

**XIII Congreso Español
de Metaheurísticas,
Algoritmos Evolutivos y
Bioinspirados
(XIII MAEB)**

MAEB 7.2:
ALGORITMOS MULTI OBJETIVO
Y APLICACIONES





Resumen extractivo de texto multidocumento usando un enfoque de optimización multiobjetivo basado en colonia de abejas artificiales

Jesus M. Sanchez-Gomez
 Departamento de Matemáticas
 Universidad de Extremadura
 Cáceres, España
 jmsanchezgonzalez@unex.es

Miguel A. Vega-Rodríguez
 Depto. Tecnología de Computadores y Comunicaciones
 Universidad de Extremadura
 Cáceres, España
 mavega@unex.es

Carlos J. Pérez
 Departamento de Matemáticas
 Universidad de Extremadura
 Cáceres, España
 carper@unex.es

Resumen—Los métodos automáticos de resumen de texto son cada vez más necesarios en la actualidad. Los enfoques de resumen extractivo multidocumento pretenden obtener el contenido principal de una colección de documentos a la vez que reducen la información redundante, lo que puede ser abordado desde un enfoque de optimización multiobjetivo. En este trabajo se ha diseñado e implementado un algoritmo MOABC (*Multi-Objective Artificial Bee Colony*) para esta tarea. Los experimentos se han realizado en base a conjuntos de datos de DUC (*Document Understanding Conferences*), y se han evaluado con las métricas ROUGE (*Recall-Oriented Understudy for Gisting Evaluation*). Los resultados muestran mejoras importantes: del 31,09 % y del 18,63 % para el ROUGE-2 y del 8,43 % y del 6,09 % para el ROUGE-L, con respecto a los mejores resultados de enfoques mono-objetivo y multiobjetivo de la literatura científica. Además, se ha demostrado que los valores ROUGE obtenidos son más estables, con una reducción en la dispersión relativa de entre 620,63 % y 1333,95 %, es decir, entre 6 y 13 veces más robustos.

Palabras clave—Resumen de texto multidocumento, optimización multiobjetivo, colonia de abejas artificiales, cobertura del contenido, reducción de la redundancia.

I. INTRODUCCIÓN

En la actualidad, la cantidad de información en Internet crece de forma exponencial en cualquier tema, y los usuarios desean obtener solo la información más relevante sobre dicho tema lo más rápido posible. Una forma de extraer esta información es mediante herramientas de minería de texto [1], con el fin de poder generar automáticamente un resumen a partir de toda la información del tema en cuestión [2], cubriendo la información más importante y evitando la redundancia de la misma.

Existen varios métodos de resumen automático: dependiendo de si la información se obtiene de uno o de más documentos, el resumen puede ser mono-documento o multidocumento [3]. Un resumen mono-documento reduce la información del documento a una breve representación, mientras que un resumen multidocumento selecciona información de toda la colección de documentos. Además, los métodos también pueden ser abstractivos o extractivos [4]. Los métodos abstractivos pueden construir un resumen con palabras o frases que no existen en el texto original, mientras que los métodos

extractivos seleccionan subconjuntos del texto original. El objetivo principal del resumen extractivo multidocumento es representar la información más importante en una versión reducida de los documentos originales, manteniendo los contenidos principales y reduciendo la redundancia. De esta manera, los usuarios pueden obtener las ideas principales del texto de forma rápida.

Como se ha definido, la naturaleza del resumen extractivo multidocumento es multiobjetivo, pero la mayoría de los enfoques en la literatura científica se basan en modelos de optimización mono-objetivo (p. ej. [5]). En estos enfoques solo se optimiza una función objetivo, la cual incluye a su vez varios objetivos que deben ser ponderados. Esta asignación de pesos es subjetiva, lo que influye mucho en la solución final. Hasta la fecha solo se ha propuesto un enfoque de optimización multiobjetivo [6], el cual ha obtenido mejores resultados que los enfoques mono-objetivo. Por lo tanto, el resumen automático de texto multidocumento mediante optimización multiobjetivo es una línea de investigación muy reciente, en la que la hipótesis principal es que al utilizar la optimización multiobjetivo se pueden obtener mejores resultados que con la optimización mono-objetivo.

En este trabajo, el resumen extractivo multidocumento es abordado mediante el algoritmo MOABC (*Multi-Objective Artificial Bee Colony*), el cual maximiza simultáneamente tanto la cobertura del contenido como la reducción de la redundancia de la colección de documentos. Los experimentos se han realizado en base al conjunto de datos de DUC (*Document Understanding Conferences*) y los rendimientos del modelo se han evaluado con las métricas ROUGE (*Recall-Oriented Understudy for Gisting Evaluation*). El enfoque propuesto ha obtenido resultados que mejoran las medias ROUGE de los mejores enfoques de la literatura científica, obteniendo además resultados más estables.

II. TRABAJO RELACIONADO

En esta sección se muestra una revisión de las principales técnicas de optimización que se han implementado para el resumen extractivo multidocumento.

En primer lugar, se revisan los trabajos mono-objetivo, donde todos los objetivos son ponderados subjetivamente para obtener una función única. [7] consideró el resumen orientado a consultas de documentos como un problema de optimización global con cuatro objetivos, en el que los experimentos se realizaron con enfoques linealizados y lexicográficos. [8] propuso un método genérico de resumen multidocumento basado en la agrupación de oraciones, resolviéndolo con una modificación del algoritmo de optimización de enjambre de partículas (PSO). [9] diseñó un modelo de resumen de texto no supervisado basado en la programación lineal entera, que resolvió con un algoritmo de ramificación y poda (B&B) y un algoritmo PSO. [10] y [11] propusieron el resumen multidocumento como un problema de la p -mediana modificada, implementando [10] un algoritmo de evolución diferencial (DE) autoadaptativo y [11] un algoritmo de evolución diferencial basado en mutación y cruce autoadaptativos (DESAMC). [12] y [13] abordaron el resumen de documentos como un problema de programación no lineal 0-1, donde la función objetivo se definió como la media heroniana de los criterios, y el problema fue resuelto con el algoritmo PSO en ambos casos. [5] consideró el resumen de documentos como un problema de optimización discreta, proponiendo un algoritmo DE adaptativo para resolverlo. [14] y [15] consideraron la programación booleana cuadrática, donde la función objetivo era una combinación ponderada de los objetivos. En ambos casos se implementó un algoritmo DE binario. [16] propuso el resumen de documentos como un problema de programación entera cuadrática, resolviéndolo con un algoritmo PSO. [17] describió un enfoque basado en optimización para el resumen de documentos, implementando un algoritmo DE mejorado. [18] presentó el resumen de documentos como un problema de optimización lineal y no lineal, utilizando un algoritmo PSO para resolverlo. [19] describió el resumen multidocumento como un problema de optimización binaria, proponiendo un algoritmo de selección elitista intergeneracional, recombinación heterogénea y mutación cataclísmica (CHC). En [20], el resumen de texto se trató como un problema de programación booleana, que se resolvió con un algoritmo DE. Finalmente, [21] consideró el resumen multidocumento basado en el enfoque de proximidad tópica, para el cual se propuso un algoritmo DE autoadaptativo.

La optimización multiobjetivo resuelve los problemas de rendimiento de los modelos mono-objetivo, optimizando cada función objetivo sin ponderaciones. En el único trabajo multiobjetivo encontrado ([6]), se propuso un modelo de resumen extractivo multidocumento basado en la optimización discreta, implementando un algoritmo genético de ordenación no-dominada II (NSGA-II) para la resolución del problema.

Los principales objetivos incluidos en estos trabajos son la cobertura del contenido y la reducción de la redundancia. Sin embargo, también existen otros criterios como la relevancia, la coherencia y la significancia, pero no son tan comunes como los anteriores. Además, todos los trabajos llevaron a cabo la experimentación utilizando los conjuntos de datos de DUC y las métricas de evaluación ROUGE.

III. DEFINICIÓN DEL PROBLEMA

En esta sección se define el resumen de texto multidocumento como un problema de optimización. Los métodos más usados en este contexto son los métodos de representación de vectores de términos (palabras), donde cada oración se representa como un vector de términos, y la similitud entre oraciones se compara por pares mediante el uso de algún criterio. El criterio mayormente utilizado es la similitud coseno, como en [5] y [6].

III-A. Similitud coseno

Dado el conjunto $T = \{t_1, t_2, \dots, t_m\}$, que contiene los m términos distintos de la colección de documentos D . Suponiendo un total de n oraciones, cada oración s_i de D se representa como un vector, $s_i = (w_{i1}, w_{i2}, \dots, w_{im})$, $i = 1, 2, \dots, n$, donde cada elemento es el peso de su término correspondiente. Este peso w_{ik} está asociado con el término t_k en la oración s_i , y puede calcularse usando el esquema *frecuencia del término - frecuencia inversa de la oración* (tf_isf), donde tf mide cuántas veces aparece el término en una oración, e isf mide cuántas oraciones de D contienen el término [22], de la siguiente forma:

$$w_{ik} = tf_{ik} \cdot \log(n/n_k), \quad (1)$$

donde tf_{ik} cuenta cuántas veces aparece el término t_k en la oración s_i , y $\log(n/n_k)$ es el factor isf , donde n_k denota el número de oraciones que contienen el término t_k .

