentan datos de calidad. En el trabajo presentado por Híjar et al. [6] se presentan tres modelos basados en el método proporcional, imputación múltiple y regresiones en un

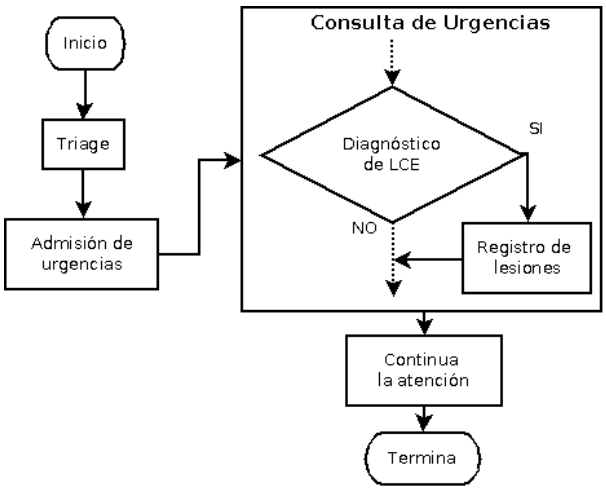


Fig. 1. Registro de lesiones en el servicio de urgencias.

intento por corregir la subestimación de muertes por tránsito debida a los códigos inespecíficos. Los autores reportan un incremento del 18 al 45 % dejando clara la magnitud del problema.

Los enfoques tradicionales para abordar el tema de la inespecificidad de los datos se basan en análisis de un conjunto de datos formado por variables cualitativas y/o cuantitativas (aquí: datos estructurados). Sin embargo, gran parte de la información hospitalaria disponible se encuentra en texto proveniente de hojas escritas por los médicos, normalmente en formato libre, ya sea manual o electrónico.

En este trabajo se propone el análisis de datos en formato de texto para abordar el problema de códigos inespecíficos. Diversas investigaciones han demostrado que el análisis de texto puede ayudar a la solución de problemas de administración hospitalaria, Lucini et al. [7] utilizaron distintas metodologías de minería de textos para entrenar modelos que permitan predecir futuras hospitalizaciones y costos de atención. Los modelos utilizados incluyen: árboles de decisión, regresión logística y máquinas de soporte vectorial, entre otras. Los autores concluyen que la minería de textos puede brindar información valiosa para la toma de decisiones.

Por otro lado, en el campo de la farmacovigilancia Abacha et al. [8] proponen el uso de minería de texto para analizar artículos científicos y registros médicos que ayuden a comprender y dar soporte al tema del suministro de medicamentos.

Kocbek et al. [9] presentan un sistema de minería de textos que utiliza información de radiología, patología y datos del paciente y de su ingreso para detectar distintos tipos de cáncer tales como el de pulmón, pecho y colon, entre otros. Concluyen que la combinación de distintas fuentes de información y el uso de minería de textos pueden mejorar las predicciones de estos padecimientos.

En nuestro caso de estudio, los datos disponibles muestran que las LCE tienen una tendencia a la baja mientras que los registros del diagnóstico inespecífico *dolor agudo* (R52.0) aumentan en proporciones similares (ver Figura 2), si bien los datos del Instituto Nacional de Estadística y Geografía (INEGI) muestran una tendencia a la baja en accidentes de tránsito [10], otros informes reportan aumentos en casos de: lesiones no intencionales, agresiones interpersonales caídas o accidentes laborales [2], [11]. Incluso la Secretaría de Salud reporta incremento en egresos hospitalarios por lesiones causadas por el tránsito, en copilotos y motociclistas, con un aumento del 62.90 % de 2010 a 2014 [12].

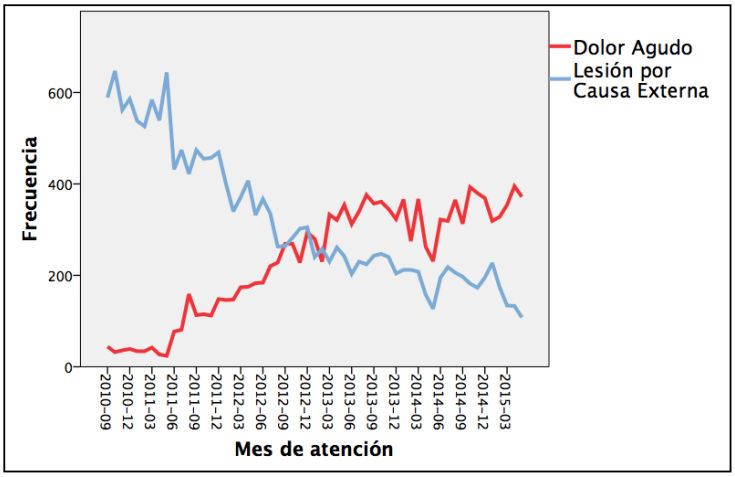


Fig. 2. Tendencia de los registros en el periodo analizado.

Este trabajo pretende estimar el porcentaje de registros con diagnóstico de dolor agudo pudieran ser LCE y hacer una comparación entre los métodos estructurados contra los basados en el análisis de textos.

2. Material y métodos

Previa aprobación ética, se extrajeron los registros electrónicos del servicio de urgencias de un hospital de tercer nivel ubicado en la ciudad de Toluca, Estado de México. La información disponible corresponde al periodo comprendido entre el 1 septiembre de 2010 al 31 de mayo de 2015 y proviene de las áreas de Triage, Admisión y Consulta, todas del servicio de urgencias y del área de Expediente Clínico que es común a todo el nosocomio.

El registro de los diagnósticos asignados a los pacientes se realizó en base al CIE-10. Utilizando el diagnóstico se definió una etiqueta con la cual se determina si el diagnóstico es Dolor Agudo (DA), Lesión por Causa Externa (LCE) o cualquier otro diagnóstico (OTRO), la Tabla 1 muestra la distribución resultante.

Tabla 1. Distribución de los datos según la etiqueta asignada.

Etiqueta	n	%
Lesión por Causa Externa (LCE)	19230	16.13 %
Dolor Agudo (DA)	15859	13.30 %
Otros diagnósticos (OTRO)	84131	70.57 %
Total	119220	100.00 %

El conjunto de datos resultante cuenta con variables estructuradas tales como: edad y sexo del paciente, escolaridad, si presenta discapacidad (ya sea mental o física), fecha de la atención, peso, talla, frecuencia cardiaca, presión arterial (sistólica y diastólica), frecuencia respiratoria, área a la que se envía al paciente después de la atención, diagnóstico (CIE-10), procedimiento realizado, medicamentos prescritos, área de atención (Consulta de urgencias, Consulta de Traumatológia de urgencia, Observación o Choque), antecedentes de Diabetes Mellitus, antecedente de alcoholismo, antecedente de tabaquismo, índice de Glasgow y forma de arribo al hospital, entre otras.

Así mismo, cada registro cuenta con variables en texto libre relativas a campos como: motivo de la consulta, antecedentes, padecimiento actual, exploración física, tratamiento y plan, observaciones, estudios realizados y pronóstico. Para el análisis propuesto, estas variables fueron concatenadas y utilizadas como un solo campo de texto.

Tabla 2. Resumen de modelos generados.

