

# META HEURÍSTICA BASADA EN COLONIA DE HORMIGAS EN N ETAPAS PARA PROBLEMAS DE ALTA DIMENSIÓN.

Daniel Hernández<sup>1</sup>, Steven Hurtado<sup>1</sup>, Henry Pérez<sup>1</sup>, Amilkar Puris<sup>1</sup>, Pavel Novoa-Hernández<sup>2</sup>  
daniel.hernandez2016@uteq.edu.ec, steven.hurtado2015@uteq.edu.ec, henry.perez2015@uteq.edu.ec,  
apuris@uteq.edu.ec, pavel.novoa@ucn.cl

<sup>1</sup>Universidad Técnica Estatal de Quevedo, EC120508, Av. Quito, Km 1,5 vía a Santo Domingo, Fax: 59352753303, Teléfono: 59352762382

<sup>2</sup>Escuela de Ciencias Empresariales, Universidad Católica del Norte, Coquimbo, Chile.

## ABSTRACT

Ant Colony Optimization is a population metaheuristic inspired by the behavior of natural ants, specifically their ability to find the shortest path between their nest and the food source. This search mechanism has been tested in discrete problems, establishing itself as a good option for this field of application. In previous works, it was shown that dividing the exploration process of these algorithms into 2 stages considerably improves their performance in terms of time and the quality of the results. In this context, we present, in this work, a generalization of the exploitation process by stages for instances of the Medium and High-Dimension of the Traveling Salesman Problem. For the tests, 5 instances of different sizes were selected and 4 variants of the algorithm were analyzed. The results corroborated that the process of division into stages is good for the performance of the algorithm, reaching the best results with 4 stages.

**KEYWORDS:** Ant Colony System (ACS), Two-Stage Ant Colony Optimization (TS-ACO), Travel Salesman Problem (TSP).

**MSC:** 91B32

## RESUMEN

La optimización basada en Colonia de Hormigas es una metaheurística poblacional inspirada en el comportamiento de las hormigas naturales, específicamente en su capacidad de encontrar el camino más corto entre su nido y la fuente de alimento. Este mecanismo de búsqueda ha sido probado en problemas discretos consolidándose como una buena opción para este campo de aplicación. En trabajos previos, se demostró que dividir en 2 etapas el proceso de exploración de estos algoritmos, mejora considerablemente su rendimiento en cuanto al tiempo y la calidad de los resultados. En este contexto, presentamos, en este trabajo una generalización del proceso de explotación por etapas para instancias del Viajante de Comercio de mediana y alta dimensión. Para las pruebas se seleccionaron 5 instancias de diferentes tamaños y se analizaron 4 variantes del algoritmo. Los resultados corroboraron que el proceso de división en etapas es beneficioso para el rendimiento del algoritmo alcanzando los mejores resultados con 4 etapas.

**PALABRAS CLAVES:** Sistema de Colonia de Hormigas (ACS), Optimización de Colonia de Hormigas en N Etapas (NS-ACO), Problema del Viajante de Comercio (TSP)

## 1. INTRODUCCIÓN

Ant Colony Optimization (ACO) (Dorigo et al., 2006) es una metaheurística inspirada en el comportamiento de búsqueda del camino más corto en las colonias de hormigas. Este algoritmo fue desarrollado para resolver problemas combinatorios como el problema del Viajante de Comercio, el problema de Asignación Cuadrática, el problema de Ordenamiento Secuencial, Programación de Producción, Horarios, Enrutamiento de Telecomunicaciones, Planificación de inversiones, entre otros. El mecanismo de exploración que propone esta metaheurística supone un conjunto de  $k$  hormigas que se mueven por un grafo que representa el problema, construyendo soluciones de manera independiente e iterativa. Luego que cada hormiga termina, se ajustan los valores de feromona de la etapa y se vuelve a realizar este mismo proceso por varias iteraciones. Logrando que mediante varias hormigas artificiales se creen soluciones a un problema de optimización. Si bien este mecanismo es el encargado de la convergencia del algoritmo, se infiere que puede ser lento si resolvemos problemas de mediana o alta dimensión.