El contenido principal de D se puede resumir cuantitativamente con la media de los pesos de los m términos en T con un vector medio $o = (o_1, o_2, \dots, o_m)$, donde cada componente se calcula como sigue:

$$o_k = \frac{1}{n} \sum_{i=1}^n w_{ik}, \quad k = 1, 2, \dots, m. \quad (2)$$

Finalmente, la similitud coseno se calcula a partir de los pesos previamente definidos, midiendo la semejanza entre las oraciones $s_i = (w_{i1}, w_{i2}, \dots, w_{im})$ y $s_j = (w_{j1}, w_{j2}, \dots, w_{jm})$ de la siguiente forma:

$$sim(s_i, s_j) = \frac{\sum_{k=1}^m w_{ik} w_{jk}}{\sqrt{\sum_{k=1}^m w_{ik}^2 \cdot \sum_{k=1}^m w_{jk}^2}}, \quad i, j = 1, 2, \dots, n. \quad (3)$$

III-B. Formulación del problema de optimización

Dado el conjunto $D = \{d_1, d_2, d_3, \dots, d_N\}$, que contiene N documentos. D también se puede representar como un conjunto que contiene las n oraciones de la colección como $D = \{s_1, s_2, \dots, s_n\}$. El objetivo es generar un resumen $\bar{D} \subset D$ teniendo en cuenta los siguientes tres aspectos:

- *Longitud*: el resumen generado \bar{D} debe tener una longitud fija L (con cierta tolerancia).
- *Cobertura del contenido*: el tema principal de la colección de documentos D debe ser cubierto en el resumen \bar{D} incluyendo las oraciones apropiadas.



- *Reducción de la redundancia*: el resumen \bar{D} no debe ser redundante, es decir, las oraciones existentes en D que son similares entre sí no deben repetirse en el resumen generado.

Este problema de resumen de texto implica la optimización simultánea de la cobertura del contenido y de la reducción de la redundancia. Sin embargo, estos dos objetivos son contradictorios entre sí, por lo que un enfoque de optimización multiobjetivo es la forma natural de abordar este problema.

Sea $x_i \in \{0,1\}$ una variable de decisión binaria que considera la presencia ($x_i = 1$) o la ausencia ($x_i = 0$) de la oración s_i en el resumen generado \bar{D} . De esta forma, la representación de la solución (el vector de decisión) es la siguiente: $x = (x_1, x_2, \dots, x_n)$.

El primer objetivo a optimizar, $\Phi_{cobertura}(x)$, se refiere al criterio de la cobertura del contenido. Dada cada oración $s_i \in \bar{D}$, la cobertura del contenido se expresa en base a la similitud entre s_i y el conjunto de oraciones de D (representado por o). Por lo tanto, la siguiente función debe ser maximizada:

$$\Phi_{cobertura}(x) = \sum_{i=1}^n sim(s_i, o) \cdot x_i. \quad (4)$$

El segundo objetivo a optimizar, $\Phi_{red_redund}(x)$, se refiere a la redundancia de la información. En este caso, se define una variable de decisión binaria y_{ij} relacionada con la presencia simultánea ($y_{ij} = 1$) o la ausencia ($y_{ij} = 0$) de las oraciones s_i y s_j en el resumen generado \bar{D} . Para cada par de oraciones $s_i, s_j \in \bar{D}$ la similitud $sim(s_i, s_j)$ debe ser minimizada. Esto es equivalente a maximizar la reducción de la redundancia, esto es, a maximizar la siguiente función:

$$\Phi_{red_redund}(x) = \frac{1}{(\sum_{i=1}^{n-1} \sum_{j=i+1}^n sim(s_i, s_j) \cdot y_{ij}) \cdot \sum_{i=1}^n x_i}. \quad (5)$$

Finalmente, el problema de optimización multiobjetivo del resumen extractivo multidocumento se formula como:

$$\text{máx } \Phi(x) = \{\Phi_{cobertura}(x), \Phi_{red_redund}(x)\}, \quad (6)$$

$$\text{sujeto a } L - \varepsilon \leq \sum_{i=1}^n l_i \cdot x_i \leq L + \varepsilon, \quad (7)$$

donde l_i es la longitud de la oración s_i y ε es la tolerancia para la restricción de longitud del resumen, definida como:

$$\varepsilon = \max_{i=1,2,\dots,n} l_i - \min_{i=1,2,\dots,n} l_i. \quad (8)$$

IV. ABC MULTI OBJETIVO

El algoritmo ABC (*Artificial Bee Colony*) es un algoritmo de optimización basado en población que se fundamenta en el comportamiento inteligente de un enjambre o colonia de abejas [23]. El ABC simula el comportamiento de las abejas de la miel para resolver problemas de optimización. Principalmente, existen tres tipos de abejas: las abejas obreras, que mantienen las soluciones actualmente conocidas del problema; las abejas observadoras, que permiten la explotación de las mejores

soluciones encontradas hasta ahora; y las abejas exploradoras, que se encargan de la exploración de nuevas soluciones cuando algunas de las soluciones actuales se agotan (cuando no se pueden mejorar más).

Este algoritmo ha sido aplicado con éxito para resolver problemas del mundo real en múltiples contextos, ver p. ej. [24] y [25]. Además, algunos autores han propuesto nuevas variantes con el fin de mejorarlo (p. ej. [26] y [27]).

En esta sección se propone un algoritmo MOABC. Antes de describirlo, se explica el preprocesamiento de los documentos de entrada y los operadores de mutación y reparación.

IV-A. Preprocesamiento de entrada

Antes de la ejecución del algoritmo, los documentos de D deben ser preprocesados siguiendo estos pasos:

1. Segmentación. Las oraciones de D se extraen por separado para marcar el comienzo y el fin de cada una.
2. Tokenización. Las palabras (términos) de cada oración son separadas token a token. Los signos de puntuación, interrogación, exclamación... se eliminan en este paso.
3. Eliminación de las palabras vacías. Las palabras vacías de cada oración son eliminadas. Estas palabras son aquellas que carecen de significado principal, como artículos, preposiciones, conjunciones, etc. La lista de palabras vacías usada está proporcionada en el paquete ROUGE, y contiene 598 palabras [28].
4. *Stemming* del resto de palabras. Las raíces del resto de palabras son extraídas mediante el algoritmo de Porter [29], el cual ha sido ampliamente utilizado, convirtiéndose en el estándar para los métodos de análisis lexicográfico y para la recuperación de información en un amplio rango de lenguajes [30].

IV-B. Operador de mutación

La operación de mutación consiste en agregar o eliminar oraciones en el resumen correspondiente. Este operador se rige por la probabilidad de mutación $p_m \in (0, 1)$. Para cada oración en una solución x se genera un número aleatorio $r_i \sim U(0, 1)$. Si $r_i \leq p_m$, la oración s_i es candidata para la mutación, y si se cumple la siguiente condición:

$$sim(s_i, o) \geq \frac{1}{n} \sum_{j=1}^n sim(s_j, o), \quad (9)$$

s_i se incluye en el resumen. De lo contrario, se elimina. Esta condición verifica si la similitud entre s_i y o es mayor o menor que la media de la similitud de las oraciones de D .

IV-C. Operador de reparación

El operador de reparación comprueba que el resumen generado no viola la restricción de longitud definida en la Ecuación 7. Antes de finalizar cada ciclo, cualquier resumen generado puede violar dicha restricción, la cual es verificada en ambas direcciones. Si el resumen generado tiene una longitud inferior a la restricción menos la tolerancia, el resumen es descartado, ya que el número de resúmenes generados en este caso es muy bajo (se ha verificado experimentalmente que estos

resúmenes se producen 6 veces menos que los resúmenes con una longitud superior a la restricción más la tolerancia). Por otro lado, si el resumen generado tiene una longitud superior a la restricción más la tolerancia, este resumen sí es reparado como se explica a continuación.

Dado el resumen generado D^* , que incumple la restricción de longitud, la operación de reparación elimina de D^* la o las oraciones más redundantes (las que tienen un alto grado de similitud entre ellas). Para esto se considera un umbral de similitud $\delta = 0,9$ (una concordancia del 90%, también usada en [6]). Las siguientes condiciones son comprobadas para cada par de oraciones s_i y s_j :

$$\{s_i, s_j \in D^*\} \wedge \{i \neq j\} \wedge \text{sim}(s_i, s_j) \geq \delta \quad i, j = 1, 2, \dots, n. \quad (10)$$

Si estas condiciones se cumplen, entonces se usa el operador de reparación, eliminando la peor oración. Para ello, se calcula el siguiente valor de *calidad* para s_i y s_j :

$$\text{calidad}_{s_i} = \text{sim}(s_i, o) + ((\text{sim}(o^{\text{sum}}, o) - \text{sim}(o^{\text{sum}-s_i}, o)) \cdot 10, \quad (11)$$

donde $\text{sim}(o^{\text{sum}}, o)$ es la similitud entre el centro del resumen generado (incluida la oración s_i) y el centro de la colección de documentos o , y $\text{sim}(o^{\text{sum}-s_i}, o)$ es la similitud entre el centro del resumen generado (excluyendo en este caso la oración s_i) y el centro de la colección de documentos o . Este segundo término tiene un orden de magnitud mayor (se multiplica por 10), ya que mide la calidad del resumen cuando se elimina la oración s_i . Finalmente, la oración con la *calidad* más baja es eliminada del resumen, y este operador es aplicado hasta que la restricción de longitud requerida se cumple.

IV-D. Principales pasos del algoritmo

En esta subsección se propone una adaptación del algoritmo ABC para optimización multiobjetivo. Mientras que la optimización mono-objetivo devuelve la mejor solución encontrada, la optimización multiobjetivo devuelve un conjunto de soluciones que son no dominadas entre sí. Una solución es no dominada cuando ninguna de las funciones objetivo se puede mejorar sin que se degrade alguna de las otras [31]. El conjunto de soluciones no dominadas se conoce como conjunto de Pareto, y su representación gráfica como frente de Pareto.

