

Algoritmo de Optimización de Colonia de Hormigas Multiobjetivo Aplicado al Problema de la Mochila Multidimensional

Daniel Soto¹, Wilson Soto^{1,2,*} y Yoan Pinzón¹

¹ Grupo de Algoritmos y Combinatoria ALGOS–UN
Universidad Nacional, Bogotá, Colombia
dsoto.7@gmail.com – ypinzon@unal.edu.co

² Grupo de Sistemas Inteligentes y de Información Espacial SIGA
Universidad Central, Bogotá, Colombia
wsotof@ucentral.edu.co

Abstract. This paper presents a multiobjective *Ant Colony Optimization* (ACO) algorithm. The proposed algorithm is applied to the multidimensional knapsack problem. The multidimensional knapsack problem is combinatorial optimization problem that consists in finding a subset of objects that maximizes total profit while satisfying some resource constraints. We show that algorithm proposed obtains better results compared with an important algorithm in a selected dataset.

Keywords: Multiobjective Ant Colony Optimization, Metaheuristic, Multidimensional Knapsack Problem.

Resumen. Este artículo presenta un algoritmo de optimización de colonia de hormigas (*Ant Colony Optimization* – ACO) multiobjetivo. El algoritmo propuesto es aplicado al problema de la mochila multidimensional. El problema de la mochila multidimensional es un problema de optimización combinatoria que consiste en encontrar un subconjunto de objetos que maximicen el beneficio total mientras se satisfacen ciertas restricciones. Se muestra como el algoritmo propuesto obtiene mejores resultados comparado con un importante algoritmo en un conjunto de datos seleccionado.

Palabras clave: Optimización de Colonia de Hormigas Multiobjetivo, Metaheurísticas, Problema de la Mochila Multidimensional.

1 Introducción

La optimización de varios objetivos tiene muchas aplicaciones en la vida real entre las cuales están la producción industrial, entornos financieros, inventarios y complejidad computacional [8]. Determinar un conjunto de soluciones óptimas para este tipo de problemas es el principal aspecto de la optimización multiobjetivo. Uno de los problemas que hace parte de este grupo es el problema de la mochila

* Autor para correspondencia

multidimensional, además, de estar clasificado como un problema de optimización combinatoria NP-hard [10,12].

Si se considera un problema de n elementos y para cada uno su correspondiente beneficio y peso. El problema de la mochila es la tarea de empacar algunos de esos elementos en una mochila m de capacidad c , tal que la suma del beneficio de esos elementos sea máxima. Ahora, si al considerar más de una mochila (objetivos) en las cuales su capacidad (restricciones) pueda ser diferente y se respete su límite logrando el máximo beneficio (optimización) se obtiene el problema de la mochila multidimensional. Por lo tanto, podemos asemejar el problema de la mochila multidimensional a un problema de optimización de múltiples objetivos.

En años recientes, algunos de los trabajos relacionados para resolver el problema de la mochila multidimensional son: desde el enfoque de los algoritmos evolutivos (EA) [14,17,18] y optimización por colonia de hormigas (ACO) [1,3,4,8,9].

El problema de la mochila multidimensional se puede aplicar en diversos problemas reales de la bioinformática, economía y robótica [2,4,6,16]

Las contribuciones del presente trabajo son: presentar un algoritmo multiobjetivo basado en ACO y en el algoritmo MMAS descrito en [15]. El algoritmo propuesto obtiene mejores resultados en varios ejemplos comparado con el algoritmo SPEA2 (Streng Pareto Evolutionary Algorithm versión 2).

Este artículo tiene la siguiente estructura: la sección 2 define formalmente el concepto del problema de la mochila multidimensional y el algoritmo colonia de hormigas. La sección 3 explica en detalle el algoritmo propuesto. La sección 4 detalla los parámetros y resultados experimentales y en la última sección aparecen las conclusiones.