Esta situación fue identificada en (Puris et al., 2010) y se presentó una estrategia de exploración para los algoritmos ACO conocida como Colonia de Hormigas en dos Etapas (Two-Stage ACO). Esta estrategia

propuso dividir la exploración que realizan las hormigas en dos fases. Para ello se incorporó un parámetro ( $r$ ) que regula el tamaño de cada etapa en el proceso de búsqueda. La idea fue probada en diferentes problemas combinatorios como el Viajante de Comercio, Asignación Cuadrática, Planificación de la Fuerza de Trabajo y Asignación de Tareas. En todos los casos de estudio se logró disminuir el costo computacional de la búsqueda manteniendo la calidad de las soluciones. La investigación anterior sirvió como premisa para el presente trabajo, cuyo objetivo fue determinar el comportamiento del proceso de división en etapas para instancias del problema de Viajante de Comercio de mediana y alta dimensión. El trabajo está estructurado de la siguiente manera; Seguidamente se hace un análisis del estado del arte, luego se presenta una descripción del proceso de división en etapas y finalmente se presenta el análisis de los resultados.

## 2. TRABAJOS RELACIONADOS

Desde el desarrollo de la metaheurística ACO varios investigadores han expuesto algunas variantes para solventar ciertos problemas tales como caer en óptimos locales y la calidad de las soluciones. Algunas de las ideas más innovadoras se discuten a continuación.

En (Pintea et al., 2016) se presenta un nuevo modelo de optimización de colonias de hormigas, en el que se enfatiza el uso de múltiples colonias con diferentes niveles de sensibilidad en la feromona de la hormiga, en el desarrollo de la experimentación se descubrió que las colonias reaccionan de manera diferente al entorno cambiante, teniendo en cuenta su nivel de sensibilidad, con lo cual consigue que se amplíe la exploración del espacio de solución. Realizando diversos experimentos numéricos con esta nueva variante sobre problemas del TSP y su versión generalizada se obtuvo que la exploración y explotación resultan significativamente mejores y conducen a una solución ACO potencialmente óptima de los problemas de optimización.

Una interesante aplicación del algoritmo de colonias de hormigas se lleva a cabo en (Zhang and Zhang, 2018) donde proponen encontrar la trayectoria más corta dinámica desde una fuente a un destino en una red de tráfico dada, de manera que se optimice el tiempo de viaje para el viajero cuando las condiciones de tráfico están en un estado de cambio dinámico. Para ello se desarrolló un algoritmo novedoso de la ruta más corta basado en algoritmos genéticos de hibridación y colonias de hormigas. Finalmente el algoritmo híbrido se probó en una red de tráfico real, y se demostró que con este algoritmo se podría encontrar la ruta más óptima de una red de tráfico dinámico y podría ser útil para la navegación de vehículos.

De forma similar en (Ibrahim et al., 2012) se crea e implementa una nueva técnica (ACOISIT) para medir los volúmenes ventriculares en la segmentación de imágenes de RMC, basado en optimización de colonias de hormigas con umbral aislado saliente, además, ésta técnica está asentada en la delimitación automática del borde sangre-miocardio. Durante la fase de pruebas se pudo conocer que la integración de la segmentación de umbral saliente en ACOISIT pudo resolver regiones ambiguas y condujo a una detección de límites precisa. En este sentido es válido afirmar que la técnica ACOISIT es significativamente más rápida (procesamiento paralelo por diferentes agentes inteligentes (hormigas)) y más consistente que las segmentaciones manuales.