El algoritmo MOABC se resume con el pseudocódigo del Algoritmo 1, que contiene los pasos principales del mismo.

En primer lugar, se inicializa el archivo de almacenamiento *Archivo_NDS*, que contiene las soluciones no dominadas (línea 1). Después, la colonia inicial se genera aleatoriamente (línea 2), es decir, para cada solución (resumen), las oraciones de D se seleccionan de forma aleatoria. Los pasos de las líneas 3 a 11 se repiten durante un número máximo de ciclos $\text{ciclos}_{\text{max}}$.

En el paso de las abejas obreras (línea 4) se aplica una mutación para mejorar la solución, que será seleccionada solo si domina a la original. En la línea 5 se utilizan dos operadores para determinar cuáles son las mejores soluciones asociadas a las abejas obreras: *rank* y *crowding*. El primero clasifica las soluciones en diferentes frentes de Pareto según

Algoritmo 1 Pseudocódigo del MOABC.

```

1: Archivo_NDS ← ∅
2: inicializar_colonia()
3: for ciclo = 1 to ciclos_max do
4:   enviar_abejas_obreras()
5:   rank_y_crowding(colonia_tam)
6:   calcular_probabilidades()
7:   enviar_abejas_observadoras()
8:   enviar_abejas_exploradoras()
9:   rank_y_crowding(2 * colonia_tam)
10:  exportar_colonia(Archivo_NDS)
11: end for

```

sus dominancias, mientras que el segundo da preferencia a las soluciones que aportan mayor diferenciación (a las más diversas) [32]. Una vez aplicados estos operadores, la selección se basa en las probabilidades calculadas en la línea 6, asignando mayores probabilidades a las mejores soluciones. En la línea 7 se envían las abejas observadoras, cuya tarea es seleccionar su abeja obrera asignada, teniendo en cuenta las probabilidades de selección, para tratar de mejorarla (el operador de mutación también es aplicado en este paso). En este punto del algoritmo, el tamaño de la colonia se duplica ($2 * \text{colonia}_{\text{tam}}$). Las últimas abejas, las exploradoras (línea 8), se encargan de verificar las soluciones agotadas, que son aquellas que no han mejorado más después de un determinado número de intentos. Estas abejas agotadas son reemplazadas por exploradoras, que son nuevas soluciones generadas aleatoriamente. Para finalizar un ciclo, el tamaño de la colonia se reduce al tamaño original ($\text{colonia}_{\text{tam}}$), aplicando los operadores *rank* y *crowding* nuevamente (línea 9), y se reparan las soluciones (cuando es necesario) para su almacenamiento en el archivo *Archivo_NDS* (línea 10).

V. RESULTADOS EXPERIMENTALES

V-A. Conjuntos de datos

Los conjuntos de datos utilizados para medir el rendimiento han sido proporcionados por DUC (*Document Understanding Conferences*), siendo un banco de pruebas abierto de referencia para la evaluación de resúmenes automáticos. Las comparaciones se han realizado con los trabajos [5] y [6], donde se han utilizado 10 temas del conjunto de datos DUC2002 (del d061j al d070f) [33].

V-B. Métricas de evaluación

El rendimiento del modelo se ha evaluado mediante el uso de las métricas ROUGE (*Recall-Oriented Understudy for Gisting Evaluation*) [34], que es considerada la métrica de evaluación oficial para los resúmenes por DUC. En este caso, las dos variantes de ROUGE usadas son: ROUGE-2 y ROUGE-L. Además de la media (M), se han considerado otras dos estadísticas para la medición de la dispersión: el rango y un coeficiente de variación estadístico. El rango (R) se calcula como $R = \text{ROUGE}_{\text{mejor}} - \text{ROUGE}_{\text{peor}}$. Sin embargo, no es adimensional, por lo que también se muestra una modificación



Tabla I

MEDIAS (M), RANGOS (R) Y COEFICIENTES DE VARIACIÓN (CV) DE LOS VALORES ROUGE-2 Y ROUGE-L. LOS MEJORES APARECEN SOMBREADOS.

Tema	ROUGE-2									ROUGE-L								
	DE adaptativo [5]			NSGA-II [6]			Enfoque propuesto			DE adaptativo [5]			NSGA-II [6]			Enfoque propuesto		
	M	R	CV	M	R	CV	M	R	CV	M	R	CV	M	R	CV	M	R	CV
d061j	0,266	0,290	109,02	0,306	0,263	85,95	0,365	0,093	25,43	0,542	0,208	38,38	0,554	0,205	37,00	0,590	0,056	9,46
d062j	0,188	0,275	146,28	0,200	0,422	211,00	0,342	0,023	6,60	0,473	0,239	50,53	0,481	0,306	63,62	0,536	0,017	3,26
d063j	0,245	0,208	84,90	0,275	0,279	101,45	0,272	0,005	1,84	0,493	0,156	31,64	0,528	0,171	32,39	0,509	0,050	9,74
d064j	0,194	0,280	144,33	0,233	0,356	152,79	0,308	0,009	2,83	0,462	0,235	50,87	0,488	0,287	58,81	0,495	0,011	2,19
d065j	0,144	0,209	145,14	0,182	0,208	114,29	0,198	0,026	13,32	0,431	0,141	32,71	0,457	0,174	38,07	0,464	0,057	12,29
d066j	0,201	0,257	127,86	0,181	0,245	135,36	0,290	0,019	6,54	0,455	0,196	43,08	0,441	0,149	33,79	0,519	0,007	1,36
d067f	0,239	0,235	98,33	0,260	0,298	114,62	0,356	0,005	1,39	0,509	0,232	45,58	0,529	0,244	46,12	0,580	0,012	2,15
d068f	0,491	0,384	78,21	0,496	0,281	56,65	0,444	0,084	18,83	0,666	0,226	33,93	0,626	0,226	36,10	0,639	0,071	11,09
d069f	0,184	0,166	90,22	0,232	0,239	103,02	0,240	0,008	3,15	0,454	0,135	29,74	0,476	0,191	40,13	0,554	0,010	1,81
d070f	0,224	0,260	116,07	0,262	0,215	82,03	0,305	0,002	0,76	0,496	0,173	34,88	0,513	0,158	30,80	0,515	0,005	0,90
Media	0,238	0,256	114,03	0,263	0,281	115,72	0,312	0,027	8,07	0,498	0,194	39,13	0,509	0,211	41,68	0,540	0,030	5,43

del coeficiente de variación tradicional de Pearson (CV), que implica la relación entre el rango y la media. Puede expresarse en términos de porcentaje como:

$$CV = \frac{R}{ROUGE_{media}} \cdot 100. \quad (12)$$

V-C. Configuración de los experimentos

Los parámetros del MOABC se han elegido para realizar comparaciones justas con los resultados de los enfoques propuestos en la literatura científica ([5] y [6]). Estos son: tamaño de colonia, $colonia_{tam} = 50$; número de ciclos, $ciclos_{max} = 1000$; probabilidad de mutación, $p_m = 0,1$; y el número de repeticiones/ejecuciones independientes, $reps_{max} = 20$.

V-D. Resultados

Los resultados del MOABC son comparados con [6] (el único enfoque multiobjetivo existente en la literatura científica) y con [5] (el mejor enfoque mono-objetivo). La Tabla I presenta la media, el rango y el CV de los valores de ROUGE-2 y ROUGE-L para cada tema y los valores medios de todos los temas para los tres enfoques comparados.

En los resultados obtenidos en la Tabla I se puede observar que el enfoque propuesto supera a los otros dos, mejorando a [5] en 9 de los 10 temas y a [6] en 8 de 10 para el ROUGE-2, y en 9 de los 10 temas a ambos para el ROUGE-L. Además, los rangos y los CV muestran que los resultados del modelo propuesto son muy robustos. Considerando la media de los diez temas, para el ROUGE-2 el enfoque propuesto produce resultados con un CV medio de 8,07%, mientras que en los otros dos enfoques son de 114,03% y 115,72%. Para el ROUGE-L, mientras que el CV está alrededor del 40% en los otros enfoques, en el modelo propuesto el valor es solo del 5,43%. Finalmente, la Tabla II muestra las mejoras en términos promedio del enfoque propuesto con respecto a los otros enfoques.

Los resultados de la Tabla II dan lugar a las siguientes observaciones. En primer lugar, el enfoque propuesto mejora al presentado en [5], donde el algoritmo es un DE adaptativo mono-objetivo. La media del ROUGE-2 es mejorada en un

Tabla II

COMPARACIÓN DE LA MEDIA (M), DEL RANGO (R) Y DEL COEFICIENTE DE VARIACIÓN (CV) ENTRE EL ENFOQUE PROPUESTO Y LOS OTROS ENFOQUES.

Enfoque	Mejora del enfoque propuesto (%)					
	ROUGE-2			ROUGE-L		
	M	R	CV	M	R	CV
DE adaptativo [5]	31,09	848,15	1313,01	8,43	546,67	620,63
NSGA-II [6]	18,63	940,74	1333,95	6,09	603,33	667,59

31,09%, y la del ROUGE-L en un 8,43%. En segundo lugar, el enfoque propuesto también supera al presentado en [6]. En este caso, el algoritmo es un NSGA-II multiobjetivo. La media del ROUGE-2 tiene una mejora del 18,63%, y la del ROUGE-L del 6,09%. Finalmente, los rangos y CV muestran que los resultados del enfoque propuesto son mucho más robustos. Para el ROUGE-2, el enfoque propuesto mejora la media de los CV en más de un 1300% (13 veces más robusto), y para el ROUGE-L mejoran en más de un 600% (6 veces más robusto). Esto significa que las soluciones del MOABC son más estables que las obtenidas en los otros dos enfoques.