2 Conceptos básicos

Los problemas de optimización multiobjetivo – en inglés *Multiobjective Optimization Problem* (MOP) – formalmente se pueden expresar como:

Definición 1. (Problemas de Optimización Multiobjetivo), *Un MOP en general incluye un conjunto de n parámetros (variables de decisión), un conjunto de k funciones objetivo y un conjunto de m restricciones. Las funciones objetivo y las restricciones son funciones de las variables de decisión. El objetivo de la optimización es*

$$\begin{array}{ll} \text{maximizar} & y = f(x) = (f_1(x), f_2(x), \dots, f_k(x)) \\ \text{sujeto a} & e(x) = (e_1(x), e_2(x), \dots, e_m(x)) \leq 0 \\ \text{donde} & x = (x_1, x_2, \dots, x_n) \in X \\ & y = (y_1, y_2, \dots, y_k) \in Y \end{array} \quad (1)$$

x es el vector de decisión, y es vector objetivo, X se denota como el espacio de decisión y Y se llama el espacio objetivo. Las restricciones $e(x) \leq 0$ determinan el conjunto de soluciones factibles.

Definición 2. (Conjunto de soluciones factibles), el conjunto factible X_f es definido como el conjunto de vectores de decisión x que satisface las restricciones $e(x)$:

$$X_f = \{x \in X \mid e(x) \leq 0\} \quad (2)$$

La imagen de X_f , i.e. la región factible dentro del espacio de búsqueda, es denotado como $Y_f = f(X_f) = \bigcup_{x \in X_f} \{f(x)\}$.

W.L.O.G. para un problema de minimización, las definiciones anteriormente dadas son similares.

El problema de la mochila multidimensional – *Multidimensional Knapsack Problem* (MKP) – es un problema de optimización multiobjetivo, el cual se compone de un número arbitrario de mochilas (objetivos) cada una con capacidad (restricción) y un conjunto de elementos que tienen asociado un peso y un beneficio. La tarea consiste en encontrar un subconjunto de estos elementos que colocados dentro de las mochilas maximice el beneficio total teniendo en cuenta las capacidades de estas [17]. Formalmente el problema MKP se define como:

Definición 3. (Problema de la mochila multidimensional), dado un conjunto n de elementos y un conjunto m de mochilas, con:

$p_{i,j}$ = beneficio del elemento j respecto a la mochila i ,

$w_{i,j}$ = peso del elemento j respecto a la mochila i ,

c_i = capacidad de la mochila i ,

encontrar un vector $x = (x_1, x_2, \dots, x_n) \in \{0,1\}^n$, tal que las restricciones de capacidad

$$e_i(x) = \sum_{j=1}^n w_{i,j} \cdot x_j \leq c_i \quad (1 \leq i \leq m) \quad (3)$$

son satisfechas y para la cual $f(x) = (f_1(x), f_2(x), \dots, f_m(x))$ es máximo, donde

$$f_i(x) = \sum_{j=1}^n p_{i,j} \cdot x_j \quad (4)$$

y $x_j = 1$ si y solo si el elemento j es seleccionado.

2.1 Optimización por colonia de hormigas

Optimización por colonia de hormigas – Ant Colony Optimization (ACO) – es una metodología propuesta por Marco Dorigo [5]. Esta metodología estudia sistemas artificiales tomando como inspiración el comportamiento real de las colonias de hormigas, las cuales se usan para resolver problemas de optimización combinatoria.

Los componentes de un algoritmo de colonia de hormigas son:

- Hormigas: tienen por objetivo realizar un recorrido (Ciclo Completo Cerrado) dentro del problema dado por el usuario.
- Memoria de feromonas: es utilizada para tener una historia de los recorridos efectuados por las hormigas y poder cambiar la intensidad de cada posible ruta a cada nueva iteración.
- Probabilidad de selección de un camino: toma de decisión de cada hormiga al momento de escoger un camino para pasar de un nodo a otro. La fórmula que define la probabilidad es llamada regla de transición de estados aleatorios o regla de Bonabeau [7]:

$$p_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}(t)^\alpha \cdot \eta_{ij}(t)^\beta}{\sum_{l \in J_i^k} \tau_{il}(t)^\alpha \cdot \eta_{il}(t)^\beta} & \text{si } j \notin J_i^k \\ 0 & \text{si } j \in J_i^k \end{cases} \quad (5)$$