Por otra parte, una experimentación llevada a cabo en (Kefi et al., 2017) puso a prueba los siguientes algoritmos: Ant System (AS), Elitist Ant System (EAS), Best-Worst Ant System (BWAS), MAX – MIN Ant System (MMAS) y Ant Colony System (ACS). Esto con el objetivo de ponerse a prueba en la resolución y comparación de enfoques meta heurísticos claves para resolver el problema del TSP, así mismo se describe un enfoque que combina ACO con búsqueda local como algoritmos de 2 y 3 opt para obtener la mejor solución en comparación con ACO tradicional. Finalmente al realizar la experimentación se observa que ACO hibridado con el algoritmo de búsqueda local es significativamente más eficaz para resolver el TSP y además, evita el fenómeno de estancamiento prematuro.

Así mismo en (Tiwari and Chande, 2019) se intenta estimar el número mínimo de hormigas necesarias para optimizar las consultas en las DDBMS distribuidas con un número variado de uniones. Esta estimación se denomina Ant Ratio, pues valora el requisito de  $x$  número de hormigas para optimizar la consulta distribuida con  $y$  número de uniones. Finalmente se realiza el análisis en términos de cardinalidad de unión y tiempo de respuesta, aquí se observa que a medida que aumenta el número de hormigas, el tiempo de ejecución también aumenta debido a la exploración no sistemática de las hormigas, y al mismo tiempo mejora la cardinalidad de unión. La cantidad mínima de hormigas necesarias para generar resultados optimizados se calcula mediante la proporción de hormigas. El número mínimo de hormigas necesarias para generar soluciones optimizadas se encuentra en el rango de 0,2 a 0,3, es decir, está en el rango de  $1/5$  a  $3/10$  del tamaño del problema.

Existen diversas variantes de los algoritmos ACO, en (Chaparro and Valdez, 2013) se realiza una experimentación en la que se revisa el comportamiento de las variantes Max-Min Ant System (MMAS) y

Ant Colony System (ACS) para la resolución del problema del viajero (TSP), ambas variantes demostraron resultados similares, sin embargo denotaron un menor tiempo de ejecución frente al método convencional. Una experimentación similar se lleva a cabo en (Li et al., 2020) donde se plantea un algoritmo basado en la acción colectiva (CAAS) partiendo del tradicional AS, este método involucra una función de umbral  $f(x)$  para la colonia de hormigas en la cual se le asigna un umbral a la hormiga, y luego cada hormiga decide si unirse a la acción colectiva. Finalmente el método se evaluó mediante una comparación entre varios algoritmos de AS aplicando un problema TSP, donde se concluye que el algoritmo puede encontrar buenas soluciones en menos iteraciones.

Otra aplicación de ACO se describe en (Mirjalili et al., 2020) donde se emplea este algoritmo para identificar la ruta óptima para un AUV, lo cual es una aplicación del mundo real sobre el problema del TSP. De esta manera, considerando hasta 200 puntos de ruta el algoritmo ACO obtuvo los resultados esperados, hallando el camino más óptimo en un periodo corto de tiempo, superando a los antiguos algoritmos de fuerza bruta utilizados en este contexto.

Partición de hormigas para la optimización de colonias de hormigas (PACO) es un nuevo método propuesto en (Lizárraga et al., 2013) donde se divide el número total de hormigas en tres subconjuntos, y cada uno se evalúa por separado haciendo énfasis a la idea de “divide y vencerás”. Al realizar las pruebas con este nuevo método sobre diferentes instancias del Problema del vendedor ambulante (TSP), se encontró que este funciona adecuadamente, demostrando la viabilidad de dividir el trabajo para la evaluación siendo incluso 64 veces más rápido que el método convencional. De manera similar en (Lizarraga et al., 2014) se propone un algoritmo ACO AVSE que divide a las hormigas de forma igualitaria en 5 variaciones de ACO, lo cual demuestra, que al realizarse los experimentos usando AVSE toman menos tiempo que aquellos experimentos realizados de forma independiente. Así mismo se propone en (Cura and Özdemir, 2019) un enfoque de AS (Ant System) para problemas de dispersión media máxima (MaxMeanDP), de esta manera la modificación del AS logra que las hormigas ejecuten el proceso de búsqueda en paralelo, en consecuencia sus resultados mostraron que el nuevo uso del rastro de feromona y la visibilidad contribuyen significativamente a la calidad de la solución.