VI. CONCLUSIONES Y TRABAJO FUTURO

El problema de resumen multidocumento requiere la optimización de más de una función objetivo, por lo que es necesario aplicar enfoques de optimización multiobjetivo. Por primera vez, un enfoque basado en el algoritmo MOABC ha sido diseñado e implementado para este caso. Los resultados obtenidos no solo han mejorado en los valores ROUGE-2 (31,09% y 18,63% mejor) y en ROUGE-L (8,43% y 6,09% mejor), sino que también han mostrado una menor dispersión (alrededor de 1300% y 600% menos, es decir, alrededor de 13 y 6 veces más robusto) y, por lo tanto, la prueba de que el enfoque propuesto es estadísticamente más sólido que los enfoques comparables publicados.

Como línea de investigación futura, el enfoque se adaptará

para su aplicación en el software NeuroK¹, que es una plataforma de *e-learning* basada en la neurodidáctica [35]. El algoritmo generará resúmenes de los contenidos de los estudiantes (mensajes, comentarios, observaciones o valoraciones), en unidades de aprendizaje o en actividades concretas, lo que será útil para los profesores de los cursos para muchos propósitos, incluida la evaluación automática para las calificaciones.

Otra línea de investigación interesante es el análisis del tiempo de CPU del algoritmo, con vistas a una posible paralelización para mejorar el tiempo de ejecución. Las técnicas basadas en programación paralela con OpenMP podrían ser muy útiles, ya que este enfoque permite explotar el paralelismo existente en las arquitecturas multinúcleo actuales.

AGRADECIMIENTOS

Esta investigación ha sido apoyada por el Ministerio de Economía y Competitividad (Centro para el Desarrollo Tecnológico Industrial, contrato IDI-20161039; Agencia Estatal de Investigación, proyectos TIN2016-76259-P y MTM2017-86875-C3-2-R), Junta de Extremadura (contrato AA-16-0017-1, y proyectos GR18108 y GR18090), Cátedra ASPgems y Unión Europea (Fondo Europeo de Desarrollo Regional).

REFERENCIAS

- [1] W. Fan y A. Bifet, "Mining big data: current status, and forecast to the future," *ACM SIGKDD Explorations Newsletter*, vol. 14, no. 2, 2013, pp. 1–5.
- [2] H. Hashimi, A. Hafez y H. Mathkour, "Selection criteria for text mining approaches," *Computers in Human Behavior*, vol. 51, 2015, pp. 729–733.
- [3] D. M. Zajic, B. J. Dorr y J. Lin, "Single-document and multi-document summarization techniques for email threads using sentence compression," *Information Processing & Management*, vol. 44, no. 4, 2008, pp. 1600–1610.
- [4] X. Wan, "An exploration of document impact on graph-based multi-document summarization," En *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2008, pp. 755–762.
- [5] R. M. Alguliev, R. M. Aliguliyev y C. A. Mehdiyev, "Sentence selection for generic document summarization using an adaptive differential evolution algorithm," *Swarm and Evolutionary Computation*, vol. 1, no. 4, 2011, pp. 213–222.
- [6] H. H. Saleh, N. J. Kadhim y B. A. Attea, "A Genetic Based Optimization Model for Extractive Multi-Document Text Summarization," *Iraqi Journal of Science*, vol. 56, no. 2, 2015 pp. 1489–1498.
- [7] L. Huang, Y. He, F. Wei y W. Li, "Modeling document summarization as multi-objective optimization," En *Intelligent Information Technology and Security Informatics (IITSI)*, 2010 Third International Symposium, IEEE, 2010, pp. 382–386.
- [8] R. M. Aliguliyev, "Clustering Techniques and Discrete Particle Swarm Optimization Algorithm for Multi-Document Summarization," *Computational Intelligence*, vol. 26, no. 4, 2010, pp. 420–448.
- [9] R. M. Alguliev, R. M. Aliguliyev, M. S. Hajirahimova y C. A. Mehdiyev, "MCMR: Maximum coverage and minimum redundant text summarization model," *Expert Systems with Applications*, vol. 38, no. 12, 2011, pp. 14514–14522.
- [10] R. M. Alguliev, R. M. Aliguliyev y C. A. Mehdiyev, "pSum-Sade: a modified p-median problem and self-adaptive differential evolution algorithm for text summarization," *Applied Computational Intelligence and Soft Computing*, vol. 2011, 2011, pp. 1–13.
- [11] R. M. Alguliev, R. M. Aliguliyev y N. R. Isazade, "DESAMC+ DocSum: Differential evolution with self-adaptive mutation and crossover parameters for multi-document summarization," *Knowledge-Based Systems*, vol. 36, 2012, pp. 21–38.
- [12] R. M. Alguliev, R. M. Aliguliyev y C. A. Mehdiyev, "An optimization model and DPSO-EDA for document summarization," *International Journal of Information Technology and Computer Science (IJITCS)*, vol. 3, no. 5, 2011, pp. 59–68.
- [13] R. M. Alguliev, R. M. Aliguliyev y N. R. Isazade, "Formulation of document summarization as a 0–1 nonlinear programming problem," *Computers & Industrial Engineering*, vol. 64, no. 1, 2013, pp. 94–102.
- [14] R. M. Alguliev, R. M. Aliguliyev y M. S. Hajirahimova, "Quadratic Boolean programming model and binary differential evolution algorithm for text summarization," *Problems of Information Technology*, vol. 3, no. 2, 2012, pp. 20–29.
- [15] R. M. Alguliev, R. M. Aliguliyev y M. S. Hajirahimova, "GenDocSum+MCLR: Generic document summarization based on maximum coverage and less redundancy," *Expert Systems with Applications*, vol. 39, no. 16, 2012, pp. 12460–12473.
- [16] R. M. Alguliev, R. M. Aliguliyev y N. R. Isazade, "CDDS: Constraint-driven document summarization models," *Expert Systems with Applications*, vol. 40, no. 2, 2013, pp. 458–465.
- [17] R. M. Alguliev, R. M. Aliguliyev y N. R. Isazade, "Multiple documents summarization based on evolutionary optimization algorithm," *Expert Systems with Applications*, vol. 40, no. 5, 2013, pp. 1675–1689.
- [18] R. M. Alguliev, R. M. Aliguliyev y C. A. Mehdiyev, "An optimization approach to automatic generic document summarization," *Computational Intelligence*, vol. 29, no. 1, 2013, pp. 129–155.
- [19] M. Mendoza, C. Cobos, E. Leon, M. Lozano, F. Rodriguez y E. Herrera-Viedma, "A new memetic algorithm for multi-document summarization based on CHC algorithm and greedy search," En *Mexican International Conference on Artificial Intelligence*, Springer, 2014, pp. 125–138.
- [20] R. M. Alguliev, R. M. Aliguliyev y N. R. Isazade, "An unsupervised approach to generating generic summaries of documents," *Applied Soft Computing*, vol. 34, 2015, pp. 236–250.
- [21] K. Umam, F. W. Putro, G. Q. O. Pratamasunu, A. Z. Arifin y D. Purwitasari, "Coverage, Diversity, and Coherence Optimization for Multi-Document Summarization," *Jurnal Ilmu Komputer dan Informasi*, vol. 8, no. 1, 2015, pp. 1–10.
- [22] G. Salton y C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing & Management*, vol. 24, no. 5, 1988, pp. 513–523.
- [23] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," *Tech. rep.*, Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
- [24] D. Karaboga y B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, 2007, pp. 459–471.
- [25] D. Karaboga, B. Gorkemli, C. Ozturk y N. Karaboga, "A comprehensive survey: artificial bee colony (ABC) algorithm and applications," *Artificial Intelligence Review*, vol. 42, no. 1, 2014, pp. 21–57.
- [26] M. S. Kiran y O. Findik, "A directed artificial bee colony algorithm," *Applied Soft Computing*, vol. 26, 2015, pp. 454–462.
- [27] M. S. Kiran, H. Hakli, M. Gunduz y H. Uguz, "Artificial bee colony algorithm with variable search strategy for continuous optimization," *Information Sciences* 300, 2015, pp. 140–157.
- [28] ROUGE Summary Evaluation Package, <http://www.berouge.com/>. [Accedido 20-Julio-2017].
- [29] Porter stemming algorithm, <http://www.tartarus.org/martin/PorterStemmer/>. [Accedido 14-Junio-2018].
- [30] P. Willett, "The Porter stemming algorithm: then and now," *Program*, vol. 40, no. 3, 2006, pp. 219–223.
- [31] C. C. Coello, C. Dhaenens y L. Jourdan, "Advances in multi-objective nature inspired computing," *SCI*, vol. 272, Springer, 2010.
- [32] K. Deb, A. Pratap, S. Agarwal y T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, 2002, pp. 182–197.
- [33] Document Understanding Conference, <http://duc.nist.gov>. [Accedido 14-Junio-2018].
- [34] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," En *Text summarization branches out: Proceedings of the ACL-04 workshop*, vol. 8, Barcelona, Spain, 2004, pp. 74–81.
- [35] F. Calle-Alonso, A. Cuenca-Guevara, D. de la Mata Lara, J. M. Sanchez-Gomez, M. A. Vega-Rodríguez y C. J. Perez Sanchez, "NeuroK: A Collaborative e-Learning Platform based on Pedagogical Principles from Neuroscience," En *Proceedings of the 9th International Conference on Computer Supported Education (CSEDU 2017)*, vol. 1, Science and Technology Publications, 2017, pp. 550–555.