Modelo	Variables	Método	Entrenamiento	Validación	Prueba	Clasificación
M1	EST	RL	28648	3666	3526	13297
M2	EST	DT	28648	3666	3526	13297
M3	EST	Boosting	28648	3666	3526	13297
M4	TFIDF	RL	30498	3880	3738	14826

EST.- Datos estructurados; RL.- Regresión Logística, DT.- árbol de decisión; TFIDF.- Frecuencia inversa de palabras

Realizando una separación de variables se crearon dos conjuntos uno para el análisis estructurado y otro para el análisis de texto. El primer modelo M1 se basa en datos estructurados utilizando Regresión logística con regularización Ridge Regression (RR) conforme a la ecuación 1 [13], así como árboles de decisión y método de boosting, con restricciones de profundidad e iteración (número de clasificadores simples), respectivamente. Para el último modelo se realizó un análisis de texto utilizando el método conocido como Frecuencia de Término - Frecuencia de Documento Inverso (TF-IDF, por sus siglas en Inglés) [14]. Para la

elección de parámetros, cada modelo fue sometido a un proceso de validación. La Tabla 2 resume los modelos generados donde la columna *Clasificación* se refiere a la cantidad de registros de *dolor agudo* que son clasificados en la parte final del trabajo:

$$\beta_0^*, \beta^* = argmin_{\beta_0, \beta b} \left(\frac{1}{N} \sum_{k=1}^{N} (y_i - (\beta_0 + x_i \beta))^2 + \lambda \beta^T \beta \right), \tag{1}$$

donde la expresión *argmin* significa "los valores β_0 y β que minimizan la expresión". β_0 es el término *intersección* y β representan el vector de coeficientes. y_i es el vector de etiquetas conocidas. La expresión $(\beta_0 + x_i \beta)$ es el valor calculado por el modelo y λ es el parámetro de regularización. La diferencia $y_i - (\beta_0 + x_i \beta)$ es el error entre el término esperado y el término calculado. Ajustando el valor de λ, la ecuación 1 busca encontrar los parámetros β_0 y β que minimicen el error de predicción [13].

Ambos conjuntos de datos requirieron preprocesamiento. En el conjunto de datos estructurados se eliminan registros duplicados o sesgados. Los datos nulos fueron tratados con imputación por promedio para las variables numéricas y a las variables categóricas se les agregó la categoría *dato no registrado*. Las variables numéricas se estandarizaron en base a la ecuación 2 mientras que las variables categóricas fueron dicotomizadas:

$$S_i' = \frac{S_i - \mu}{\sigma}, \tag{2}$$

donde S' es el nuevo valor estandarizado para el *i-ésimo* registro, S_i es el valor actual, μ es la media del conjunto de datos y σ su desviación estándar. Con esta ecuación se busca homogeneizar los valores para evitar que ciertas variables tengan más peso que otras como resultado de las diferencias en sus escalas de medición.

Por otro lado en el conjunto con datos de texto, para eliminar palabras mal escritas se formó un listado tomando como base notas médicas, un diccionario de la lengua española y un glosario de términos médicos [15]. También se eliminaron palabras poco relevantes para el análisis (que, por, de, el, etc.). Todas las palabras se consideraron en mayúsculas y sin acentos. Para este análisis se utilizó el método TF-IDF que se basa en la ecuación 3:

$$F_i(p) = \log \left(\frac{N}{1+n} \right), \tag{3}$$

donde la frecuencia inversa de la palabra p es el logaritmo del total de registros N entre *1* mas el número de registros n que contienen la palabra p [16] [17]. Con esto se forma un conjunto estructurado con el que posteriormente se entrenó, validó y probó un modelo de regresión logística.

Los registros con diagnóstico de *dolor agudo* fueron separados para ser clasificados al final; constan de 13297 registros, para datos estructurados y 14826 para texto. El entrenamiento, validación y prueba se realizó con los datos etiquetados como LCE y una parte proporcional de los etiquetados como OTROS,

ambos conjuntos fueron balanceados utilizando el método *submuestreo* [18]. El total de registros fue de 35580 y 38116 para datos estructurados y de texto respectivamente. Finalmente, los conjuntos de datos resultantes se dividieron para entrenamiento (80 %), validación (10 %) y prueba (10 %) de los modelos. (ver Tabla 2).

Para cada modelo se realizó un ajuste de parámetros que varía según el clasificador: a los modelos M1 y M4, basados en regresión logística, se les aplicó regularización RR para evitar el sobre ajuste de los coeficientes. Los modelos M2 y M3 variaron en profundidad de árbol y número de árboles, respectivamente. Para estos ajustes se utilizó el conjunto de datos de entrenamiento y de validación [13]. El método de *boosting*, del modelo M3 esta basado en una colección de clasificadores básicos combinados a través de una técnica conocida como gradient boosting [19].

Utilizando los parámetros encontrados en la fase de validación, se clasifican los datos de prueba para realizar la evaluación de los modelos. Finalmente, se realiza la clasificación de los datos de dolor agudo, separado previamente.

Para la programación, tratamiento de datos y entrenamiento de los modelos se utilizó Python 2.5 y GraphLab Create[19].

3. Resultados

En esta sección se muestran los resultados de los análisis realizados, se describe cada modelo en el orden presentado en la Tabla 2. El rendimiento de todos los modelos se muestra en la Tabla 5 y fue calculado utilizando el conjunto de datos de prueba.

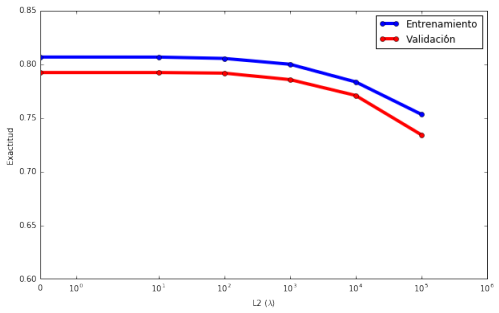


Fig. 3. Exactitud del modelo M1 con diferentes valores de regularización λ.

El modelo M1 se basa en el conjunto de datos estructurados y se realiza una regresión logística que incluye el uso de la ecuación 1 para la validación mediante una regularización RR. La gráfica de la Figura 3 muestra la exactitud para cada valor de λ tanto de los valores de entrenamiento como de validación. En este

proceso se busca que el valor de λ maximice la exactitud del conjunto de datos de validación.

Tabla 3. Variables con coeficientes más representativos del modelo M1, $\lambda = 100$.

Variable	Coeficiente	Descripción
proc.3809	10.0069	Incisión de venas, miembros inferiores
proc.8149	8.3752	Otra reparación de tobillo
med.02608	8.3734	Carbamazepina
med.00569	7.3166	Nitroprusiato de sodio
med.05506	7.2910	Celecoxib
med.03253	-6.2843	Haloperidol
proc.9393	-6.0935	Métodos de resucitación no mecánicos
proc.9908	-5.9974	Transfusión de expansor sanguíneo
med.03631	-5.9859	Solución de glucosa
med.01901	-5.7695	Sulfadiazina

proc.- Procedimiento médico, med.- Medicamento.

En este caso hay poca variabilidad en la exactitud del modelo (ver Figura 3). El valor de λ que minimiza el error y que por lo tanto aumenta la exactitud es 100 (10^2); este valor fue utilizado para entrenar la versión final del modelo M1. Los coeficientes del modelo resultante con pesos mas representativos (positivos y negativos) son los mostrados en la Tabla 3.

En la Tabla 3 los prefijos *proc* y *med*, hacen referencia al procedimiento aplicado al paciente y a los medicamentos prescritos, ambos valores basados en sus respectivos catálogos. Debido a que el nombre de la variable es largo y para mejorar la lectura, se han acotado al prefijo *proc* y *med* más la clave correspondiente. El comportamiento de estos coeficientes a diferentes valores λ se muestra en la Figura 4.

El modelo M2 también utiliza datos estructurados para entrenar un árbol de decisión. En este caso el parámetro a considerar es la profundidad del árbol. En la Figura 5 se puede apreciar que el mejor valor de profundidad es 10. Con este valor se realiza el entrenamiento final del modelo y posteriormente se realiza la prueba.

El modelo M3 utiliza un conjunto de árboles (boosting). En este modelo el parámetro a encontrar es la cantidad de árboles o *iteración* que conformarán el modelo. En la Figura 6 se muestra sus respectivos valores de exactitud a diferente número de árboles. Aunque se puede apreciar que al aumentar indefinidamente el número de árboles la exactitud en el conjunto de entrenamiento se hace casi perfecta, también se aprecia que la exactitud en la validación disminuye a partir del valor 50, por lo que se ocupa este como parámetro del modelo.

Finalmente, para el modelo M4 una vez realizada la representación del texto con el método de TF-IDF, se calcula la inversa mediante la ecuación 3. Posteriormente se entrena un modelo de regresión logística donde el proceso es similar al

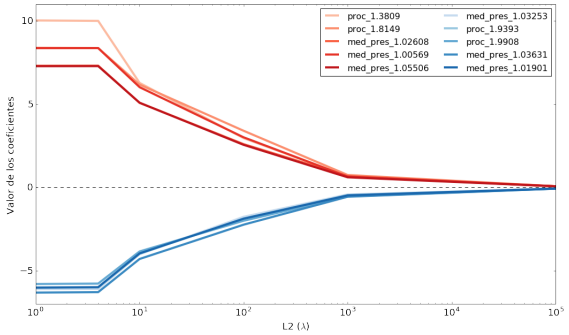


Fig. 4. Comportamiento de los coeficientes del modelo M1 a distintos valores de regularización λ.