En un contexto de ACO híbrido (Blum et al., 2014) presenta un enfoque de optimización de colonias de hormigas puro y una variante de éste, al ponerlo a prueba en la solución del problema del bombero que está orientado a la propagación de fuego, en un gráfico se obtuvo que, la técnica híbrida resulta superior al enfoque ACO puro.

Por otra parte, con la finalidad de convertir el proceso de corrección de programa en una tarea más sencilla y segura en (Vescan et al., 2020) se describe un método optimizado de priorización de casos de prueba inspirado en la optimización de colonias de hormigas denominado, priorización de casos de prueba ANT y denotado como TCP-ANT. Finalizando el desarrollo se logró visualizar que TCP-ANT supera las técnicas existentes de priorización inversa, priorización aleatoria y sin orden de la literatura, así mismo se asegura que la gravedad de las fallas, también se utiliza como criterio al seleccionar el siguiente caso de prueba, con el objetivo de reducir el costo de ejecutar la priorización encontrada, se introduce un nuevo criterio referido al número de casos de prueba para cubrir las fallas.

Así mismo en (Jalali et al., 2007) se realiza una experimentación que involucra una versión especial del algoritmo multi-colonia, para generar una malla homogénea en todo el espacio de búsqueda y así reducir la posibilidad de perder el dominio óptimo local. El algoritmo que se propone puede trabajar de manera eficiente con las variables de decisión discretas y continuas. En este sentido, para conocer el rendimiento del algoritmo se consideró la función Fletcher-Powell multimodal, continua, no separable, no lineal e ilegal (CNLI) y la optimización de la operación del problema complejo de 10 yacimientos, obteniendo como resultado que la aplicación del algoritmo propuesto reveló su potencial para mejorar los resultados finales en problemas complejos de optimización continua.

Las investigaciones presentadas anteriormente concentran su esfuerzo en conseguir soluciones de mejor calidad sin analizar otros aspectos importantes como la velocidad de convergencia. Esta característica de la meta-heurística apunta a la rapidez con que estos algoritmos son capaces de encontrar soluciones de calidad.

### 3. PROPUESTA ALGORÍTMICA

Desde su origen en 1992 la metaheurística ACO ha sido muy difundida por la comunidad científica como una estrategia a considerar para resolver problemas discretos. Algunos algoritmos forman parte de esta metaheurística como por ejemplo, Ant System (Dorigo et al., 1996), Ant Colony System (Dorigo and Gambardella, 1997), Max-Min Ant System (Stützle and Hoos, 2000) entre otros. Todos estos algoritmos presentan el mismo principio básico de exploración, que consiste en un grupo de hormigas se posicionan de manera aleatoria en algún nodo del grafo que representa al problema de estudio, luego estas hormigas

se mueven por el grafo siguiendo una regla probabilística de decisión  $P_{ij}$  que involucra una función heurística de cambio de estado ( $\eta_{ij}$ ) y la deseabilidad del cambio (rastros de feromona,  $\tau_{ij}$ ). Luego de que todas las hormigas terminan, se aplica una regla de actualización de los rastros de feromona que puede involucrar a todas las soluciones, a la mejor de la iteración o a la mejor encontrada hasta el momento (depende del algoritmo ACO implementado). Este proceso se repite en varias ocasiones hasta cumplir con una condición de parada, que puede ser cantidad máxima de repeticiones, grado de convergencia, calidad de soluciones encontradas, entre otras.