¹<https://neurok.es/>



Algoritmo Multiobjetivo de Colonia de Abejas Artificiales aplicado al Problema de Orientación

Rodrigo Martín-Moreno

*Depto. Tecnología de Computadores y Comunicaciones
Universidad de Extremadura
Cáceres, España
rmartinky@alumnos.unex.es*

Miguel A. Vega-Rodríguez

*Depto. Tecnología de Computadores y Comunicaciones
Universidad de Extremadura
Cáceres, España
mavega@unex.es*

Resumen—Hemos desarrollado un nuevo algoritmo para la resolución de problemas de orientación multiobjetivo, que tienen aplicación directa en el cálculo de rutas óptimas, logística, o bien el diseño de rutas turísticas, nuestra motivación principal. Los turistas priorizan según sus gustos el visitar puntos de interés, los cuales se pueden categorizar (p. ej. culturales u ocio), por lo que es necesaria la utilización de sistemas expertos multiobjetivo. Para conseguir las mejores soluciones de Pareto, hemos adaptado el algoritmo de colonia de abejas artificiales al contexto multiobjetivo. El rendimiento del algoritmo se ha comparado con dos algoritmos usados previamente en la literatura de los problemas de orientación multiobjetivo (P-ACO y P-VNS), y los resultados obtenidos indican que este nuevo algoritmo es acertado para resolver problemas de orientación multiobjetivo.

Palabras clave—Algoritmo de colonia de abejas artificiales, Optimización multiobjetivo, Problema de orientación, Inteligencia de enjambre, Computación evolutiva.

I. INTRODUCCIÓN

Existe una gran variedad de sistemas de apoyo a la toma de decisiones, usados para diferentes propósitos como marketing, finanzas, logística o recursos humanos, por ejemplo. Sin embargo, existen pocos sistemas para ayudar a la gente a diseñar rutas turísticas que encajen en sus preferencias, más bien se les ofrece un conjunto de rutas prefijadas poco personalizables. Cuando los visitantes planean una visita a un destino, priorizan qué lugares o puntos de interés (POIs) merecen ser visitados, teniendo en cuenta su presupuesto, tiempo e interés y decidiendo su orden en la ruta. Por ello, un sistema de apoyo a la toma de decisiones, que ayude a los visitantes en este proceso, puede ser muy interesante. En todo caso, la decisión final recaerá siempre sobre el visitante, ya que el sistema solo ofrecerá las mejores soluciones para sus requerimientos, pero no contemplará la parte emocional de planear una visita a un destino.

El hecho de crear rutas que conecten POIs se define como un Problema de Orientación (OP) [1]. En este caso, nos centramos en el Problema de Orientación MultiObjetivo (MOOP), donde existen varias categorías para cada POI (p. ej.: cultural u ocio) y cada uno de ellos tiene diferentes beneficios para cada categoría. Hemos desarrollado un algoritmo MultiObjetivo de Colonia de Abejas Artificiales (MOABC), basándonos en el algoritmo mono-objetivo ABC propuesto por [2], para resolver el MOOP de una manera competitiva. Es la primera vez que

el algoritmo MOABC se aplica en la resolución de problemas de orientación multiobjetivo. Como veremos, los resultados obtenidos son muy competitivos al compararlos con los de otros algoritmos multiobjetivo (P-ACO y P-VNS) del estado del arte en este campo. Los experimentos han sido realizados en instancias de test e instancias del mundo real (un total de 216 instancias agrupadas en 10 conjuntos), y hemos usado tres indicadores de calidad para comparar los resultados.

Este artículo se estructura de la siguiente manera. La sección II introduce al lector en la investigación desarrollada en este campo. La sección III presenta la definición formal del problema y su formulación matemática. La sección IV desgana nuestra estrategia multiobjetivo para resolver el problema. La sección V muestra los resultados obtenidos y compara los resultados de nuestro algoritmo con los de otros algoritmos previamente publicados. Por último, la sección VI culmina este artículo y muestra también posibles líneas de trabajo futuras.

II. TRABAJO RELACIONADO

La optimización multiobjetivo ha sido un campo importante, con bastantes trabajos de investigación, en las dos últimas décadas. Dentro de ella, la resolución de MOOP es una área de gran aplicabilidad, aunque con poca actividad de investigación.

Hemos basado nuestro trabajo en el algoritmo ABC y lo hemos adaptado al contexto multiobjetivo. La elección del ABC se debe a que ha sido extensamente estudiado y aplicado para resolver problemas reales en múltiples campos [3], incluyendo optimización mono-objetivo (p. ej. [4], [5] y [6]) y optimización multiobjetivo (p. ej. [7], [8] y [9]).

Por otro lado, OP fue definido por [1], y algunos autores lo han considerado como un tipo de TSP (Problema del Viajante) con beneficios (ver [10]) o TSP selectivo. En OP, cada vértice tiene asociado algún beneficio, y la finalidad es visitar un grupo de vértices que maximice la suma de los beneficios, mientras se satisfaga la restricción de coste/longitud del tour. Dos estudios completos del estado del arte del problema de orientación se pueden encontrar en [11] y [12].

Relacionado con OP está el problema de orientación de equipo o TOP (ver [13] y [14]), donde el problema se extiende a múltiples tours.

Con respecto a MOOP, el problema abordado en este trabajo, muy pocas propuestas se pueden encontrar en la

literatura. Dentro del problema de orientación biobjetivo podemos destacar [15] que usó el algoritmo P-ACO (*Pareto Ant Colony Optimization*) y el algoritmo P-VNS (*Pareto Variable Neighborhood Search*) combinados con PR (*Path Relinking*), [16] que usó un algoritmo evolutivo también combinado con PR y [17] que utilizó GRASP combinado con PR.

III. DEFINICIÓN DEL PROBLEMA MOOP

Puede modelarse como un grafo dirigido $G = (V, A)$ con un conjunto de vértices, $V = \{v_0, v_1, v_2, \dots, v_{n+1}\}$, y un conjunto de aristas, $A = \{(v_i, v_j) : v_i, v_j \in V \wedge v_i \neq v_j \wedge v_i \neq v_{n+1} \wedge v_j \neq v_0\}$. Cada vértice $v_i \in V \setminus \{v_0, v_{n+1}\}$ tiene asociados K beneficios b_{ik} ($k = 1, \dots, K$). Los vértices inicial y final, v_0 y v_{n+1} , no tienen beneficios asociados. Además, cada arista $(v_i, v_j) \in A$ tiene un coste c_{ij} que puede ser interpretado como distancia, dinero o tiempo invertido para ir desde v_i hasta v_j .

En todas las instancias usadas en este trabajo, v_0 y v_{n+1} representan el mismo punto, por lo tanto, entendemos que cada solución es un “tour” en lugar de “ruta”. La finalidad del problema de orientación multiobjetivo es encontrar los mejores tours (que maximicen los beneficios para todos los objetivos) desde v_0 hasta v_{n+1} , cumpliendo la restricción marcada de coste/longitud C_{max} .

Podemos definir matemáticamente el problema como:

$$\text{maximizar } F(x) = (f_1(x), \dots, f_K(x)), \quad (1)$$

$$f_k = \sum_{v_i \in V \setminus \{v_0, v_{n+1}\}} (b_{ik} \cdot y_i) \quad (k = 1, \dots, K), \quad (2)$$

donde y_i , una variable binaria, toma valor 1 cuando v_i es visitado, y 0 en caso contrario. Además, hay que tener en cuenta que:

$$\sum_{v_i \in V \setminus \{v_j\}} a_{ij} = y_j \quad (v_j \in V \setminus \{v_0\}), \quad (3)$$

$$\sum_{v_j \in V \setminus \{v_i\}} a_{ij} = y_i \quad (v_i \in V \setminus \{v_{n+1}\}), \quad (4)$$

$$\sum_{\{v_i, v_j\} \in S} a_{ij} \leq |S| - 1 \quad (S \subseteq V \wedge S \neq \emptyset), \quad (5)$$

$$y_0 = y_{n+1} = 1, \quad (6)$$

$$\sum_{(v_i, v_j) \in A} c_{ij} \cdot a_{ij} \leq C_{max}, \quad (7)$$

$$a_{ij} \in \{0, 1\} \quad \left((v_i, v_j) \in A \right), \quad (8)$$

$$y_i \in \{0, 1\} \quad (v_i \in V). \quad (9)$$

La variable binaria a_{ij} toma valor 1 cuando $(v_i, v_j) \in A$ es usado, y 0 en caso contrario. La ecuación 1 indica que para resolver MOOP hay que maximizar las diferentes funciones objetivo. La ecuación 2 define cada función objetivo como la suma de sus beneficios correspondientes. Las ecuaciones 3 y 4 implican que para cada vértice visitado solo existe una arista entrante y una arista saliente. La ecuación 5 impide subtours. La ecuación 6 implica que los puntos de comienzo y fin se

usan en todos los tours. La ecuación 7 garantiza que el coste del tour no es superior al límite establecido C_{max} . En este caso, resolveremos el problema de orientación biobjetivo, por lo que $K = 2$.

IV. SOLUCIÓN. NUEVO ALGORITMO MULTI OBJETIVO

En [18] se demostró que OP es NP-hard. Por ello, es necesario aplicar técnicas metaheurísticas en la resolución del problema de orientación multiobjetivo.