La regla de Bonabeau (5) está basada en los conceptos de visibilidad (la visibilidad

es el numero inverso de la distancia entre dos nodos i y j , y $\eta_{ij} = \frac{1}{d_{ij}}$ de la traza de feromona dejada sobre un arco del camino, es decir, la traza de feromona dejada entre

un nodo i y una j en una iteración t es $\tau_{ij}(t)^\alpha$, α y β son dos parámetros que tiene el control sobre la importancia relativa de la intensidad de la feromona y la visibilidad, así se puede decir que con $\alpha = 0$ solamente la visibilidad es evaluada, por el contrario si $\beta = 0$ solamente la intensidad de las feromonas es evaluada.

Como resultado de la evaluación de la regla se tiene una probabilidad para cada posible camino, $p_{ij}^k(t)$ es la probabilidad que la hormiga k tome la ruta hacia j desde i en la iteración t , la probabilidad solamente es calculada para las villas j que no han sido visitadas aún por la hormiga k .

Después de haber completado los recorridos de todas las hormigas escogidas es necesario dejar una cantidad de feromona sobre cada arco de cada recorrido en función de la calidad de la solución encontrada para cada recorrido, esta función es llamada impregnación de las feromonas y se usa la regla de intensificación [7]:

$$\Delta \tau_{ij}^k(t) = \begin{cases} \frac{Q}{L^k(t)} & \text{si } (i, j) \in T^k \\ 0 & \text{si } (i, j) \notin T^k \end{cases} \quad (6)$$

En (6), $L^k(t)$ es la longitud del recorrido de la hormiga k en la iteración t y Q es una constante, la regla es aplicada a todos los arcos que pertenecen al recorrido realizado por la hormiga k en la iteración t .

Otra función importante es la función de atenuación de las feromonas, proceso que evapora las trazas de feromonas garantizando que la memoria global del sistema olvida las malas soluciones encontradas con la regla de evaporación enunciada en [7]:

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (7)$$

En (7) ρ es el índice de evaporación de la feromona, es decir, el índice con el cual la traza de feromonas se va a evaporar en relación con las iteraciones y $\Delta\tau_{ij}(t)$ es la función de impregnación.

2.2 Métricas

Para el análisis de resultados se utilizaron las métricas de desempeño de [17], que están basadas en los conceptos de óptimo de Pareto y Dominancia (Ecuación 8).

El óptimo de Pareto en un problema multiobjetivo, es una solución óptima cuando no existe otra solución tal que mejore en un objetivo sin empeorar al menos uno de los otros. Esta solución óptima se determina con el concepto de dominancia.

Definición 4. (Dominancia), para dos vectores de decisión cualesquiera p y q ,

$$\begin{aligned} p > q \text{ (} p \text{ domina } q \text{)} & \Leftrightarrow f(p) > f(q) \\ p \geq q \text{ (} p \text{ domina débilmente } q \text{)} & \Leftrightarrow f(p) \geq f(q) \\ p \approx q \text{ (} p \text{ es indiferente a } q \text{)} & \Leftrightarrow f(p) \not\geq f(q) \wedge f(q) \not\geq f(p) \end{aligned} \quad (8)$$

Cuando las soluciones son dominantes en el sentido de que no hay otras soluciones superiores a estas en el espacio de búsqueda y cuando todos los objetivos son considerados, este conjunto se denomina conjunto no dominado o frente óptimo de Pareto y es la mejor colección de soluciones al problema.

Las métricas usadas básicamente evalúan la distancia, la distribución y la extensión de la solución generada por los algoritmos.

