

Un enfoque multi-objetivo para el problema de la máxima diversidad

Pedro Casas-Martínez Dept. Computer Science Universidad Rey Juan Carlos Móstoles, España p.casas.2017@alumnos.urjc.es 0000-0002-7859-2988	Alejandra Casado-Ceballos Dept. Computer Science Universidad Rey Juan Carlos Móstoles, España a.casadoce.2017@alumnos.urjc.es 0000-0003-3417-6859	Jesús Sánchez-Oro Dept. Computer Science Universidad Rey Juan Carlos Móstoles, España jesus.sanchezoro@urjc.es 0000-0003-1702-4941	Eduardo G. Pardo Dept. Computer Science Universidad Rey Juan Carlos Móstoles, España eduardo.pardo@urjc.es 0000-0002-6247-5269
--	--	---	---

Resumen—El problema de la máxima diversidad ha sido ampliamente estudiado desde diferentes perspectivas, haciendo uso de técnicas exactas y heurísticas. En este artículo, nos enfrentamos a una variante multi-objetivo del problema, la cual intenta optimizar simultáneamente las cinco métricas de diversidad cuyo uso está más extendido en la literatura. Para lograrlo, proponemos una novedosa adaptación del ya conocido *Greedy Randomized Adaptive Search Procedure*, que ha sido usado tradicionalmente para optimización mono-objetivo. Con este nuevo enfoque, un conjunto de soluciones eficientes es generado siguiendo dos estrategias constructivas diferentes, analizando el rendimiento de ambas. La fase de mejora del algoritmo propuesto consiste en un procedimiento de búsqueda local basado en una estructura de intercambios siguiendo un enfoque *first improvement*. Además, se propone una exploración inteligente de la vecindad que permita limitar el espacio de búsqueda. Los experimentos computacionales muestran la calidad del algoritmo propuesto, comparado con los resultados obtenidos con el mejor método previo de la literatura. Las diferencias encontradas han sido respaldadas por pruebas estadísticas no paramétricas.

Index Terms—diversidad, multi-objetivo, GRASP, búsqueda local

I. INTRODUCCIÓN

La comunidad científica lleva estudiando la familia de problemas sobre la máxima diversidad desde 1988 [9], mientras que los primeros modelos de programación entera fueron introducidos en Dhir et al. [2] y Kuo et al. [10]. El concepto de la diversidad ha evolucionado con el paso del tiempo, y muchas propuestas nuevas en la literatura han mejorado su definición [17] construyendo así una familia de problemas de optimización. El alto interés en este tipo de familia de problemas se debe principalmente a sus aplicaciones prácticas en una gran variedad de áreas: análisis de redes sociales, problemas de localización de instalaciones, generación de un conjunto de estudiantes heterogéneos en clase, etc. Ha sido demostrado que todos estos problemas son problemas de optimización combinatoria \mathcal{NP} -difíciles [16]. La idea principal en la que se basan los problemas de optimización de diversidad consiste en seleccionar un subconjunto de elementos de un

conjunto dado, con el objetivo de maximizar una cierta medida de distancia. La principal diferencia entre la mayoría de enfoques propuestos es la forma de calcular la diversidad de los elementos seleccionados.

Antes de definir las métricas estudiadas en este artículo, vamos a explicar algunas definiciones preliminares comunes en la familia de problemas de optimización de la diversidad. Dado un conjunto N de elementos, el objetivo es encontrar un subconjunto S de p elementos, siendo $p < |N|$, de tal forma que la diversidad de los elementos incluidos en S sea maximizada. Para evaluar la diversidad se debe definir una métrica de distancia d_{ij} , con $i, j \in N$, entre dos elementos. Para preservar la generalidad, no consideramos ninguna métrica de distancia específica, sino que cualquier métrica puede ser seleccionada: distancia euclídea, distancia del coseno, índice de similitud de jaccard, entre otras [20, 19]. Tras esto, el problema de la máxima diversidad busca encontrar una solución S^* que maximiza una cierta métrica de diversidad. Más formalmente,

$$S^* \leftarrow \arg \max_{S \in \mathcal{SS}} d(S) \quad (1)$$

donde \mathcal{SS} representa el espacio de búsqueda, i.e., el conjunto de todas las posibles combinaciones de p elementos distintos de N .