Por su parte, la propuesta en dos Etapas (TS-ACO) divide el espacio de búsqueda en dos subespacios, para ello define el parámetro  $r \in [0,1]$ , que determina la dimensión de cada subespacio. Este parámetro divide en dos el conjunto de hormigas (Ecuación 1), la dimensión del problema (Ecuación 2) y la cantidad de iteraciones (Ecuación 3).

$$k_1 = r * k, \quad k_2 = k - k_1 \quad (1)$$

$$n_1 = r * n, \quad n_2 = n - n_1 \quad (2)$$

$$cp_1 = r * cp, \quad cp_2 = cp - cp_1 \quad (3)$$

Según este enfoque, en la primera etapa las  $k_1$  hormigas exploran en grafo para encontrar soluciones de tamaño  $n_1$  y la cantidad de veces que se repite la búsqueda está condicionada por  $cp_1$ . Luego de aquello, se seleccionan los mejores recorridos para tomarlos como punto de partida para la segunda etapa. En la segunda etapa las  $k_2$  hormigas completan los mejores recorridos de la etapa anterior agregando los  $n_2$  nodos restantes y el mismo proceso se repite hasta que se cumple  $cp_2$ . Como se puede apreciar, este enfoque es muy simple pero según los autores reduce considerablemente el esfuerzo computacional para encontrar buenas soluciones si lo comparamos con la estrategia de ACO original.

La idea de ACO en  $n$  etapas aparece como una generalización de la estrategia en dos etapas. En este nuevo enfoque se definen varios valores de  $r$  para controlar el tamaño de cada etapa en el proceso de búsqueda. De esta manera, los parámetros ( $k, cp, n$ ) de la etapa  $i$  se calculan aplicando las ecuaciones 4, 5 y 6 a la cantidad no utilizada en la etapa anterior. Análogamente, en la etapa  $i$  se utilizan los mejores resultados de la etapa  $i-1$  y se construyen caminos que son utilizados en la siguiente etapa hasta completar la solución del problema.

$$k_i = r_i * \overline{k_{i-1}} \quad (4)$$

$$cp_i = r_i * \overline{cp_{i-1}} \quad (5)$$

$$n_i = r_i * \overline{n_{i-1}} \quad (6)$$

#### 4. METODOLOGÍA

Los experimentos se llevaron a cabo utilizando el algoritmo Ant Colony System (ACS) debido a que en (Dorigo and Gambardella, 1997) se describe como uno de los algoritmos ACO que mejor desempeño tiene en el TSP. Se utilizaron 5 instancias del problema de mediana y alta dimensión como son: gr431, gr666, pr1002, pr2392 y pcb3038 (los números en el nombre representan la dimensión) disponibles en la librería TSPLIB<sup>1</sup>. Se realizaron 5 ejecuciones independientes de cada algoritmo debido al costo computacional del estudio en las condiciones tecnológicas que fueron realizados y por último se utilizó un esquema comparativo no paramétrico basado un test de Friedman para detectar diferencias significativas en los resultados y el test de Holm para identificar las diferencias.

#### 5. RESULTADOS

La tabla 1 presenta los parámetros utilizados en el proceso de experimentación, donde la “Cantidad de etapas” y el “Tamaño de cada etapa” representan parámetros propios de la propuesta. La codificación de este estudio se encuentra disponible en el proyecto Metaheuristics N Stages ACO<sup>2</sup> para replicar los resultados.