- **Métrica M_1** distancia promedio al conjunto Óptimo de Pareto X_p :

$$M_1(A) = \frac{1}{|A|} \sum_{a \in A} \min \{\|a - x\| \mid x \in X_p\} \quad (9)$$

- **Métrica M_2** distribución del conjunto A , donde, $\sigma > 0$ corresponde a un parámetro de vecindad:

$$M_2(A) = \frac{1}{|A|-1} \sum_{a \in A} |\{b \in A \mid \|a - b\| > \sigma\}| \quad (10)$$

- **Métrica M_3** extensión del conjunto A :

$$M_3(A) = \sqrt{\sum_{i=1}^n \max \{\|a_i - b_i\| \mid a, b \in A\}} \quad (11)$$

3 Algoritmo propuesto

El algoritmo propuesto (Algoritmo P–MOACO¹) está basado en MMAS [15] y en el trabajo de Khichane *et al.* [11]. El algoritmo propuesto (Algoritmo 1) realiza dos cambios en el algoritmo base y crea el planteamiento de la visibilidad calculada como una propiedad variable de cada elemento. A continuación se muestra en detalle las principales características del algoritmo propuesto:

- ✓ Una variabilidad adaptativa que refiere al cambio de los parámetros α (alfa) y β (beta) para la población (– valores entre 0 y 5 – [13]),
- ✓ el cambio de la importancia relativa de los objetivos a optimizar y
- ✓ la evaluación de las propiedades de los elementos.

Algoritmo 1 Algoritmo P–MOACO

```

1: Iniciar  $\tau$ 
2: Generar 2 especies para la población (Evaluación diferente del objetivo)
3: Para  $t = 1 \dots t_{max}$ 
4:   Genera_Probabilidad
5:   Generar número global  $\alpha$  (alfa) de forma aleatoria
6:   Generar número global  $\beta$  (beta) de forma aleatoria
7:   Para toda hormiga  $k = 1 \dots k_{max}$ 
8:      $V_{recorrido}[ ] = 0$ 
9:     Elegir un elemento de forma aleatoria
10:    Para todo elemento no seleccionado
11:      Selección_Elemento
12:      Intensificar la traza de feromonas (Ecuación 6)
13:      Evaporar la traza de forma uniforme (Ecuación 7)
14:    Fin Para
15:    Ingresar al frente global aplicando dominancia
16:  Fin Para
17: Fin Para
18: Refinamiento del frente global

```

Sobresalen los siguientes detalles del algoritmo P–MOACO (Algoritmo 1):

- En la línea 2 se garantiza una evaluación equitativa (E) de la solución en dos formas diferentes: beneficio ($p_{i,j}$) o beneficio/peso $\left(\frac{p_{i,j}}{w_{i,j}} \right)$.
- En la línea 4 se encuentra el procedimiento Genera_Probabilidad, que genera números de forma aleatoria que sirven como probabilidad P para determinar la importancia de los m objetivos, según la Ecuación 12.

¹ Pareto Multi–Objective Ant Colony Optimization

$$\sum_{\ell=0}^m P_{\ell} = 1 \quad (12)$$

- En la línea 11 se encuentra el procedimiento *Seleccion_Elemento*, que selecciona un elemento dentro de la lista de elementos restantes basado en la regla de transición (véase Ecuación 5), donde:

$$\eta_{i,j} = \sum_{\ell=0}^m E_{\ell} \cdot P_{\ell} \quad (13)$$

- Las líneas 15 y 18 están relacionadas con el *óptimo de Pareto*.

El algoritmo propuesto se puede clasificar en la taxonomía MOACO (*Multi-Objective Ant Colony Optimization*) [4] de la siguiente forma (Tabla 1):

Tabla 1. Algoritmo propuesto en la taxonomía MOACO

Algoritmo	$[\tau]$	$[\eta]$	Colonia	Agregación/Peso	Actualización	Archivo Pareto
P-MOACO	1	–	simple	– / dinámico	No-dominado	–

4 Experimentos y resultados

Los resultados experimentales del algoritmo propuesto son comparados con los resultados del algoritmo SPEA2 (*Streng Pareto Evolutionary Algorithm* versión 2). El algoritmo SPEA2 es un referente para solucionar problemas multiobjetivo que usan el *óptimo de Pareto* [2].