Los problemas de maximización de la diversidad han sido tradicionalmente abordados desde una perspectiva mono-objetivo, enfocándose en encontrar un subconjunto de los elementos más diversos con respecto a una sola métrica de diversidad. Sin embargo, es interesante analizar los resultados obtenidos al modelar la diversidad como un problema de optimización multi-objetivo. Además, en el contexto de aplicaciones en la vida real, es difícil seleccionar sólo una métrica de diversidad que satisfaga todos los requerimientos del problema. Por tanto, es deseable la capacidad de optimizar simultáneamente más de una métrica. En ese caso, estamos lidiando con un problema de optimización multi-objetivo (MOP, del inglés *Multi-objective Optimization Problem*), en el cual varias funciones objetivo deben ser optimizadas al mismo tiempo. La principal característica al lidiar con un MOP es que las funciones objetivo consideradas están en conflicto, i.e., optimizar una de ellas suele deteriorar otras. Debido a esto,

Esta investigación está financiada por el Ministerio de Ciencia, Innovación y Universidades con el proyecto con referencia PGC2018-095322-B-C22, y por la Comunidad de Madrid y los Fondos Estructurales de la Unión Europea con los proyectos con referencias S2018/TCS-4566 y Y2018/EMT-5062.

no existe una solución única que optimiza todos los objetivos al mismo tiempo. Por tanto, el resultado de un MOP no es una única solución sino un conjunto de soluciones eficientes.

El conjunto de soluciones eficientes es construido siguiendo el principio de optimalidad de Pareto, que presenta un balance para una solución en la cual una función objetivo no puede ser mejorada sin deteriorar otra. Por tanto, la comparación de dos soluciones es dada por el concepto de dominancia. Específicamente, dadas dos soluciones S_1 y S_2 , podemos afirmar que S_1 domina a S_2 si y solo si la evaluación de S_1 , considerando cada función objetivo, es mejor o igual que el valor de la función objetivo correspondiente en S_2 . De no ser así, se dice que S_1 y S_2 son recíprocamente no dominadas. Según esta definición, una solución S_1 es considerada eficiente si no existe ninguna otra solución S_2 en el espacio de soluciones tal que S_2 domine a S_1 . El conjunto que contiene las mejores soluciones no dominadas del espacio de búsqueda se denomina Frente de Pareto.

Este trabajo está centrado en proponer un algoritmo que sea capaz de optimizar más de una medida de diversidad al mismo tiempo con el objetivo de encontrar diferencias y similitudes entre las medidas de diversidad más extendidas. Este enfoque fue presentado originalmente en Vera et al. [22], donde los autores estudiaron cinco medidas de diversidad diferentes simultáneamente. Este problema fue llamado *Multi-Objective Maximum Diversity Problem* (MOMDP). Concretamente, los autores en Vera et al. [22] propusieron un algoritmo evolutivo multi-objetivo para resolver el MOMDP.

En particular, se propone una adaptación del ya conocido *Greedy Randomized Adaptive Search Procedure* (GRASP) para resolver el MOMDP. El resto del trabajo está organizado de la siguiente forma: la Sección II describe las métricas de diversidad consideradas en este trabajo, la Sección III detalla la propuesta algorítmica, la Sección IV presenta los experimentos computacionales llevados a cabo para analizar el desempeño del algoritmo propuesto y, finalmente, la Sección V detalla las conclusiones más relevantes de esta investigación.

II. MÉTRICAS DE DIVERSIDAD

En esta sección se definen las medidas de diversidad estudiadas en este trabajo, las cuales han sido consideradas juntas previamente en [22]. En dicho estudio, los autores muestran como los objetivos propuestos están en conflicto, es decir, no es posible mejorar en uno de ellos sin deteriorar la calidad de los otros.

La primera medida de diversidad es *Max-Sum Diversity* (MSD). En este caso, la diversidad de un subconjunto de elementos es evaluada como la suma de las distancias entre cada par de elementos seleccionados [15], tal como está definido en la Ecuación 2.

$$MSD(S) = \sum_{\substack{i,j \in S \\ i < j}} d_{ij} \quad (2)$$

El objetivo es, por tanto, encontrar una solución S^* que maximice el valor de MSD. Esta métrica ha sido ampliamente

usada en el contexto de la localización competitiva de instalaciones [8] y la selección de miembros de jurado [14].

La segunda medida de diversidad considerada, llamada *Max-Min Diversity* (MMD), es evaluada como la distancia mínima entre elementos en S (ver la Ecuación 3).

$$MMD(S) = \min_{\substack{i,j \in S \\ i < j}} d_{ij} \quad (3)$$

De nuevo, el objetivo es encontrar una solución S^* que maximice el MMD. Esta métrica ha sido estudiada en problemas relacionados con la localización de instalaciones nocivas [5].