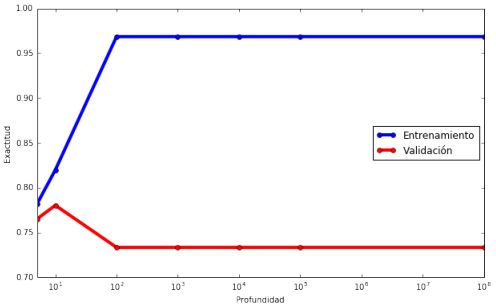


Fig. 5. Exactitud del modelo M2 con diferentes valores de profundidad del árbol de decisión

del modelo M1. En la Figura 7 se muestran los valores de la exactitud según varia el valor del parámetro λ. En este caso el valor que presenta el mejor rendimiento en el conjunto de validación es con $\lambda = 1000(10^3)$.

A diferencia del modelo M1, en este caso el lugar de las variables es tomado por las palabras utilizadas en el análisis. Las que tienen coeficientes más representativos se muestran en la Tabla 4 (Las palabras fueron analizadas en mayúsculas y sin acentos).

Los resultados muestran que los modelos implementados obtienen valores de evaluación superiores al 80.00 %. La Tabla 5 presenta los resultados de las pruebas de rendimiento de los modelos donde se puede apreciar que el modelo M4 de análisis de texto tiene los valores más altos.

La parte final del trabajo consiste en utilizar los modelos implementados como resultado del proceso de validación y prueba para clasificar los registros de atenciones médicas a pacientes que fueron diagnosticados con dolor agudo. En la columna *clasificación* de la Tabla 2, se muestra la cantidad de datos estructurados y de texto a clasificar. Los datos son sometidos a cada uno de los modelos según su tipo de dato para finalmente obtener el porcentaje de registros

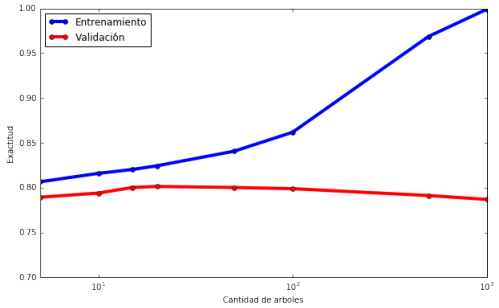


Fig. 6. Exactitud del modelo M3 para distintos valores de regularización

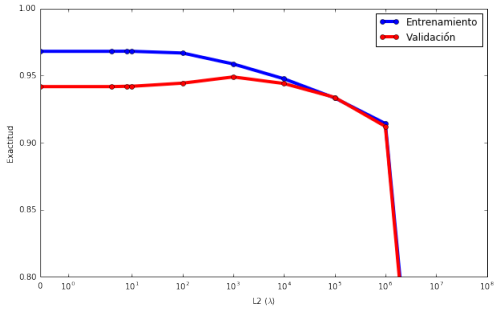


Fig. 7. Exactitud del modelo M4 con diferentes valores de regularización λ.

que pertenecen a LCE. Los resultados de este análisis se muestran en la Tabla 6 y en la gráfica de la Figura 8.

4. Discusión

El problema de los registros con valores inespecíficos es común a distintas áreas. En el sector salud se puede presentar en registros de mortalidad, de natalidad, de LCE o de prevalencia y/o incidencia de enfermedades, solo por mencionar algunas. Este trabajo se enfoca en el subregistro de LCE como consecuencia del código CIE-10 del dolor agudo, sin embargo, otros padecimientos pueden verse afectados por este o por algún otro código inespecífico [4].

Los datos disponibles para este trabajo muestran que en el periodo analizado, el 13.30 % de los registros totales en el servicio de urgencias del hospital son diagnósticos de dolor agudo. Si se compara con el 16.13 % que corresponde a LCE en el mismo periodo, se puede considerar como factor importante en las estadísticas que se reportan estatal y nacionalmente. Los resultados muestran que en promedio 83.10 % de los registros de dolor agudo son LCE lo que aumentaría su prevalencia del 16.13 % al 25.66 % del total de atenciones en el servicio de urgencias del hospital analizado.

Tabla 4. Palabras con coeficientes más representativos del modelo M4 con $\lambda = 1000$.

Palabra	Coeficiente
Privación	0.5067
Minimizar	0.4201
Difteria	0.3803
Refuerza	0.3747
Arteriovenoso	0.3392
Alcali	-0.3454
Escotoma	-0.3140
Preparan	-0.3110
Leucemias	-0.2905
Purgante	-0.2692

Tabla 5. Desempeño de los modelos contra los datos de prueba.

Modelo	Exactitud	F1-score	Precisión	Recall	AUC
M1	0.8117	0.8184	0.7941	0.8442	0.8758
M2	0.8001	0.8006	0.8026	0.7985	0.8702
M3	0.8142	0.8181	0.8054	0.8313	0.8878
M4	0.9393	0.9392	0.9538	0.9250	0.9729

Gran parte de la información hospitalaria se encuentra en formato no estructurado y suele dejarse fuera del análisis de datos tradicional (con la pérdida de información que esto implica) sin embargo, los resultados obtenidos en este trabajo sugieren que la minería de textos puede ser de utilidad en este tipo de situaciones ya sea como complemento a las técnicas tradicionales o como una alternativa confiable cuando no existen datos estructurados suficientes y de calidad.

La dicotomización de las variables categóricas crean conjuntos de datos con altas dimensiones, por tal motivo solo se presentan los coeficientes de las variables más relevantes, y se realiza una regularización RR para evitar sobre ajuste (overfitting), sin embargo, no se realizaron regularizaciones para selección de variables. Así mismo, para mejorar la velocidad de entrenamiento, validación y prueba, en este trabajo se utilizaron conjuntos de validación fijos y seleccionados al azar, no obstante, para mejorar la confiabilidad de los resultados conviene realizar mas pruebas de entrenamiento-validación, por ejemplo utilizando el método de validación cruzada.

Normalmente, este tipo de estudios se realiza considerando los conjuntos de entrenamiento, validación y pruebas. En este trabajo también se muestran los resultados de datos reales con diagnósticos desconocidos y se utilizan para hacer

Tabla 6. Clasificaciones realizadas con datos reales de dolor agudo.

Modelo	LCE(%)	OTRO(%)
M1	11172(84.02 %)	2125(15.98 %)
M2	10848(81.58 %)	2449(18.42 %)
M3	11199(84.22 %)	2098(15.78 %)
M4	12240(82.56 %)	2586(17.44 %)

LCE.- Cantidad de registros clasificados como Lesión por Causa Externa, OTRO.- Cantidad de registros clasificados como no Lesión por Causa Externa.

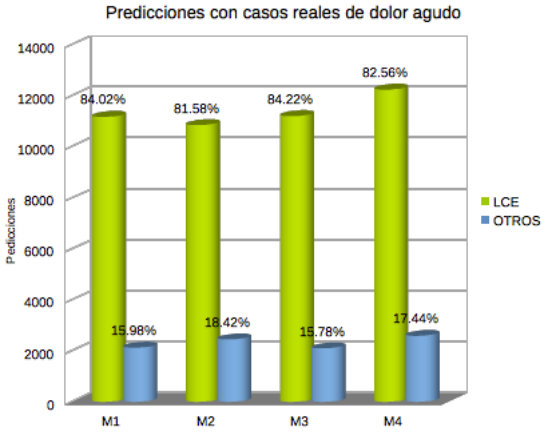


Fig. 8. Resultados de la clasificación de dolores agudos.

estimaciones de los porcentajes de LCE registradas como dolor agudo, como se puede apreciar en la gráfica de la Figura 8.