En esta sección explicamos el algoritmo MOABC que hemos diseñado y desarrollado. Inicialmente el algoritmo Colonia de Abejas Artificiales fue propuesto por [2] en el contexto mono-objetivo. Nosotros lo hemos adaptado al contexto multiobjetivo y particularmente a la resolución del problema de orientación multiobjetivo. Modelamos cada solución en MOOP como una lista de puntos (aquellos que se han incluido en el tour correspondiente), lo cual es la manera más natural para representar una solución para este problema.

IV-A. Optimización Multiobjetivo

Debido a la naturaleza multiobjetivo del problema a resolver, es muy difícil elegir una solución óptima donde todos los objetivos se maximicen. Sin embargo, si restringimos a soluciones no dominadas, la elección se limitará a un conjunto razonable de soluciones candidatas. Las siguientes definiciones ayudarán a clarificar este aspecto.

Una solución x domina a una solución x' si x no es peor que x' en ninguna de las funciones objetivo, y es mejor en al menos una de las funciones objetivo. Formalmente: para maximizar $F(x) = (f_1(x), \dots, f_K(x))$, x domina x' si $f_k(x) \geq f_k(x')$ para todo $k = 1, \dots, K$, y $f_k(x) > f_k(x')$ para al menos un k . Si esto ocurre, escribimos $x \succ x'$.

Si ninguna solución domina a la solución x^* , decimos que x^* es no dominada o Pareto eficiente. En este caso, decimos que $z^* = F(x^*) = (f_1(x^*), \dots, f_K(x^*))$ es un vector no dominado. El conjunto de todos los vectores no dominados es llamado frente de Pareto. La relación \succ puede ser extendida desde el espacio de soluciones al espacio de objetivos. En ese caso, dados dos vectores $z = (z_1, \dots, z_K)$ y $z' = (z'_1, \dots, z'_K)$, escribimos $z \succ z'$ si $z_k \geq z'_k$ para todo $k = 1, \dots, K$ y $z_k > z'_k$ en al menos un k .

IV-B. MOABC

El algoritmo ABC es un algoritmo metaheurístico, basado en inteligencia de enjambre, que se inspira en el comportamiento alimenticio de las colonias de abejas melíferas y se compone de tres componentes principales: abejas obreras, abejas observadoras y abejas exploradoras. Las abejas obreras buscan fuentes de alimento (soluciones). Existen otros dos comportamientos interesantes que se requieren para la auto-organización y la inteligencia de enjambre: enviar nuevas abejas (abejas observadoras) hacia las fuentes de alimento prometedoras (feedback positivo) y abandonar fuentes de alimento agotadas (feedback negativo), generando abejas exploradoras.

Como hemos mencionado anteriormente, hemos adaptado el algoritmo ABC original al contexto multiobjetivo y en particular a la resolución de MOOP. Un pseudocódigo de nuestro



algoritmo MOABC se muestra en algoritmo 1. Las siguientes subsecciones detallan todas las partes de este algoritmo.

Algoritmo 1 Colonia de Abejas Artificiales Multiobjetivo

entrada: T_{max} (tiempo máximo de ejecución), TP (tamaño de la población) y $limite$ (criterio de abandono)

salida: $Fichero_SND$ (fichero con las soluciones no dominadas)

```

1:  $t \leftarrow 0$ ;  $Fichero\_SND \leftarrow \emptyset$ 
2: IniciarFuentesAlimento( $TP$ )           ▷ ver IV-C
3: while  $t < T_{max}$  do
4:   FaseAbejasObreras( $TP$ )             ▷ ver IV-D
5:   CalcularCantidadNectar( $TP$ )       ▷ ver IV-E
6:   FaseAbejasObservadoras( $TP$ )      ▷ ver IV-F
7:   FaseAbejasExploradoras( $TP$ ,  $limite$ ) ▷ ver IV-G
8:   GuardarSND( $Fichero\_SND$ )        ▷ ver IV-H
9: end while

```

IV-C. Iniciar Fuentes de Alimento

Todas las soluciones (fuentes de alimento) de la población (TP : tamaño de la población) se generan aleatoriamente por las abejas obreras siguiendo estos pasos: primero calculamos los 20 mejores movimientos desde cada punto a cualquier otro. Este cálculo solo se realiza una vez y se usa cuando sea necesario. Los mejores movimientos son calculados siguiendo la ecuación 10. Para ir desde i a j se calcula el $ratio_{ij}$, donde b_{j1} y b_{j2} son los beneficios del punto j y c_{ij} es el coste para ir desde i hasta j . Por lo tanto, la ecuación 10 relaciona los beneficios con el coste. Cabe destacar que los valores b_{j1} , b_{j2} y c_{ij} son normalizados para evitar sesgos. Los movimientos con mejor ratio se consideran mejores movimientos.

$$ratio_{ij} = \frac{(b_{j1} + b_{j2})}{c_{ij}}. \quad (10)$$

Después, comenzando por el punto inicial, seleccionamos un movimiento de la lista de 20 movimientos. Este proceso se repite hasta que no se pueden añadir más puntos por la restricción de coste C_{max} . Indicar que todos los tours finalizan en el punto de fin y que el resto de restricciones son también comprobadas durante la generación aleatoria de una solución.

IV-D. Fase Abejas Obreras

Las abejas obreras buscan mejores soluciones, dentro de sus vecindarios. Esto consiste en encontrar una solución vecina y evaluar su calidad. Para obtener soluciones vecinas para las abejas obreras, usamos dos operadores: acortamiento e inserción. El acortamiento intenta reordenar los vértices, dentro de un tour, para minimizar su coste. En este caso usamos el operador de acortamiento 2-opt propuesto por [19]. La inserción comprueba si es factible insertar un nuevo punto después de realizar el acortamiento. Para cada posible inserción, se elige de manera aleatoria uno de los mejores 10 puntos candidatos (según ecuación 10) de los puntos no visitados. Cualquier posible inserción se realiza teniendo en cuenta la restricción de coste C_{max} . Después de generar la solución vecina, se

compara con la solución actual y si la primera domina a la segunda, entonces se reemplaza la solución actual para la abeja obrera.

IV-E. Calcular Cantidad de Néctar

Usamos dos operadores multiobjetivo para determinar cuáles son las mejores soluciones asociadas a las abejas obreras: *rank* y *crowding*. El primero clasifica las soluciones en los diferentes frentes de Pareto acorde a sus relaciones de dominancia, mientras que el segundo tiene en cuenta la distancia de amontonamiento, que consigue variedad en las soluciones. Una explicación más detallada de estos dos operadores puede encontrarse en [20]. Una vez calculados, los combinamos para cada solución en un único valor (llamado fitness multiobjetivo o *MOfitness*), según la ecuación 11. *MOfitness* es un valor importante porque un mayor fitness multiobjetivo implicará una mayor probabilidad de selección, fundamental en la siguiente subsección.

$$MOfitness(x) = \frac{1}{2^{rank(x)} + \frac{1}{1 + crowding(x)}}. \quad (11)$$

IV-F. Fase Abejas Observadoras

Las abejas obreras comparten la información del fitness de sus soluciones con las abejas observadoras, y estas eligen sus soluciones de manera probabilística basándose en las probabilidades calculadas en base a los valores *MOfitness* (ver ecuación 11). Aplicamos el método de selección de ruleta basado en fitness propuesto por [21]. Por ello, las mejores soluciones reclutarán más abejas observadoras (feedback positivo).

La probabilidad $p(x^i)$ por la que una solución x^i es elegida por una abeja observadora se calcula con la ecuación 12.

$$p(x^i) = \frac{MOfitness(x^i)}{\sum_{m=1}^{TP} MOfitness(x^m)}. \quad (12)$$

Después de que una solución x^i sea probabilísticamente elegida por una abeja observadora, se genera una solución vecina usando dos operadores: intercambio de vértices y acortamiento. Intercambio de vértices intenta reemplazar cada punto existente del tour por un punto no visitado. En cada posición, el punto no visitado se selecciona a través de los mejores 10 puntos (de acuerdo a la ecuación 10) no visitados para esa posición. Después, se usa la misma operación de acortamiento de la subsección IV-D para mejorar la solución.

Tras generar la solución vecina, se compara con la solución original y si la primera no es dominada por la segunda, entonces se reemplaza la solución original para la abeja observadora.

IV-G. Fase Abejas Exploradoras

Las abejas obreras y observadoras cuyas soluciones no pueden ser mejoradas después de un número de intentos, indicado por un parámetro de configuración del algoritmo MOABC llamado *limite*, se convierten en abejas exploradoras y sus soluciones son abandonadas. Destacar que MOABC utiliza más abejas exploradoras que el algoritmo ABC original, donde se utiliza

como mucho una por iteración [22]. En MOABC todas las abejas obreras y observadoras con intentos $\geq \text{limite}$ se convierten en exploradoras. En todo caso, las comparaciones con P-ACO y P-VNS (subsección V-C) son justas ya que se basan en la utilización del mismo tiempo de ejecución. Las abejas exploradoras buscan de manera aleatoria nuevas soluciones (mejorando la exploración del algoritmo). Por lo tanto, aquellas soluciones que no pueden ser mejoradas por explotación, son abandonadas (feedback negativo). La nueva solución asociada a la abeja exploradora se genera aleatoriamente de la misma manera que en la subsección IV-C. Posteriormente, la abeja exploradora intenta mejorar la solución aplicando los tres operadores vistos anteriormente: inserción, intercambio y acortamiento.