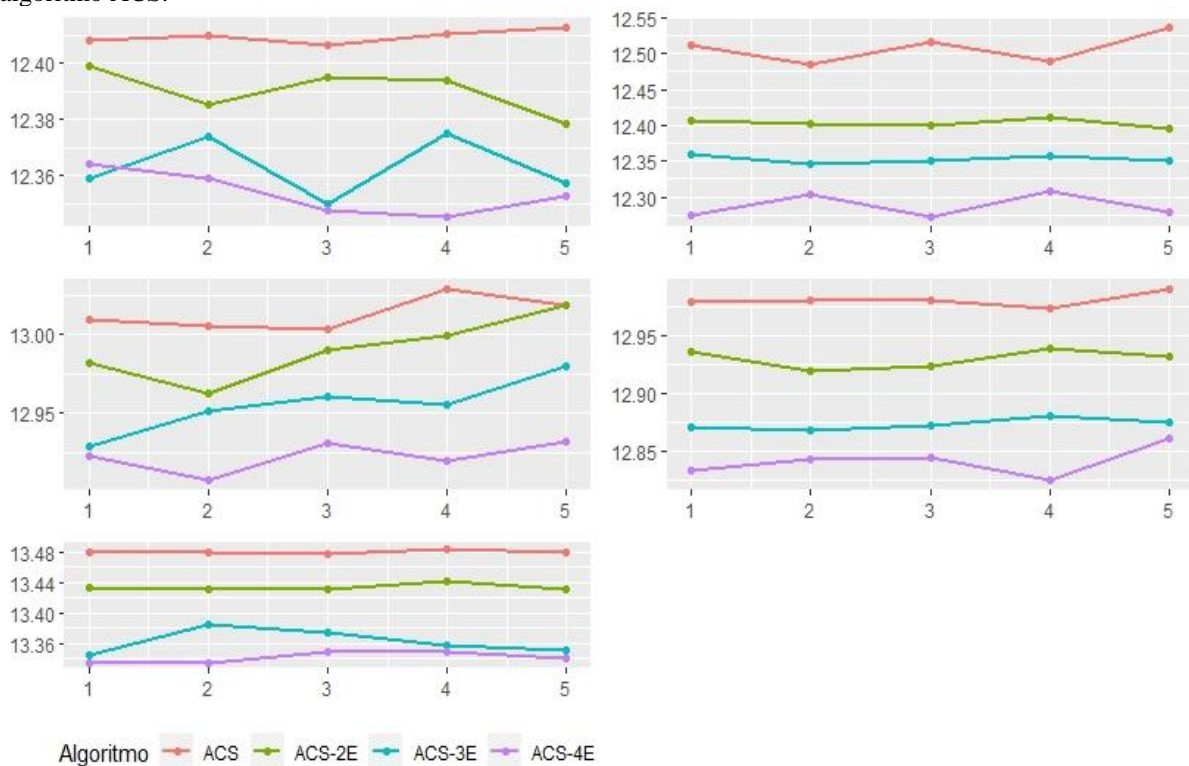
<sup>1</sup> <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/>

<sup>2</sup> <https://github.com/dhernandezl/Metaheuristics-N-Stages-ACO>

**Tabla 1** Parámetros establecidos para la experimentación

Parámetro	ACS	ACS-2E	ACS-3E	ACS-4E
Cantidad de etapas	1	2	3	4
Iteraciones (cp)	10000	10000	10000	10000
Número de Hormigas (k)	15	15	15	15
Alfa ( $\alpha$ )	1	1	1	1
Beta ( $\beta$ )	2	2	2	2
Constancia de Evaporación ( $\rho$ )	0.1	0.1	0.1	0.1
Feromona Inicial ( $\tau$ )	0.1	0.1	0.1	0.1
Variable Real (q0)	0.1	0.1	0.1	0.1
Tamaño de cada Etapas	-	r=25	r=[25, 25]	r=[25, 25, 25]

La figura 1 muestra el desempeño (valor escalado con función logarítmica) de cada estrategia de exploración para las 5 ejecuciones independientes que se desarrollaron. Se puede observar que a medida que va aumentando la dimensión del problema va siendo mayor la diferencia en el desempeño de cada propuesta. En todos los casos, dividir el proceso de exploración determina mejor rendimiento del algoritmo y específicamente la variante de 4 etapas es la que mejores resultados obtiene en todos los casos de estudio. De la misma manera se puede observar que la dispersión entre las 5 ejecuciones es muy similar para todos los algoritmos, de lo que se puede inferir que la propuesta no altera la convergencia del algoritmo ACS.