5. Conclusiones

Se presentan cuatro modelos con distintas técnicas de clasificación: 1) regresión logística con datos estructurados, 2) árboles de decisión, 3) el método boosting y 4) regresión logística con TF-IDF para el análisis de datos en formato de texto. Las pruebas realizadas y la comparación entre modelos muestran que la minería de textos es una alternativa confiable para abordar el tema de los registros inespecíficos, particularmente del subregistro de diagnósticos relativos a LCE.

Además del método tradicional de entrenamiento, validación y prueba de los algoritmos, también se utilizan los modelos validados y probados para la

clasificación de un conjunto de datos del que no se conocen las etiquetas y que representan casos reales de inespecificidad. Dada la similitud de los resultados obtenidos se puede presumir cierto grado de confianza, sin embargo, se recomienda que un porcentaje de dichos registros sean analizados y clasificados por personal experto para comparar resultados.

Considerando a la prevención como una mejor alternativa al remedio, se sugiere la revisión de catálogos y SI para evitar el uso de valores inespecificos o en su defecto, cuando la situación lo requiera, pedir al usuario una aclaración de dicho valor. Sin embargo hay que considerar que los servicios de urgencias suelen ser caóticos y se que se cuenta con poco tiempo para atención al paciente, por lo tanto los SI tendrán que ser fíciles, rápidos y asegurar la calidad de la información.

Como trabajo futuro se pretende analizar el porcentaje de diagnósticos de dolor agudo que se clasifica como LCE mediante los modelos implementados con intención de estimar a que tipo de causa externa pertenece (Accidente de tránsito, caídas, agresiones, etc.) y obtener resultados más específicos.

Agradecimientos. El presente trabajo se realizó con el apoyo del Centro Médico "Lic. Adolfo López Mateos"del Instituto de Salud del Estado de México y con la participación de la Dirección de Posgrado de la Facultad de Ingeniería de la Universidad Autónoma del Estado de México.

Referencias

1. Bhalla, K., Harrison, J., Shahraz, S., Fingerhut, L.: Availability and quality of cause-of-death data for estimating the global burden of injuries. Bulletin of the World Health Organization 88:831–838C (2010), doi: 10.2471/blt.09.068809.
2. Instituto Nacional de Salud Pública: Las lesiones por causa externa en México. Lecciones aprendidas y desafíos para el Sistema Nacional de Salud. INSP, Ciudad de México/Cuernavaca(MX) (2010)
3. Clasificación Internacional de Enfermedades (CIE-10) C OPS/OMS Colombia - OPS/OMS Colombia. En: Paho.org (Consultado 4 Apr 2017)
4. Pérez-Núñez, R., Mojarro-Íñiguez, M., Mendoza-García, M., et al.: Subestimación de la mortalidad causada por el tránsito en México: análisis subnacional. Salud Pública de México 412–420 (2016), doi: 10.21149/spm.v58i4.8021.
5. Royston, P.: Multiple imputation of missing values. Stata journal 4:227–241 (2004)
6. Híjar, M., Chandran, A., Pérez-Núñez, R. et al.: Quantifying the Underestimated Burden of Road Traffic Mortality in Mexico: A Comparison of Three Approaches. Traffic Injury Prevention 13:5–10 (2012), doi: 10.1080/15389588.2011.631065.
7. Lucini, F. S., Fogliatto, F. C., da Silveira, G. et al.: Text mining approach to predict hospital admissions using early medical records from the emergency department. International Journal of Medical Informatics 100:1–8 (2017), doi: 10.1016/j.ijmedinf.2017.01.001.
8. Ben Abacha, A., Chowdhury, M., Karanasiou, A. et al.: Text mining for pharmacovigilance: Using machine learning for drug name recognition and drug-drug interaction extraction and classification. Journal of Biomedical Informatics 58:122–132 (2015), doi: 10.1016/j.jbi.2015.09.015.

9. Kocbek, S., Cavedon, L., Martinez, D. et al.: Text mining electronic hospital records to automatically classify admissions against disease: Measuring the impact of linking data sources. Journal of Biomedical Informatics 64:158–167 (2016), doi: 10.1016/j.jbi.2016.10.008.
10. Accidentes de tránsito terrestre en zonas urbanas y suburbanas. En: Inegi.org.mx. http://www.inegi.org.mx/est/contenidos/proyectos/registros/economicas/accidentes/ (Consultado 2 Abril 2017)
11. Baldeón Calisto, M.: Análisis estadístico de accidentalidad laboral del Ecuador y comparación con la accidentalidad laboral de Colombia del año 2013. Maestría, Universidad San Francisco de Quito, Colegio de Postgrados; Quito, Ecuador (2014)
12. Secretaría de Salud/STCONAPRA: Informe sobre la Situación de la Seguridad Vial, México 2015. Secretaría de Salud, Ciudad de México (2017)
13. Bowles, M.: Machine learning in Python. 1st ed., John Wiley & Sons, Indianapolis (2015)
14. Feldman, R., Sanger, J.: The text mining Handbook. 1st ed. University Press, Cambridge (2007) ,
15. Glosario de Terminos Medicos. En: Dra Laura Cartuccia. https://auditoriamedica.wordpress.com/2009/05/24/glosario-de-terminos-medicos (Consultado 3 Abril 2017)
16. Aizawa, A.: The Feature Quantity: An Information Theoretic Perspective of Tfidf-like Measures. In: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, pp. 104–111 (2000)
17. Joachims, T.: A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. In: Proceedings of ICML-97, 14th International Conference on Machine Learning, pp. 143–151 (1997)
18. Haibo, H., Garcia, E.: Learning from Imbalanced Data. IEEE Transactions on Knowledge and Data Engineering 21:1263–1284 (2009), doi: 10.1109/tkde.2008.239
19. GraphLab Create API Documentation GraphLab Create API 1.10 documentation. In: Turi.com. https://turi.com/products/create/docs/index.html (Consultado 4 Abril 2017)

ISSN 1870-4069

DW4SN: A Tool for Dynamic Data Warehouse Building from Social Network

Rania Yangui, Ahlem Nabli, Faiez Gargouri

Sfax University, Miracl Labortory, Tunisia

Abstract. Analyzing social networks data becomes increasingly important for many companies day-to-day operations. However, due to their scale, complexity and dynamics, these networks are difficult to be processed by means of traditional warehouse systems. In fact, the rapid growth of data with the frequent arrival of new needs requires that the system should be adaptable to changes. This work presents DW4SN (Data Warehouse for Social Network) tool for building data warehouse from social network, where clustering methods are used to define the DW schema and NoSQL systems are used to implement the warehouse. We validate our system on a real data set concerning crafts women social network. The main benefits were obtained in terms of scalability and dynamicity.

Keywords: Social networks, clustering, data mining.

1 Introduction

During the last decade the explosion of social media has lead to the generation of massive volumes of user-generated data that has given birth to a novel area of research, namely data warehousing for social network. One big challenge in this domain is that companies can effectively use published data. This puts emphasis on how classical data warehouse (DW) methodologies can be extended in order to deal with novel features of social network data, such as volume, dynamicity and heterogeneity. Developing a such warehouse is a complex and costly activity. It requires strategies, which should be specific to the social network characteristics and user's needs.

Recently, Online Social Networks started to be modeled with rich structured data that incorporate semantics using ontologies like Friend of a Friend (FOAF) to describe users, content and their relationships [1]. In the literature, few FOAF-based approaches for the design of DW systems from social network have been proposed with various degrees of automation [2] [3]. However, these approaches have shown some limitations. In fact, the rapid growth of data with the frequent arrival of new needs, which requires that the system should be adaptable to changes, is not considered. Moreover, these works are based on a relational approach which presents drawbacks concerning the DW scalability.

Thereby, in this paper we present a new prototype comosed of five modules for data warehouse building from social network. In particular, we detail three

main modules which enable the DW schema building via dynamic discovery of Multidimensional Concepts (MCs) using a semi-supervised hierarchical clustering and then implementing it under document-oriented NoSQL system. This system is attractive for developers due to its ability to handle large volumes of heterogeneous data. Thereafter, we validate our system on a real data set concerning crafts women SN.