IV-H. Guardar Soluciones No Dominadas

En cada iteración, las soluciones no dominadas se guardan en un fichero para preservarlas mientras se generan nuevas soluciones. El operador utilizado para calcular las soluciones no dominadas es el *rank*, el cual clasifica las soluciones en diferentes frentes de Pareto de acuerdo a sus relaciones de dominancia. Además se usa el operador *crowding* que junto con el operador *rank* es usado para ordenar las soluciones actuales. Una explicación más detallada de estos operadores puede encontrarse en [20]. Después, las mejores *TP* (tamaño de la población) soluciones son tomadas por las abejas obreras para la siguiente iteración y las soluciones no dominadas (con $\text{rank} = 0$) se guardan en *Fichero_SND*.

V. RESULTADOS

Hemos implementado nuestro algoritmo MOABC en C++, con el compilador GNU g++ 4.8.4 en Ubuntu Server 14.04 LTS 64 bits. Todas las ejecuciones se han realizado en un procesador Intel Pentium 4 D de 3,2 GHz y 1 GB de RAM, un entorno computacional muy similar al usado para los algoritmos P-ACO y P-VNS.

V-A. Instancias de Comparación

Hemos utilizado las mismas instancias que fueron usadas por [15] y que se encuentran disponibles en <http://prolog.univie.ac.at/research/OP>. Los conjuntos de instancias se pueden categorizar en 10 grupos, de acuerdo al número de vértices. En cada grupo existen varias instancias con distinta restricción de longitud máxima de tour (C_{max}). Los primeros 3 grupos (2_p21, 2_p32 y 2_p33) con 21, 32 y 33 vértices, y 11, 18 y 20 instancias respectivamente fueron creados por [1]. Los 2 siguientes grupos (2_p64 y 2_p66) con 64 y 66 vértices, y 14 y 26 instancias fueron propuestos por [23]. Estos 5 grupos fueron creados por sus autores con un único objetivo (beneficio), por lo que [15] añadió un segundo objetivo. La distancia entre vértices se calcula usando la fórmula de la distancia Euclídea. Estos 5 grupos pueden considerarse como instancias de test (benchmark).

Los siguientes 5 grupos fueron creados totalmente por [15] y son instancias reales de algunas ciudades y regiones europeas. Dentro de estos 5 grupos tenemos 97 y 273 vértices (2_p97 y

2_p273) con 20 instancias cada uno y con una restricción de distancia máxima de tour (C_{max}) entre 1 y 20 kilómetros; y 559, 562 y 2143 vértices (2_p559, 2_p562 y 2_p2143) con 29 instancias y una C_{max} desde 10 hasta 150 kilómetros. Todas estas se proporcionan con una matriz de distancias entre vértices. Además, para estos 5 grupos de instancias se usa una distancia de servicio entre vértices de 0,5 kilómetros que se puede entender como el tiempo necesario para visitar un POI.

V-B. Indicadores o Métricas de Calidad

Hemos usado el software disponible en <http://www.tik.ee.ethz.ch/sop/pisa/?page=assessment.php> para calcular los indicadores, según las instrucciones proporcionadas por [24]. Todos los indicadores usados en este trabajo han sido calculados usando valores normalizados de las funciones objetivo. Para algunos indicadores unarios se necesita un conjunto de referencia, por lo que hemos generado un conjunto de referencia con las soluciones no dominadas de todas las ejecuciones de los algoritmos a comparar (MOABC, P-ACO y P-VNS). Hemos usado las soluciones proporcionadas por [15] para cada instancia calculada por los algoritmos P-ACO y P-VNS, y realizado una comparación de acuerdo a varios indicadores o métricas utilizadas habitualmente en la literatura de optimización multiobjetivo: hipervolumen, épsilon unario y R3. Cada uno de estos tres indicadores se basa en diferentes propiedades, lo que permite una evaluación completa y balanceada que incrementa su fiabilidad. Puede encontrarse información detallada sobre los indicadores hipervolumen, épsilon unario y R3 en [25], [26] y [27] respectivamente.

V-C. Comparación entre Algoritmos

En esta subsección, comparamos los resultados obtenidos por MOABC con los resultados de los algoritmos P-ACO y P-VNS [15]. Hemos usado el mismo número de ejecuciones independientes que fueron usadas en esos algoritmos, 10 ejecuciones por instancia. También hemos usado los resultados obtenidos por P-ACO y P-VNS en cada instancia, los cuales utilizaron los parámetros de configuración más óptimos, como puede comprobarse en [15]. En el caso de nuestro algoritmo MOABC, siguiendo las recomendaciones de [28], hemos estudiado no solo el parámetro *TP* (tamaño de la población), sino también el parámetro *limite* (criterio de abandono). Después de realizar tests preliminares con un rango de diferentes valores para cada parámetro, hemos establecido la configuración de los parámetros MOABC a un *TP* de 60 y un *limite* de 10. Todos los algoritmos fueron parados tras consumir el mismo tiempo de ejecución, el usado por el algoritmo P-ACO. Los tiempos de ejecución y resultados se encuentran disponibles en <http://prolog.univie.ac.at/research/OP> para todas las ejecuciones e instancias de los algoritmos P-ACO y P-VNS.

En primer lugar, mostramos una comparación de los resultados obtenidos para el indicador hipervolumen (I_H). La tabla I muestra los resultados de los tres algoritmos para cada conjunto de instancias.



Tabla I
RESULTADOS DEL INDICADOR HIPERVOLUMEN. LOS VALORES EN NEGRITA INDICAN SUPERIORIDAD. LOS VALORES ALTOS SON MEJORES.

Conjuntos	I_H (MOABC)	I_H (P-ACO)	I_H (P-VNS)
2_p21	3,525	3,390	3,471
2_p32	3,366	3,375	3,250
2_p33	3,575	3,638	3,615
Media (Pequeño)	3,489	3,468	3,445
2_p64	3,363	3,586	3,569
2_p66	3,346	3,609	3,554
2_p97	3,501	3,315	3,335
Media (Mediano)	3,403	3,503	3,486
2_p273	3,472	3,329	3,320
2_p559	3,648	3,415	3,441
2_p562	3,634	3,451	3,370
2_p2143	3,640	3,144	2,839
Media (Grande)	3,598	3,335	3,242
Media (Todos)	3,497	3,435	3,391

La tabla I muestra la media de los resultados obtenidos por conjunto de instancia. Por ejemplo, el conjunto de instancias 2_p21 tiene 11 instancias, por lo tanto, esta tabla muestra la media de los resultados de 110 ejecuciones. Además, hemos agrupado los conjuntos de instancias en tres grupos, teniendo en cuenta sus tamaños (número de vértices): pequeño, mediano y grande; y mostrado su media de resultados. Finalmente, mostramos la media para todas las instancias. En instancias pequeñas, MOABC obtiene la mejor media de hipervolumen, y consigue mejores resultados en 2_p21 que P-ACO y P-VNS. En instancias medianas, el algoritmo MOABC obtiene el mejor hipervolumen para 2_p97. En instancias grandes, el algoritmo MOABC es definitivamente superior que P-ACO y P-VNS, obteniendo los mejores hipervolumenes para todas las instancias grandes (las más complicadas). De hecho, es aquí donde las diferencias entre los tres algoritmos son mayores, especialmente en el conjunto de instancias con 2143 vértices (2_p2143). Como conclusión, en la tabla I podemos ver que MOABC tiene la mejor media global de hipervolumen.

Tabla II
RESULTADOS DEL INDICADOR ÉPSILON UNARIO. LOS VALORES EN NEGRITA INDICAN SUPERIORIDAD. LOS VALORES BAJOS SON MEJORES.

Conjuntos	I_ϵ (MOABC)	I_ϵ (P-ACO)	I_ϵ (P-VNS)
2_p21	1,000	1,108	1,034
2_p32	1,024	1,028	1,124
2_p33	1,074	1,032	1,050
Media (Pequeño)	1,032	1,056	1,069
2_p64	1,085	1,036	1,038
2_p66	1,104	1,033	1,049
2_p97	1,000	1,158	1,137
Media (Mediano)	1,063	1,076	1,074
2_p273	1,072	1,196	1,161
2_p559	1,037	1,118	1,113
2_p562	1,039	1,112	1,128
2_p2143	1,076	1,188	1,265
Media (Grande)	1,056	1,153	1,167
Media (Todos)	1,050	1,095	1,103

En segundo lugar, y usando la misma metodología que en el indicador anterior, la tabla II muestra los resultados para

los tres algoritmos y su comparación basada en el indicador épsilon unario (I_ϵ). En pequeñas instancias, MOABC obtiene la mejor media en el indicador épsilon, y también obtiene mejores resultados para los conjuntos de instancias 2_p21 y 2_p32 que P-ACO y P-VNS. En instancias medianas, el algoritmo MOABC nuevamente consigue la mejor media en el indicador épsilon, y también los mejores resultados para 2_p97. En instancias grandes, MOABC es claramente mejor que P-ACO y P-VNS ya que obtiene los mejores resultados para todos los conjuntos de instancias grandes (las más complejas). Analizando la tabla II podemos observar que MOABC tiene la mejor media global del indicador épsilon, consiguiendo el mejor resultado en 7 de 10 conjuntos de instancias.

Tabla III
RESULTADOS DEL INDICADOR R3. LOS VALORES EN NEGRITA INDICAN SUPERIORIDAD. LOS VALORES BAJOS SON MEJORES.