Además, el algoritmo SPEA2, en [20] mostró como obtuvo un comportamiento similar a NSGA-II (Nondominated Sorting Genetic Algorithm) y superior a los algoritmos SPEA (primera versión de SPEA2) y PESA (Pareto Envelope-Based Selection Algorithm) para el problema de la mochila multidimensional con 2,3 y 4 objetivos, incluso en algunos casos obtuvo mejores resultados experimentales con respecto a NSGA-II. Los resultados reportados corresponden a la mediana después de 25 ejecuciones. El tiempo de ejecución promedio para 100 elementos es 36.37, para 250 elementos 73.13 y para 500 elementos es 243.51 minutos. Los experimentos fueron realizados en un equipo Intel Core 2 Duo de GHz, 3GB de memoria y sobre Ubuntu Karmic Koala.

4.1 Parámetros

Las metaheurísticas como método heurístico especialmente para resolver problemas de optimización combinatoria deben tener en cuenta el balance entre la exploración (búsquedas diversas para descubrir nuevas áreas) y la explotación (búsqueda intensa en el área actual para encontrar soluciones óptimas). Los parámetros en ACO para modificar el comportamiento de las hormigas y lograr el equilibrio entre la búsqueda local y global son la intensificación y la diversificación, respectivamente.

La diversificación se refiere a la sensibilidad de las hormigas con respecto a la huella dejada por la feromona, si este valor decrece, la sensibilidad de las hormigas a la traza de la feromona es baja. La intensificación se relaciona con la tasa de evaporación de la feromona, si existe un decremento en este valor, implica que la feromona se evapora más lentamente. Entonces, si se incrementa la habilidad de exploración de las hormigas, usualmente se encuentran mejores soluciones, pero se ve afectado el tiempo para encontrarlas.

Para los experimentos reportados a continuación, se tiene a α , β y ρ calculados de forma aleatoria y un número de hormigas de 100. Un número de 1000 iteraciones o ciclos fue determinado en base a las métricas usadas (Figura 1).

4.2 Conjuntos de datos

El algoritmo P-MOACO fue probado con datos extraídos del Swiss Federal Institute of Technology Zurich ETH2 y comparado con los resultados del algoritmo SPEA2 sobre los mismos conjuntos de datos.

Los conjuntos de datos seleccionados tienen 3 objetivos (m) y 100–250–500 elementos (n). La elección de estos conjuntos es para poder visualizar gráficamente los resultados.

4.3 Resultados

Para los problemas seleccionados no se tiene un conjunto óptimo reportado, es por ello que la métrica M1 no es tomada en cuenta. Con respecto a las restantes métricas (Figura 1), se observa como el incremento en el número de iteraciones o ciclos mejoran los resultados correspondientes a la distribución y extensión de la solución.

Los resultados obtenidos con el algoritmo P-MOACO son comparados con el algoritmo SPEA2 para cada uno de los conjuntos de datos seleccionados. Fueron seleccionados los mejores resultados según las métricas para todas las ejecuciones realizadas (Tabla 2). Los resultados para 100 elementos se pueden ver en la Figura 2, 250 elementos en la Figura 3 y 500 elementos en la Figura 4.

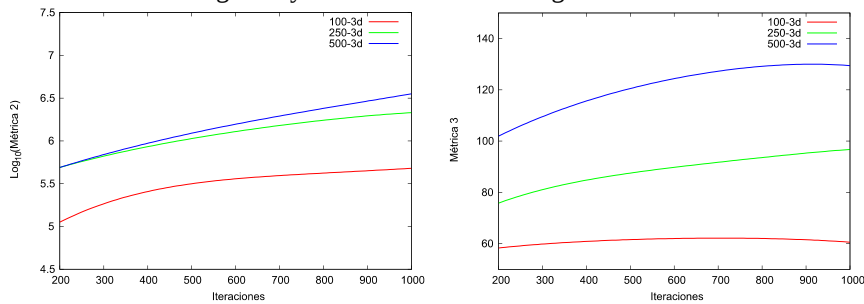


Fig. 1. Resultados métrica M_2 (izq.) y métrica M_3 (der.) del algoritmo P-MOACO para los conjuntos de datos de 100, 250 y 500 elementos

² <http://www.tik.ee.ethz.ch/sop/download/supplementary/testProblemSuite/>

Tabla 2. Resultados de la mejor evaluación del algoritmo P-MOACO y el algoritmo SPEA para el conjunto de 100, 250 y 500 elementos con 3 mochilas.