Max-MinSum Diversity (MMSD) es la tercera métrica de diversidad considerada en este trabajo. Para cada elemento seleccionado $i \in S$, se calcula la suma de distancias desde i a los elementos en $S \setminus \{i\}$, devolviendo el valor mínimo obtenido de entre todas las posibles sumas (ver la Ecuación 4).

$$MMSD(S) = \min_{i \in S} \sum_{j \in (S \setminus \{i\})} d_{ij} \quad (4)$$

El objetivo en este caso es maximizar el valor de MMSD. Esta métrica ha sido usada tradicionalmente en problemas relacionados con la equidad o en la selección de grupos homogéneos [3].

La cuarta medida de diversidad considerada en este trabajo es llamada *Minimum Differential Dispersion* (MDD). Para evaluar esta medida, se utiliza un método similar al usado para evaluar el MMSD. Para cada elemento seleccionado $i \in S$, se calcula la suma de distancias al resto de elementos en $S \setminus \{i\}$, evaluando el MDD como la diferencia entre el máximo y el mínimo valor obtenido entre todas las posibles sumas (ver la Ecuación 5). El objetivo de esta medida es, por tanto, encontrar el valor mínimo del MDD.

$$MDD(S) = \max_{i \in S} \sum_{j \in (S \setminus \{i\})} d_{ij} - \min_{i \in S} \sum_{j \in (S \setminus \{i\})} d_{ij} \quad (5)$$

La quinta y última medida de diversidad considerada en este trabajo se denomina *Minimum p-Center Diversity* (MPCD). Evaluar esta medida requiere el conjunto de elementos seleccionados S y también el conjunto de elementos no seleccionados $N \setminus S$. Para cada elemento no seleccionado $i \in N \setminus S$, se calcula la distancia a su elemento seleccionado más cercano $j \in S$, devolviendo la mayor distancia encontrada (ver la Ecuación 6).

$$MPCD(S) = \max_{i \in (N \setminus S)} \left\{ \min_{j \in S} d_{ij} \right\} \quad (6)$$

En este caso, el objetivo es minimizar el valor del MPCD. Este problema surge en el contexto de localización de instalaciones, donde el conjunto de elementos seleccionados hace referencia a las instalaciones que serían abiertas, mientras que el conjunto de no seleccionados representa a los puntos de demanda que dependen de los servicios de las instalaciones abiertas.

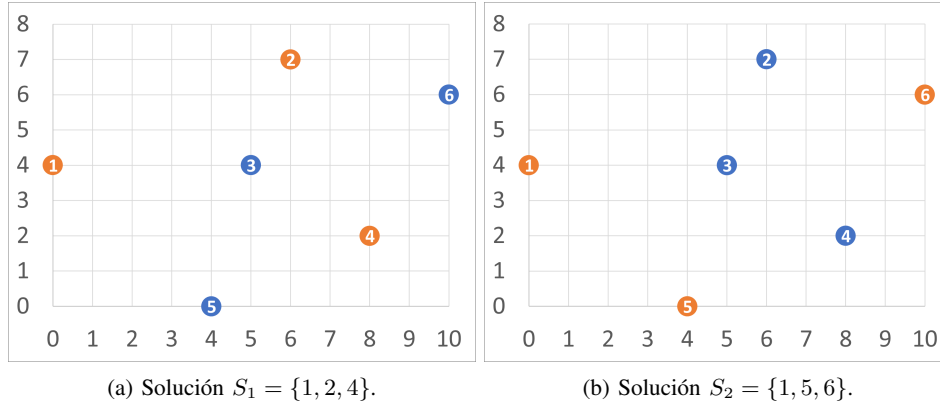


Figura 1: Dos posibles soluciones para una instancia con 6 elementos disponibles de los cuales hay que seleccionar 3.

A continuación, se muestra la evaluación de estas métricas con un ejemplo gráfico. La Figura 1 muestra una instancia de ejemplo con $N = \{1, 2, 3, 4, 5, 6\}$ y $p = 3$. Las Figuras 1a y 1b ilustran dos posibles soluciones $S_1 = \{1, 2, 4\}$ y $S_2 = \{1, 5, 6\}$ para este ejemplo, respectivamente. Debemos destacar que los puntos están representados en un espacio bidimensional euclídeo para facilitar la evaluación de la distancia entre los puntos.