The paper is organized in the following way. Section 2 describes the case study of the paper and some related works. Section 3 introduces the generic prototyping methodology. Section 4 describes a semi-automatic tool developed to support the proposed methodology. Section 5 evaluates the proposed system and discusses the results. Section 6 concludes the paper and gives some hints to future work.

2 Research Context and Motivations

In this section, we present an example from BWEC[1] (Business for women in emerging countries) project that aims to improve the social-economic situation of crafts women in emerging countries. These women are confronted with problems as lack of capital, unavailability of raw materials, lack of knowledge and skill of modern techniques and marketing problems.

The use of social networks could positively contribute to this population by getting to know each other and sharing their business experiences and problems. Indeed, SNs help handicraft women to share information, resources and equipments. By getting to know each other and sharing their business experiences and problems in public groups, women became aware of the need to reconcile efforts to strengthen their self-help initiatives and networks in the communities. Using data available on SNs let handicraft women increase knowledge they have about production process and commercialization strategies. Therefore, we need to analyze womens interests and changes in their relationships. In order to carry this out, it is necessary to accumulate womens profiles, interactions and the information about their activities in a data warehouse.

Nevertheless, as claimed by many works [9] [10], user generated data from social network are very difficult to store and analyze in terms of traditional data warehousing methodologies. Moreover, decision-makers of that project (crafts women) are unskilled DW users, and then they need DW prototypes to validate their analysis needs.

In the literature, many researchers have proposed approaches for DW building from SNs. Nevertheless, the majority of these works focused only on the intermediate logical design phase [6] [7] [8], while few others provided approaches including both conceptual and logical design levels [9] [10]. As the conceptual modeling is the core stone of a successful DW, special attention should be paid to it. In this context, the rapid growth of data with the frequent arrival of

[1] Towards a new Manner to use Affordable Technologies and Social Networks to Improve Business for Women in Emerging Countries http://projetat.cerist.dz/artisanat/

new needs requires that the system should be adaptable to changes. The data warehouse designer, with his/her limited knowledge of the domain can bring the some multidimensional concepts, but in other cases, such process needs to be automated. Clustering techniques can help in designing dynamic DW schema.

Moreover, as NoSQL data stores are becoming increasingly popular, some works have shown that it is possible to convert a multidimensional conceptual model into NoSQL storage [11,12,6,7,8]. These systems are attractive for developers due to their ability to handle large volumes of data, as well as data with a high degree of structural variety. However, these approaches have focused only on the rules for transforming the concepts of fact and dimension in a NoSQL structure. Furthermore, it appears that the majority of researchers use HBase for implementing a NoSQL DW. This is justified by the resemblance between the logic model of HBase and that of relational databases, particularly in terms of concepts of tables and rows.

Based on the above discussion, there is a strong need for a significant prototype that allows a semi-automatic building of dynamic DW from SN. Semi-Automating data warehouse conceptual and logical design can provide real value to DW development projects, and increase their chance of success while reducing cost, risk and manual effort.

3 Building Data Warehouse from Semantic Social Network: Generic Prototype

Given the sheer volume of social network data normally involved in the building of a data warehouse, a case can be made for semi-automated support in the construction of a dynamic warehouse. We build a prototype that can load a company's data warehouse and a FOAF ontology, design new multidimensional concepts in the data warehouse schema and implement the obtained warehouse under document-oriented NoSQL system. The prototype uses a clustering method for the dynamic discovery of MCs, and uses transformation rules for the mapping to NoSQL data base. The prototype is composed of five modules as depicted in Fig.1.

1. Data warehouse schema crawler: the system recovers and displays the multidimensional concepts from the company's data warehouse schema. This warehouse is created following a classical mixed approach [13].
2. FOAF Generator: The system loads the FOAF ontologies from livejournal social network and merges them.
3. Clustering Performer: the system computes the similarity measure and performs the hierarchical clustering.
4. Multidimensional Concepts Determiner: According to the result of clustering and guided by the designer, the system determines new MCs. Then the system updates the le that describes the DW schema by adding new MCs.
5. NoSQL mapper: The system applies a set of transformation rules at ETL (Extract Transform Load) lavel to implement the NoSQL DW.

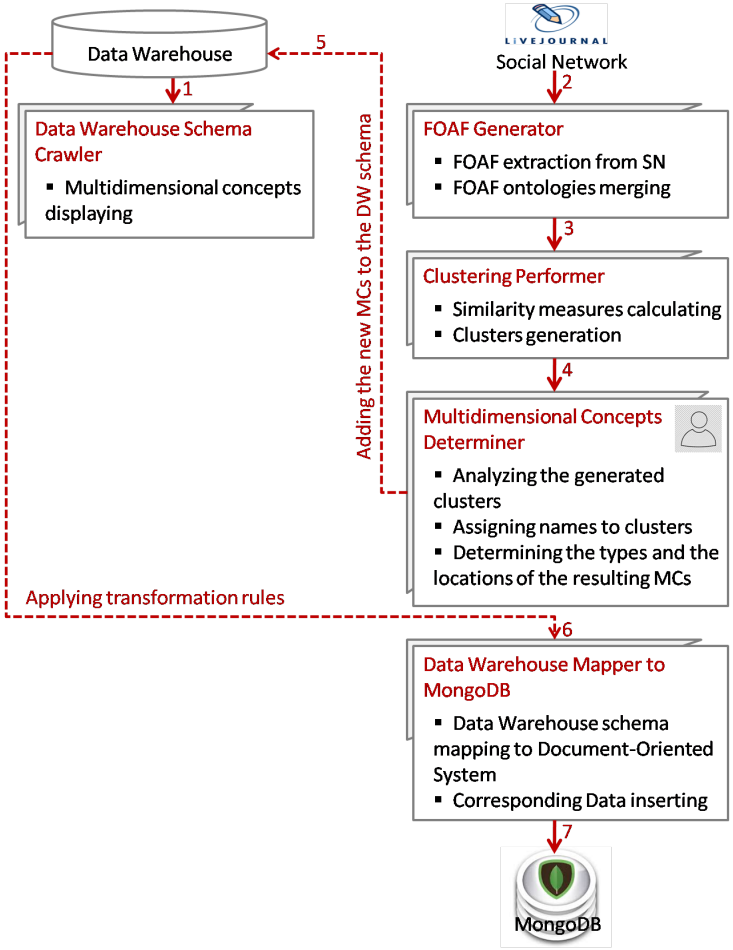


Fig. 1. Prototype woring.

Our prototype allows designer to easily define and validate their DW in an incremental way. In the next subsection, we will detail the three last modules.

3.1 Clustering Performer

In the implementation of our prototype we have chosen to use SHICARO [14] (Semi-supervised Hierarchical Clustering based on Ranking features using Ontology) method that aims to cluster objects based on scheduled features in order to get the optimal results that meet the designer needs.

Since the expert knows the goal behind which the clustering is performed, SHICARO method performs clustering under the designer guidance. Thus, the designer is required to order the features from the highest to the lowest ones.

As input to this method, we have a set of FOAF instances and a set of scheduled features. At each iteration, a set of features $F = \{(f_1, r_1), \ldots, (f_n, r_n)\}$ that have the same rank r_i is applied to cluster objects. The number of iterations is equal to the maximum of the ranks. The N number of clusters to be generated is expressed in terms of the number of instances and of iterations:

$$N = Round\left(InstancesNumber/2^{Max(r_i)}\right).$$

Steps followed during implementation of SHICARO method are described by Algorithm 1.

Data: FOAF ontology, $F = \{(f_i, r_i)\}$
Result: $C = \{c_k\}$
initialization: Each instance is placed in its own cluster c_k ;
repeat
1. Find min r_i ;
2. Select the current future set F_c having min rang ;
3. Calculate p the number of clusters to generate at each iteration ;
4. Compute similarities based on the current features set F_c ;
5. Merge c_k to p clusters ;
6. Update C and F ;

until $F = \emptyset$;
Return C ;

Algorithm 1: SHICARO Algorithm.

This process is repeated for each generated cluster (step 1) based on the next current feature set until the number of features reaches 0. The extracted clusters become the input for the dynamic schema upgrading in the next subsection.

3.2 Multidimensional Concepts Determiner

This module is performed by the designer and consists in analyzing the generated clusters, determining the type of multidimensional concept, assigning names to clusters and specifying the location of insertion in the multidimensional schema. The multidimensional concepts determination module is described by Algorithm 2.