	100		250		500	
	SPEA2	P-MOACO	SPEA2	P-MOACO	SPEA2	P-MOACO
No. soluciones óptimas	354	1090	5.028	5.791	38.961	63.796
Métrica M_2	330	1176	5.171	6.303	49.335	95.094
Métrica M_3	410	1133	5.464	6.572	63.166	137.542

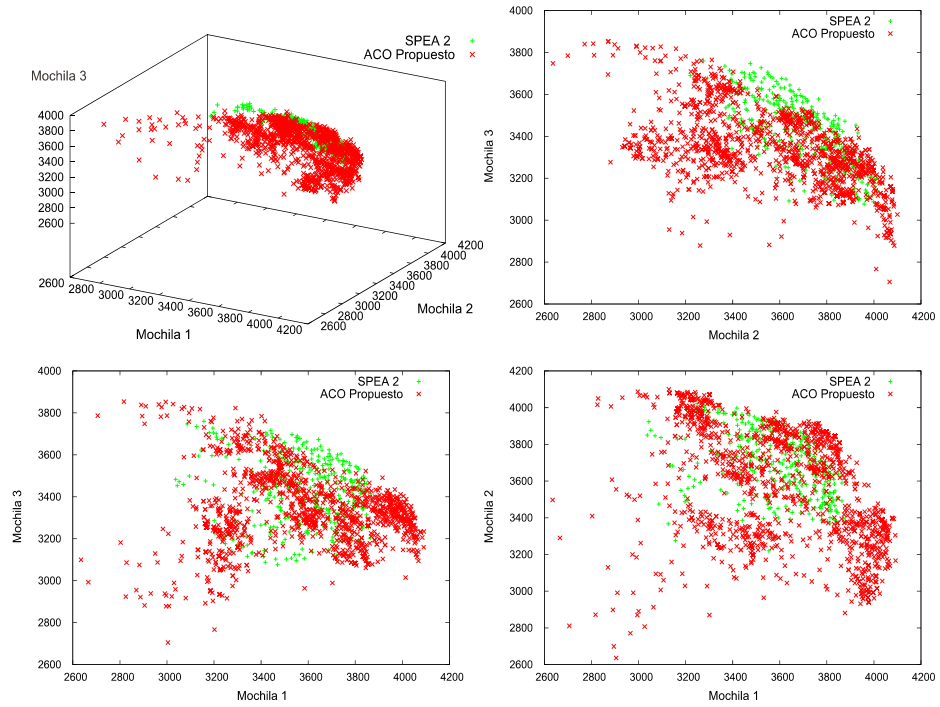


Fig. 2. Comportamiento de los algoritmos SPEA2 y P-MOACO con $m=3$ y $n=100$

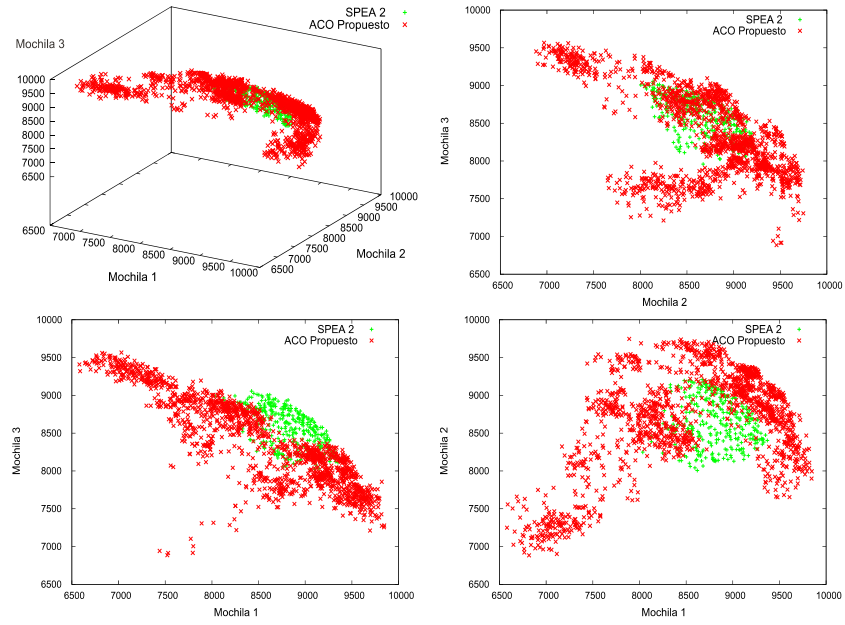


Fig. 3. Comportamiento de los algoritmos SPE2 y P-MOACO con $m=3$ y $n=250$

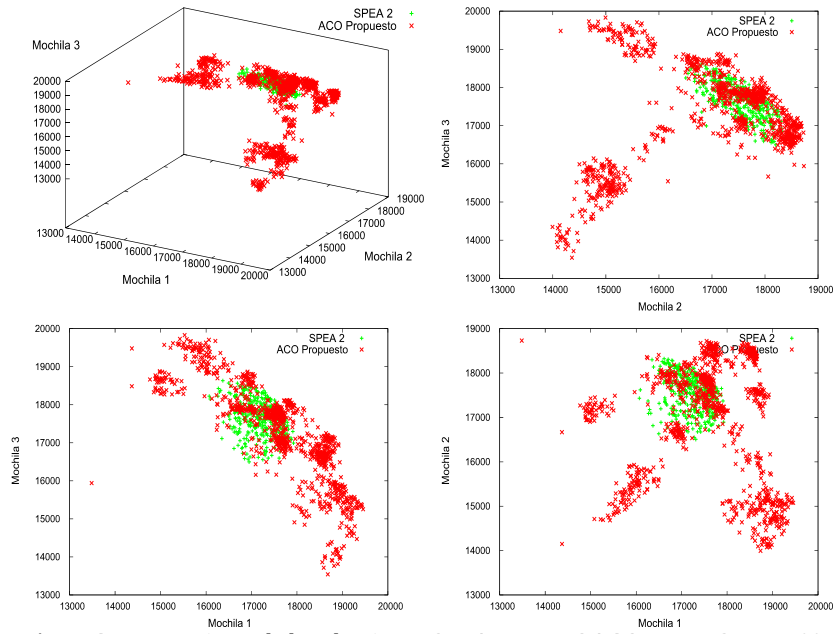


Fig. 4. Comportamiento de los algoritmos SPEA2 y P-MOACO con $m=3$ y $n=500$

En las figuras (Figura 2, Figura 3 y Figura 4) es evidente observar como el algoritmo P-MOACO tiene una extensión más grande de soluciones óptimas y una distribución entre soluciones más concentrada con respecto al algoritmo SPEA2. Las diferentes vistas de cada una de las figuras reflejan dicha afirmación, al igual que los resultados presentados en la Tabla 2.

CONCLUSIONES

Este artículo tiene descrita una versión modificada de un algoritmo ACO multiobjetivo y su aplicación a la solución óptima de búsqueda en un problema de optimización combinatoria. Los resultados muestran que el algoritmo propuesto tiene un mejor desempeño a un algoritmo evolutivo basado en el frente de pareto y de igual forma indican el potencial de la aproximación de ACO para resolver problemas de restricción, específicamente, en el problema de la mochila multidimensional.

La importancia del algoritmo propuesto está en su aplicación a problemas reales relacionados con optimización combinatoria, donde, para cualquier número de elementos n y cualquier número de objetivos m se pretende minimizar o maximizar una función objetivo. Uno de los trabajos futuros es mejorar el algoritmo propuesto usando el paradigma de programación paralela esperando que se amplíe el espacio de búsqueda y encontrar de forma más rápida soluciones óptimas del problema.