La Tabla I resume los resultados obtenidos en cada una de las métricas de diversidad en S_1 y S_2 . Debemos destacar que el mejor resultado para cada métrica de diversidad se resalta en negrita.

Solución	MSD	MMD	MMSD	MDD	MPCD
S_1	20.34	5.39	12.09	2.86	4.47
S_2	24.34	5.66	14.14	4.54	4.47

Tabla I: Evaluación de las cinco métricas de diversidad propuestas sobre las soluciones presentadas en la Figura 1.

Los resultados presentados en la Tabla I confirman el comportamiento de un MOP. En concreto, S_1 obtiene mejor calidad que S_2 en la métrica MDD, mientras que S_2 supera a S_1 en MSD, MMD y MMSD. Finalmente, ambas soluciones presentan la misma calidad en la métrica MPCD. Por tanto, no es posible determinar que S_1 sea mejor que S_2 o viceversa, ya que son soluciones no dominadas.

III. PROPUESTA ALGORÍTMICA

Este trabajo propone un *Greedy Randomized Adaptive Search Procedure (GRASP)* para resolver el MOMDP. GRASP fue presentado originalmente en [6] pero no fue formalmente definido hasta [7]. Es una metaheurística muy extendida que ha sido aplicada con éxito a muchos problemas de optimización complejos de una gran variedad de áreas [13, 18, 21]. GRASP sigue una estrategia multi-arranque formada por dos etapas diferentes: construcción y mejora. La primera consiste en una fase de construcción voraz aleatorizada para generar una solución inicial, mientras que la mejora es responsable de optimizar localmente la solución construida. El éxito de esta

metodología se basa en el balance entre intensificación y diversificación en la generación de soluciones, que se controla por un parámetro de búsqueda llamado $\alpha \in [0, 1]$, que se describe en detalle en la Sección III-A.

III-A. Procedimiento constructivo

El procedimiento constructivo en el contexto de GRASP debe generar una solución inicial diversa y de alta calidad para comenzar la búsqueda. En lugar de usar un procedimiento completamente voraz, GRASP favorece la diversificación incluyendo aleatoriedad en este paso. El Algoritmo 1 presenta el pseudocódigo del procedimiento constructivo propuesto.

Algorithm 1 GRASP Constructive(N, p, α)

```

1:  $i \leftarrow RND(N)$ 
2:  $S \leftarrow \{i\}$ 
3:  $CL \leftarrow N \setminus \{i\}$ 
4: while  $|S| < p$  do
5:    $g_{\min} \leftarrow \min_{c \in CL} g(c)$ 
6:    $g_{\max} \leftarrow \max_{c \in CL} g(c)$ 
7:    $\mu \leftarrow g_{\max} - \alpha \cdot (g_{\max} - g_{\min})$ 
8:    $RCL \leftarrow \{c \in CL : g(c) \geq \mu\}$ 
9:    $i \leftarrow RND(RCL)$ 
10:   $CL \leftarrow CL \setminus \{i\}$ 
11:   $S \leftarrow S \cup \{i\}$ 
12: end while
13: return  $S$ 

```

El procedimiento constructivo requiere tres parámetros de entrada: N , el conjunto de elementos candidatos para ser seleccionados; p , el número de elementos que deben ser seleccionados; y α , un número real en el rango $[0, 1]$ que controla el balance entre diversificación e intensificación. En el contexto de GRASP, el primer elemento que se incluye en la solución suele ser seleccionado de forma aleatoria del conjunto de disponibles N para favorecer la diversificación (paso 1). Este elemento es incluido en la solución que se está construyendo (paso 2), y se crea una lista de candidatos CL con todos los elementos menos i (paso 3). Hecho esto, el método añade de forma iterativa un nuevo elemento a la solución hasta

que se convierte en factible (pasos 4-12). En cada iteración, se calcula el valor mínimo (g_{\min}) y el máximo (g_{\max}) de una cierta función voraz (pasos 5-6). Esta función voraz $g(c)$ determina la contribución de cada elemento candidato c a la calidad de la solución en construcción.