As described by Algorithm 2, fives cases are possible: adding new fact, adding new measure to an existing fact, adding new dimension, adding new week attribute or adding new parameter to an existing or a new hierarchy.

Data: A set of clusters $C = \{c_i\}$, Initial DW Schema $MS_{DW} = (F, D)$ where $F = \{f_1, \ldots, f_n\}$ and $D = \{d_1, \ldots, d_k\}$
Result: discovered MCs integrated in the MS_{DW}
forall the *cluster* c_i *in* C **do**
1. Display the cluster c_i to the user ;
2. The user input a name N_{c_i} to c_i ;
3. Identify a type T_{c_i} of c_i ;
4. **switch** *the value of* T_{c_i} **do**
   - **case** *"Fact"*
     - Assign N_{c_i} to the fact f ;
     - Add f to F ;
     - Display all dimensions D from MS_{DW} ;
     - Select the dimensions $d \in D$ related to f ;
     - Associate d to f in MS_{DW} ;
   - **case** *"Dimension"*
     - Assign N_{c_i} to the dimension d ;
     - Add d to D ;
     - Display all fact F from MS_{DW} ;
     - Choose the facts $f \in F$ to which d is related ;
     - Associate d to f in MS_{DW} ;
   - **case** *"Measure"*
     - Assign N_{c_i} to the measure m ;
     - Display all fact F from MS_{DW} ;
     - Choose the fact $f \in F$ to which m will be added ;
     - Add m to f ;
   - **case** *"Week Attribute"*
     - Assign N_{c_i} to the week attribute wa ;
     - Display the set of dimension D from MS_{DW} ;
     - Select the dimension d to which wa will be added ;
     - Display all parameters of d ;
     - Select the parameter p to which wa will be added ;
     - Add wa to p ;
   - **otherwise**
     - Assign N_{c_i} to the parameter p ;
     - Display the set of dimension D from MS_{DW} ;
     - Select the dimension d to which p will be added ;
     - Display all the hierarchies of d ;
     - **if** *p will be added to a new hierarchy h* **then**
       - Add h to d ;
       - Add p to h ;
     - **else**
       - Choose the hierarchy he to which p will be added ;
       - Choose the level l at which p will be added ;
       - Add p to he at level l ;
     - **end**
   - **end**
4. **endsw**

end

Algorithm 2: MCs Definition.

3.3 NoSQL Mapper

Given that a well-designed DW requires a well planned logical design, all updates and versions of a DW lead to a revision of the logical design. Generally, the mapping from the conceptual to the logical model is made according to three approaches: ROLAP (Relational-OLAP), MOLAP (Multidimensional-OLAP) and HOLAP (Hybrid-OLAP) [15]. However, these systems suffer from scaling-up to large data volume [7]. As an alternative, NoSQL systems begin to grow.

NoSQL is a dynamic and cloud friendly approach to dynamically process social network generated data. NoSQL Database, also known as Not Only SQL is an alternative to SQL database which does not require any kind of fixed schemas. NoSQL generally scales horizontally and avoids major join operations on the data.

In the literature, four types of NoSQL data stores exist [16]: Key-value Stores, Document Stores, Columnar Stores and Graph Stores. In the absence of a clear approach which allows the implementation of data warehouses under NoSQL model, we have compared in previous work [17], two NoSQL systems: columns-oriented and documents-oriented with two types of transformation: simple and hierarchical. The results showed that the documents-oriented NoSQL data warehouse with hierarchical transformation is more efficient in terms of interrogation. This is justified by the fact that data in the column-oriented systems are not available in the same place.

The hierarchical transformation to document-oriented system uses different collections for storing facts and dimensions, and uses the simple documents for representing measure and the composed attributes for representing dimension attributes while explaining hierarchies.

Rule 1 represents the transformation of a fact and its measures to the document-oriented model.

Rule 1: Each fact $F \in MS^{Fact}$ is transformed to a document DC (DC^N, DC^{Att}) where :

- **The name of the document is the name of the fact** $/DC^N \leftarrow F^N$**;**
- **Each measure** $M \in F$ **is transformed to a simple attribute** $SA \in DC^{att}/SA^N \leftarrow M$**;**
- **Each identifier of a related dimensions is transformed to a simple attribute** $SA \in DC^{att}/SA^N \leftarrow D^{id}$**;**

Moreover, the hierarchical transformation of a dimension and its attributes (Strong and Weak) to the document-oriented model is mentioned by Rule 2.

Rule 2: Each dimension $D(D^N, D^{Att}, D^{Hier})/D \in MS^{Dim}$ is transformed to a document $DC(DC^N, DC^{Att})$ where:

- **The name of the document D is equivalent to the name of the dimension $/DC^N \leftarrow D^N$;**
- **Each hierarchy H(H^N, H^P, P^{FAtt}) is transformed to a composed attribute $CA \in DC^{att}$ where :**
  - **The name of the composed attribute is the hierarchy name $/CA^N \leftarrow H^N$**
  - **The values of composed attributes are the simple attributes that represent the weak and the strong attributes.**

Based on the above defined rules (Rule 1 and Rule 2), we deduce the hierarchical transformation of a multidimensional schema (MS) as mentioned by Rule 3.

Rule 3: Each multidimensional schema MS (MS^N, MS^{Fact}, MS^{Dim}, $Func$) is transformed to a documents collection $DCC(DCC^N, DCC^{Val})$ where:

- **The name of the collection is the name of the MS $/DCC^N \leftarrow MS^N$;**
- **Each fact $F \in MS^{Fact}$ is transformed to a document $DC(DC^N, DC^{Att})$ (Rule 1);**
- **Each dimension $D(D^N; D^{Att}; D^{Hier}) \in MS^{Dim}$ is transformed to a document $DC(DC^N, DC^{Att})$ (Rule 2).**

4 DW4SN Tool

In this section, we present the propose system DW4SN(Data Warehouse for Social Network). This system is composed of two main operational phases (Fig.2).

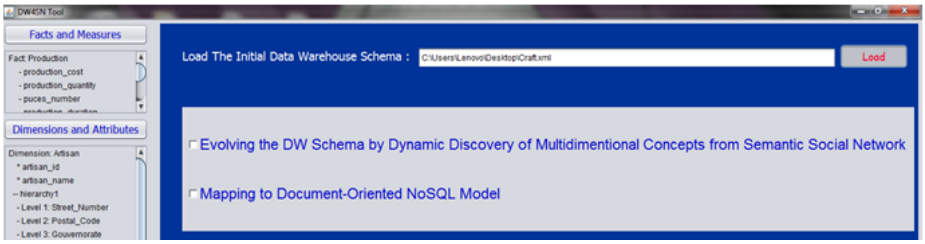


Fig. 2. DW4SN operational phases.

The first one is the Dynamic Discovery of MCs during which DW4SN enriches the warehouse schema using hierarchical clustering with ranking features. The second phase of DW4SN is the Migration to Document-Oriented NoSQL system,

during which the system apply a set of rules that create the warehouse under MongoDB.

This work is directed by means of JAVA programming language. To access social network pages, we extracted the FOAF ontology for CraftsWomen group in livejournal. This group is created by the project community to gather crafts women from Tunisia and Algeria. For each instance of foaf: Person in the generated FOAF ontology, we extracted the corresponding personal FOAF. Then, we used the PROMPT plug-in integrated in Protg2000 Tool to merge the extracted FOAF files. We obtained an ontology with hundreds of instances.

To perform the clustering on FOAF instances, we used the Weka API integrated with java. At each iteration, we used the hierarchical clustering. The output of each iteration becomes the input of the next one. The number of clusters at each iteration is calculated in terms of the number of instances and the max of rank in the feature set. The user is required to load the initial DW, and then he is required to load the FOAF ontology and select the object on which he wants to perform the clustering. After that, the user is required to enter an ordered feature set that meets his need and then apply the clustering button. An interface that contains the clustering results is so displayed.

Fig.3 shows the Clustering configuration and results in our tool.