Conjuntos	I_{R3} (MOABC)	I_{R3} (P-ACO)	I_{R3} (P-VNS)
2_p21	0,000	0,031	0,010
2_p32	0,003	0,003	0,030
2_p33	0,018	0,005	0,010
Media (Pequeño)	0,007	0,013	0,017
2_p64	0,032	0,011	0,007
2_p66	0,039	0,008	0,011
2_p97	0,000	0,046	0,041
Media (Mediano)	0,024	0,022	0,020
2_p273	0,030	0,067	0,063
2_p559	0,011	0,054	0,048
2_p562	0,010	0,041	0,059
2_p2143	0,038	0,135	0,204
Media (Grande)	0,022	0,074	0,093
Media (Todos)	0,018	0,036	0,043

Finalmente, y usando la misma metodología, la tabla III muestra los resultados para los tres algoritmos y su comparación del indicador R3 (I_{R3}). En pequeñas instancias, MOABC obtiene la mejor media del indicador R3, y también los mejores resultados para los conjuntos de instancias 2_p21 y 2_p32. En instancias medianas, MOABC consigue los mejores resultados para el conjunto de instancias 2_p97. En instancias grandes, MOABC es definitivamente mejor que P-ACO y P-VNS, ya que como en los anteriores indicadores consigue los mejores resultados para todas las instancias grandes. Teniendo en cuenta la tabla III, podemos afirmar que MOABC tiene la mejor media global del indicador R3, obteniendo los mejores resultados para 7 de 10 conjuntos de instancias.

Resumiendo los resultados de las tres tablas anteriores podemos concluir que:

- MOABC tiene la mejor media global para los tres indicadores (I_H , I_ϵ y I_{R3}).
- En las instancias pequeñas, MOABC obtiene la mejor media para los tres indicadores, y generalmente, proporciona mejores resultados que P-ACO y P-VNS en 2 de 3 conjuntos de instancias (2_p21 y 2_p32).
- En las instancias medianas, de acuerdo con los tres indicadores, MOABC claramente supera a los otros dos algoritmos (P-ACO y P-VNS) en el conjunto de instan-

cias 2_p97 (uno de los tres conjuntos de instancias del grupo).

- En las instancias grandes, MOABC es el mejor algoritmo, teniendo en cuenta los tres indicadores, sus medias correspondientes, y también los resultados individuales para cada uno de los 4 conjuntos de instancias de este grupo. Las diferencias respecto a los tres algoritmos son especialmente mayores en el último conjunto de instancias con 2143 vértices (el más difícil).

VI. CONCLUSIONES

Hemos desarrollado un enfoque metaheurístico para solucionar el problema de orientación multiobjetivo. Nuestra propuesta, MOABC, es un algoritmo de optimización multiobjetivo basado en el comportamiento de las abejas melíferas. MOABC ha sido comparado con dos algoritmos multiobjetivo que fueron publicados anteriormente, P-ACO y P-VNS. Las comparaciones fueron realizadas usando instancias de test e instancias del mundo real del problema de orientación, con un total de 216 instancias agrupadas en 10 conjuntos. Utilizamos tres indicadores de calidad multiobjetivo, utilizados habitualmente en la literatura, para comparar los resultados de los tres algoritmos. MOABC es mejor en general que P-ACO y P-VNS, de acuerdo a la media de los indicadores y sensiblemente mejor cuanto mayor (más compleja) es la instancia del problema a resolver (mayor número de vértices). Por lo tanto, podemos concluir que MOABC es una aproximación muy interesante para resolver el problema de orientación.

En el futuro intentaremos aplicar este enfoque a un planificador de rutas turísticas. Además, trataremos de mejorar estos resultados usando otras estrategias distintas (p. ej.: otras variantes de ABC u otros algoritmos evolutivos). Su diseño, implementación y ejecución, y la comparación con algoritmos multiobjetivo es interesante para futuras investigaciones. Finalmente, la aplicación de MOABC en otros tipos de problemas de orientación también supone una línea interesante de trabajo.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por la AEI (Agencia Estatal de Investigación, España) y el FEDER (Fondo Europeo de Desarrollo Regional, UE), bajo el proyecto TIN2016-76259-P (proyecto PROTEIN). Gracias también a la Junta de Extremadura por la ayuda GR18090 otorgada al grupo de investigación TIC015.

REFERENCIAS

- [1] T. Tsiligirides, Heuristic methods applied to orienteering, *Journal of the Operational Research Society* 35 (9) (1984) 797–809.
- [2] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, *Journal of Global Optimization* 39 (3) (2007) 459–471.
- [3] D. Karaboga, B. Gorkemli, C. Ozturk, N. Karaboga, A comprehensive survey: artificial bee colony (ABC) algorithm and applications, *Artificial Intelligence Review* 42 (1) (2014) 21–57.
- [4] Y. Zhang, L. Wu, Optimal multi-level thresholding based on maximum Tsallis entropy via an artificial bee colony approach, *Entropy* 13 (4) (2011) 841–859.
- [5] Y. Zhang, L. Wu, S. Wang, Magnetic resonance brain image classification by an improved artificial bee colony algorithm, *Progress In Electromagnetics Research* 116 (2011) 65–79.
- [6] S. Wang, Y. Zhang, Z. Dong, S. Du, G. Ji, J. Yan, J. Yang, Q. Wang, C. Feng, P. Phillips, Feed-forward neural network optimized by hybridization of PSO and ABC for abnormal brain detection, *International Journal of Imaging Systems and Technology* 25 (2) (2015) 153–164.
- [7] W. Zou, Y. Zhu, H. Chen, B. Zhang, Solving multiobjective optimization problems using artificial bee colony algorithm, *Discrete Dynamics in Nature and Society* 2011 (2011) 569784.
- [8] H. Zhang, Y. Zhu, W. Zou, X. Yan, A hybrid multi-objective artificial bee colony algorithm for burdening optimization of copper strip production, *Applied Mathematical Modelling* 36 (6) (2012) 2578–2591.
- [9] J. Huo, L. Liu, An improved multi-objective artificial bee colony optimization algorithm with regulation operators, *Information* 8 (1) (2017) 18.
- [10] D. Feillet, P. Dejax, M. Gendreau, Traveling salesman problems with profits, *Transportation Science* 39 (2) (2005) 188–205.
- [11] A. Gunawan, H. C. Lau, P. Vansteenwegen, Orienteering problem: A survey of recent variants, solution approaches and applications, *European Journal of Operational Research* 255 (2) (2016) 315–332.
- [12] P. Vansteenwegen, W. Souffriau, D. V. Oudheusden, The orienteering problem: A survey, *European Journal of Operational Research* 209 (1) (2011) 1–10.
- [13] P. Vansteenwegen, W. Souffriau, G. V. Berghe, D. V. Oudheusden, A guided local search metaheuristic for the team orienteering problem, *European Journal of Operational Research* 196 (1) (2009) 118–127.
- [14] I.-M. Chao, B. L. Golden, E. A. Wasil, The team orienteering problem, *European Journal of Operational Research* 88 (3) (1996) 464–474.
- [15] M. Schilde, K. F. Doerner, R. F. Hartl, G. Kiechle, Metaheuristics for the bi-objective orienteering problem, *Swarm Intelligence* 3 (3) (2009) 179–201.
- [16] D. Purevsuren, G. Cui, S. ur Rehman, Evolutionary multiobjective optimization algorithms with path relinking for bi-orienteering problem, in: *Software Engineering and Service Science (ICSESS)*, 2015 6th IEEE International Conference on, 2015, pp. 132–135.
- [17] R. Martí, V. Campos, M. G. Resende, A. Duarte, Multiobjective GRASP with path relinking, *European Journal of Operational Research* 240 (1) (2015) 54–71.
- [18] B. L. Golden, L. Levy, R. Vohra, The orienteering problem, *Naval Research Logistics (NRL)* 34 (3) (1987) 307–318.
- [19] G. A. Croes, A method for solving traveling-salesman problems, *Operations Research* 6 (6) (1958) 791–812.
- [20] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation* 6 (2) (2002) 182–197.
- [21] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.
- [22] M. Mernik, S.-H. Liu, D. Karaboga, M. Črepinšek, On clarifying misconceptions when comparing variants of the Artificial Bee Colony algorithm by offering a new implementation, *Information Sciences* 291 (2015) 115–127.
- [23] I.-M. Chao, B. L. Golden, E. A. Wasil, A fast and effective heuristic for the orienteering problem, *European Journal of Operational Research* 88 (3) (1996) 475–489.
- [24] J. Knowles, L. Thiele, E. Zitzler, A tutorial on the performance assessment of stochastic multiobjective optimizers, Technical report no. 214, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland, revised version (Feb 2006).
- [25] N. Beume, C. M. Fonseca, M. López-Ibáñez, L. Paquete, J. Vahrenhold, On the complexity of computing the hypervolume indicator, *IEEE Transactions on Evolutionary Computation* 13 (5) (2009) 1075–1082.
- [26] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, V. G. da Fonseca, Performance assessment of multiobjective optimizers: an analysis and review, *IEEE Transactions on Evolutionary Computation* 7 (2) (2003) 117–132.
- [27] M. P. Hansen, A. Jaszkiewicz, Evaluating the quality of approximations to the non-dominated set, Tech. Rep. IMM-REP-1998-7, Institute of Mathematical Modelling, Technical University of Denmark (1998).
- [28] N. Veček, S.-H. Liu, M. Črepinšek, M. Mernik, On the importance of the artificial bee colony control parameter ‘limit’, *Information Technology and Control* 46 (4) (2017) 566–604.