Tras evaluar los candidatos, se calcula un umbral μ (paso 7) con el objetivo de construir una lista de candidatos restringida *RCL* (paso 8) con los elementos más prometedores. El umbral depende del valor del parámetro de entrada de búsqueda α , que controla la voracidad / aleatoriedad del método. Por un lado, si $\alpha = 0$, entonces $\mu = g_{\max}$, por lo que la *RCL* solo contiene aquellos elementos con el valor máximo de la función voraz, lo que resulta en un criterio totalmente voraz. Por otro lado, si $\alpha = 1$, entonces $\mu = g_{\min}$, lo que resulta en un criterio totalmente aleatorio ya que todos los elementos disponibles pueden entrar en *RCL*. El valor más adecuado para este parámetro de entrada se estudia en la Sección IV. Una vez la *RCL* ha sido construida con los elementos más prometedores, se escoge uno al azar para ser incluido en la solución en construcción (paso 9). Finalmente, *CL* se actualiza eliminando el elemento seleccionado (paso 10) e incluyéndolo en la solución (paso 11). El método termina cuándo se han seleccionado p elementos, devolviendo la solución S (paso 13).

Este esquema tradicional del constructivo GRASP debe ser adaptado a un problema de optimización multi-objetivo como MOMDP. En concreto, proponemos dos adaptaciones diferentes para construir soluciones factibles. La primera adaptación, llamada *Joint Alternate Evaluation of Objectives (JALEO)*, se centra en usar la métricas de diversidad presentadas en la Sección II como funciones voraces en cada construcción independientemente, generando el mismo número de soluciones para cada objetivo. La segunda adaptación, llamada *Mixed Objective Function Algorithm (MOFA)*, usa todas las métricas de diversidad como funciones voraces simultáneamente en cada construcción. Para esto, la función voraz varía de una métrica de diversidad a otra en cada iteración del procedimiento constructivo. En la Sección IV se compara el desempeño de ambas estrategias para construir soluciones iniciales. Finalmente, cabe destacar que, para cada solución construida, se evalúa la posibilidad de ser incluida en el conjunto de soluciones eficientes.

III-B. Búsqueda local

Las soluciones iniciales construidas con el método descrito en la Sección III-A no son necesariamente óptimos locales y, por tanto, pueden ser mejoradas para encontrar un óptimo local respecto a un cierto vecindario. El segundo paso del algoritmo GRASP es responsable de encontrar estos óptimos locales para cada solución construida.

El primer elemento requerido para definir una búsqueda local es un operador de movimiento. En el contexto del problema en consideración, proponemos un movimiento de intercambio el cual, dada una solución S y un conjunto de elementos N , elimina un elemento seleccionado $i \in S$ e incluye uno no seleccionado $j \in N \setminus S$. Cabe destacar que el

movimiento de intercambio siempre resulta en una solución factible. En términos matemáticos,

$$\text{Intercambio}(S, i, j) = (S \setminus \{i\}) \cup \{j\}$$

Por tanto, el vecindario $\mathcal{N}(S)$ de una solución dada S se define como el conjunto de soluciones S' que pueden ser alcanzadas aplicando un solo intercambio a S . Formalmente,

$$\mathcal{N}(S) \leftarrow \{S' \leftarrow \text{Intercambio}(S, i, j) \mid \forall i \in S \wedge \forall j \in N \setminus S\}$$

La definición de una búsqueda local consiste en determinar cómo se recorre el vecindario en la búsqueda. En este trabajo se propone un enfoque basado en *first improvement*, que realiza el primer movimiento que conlleve una mejora, reduciendo los tiempos de cómputo. Debido a que se trata de un MOP, consideramos que se ha producido una mejora si la solución explorada entra en el conjunto de soluciones eficientes.

Al utilizar la estrategia *first improvement* es relevante determinar el orden en el cual el vecindario es explorado. Como una solución en $\mathcal{N}(S)$ se obtiene realizando un solo movimiento de *Intercambio*, el cual involucra a un elemento seleccionado $i \in S$, y a uno no seleccionado $j \in (N \setminus S)$, podemos determinar el orden el cual se exploran los elementos en S y, de la misma manera, con los elementos en $N \setminus S$. En particular, para cada i , podemos calcular la distancia mínima a los demás elementos seleccionados, i.e., $\min_{i' \in (S \setminus \{i\})} d_{i, i'}$. De esta forma, los elementos seleccionados se exploran en orden ascendente respecto a esta distancia, mientras que los no seleccionados se exploran en orden descendente. Este ordenamiento nos permite explorar primero los movimientos más prometedores.