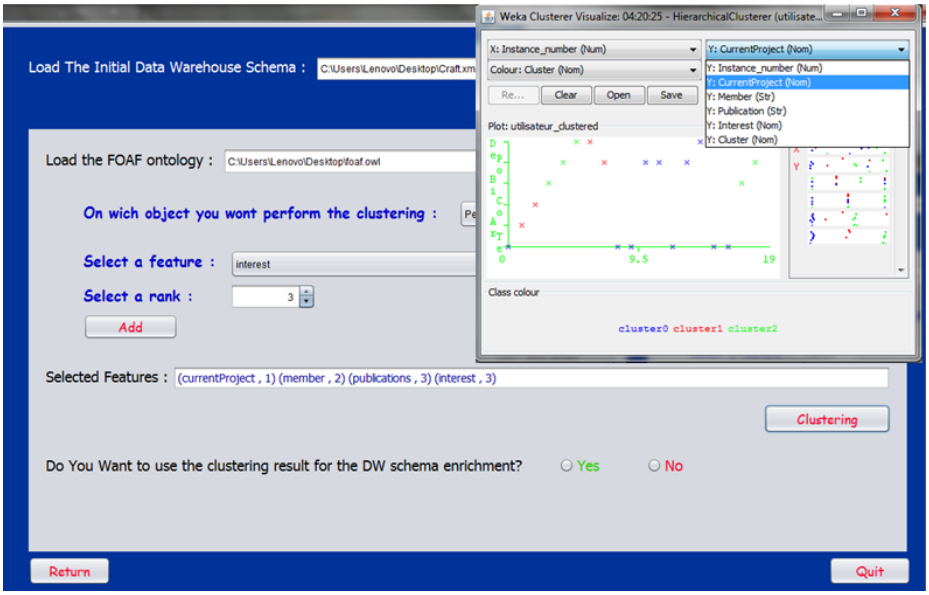


Fig. 3. Clustering configuration and results.

Thereafter, the system asks the user if he wants to use the clustering results for the DW schema enrichment. If the response of the user is positive, a new

interface is displayed. Then, the system asks the user to appoint the new MC and specify its type and its position in the DW schema. Many cases are possible:

- Adding a new fact. In this case the designer should select the related dimensions. In fact, due to the dynamicity of SN, the analysis axes may change and increase.
- Adding a new measure to an existing fact.
- Adding a new dimension. In this case, the user must select the fact to which the dimension will be related.
- Adding a new week attribute. In this case, the designer must select the concerned dimension.
- Adding a new parameter. In this case, the user should select the desired dimension and specify if the parameter will be added to an existing or a new hierarchy. If it is a new hierarchy, the system creates a hierarchy and inserts the parameter at the first level. Else (if it is an existing hierarchy), the designer should select the level at which he wants to add the parameter.

Fig.4 represents examples of adding a new MCs to the DW schema.

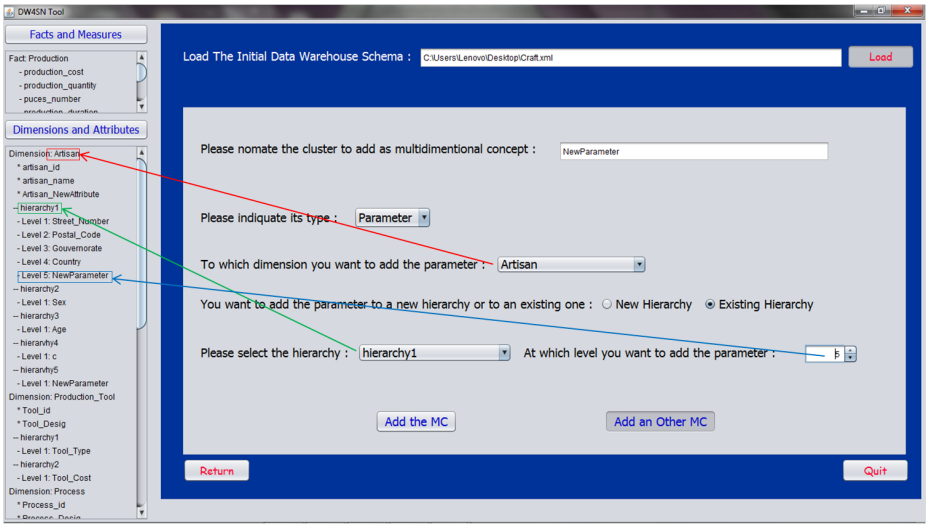


Fig. 4. Adding new MCs.

The screenshot shows an example of adding a new parameter to an existing hierarchy (hierarchy 1) at the level 5 in the "Artisan" dimension.

To implement the DW under MongoDB, we used the java routines integrated with the data integration tool Talend for Big Data. In fact, the schema-less nature of the document-oriented data base means that we can store documents in any shape but the notion of schema itself doesnt disappear from the model.

Therefore, to create a NoSQL DW, we are required to write a program that relies on some form of implicit schema (Fig.5).

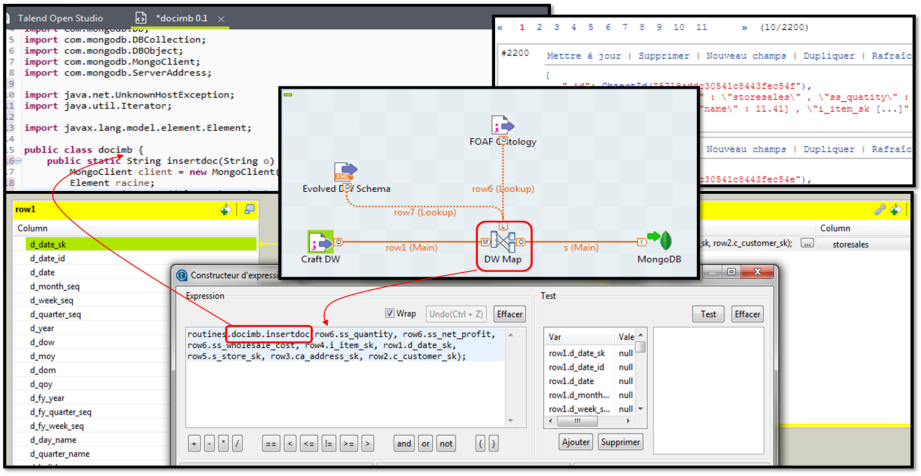


Fig. 5. Example of ETL routines.

Fig.5 shows four main interfaces: the created Job, an excerpt of the implemented java routine, the expression editor, and an excerpt of the resulting MongoDB data base.

A Job is a graphical design, of one or more components connected together such as tFileInputDelimited (Craft DW, FOAF Ontology, Evolved DW Schema), tMap (DWMap) and FileOutputDelimited (MongoDB). Otherwise, a routine is a complex Java code, generally used to optimize data processing and improve Job capacities. In this work, the routine is used for implementing the transformation rules. The implemented routine is then called and edited in the expression editor. This editor provides visual interface to write any function or transformation in a handy dedicated view.

5 Discussion

In this paper, we proposed a novel tool for building data warehouse from social network. The focus of this work is to overcome the existing limitation identified in the literature and to accomplish the ever growing requirement of modern analytical systems. In this part, we evaluate the proposed system and discuss the results. For that, we applied it in a real case study: BWEC project. In this project our task is to build a data warehouse from craft women social network.

One of the important aspects of the proposed prototype is the dynamic management of data warehouse schema. To get the optimal results that meet the

designer needs, we proposed, an algorithm for semi-supervised hierarchical clustering based on scheduling features using FOAF ontology. Then, we proposed an algorithm for multidimensional concepts determination based on the generated clusters. For the sake of discussion, we compared the grouping among the simple hierarchical clustering (HC) and SHICARO method for 11 month in 2016. We compared the F-score measures per month.

Fig.6 shows the F-Score results per month for HC and SHICARO algorithms.