Referencias

- [1] Aghezzaf Brahim and Naimi Mohamed. The two-stage recombination operator and its application to the multiobjective 0/1 knapsack problem: A comparative study. *Computers and Operations Research*. Vol. 36(12):3247–3262, December 2009.
- [2] Aimin Zhou, Bo--Yang Qu, Hui Li, Shi-Zheng Zhao, Ponnuthurai Nagarathnam Suganthan and Qingfu Zhang. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32–49, March 2011.
- [3] Alaya Inés, Solnon Christine and Ghédira Khaled. Ant Colony Optimization for Multi-objective Optimization Problems, in *Proc. of the 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007)*, 1:450–457, IEEE Computer Society Press, October 2007.
- [4] Angelo, Jaqueline S. y Barbosa, Helio J.C., On Ant Colony Optimization Algorithms for Multiobjective Problems. *Ant Colony Optimization: Methods and Applications*. Ed. Avi Ostfeld, InTech. pp. 53–74, 2011.
- [5] Dorigo M. *Optimization, Learning and Natural Algorithms*, PhD thesis. Politecnico di Milano, Italie, 1992.
- [6] Dréo, Johann, “Adaptation de la méthode des colonies de fourmis pour l'optimisation en variables continues. Application en génie biomédica”, Ph.D. thesis, Universidad de Paris XII, France, 2003.
- [7] Dréo, Johann, Pérowsky Alain, Siarry Patrick and Taillard Éric. “Métaheuristiques pour l'optimisation difficile”, Eyrolles, 2003.
- [8] Fidanova, S., Probabilistic Model of Ant Colony Optimization for Multiple Knapsack Problem, *Large Scale Scientific Computing, Lecture Notes in Computer Science* 4818 (2008), 545–552.

- [9] Fidanova, S., Ant Colony Optimization for Multiple Knapsack Problem and Model Bias, Numerical Analysis and Applications 2004, Lecture Notes in Computer Science 3401 (2005), 282–289.
- [10] Kellerer, H., U. Pferschy and D. Pisinger, “Knapsack Problems”, Springer Verlag, Berlin, Germany. 2004.
- [11] Khichane, Madjid, Patrick Albert and Christine Solnon, “Un modèle réactif pour l'optimisation par colonies de fourmis: application à la satisfaction de contraintes”, Journées Francophones de Programmation par Contraintes JFPC'09, France, 2009.
- [12] Martello, Silvano and Paolo Toth, “Knapsack problems: algorithms and computer implementations”, John Wiley & Sons Inc., New York, USA, 1990.
- [13] Olive, Xavier, Résolution de conflits par algorithmes stochastiques parallèles, Master Recherche, École nationale supérieure de l'aéronautique et de l'espace, Toulouse, France, 2006.
- [14] Shah Ruchit and Reed Patrick. Comparative analysis of multiobjective evolutionary algorithms for random and correlated instances of multiobjective d--dimensional knapsack problems. European Journal of Operational Research, Elsevier, 211(3):466–479, June 2011.
- [15] Stutzle, Thomas y Holger Hoos, MAX--MIN Ant System. Future Generation Computer Systems, 16 (2000), 889–914.
- [16] Tomoyuki Hiroyasu, Mitsunori Miki, Seiichi Nakayama and Yoshiko Hanada. Multi-Objective Optimization of Diesel Engine Emissions and Fuel Economy Using SPEA2+, in Hans-Georg Beyer et al. (editors), 2005 Genetic and Evolutionary Computation Conference (GECCO'2005), 2:2195–2196, ACM Press, New York, USA, June 2005 .
- [17] Zitzler, E., “Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications”. Ph.D. thesis, Swiss Federal Institute of Technology (ETH) Zurich, Switzerland, 1999.
- [18] Zitzler, E. and L. Thiele, Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach, IEEE Transactions on Evolutionary Computation, 3 (1999), 257–271.
- [19] Zitzler, E. and L. Thiele, “An Evolutionary Algorithm for Multiobjective Optimization: The Strength Pareto Approach”, Technical Report, Swiss Federal Institute of Technology (ETH) Zurich, Switzerland, 1998.
- [20] Zitzler, E., M. Laumanns and L. Thiele, “SPEA2: Improving the Strength Pareto Evolutionary Algorithm”, Technical Report, Swiss Federal Institute of Technology (ETH) Zurich, Switzerland, 2001.