IV. RESULTADOS EXPERIMENTALES

Esta sección tiene dos objetivos principales: (i) ajustar los parámetros de entrada para encontrar la mejor configuración del algoritmo propuesto y (ii) realizar un análisis comparativo con el mejor método encontrado en el estado del arte para analizar el desempeño del algoritmo. Todos los algoritmos han sido implementados en Java SE 11, realizando los experimentos en un AMD EPYC 7282 (2.8 GHz) y 8 GB RAM.

El conjunto de instancias consideradas en este trabajo es el mismo que el presentado en el mejor método encontrado en la literatura [22], formado por el subconjunto GKD de la MDPLIB¹. En concreto, se trata de 145 instancias que van desde 10 a 500 elementos donde el número de elementos a seleccionar va desde 2 hasta 50.

Al trabajar con problemas de optimización mono-objetivo, es fácil realizar una comparación entre dos o más algoritmos, ya que sólo es necesario comparar el valor de la función objetivo para cada algoritmo que se desee analizar. Sin embargo, en el contexto de la optimización multi-objetivo, como en el MOMDP, el resultado de un algoritmo es un conjunto de soluciones eficientes y, por tanto, es necesario definir métricas específicas para comparar conjuntos de soluciones eficientes.

¹<https://www.uv.es/rmarti/paper/mdp.html>

En este trabajo se utilizan algunas métricas bien conocidas en la literatura de MOP: cobertura, hipervolumen, indicador ϵ , y distancia generacional inversa [12]. A continuación, se explica brevemente cada una de ellas.

La métrica cobertura, $C(R, \hat{E})$, calcula el número de soluciones del frente de aproximación en evaluación \hat{E} que son dominadas por un frente de referencia R . Este frente de referencia se calcula como el conjunto de soluciones eficientes resultante de la unión de todos los frentes involucrados en la comparación. El hipervolumen, HV , evalúa el volumen en el espacio objetivo cubierto por el conjunto de soluciones eficientes en evaluación. El indicador ϵ , EPS , calcula cuál es la distancia más pequeña necesaria para transformar cada punto del conjunto de soluciones eficientes en evaluación al punto más cercano en el frente de referencia. La última métrica considerada es la distancia generacional inversa, $IGD+$, que consiste en invertir la métrica de distancia generacional con el objetivo de medir la distancia entre el conjunto de soluciones eficientes en evaluación con el frente de referencia. Finalmente, se ha incluido también el tamaño del conjunto de soluciones eficientes, $|\hat{E}|$ y el tiempo de cómputo requerido por cada algoritmo medido en segundos, $Tiempo (s)$. Todas las tablas presentadas en esta sección incluyen el valor medio de cada métrica respecto al conjunto de instancias consideradas, donde el mejor valor es resaltado en negrita.

Para empezar, es necesario ajustar los parámetros de entrada del algoritmo: α y el número de soluciones generadas. Para evitar un sobreajuste, se ha seleccionado un subconjunto de 20 instancias representativas (13% del conjunto completo de instancias) para realizar todos los experimentos preliminares. En concreto, se han probado los valores de $\alpha = \{0,25, 0,50, 0,75, RND\}$, donde RND indica que el valor de α se selecciona aleatoriamente para cada construcción, resultando el constructivo JALEO con un valor de $\alpha = RND$ como la mejor opción para generar soluciones. Para el número de construcciones, se ha probado desde 100 hasta 1000, en intervalos de 100 construcciones, resultando 700 el mejor valor en cuanto al compromiso calidad-tiempo.

IV-A. Pruebas competitivas

El objetivo principal de esta sección es realizar una prueba competitiva entre el algoritmo propuesto y el mejor método previo encontrado en el estado del arte para analizar la eficiencia y eficacia el algoritmo GRASP propuesto (JALEO). El mejor método encontrado en la literatura es Vera et al. [22], donde se presenta un algoritmo evolutivo multi-objetivo (MOEA) para resolver el MOMDP. En concreto, los autores diseñan un *Non-Dominated Sorting Genetic Algorithm* (NSGA-II) [1] aprovechando la implementación ofrecida en la librería JMetal [4]. Para hacer una comparación justa, todos los experimentos han sido ejecutados en el mismo ordenador, utilizando la misma implementación de NSGA-II. Adicionalmente, se incluye el algoritmo evolutivo multi-objetivo basado en descomposición (MOEA/D) [23], para el cual se ha demostrado que es mejor que NSGA-II en rendimiento en trabajos recientes [11]. Ambos algoritmos

han sido ejecutados con los parámetros determinados por los autores del artículo previo, es decir, 250000 evaluaciones y un tamaño de población de 500. En este experimento final se ha usado el conjunto completo de 145 instancias descritas en la Sección IV. La Tabla II muestra los resultados finales de la prueba competitiva.