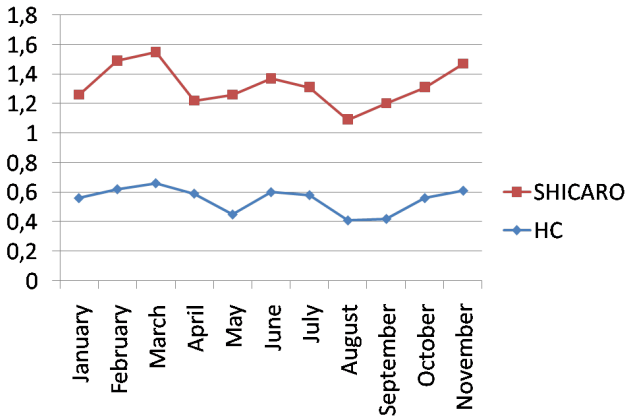


Fig. 6. F-Score results per month for HC and SHICARO algorithms.

It has been observed that SHICARO algorithm is the best (the average of F-Score is using SHICARO is while 11 for HC) . The proposed system provides a high degree of automation that enables designers to quickly and easily designing and building conceptual DW schema and to reduce the deployment costs.

Moreover, as NoSQL data stores are becoming increasingly popular in application development, we proposed transformation rules to map conceptual schema to document-oriented model. Document-oriented DB is attractive for developers due to their ability to handle large volumes of data with a high degree of structural variety. Typically, NoSQL data stores are accessed programmatically since they are schema-less. Thus, the implementation of the transformation rules is performed at the ETL level while integrating data.

In order to evaluate the implemented NoSQL DWs, we choose to use Read Request Latency (RRL) metric. This metric has the purpose to evaluate the systems ability to respond quickly to user requests. We built a sample data warehouse which was stored in both MySQL and MongoDB in order to be compared for disk space usage and Read Request Latency (Fig. 7).

Regarding requests, we choose to use four queries classified into two categories. The first category consists on increasing the number of dimensions and attributes to test the performance of the decision system with the presence of

joins in the users queries. As for the second category, it is more complex and consists on using some operators.

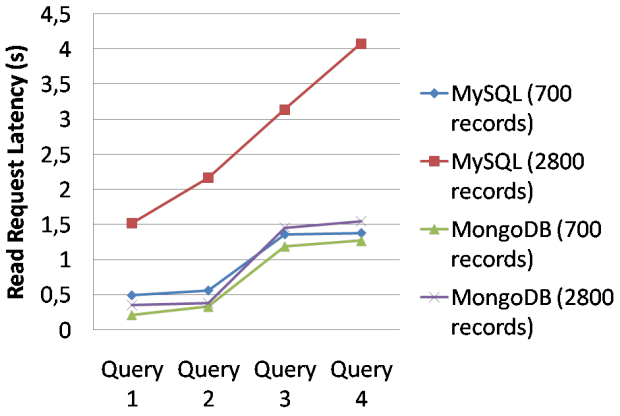


Fig. 7. Read Request Latency Values.

We found that the DW implemented under MySQL consumed less storage space. This is justified by the fact that MongoDB allows every record to have a completely different structure than every other. It stores the schema of every record with the record itself. However, DW implemented under SQL is lack of scaling and inefficient of handling big data.

We deducted that, the NoSQL data warehouse can respond to increasing queries without performance degradation (0.21 for Q1 and 1.27 for Q4). Moreover, the NoSQL data warehouse can rapidly adapt to growth in data volume and query intensity without degrading performance (0.21 for 700 records while 0.35 for 2800 records using the same query). The NoSQL data warehouse can quickly adapt to changes in data structure and content without requiring any schema redesign, additional data migration, or new data storage structures.

6 Conclusion

In this work, we have presented a prototyping methodology and the associated tool, to build a data warehouse from social network. To do that, we proposed a method to build DW schema via content-based discovery of facts, dimensions, hierarchies and measures using hierarchical clustering. This method uses a Semi-supervised Hierarchical Clustering based on ranking features using ontology.

The second contribution of this work is to propose rules for transforming a multidimensional conceptual schema into NoSQL system. As NoSQL databases are schema-less, the creation of a schema occurs while inserting data at ETL

level. In this level, we implemented the defined transformation rules as routines that reflect the implicit schema.

The overall proposed methodology meets the majority of the posed challenges. In fact, NoSQL allows an effective modeling of the massive data volumes generated by the SN. Moreover, the dynamic discovery of MC using semi-supervised hierarchical clustering handles the rapid growth of the mass of information. As perspectives, we aim implementing OLAP operators (D-Roll up, D-Slice, D-Dice) with the phases of the invisible join technique for a document-oriented data warehouse.

References

1. Chelmis, C., Wu H., Sorathia, V., Prasanna V.K.: Semantic Social Network Analysis for the Enterprise. In: Computing and Informatics, pp. 479–502 (2014)
2. Sohn, J.S., In-Jeong, C.: Dynamic FOAF management method for social networks in the social web environment. The journal of supercomputing, pp. 1–17 (2013)
3. Wojciech, K.: Facebook crawler as software agent for business intelligence system. Studia informatica, pp. 89–110 (2014)
4. U. Rehman, N., Mansmann, S. Scholl, M. H.: Discovering dynamic classification hierarchies in OLAP dimensions. In: Proceedings of the 20th international conference on Foundations of Intelligent Systems, pp. 425-434 (2012)
5. Gallinucci, E., Golfarelli, M. Rizzi, S.: Meta-Stars: Dynamic, Schemaless, and Semantically-Rich Topic Hierarchies in Social BI. In: Proceeding of the 18th International Conference on Extending Database Technology, pp. 529–532 (2015)
6. Dehdouh, K., Boussaid, O., Bentayeb, F.: Using the column oriented NoSQL model for implementing big data warehouses. In: Proceedings of the 21st International Conference on Parallel and Distributed Processing Techniques and Applications, pp. 469–475 (2015)
7. Chevalier, M., El Malki, M., Kopliku, A., Teste, O., Tournier, R.: Implementing Multidimensional Data Warehouses into NoSQL. In: Proceedings of the 17th International Conference on Enterprise Information Systems, pp. 172–183 (2015)
8. Santos, M.Y., Martinho, B. Costa, C.: Modelling and implementing big data warehouses for decision support. Journal of Management Analytics, pp. 1–19 (2017)
9. U. Rehman, N., Mansmann, S. Scholl, M. H.: Discovering dynamic classification hierarchies in OLAP dimensions. In: Proceedings of the 20th international conference on Foundations of Intelligent Systems, pp. 425-434 (2012)
10. Gallinucci, E., Golfarelli, M. Rizzi, S.: Meta-Stars: Dynamic, Schemaless, and Semantically-Rich Topic Hierarchies in Social BI. In: Proceeding of the 18th International Conference on Extending Database Technology, pp. 529–532 (2015)
11. Bringay, S., Laurent, A., Poncelet, P., Roche, M., Teisseire, M.: Towards an On-Line Analysis of Tweets Processing. In: Proceedings of the 22nd International Conference on Database and Expert Systems Applications, pp. 154–161 (2011)
12. Dede, E., Govindaraju, M., Gunter, D., Canon, R., Ramakrishnan L.: Performance evaluation of a MongoDB and hadoop platform for scientific data analysis. In: Proceedings of the 4th ACM workshop on Scientific cloud computing, pp. 13–20 (2013)
13. Ahlem, N., Senda B., Rania Y., Faiyez G.: Two-ETL Phases for Data Warehouse Creation: Design and Implementation. In: Proceedings of the 19th East European

Conference in Advances in Databases and Information Systems, Poitiers, France, pp. 138–150 (2015)
14. Yangui, R., Nabli, A., Gargouri, F.: Semi-supervised Hierarchical Clustering bAsed on Ranking features using Ontology. In: The 2d International Conference on Management and Technology in Knowledge, Service, Tourism and Hospitality, pp.225-231 (2014)
15. Chaudhuri, S., Dayal, U., Ganti, V.: Database technology for decision support systems. IEEE Computer Society, pp. 48–55 (2002)
16. Sharma, V.: SQL and NoSQL Databases. International Journal of Advanced Research in Computer Science and Software Engineering, pp. 20–27 (2012)
17. Yangui, R., Nabli, A., Gargouri, F.:Automatic Transformation of Data Warehouse Schema to NoSQL Data Base: Comparative Study. In: Knowledge-Based and Intelligent Information and Engineering Systems, Proceedings of the 20th International Conference, pp. 255-264 (2016)

Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
noviembre de 2017
Printing 500 / Edición 500 ejemplares