Algoritmo	$C(R, \hat{E})$	HV	EPS	$IGD+$	$Tiempo (s)$	$ \hat{E} $
JALEO	0.22	0.41	0.15	0.03	1.72	804.54
NSGA-II	0.43	0.34	0.27	0.08	22.23	87.07
MOEA/D	0.26	0.35	0.28	0.12	25.29	20.57

Tabla II: Comparativa de GRASP, NSGA-II, y MOEA/D.

La primera conclusión de estos resultados es que MOEA/D es un método más competitivo que NSGA-II al considerar algoritmos evolutivos multi-objetivo, como sugería Li et al. [11]. Analizando primero la cobertura, JALEO obtiene el mejor valor (0.2223), seguido de cerca por MOEA/D. Los resultados obtenidos por NSGA-II indican que casi la mitad de las soluciones descubiertas por este algoritmo son dominadas por una o más soluciones del frente de referencia. En cuanto al hipervolumen, tanto MOEA/D como NSGA-II alcanzan valores similares (0.3495 vs 0.3511), siendo JALEO considerablemente mejor que ambos en este caso (0.4121). El mismo análisis es realizado al considerar el indicador ϵ , siendo de nuevo JALEO el mejor algoritmo. Además, JALEO es capaz de alcanzar el mejor valor para la métrica $IGD+$. Cabe mencionar que JALEO es capaz de incluir en el conjunto de soluciones eficientes una media de 804 soluciones, mientras que MOEA/D es capaz de incluir una media de solo 20 soluciones. Además, cabe destacar que JALEO es más de 10 veces más rápido que NSGA-II y MOEA/D, requiriendo, de media, menos de 2 segundos para resolver una instancia del MOMDP.

Finalmente, se realiza un test de Friedman no paramétrico para confirmar si hay diferencias estadísticas significativas entre los algoritmos comparados, utilizando las métricas de cobertura e hipervolumen. El valor- p obtenido menor de 0.0001 confirma esta hipótesis, resultando JALEO como el mejor del ranking (1.78), seguido de MOEA/D (1.95) y de NSGA-II (2.28). Analizando estos resultados, JALEO destaca como uno de los algoritmos más competitivos para resolver el MOMDP en cortos tiempos de cómputo. Finalmente, se aplica un test de Wilcoxon no paramétrico por pares para confirmar la superioridad de JALEO sobre MOEA/D. El valor- p menor de 0.0001 respalda esta hipótesis.

V. CONCLUSIONES

En este artículo hemos presentado un enfoque original para resolver una variante multi-objetivo del problema de la máxima diversidad (MOMDP). Los problemas multi-objetivo se adaptan mejor a los problemas de la vida real, donde se considera normalmente más de un solo objetivo. Se proponen dos procedimientos constructivos diferentes siguiendo dos ideas completamente opuestas. Por un lado, JALEO genera soluciones optimizando cada una de las funciones voraces independientemente. Por otro lado, MOFA optimiza

simultáneamente todas las métricas de diversidad consideradas en cada construcción. El análisis experimental muestra que la estrategia seguida en JALEO es capaz de obtener mejores resultados. Se presenta además un procedimiento de búsqueda local eficiente que es capaz de limitar el vecindario explorado para encontrar un óptimo local que respeta las soluciones originalmente incluidas en el conjunto de soluciones eficientes. Se trata de una adaptación original de la metodología GRASP para los problemas de optimización multi-objetivo. La rigurosa experimentación computacional está respaldada con test estadísticos no paramétricos que confirman la calidad de la propuesta, destacando JALEO como el mejor algoritmo metaheurístico para resolver el MOMDP.

Este esquema GRASP se puede aplicar a cualquier problema multi-objetivo de la literatura. El código fuente sobre el que se han ejecutado los experimentos de este artículo está disponible en <https://grafo.etsii.urjc.es/momdp>.

REFERENCIAS

- [1] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [2] K. Dhir, F. Glover, and C.-C. Kuo, "Optimizing diversity for engineering management," in *Proceedings of Engineering Management Society Conference on Managing Projects in a Borderless World*. IEEE, 1993, pp. 23–26.
- [3] A. Duarte, J. Sánchez-Oro, M. G. Resende, F. Glover, and R. Martí, "Greedy randomized adaptive search procedure with exterior path relinking for differential dispersion minimization," *Information Sciences*, vol. 296, pp. 46–60, 2015.
- [4] J. J. Durillo and A. J. Nebro, "jmetal: A java framework for multi-objective optimization," *Advances in Engineering Software*, vol. 42, no. 10, pp. 760–771, 2011.
- [5] E. Erkut and S. Neuman, "Analytical models for locating undesirable facilities," *European Journal of Operational Research*, vol. 40, no. 3, pp. 275–291, 1989.
- [6] T. A. Feo and M. G. Resende, "A probabilistic heuristic for a computationally difficult set covering problem," *Operations Research Letters*, vol. 8, no. 2, pp. 67–71, 1989.
- [7] T. A. Feo, M. G. C. Resende, and S. H. Smith, "A greedy randomized adaptive search procedure for maximum independent set," *Operations Research*, vol. 42, no. 5, pp. 860–878, 1994.
- [8] F. Glover, C.-C. Kuo, and K. S. Dhir, "Heuristic algorithms for the maximum diversity problem," *Journal of information and Optimization Sciences*, vol. 19, no. 1, pp. 109–132, 1998.
- [9] M. J. Kuby, "Programming models for facility dispersion: the p-dispersion and maximum dispersion problems," *Mathematical and Computer Modelling*, vol. 10, no. 10, p. 792, 1988.
- [10] C.-C. Kuo, F. Glover, and K. S. Dhir, "Analyzing and modeling the maximum diversity problem by zero-one programming," *Decision Sciences*, vol. 24, no. 6, pp. 1171–1185, 1993.
- [11] H. Li, K. Deb, Q. Zhang, P. N. Suganthan, and L. Chen, "Comparison between MOEA/D and NSGA-III on a set of novel many and multi-objective benchmark problems with challenging difficulties," *Swarm and Evolutionary Computation*, vol. 46, pp. 104–117, 2019.
- [12] M. Li and X. Yao, "Quality evaluation of solution sets in multiobjective optimisation: A survey," *ACM Computing Surveys (CSUR)*, vol. 52, no. 2, p. 26, 2019.
- [13] A. López-Sánchez, J. Sánchez-Oro, and A. Hernández-Díaz, "Grasp and vns for solving the p-next center problem," *Computers & Operations Research*, vol. 104, pp. 295 – 303, 2019.
- [14] M. Lozano, D. Molina, C. García *et al.*, "Iterated greedy for the maximum diversity problem," *European Journal of Operational Research*, vol. 214, no. 1, pp. 31–38, 2011.
- [15] R. Martí and F. Sandoya, "Grasp and path relinking for the equitable dispersion problem," *Computers & Operations Research*, vol. 40, no. 12, pp. 3091–3099, 2013.
- [16] R. Martí, M. Gallego, A. Duarte, and E. G. Pardo, "Heuristics and metaheuristics for the maximum diversity problem," *Journal of Heuristics*, vol. 19, no. 4, pp. 591–615, 2013.
- [17] F. Parreño, R. Álvarez Valdés, and R. Martí, "Measuring diversity. a review and an empirical analysis," *European Journal of Operational Research*, vol. 289, no. 2, pp. 515–532, 2021.
- [18] S. Pérez-Peló, J. Sánchez-Oro, and A. Duarte, "Finding weaknesses in networks using greedy randomized adaptive search procedure and path relinking," *Expert Systems*, vol. 37, no. 6, p. e12540, 2020.
- [19] G. Qian, S. Sural, Y. Gu, and S. Pramanik, "Similarity between euclidean and cosine angle distance for nearest neighbor queries," in *Proceedings of the 2004 ACM symposium on Applied computing*. New York, NY, USA: Association for Computing Machinery, 2004.
- [20] R. Real and J. M. Vargas, "The probabilistic basis of jaccard's index of similarity," *Systematic Biology*, vol. 45, no. 3, pp. 380–385, 1996.
- [21] S. Sohrabi, K. Ziarati, and M. Keshtkaran, "A greedy randomized adaptive search procedure for the orienteering problem with hotel selection," *European Journal of Operational Research*, vol. 283, no. 2, pp. 426–440, 2020.
- [22] K. Vera, F. Lopez-Pires, B. Baran, and F. Sandoya, "Multi-objective maximum diversity problem," in *2017 XLIII Latin American Computer Conference (CLEI)*, 2017, pp. 1–9.
- [23] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.