**Figura 1** Comparación de las ejecuciones independientes entre las técnicas estudiadas. Para corroborar la existencia de diferencias significativas entre las soluciones, aplicamos un test de Friedman con un nivel de confianza de 0.05. El valor p (0.00181) que se obtuvo fue menor que el valor de confianza fijado para el estudio, concluyendo que son significativas las diferencias que existen en los resultados. Como consecuencia, aplicamos el test Holm tomando el algoritmo ACS-4E como algoritmo de control o referencia con un nivel de confianza del 0.05. La tabla muestra que los resultados del

algoritmo de Control son significativamente mejores que la variante ACS y ACS-2E y similares que los obtenidos por ACS-3E. De manera general se pudo determinar que el algoritmo obtiene mejor rendimiento cuando el proceso de exploración se divide en etapas y la cantidad de etapas es un factor fundamental para alcanzar buenos resultados.

. **Tabla 2** Comparación del test de Friedman aplicando post-hoc

ACS-4E vs.	valor-p	Holm	¿Se acepta la hipótesis nula de igualdad?
ACS	2.385E-04	0.0166	No
ACS-2E	0.0143	0.025	No
ACS-3E	0.2206	0.05	Si

De esta forma se puede confirmar que dividir la exploración de los algoritmos ACO en etapas, supone un mejor desempeño al momento de resolver problemas de mediana y alta dimensión. Además, es importante estudiar si existe alguna relación entre el tamaño y el número de etapas.

## 6. CONCLUSIONES

En la investigación se presentó un estudio para mejorar la estrategia de exploración que presentan los algoritmos ACO. Específicamente se dividió el proceso de búsqueda de las hormigas en N etapas como una forma de reducir el costo computacional de la búsqueda. La idea fue introducir un parámetro para establecer el tamaño de cada etapa en el proceso. En la experimentación se utilizaron 5 instancias del TSP de mediana y alta dimensión. Los resultados mostraron que:

- El modelo basado en 4 etapas alcanza los resultados de mejor calidad en todos los casos de estudio, siendo estos significativamente mejores que los encontrados por el algoritmo sin etapas y en 2 etapas.
- En todos los casos, las variantes que dividieron el proceso en etapas, alcanzaron los mejores resultados.
- Por último, un análisis de estabilidad de las soluciones ratificó la eficiencia del modelo en n etapas al mantener resultados no muy dispersos en relación al algoritmo original.

De manera general se logró constatar la aplicabilidad de N etapas para problemas de media y alta dimensionalidad permitiendo mejorar la convergencia de los algoritmos ACO, llegando a evidenciarse alguna relación entre la cantidad de etapas y la dimensión del problema. Esta última consideración forma parte de nuestro trabajo futuro, donde trataremos de analizar instancias más grandes y aumentar la cantidad de etapas para el proceso de búsqueda.

**RECEIVED: JANUARY, 2021.**

**REVISED: SEPTEMBER, 2021.**

## REFERENCIAS

- [1] BLUM, C., BLESÁ, M. J., GARCÍA-MARTÍNEZ, C., RODRÍGUEZ, F. J., and LOZANO, M. (2014): The Firefighter Problem: Application of Hybrid Ant Colony Optimization Algorithms. **Evolutionary Computation in Combinatorial Optimisation**, 218-229. [https://doi.org/10.1007/978-3-662-44320-0\\_19](https://doi.org/10.1007/978-3-662-44320-0_19)
- [2] CHAPARRO, I., and VALDEZ, F. (2013): Variants of Ant Colony Optimization: A Metaheuristic for Solving the Traveling Salesman Problem. En O. Castillo, P. Melin, and J. Kacprzyk (Eds.), **Recent Advances on Hybrid Intelligent Systems**, 323-331. Springer. [https://doi.org/10.1007/978-3-642-33021-6\\_26](https://doi.org/10.1007/978-3-642-33021-6_26)
- [3] CURA, T., and ÖZDEMİR, M. (2019): A new use of the ant system algorithm for the max-mean dispersion problem. **Computers and Industrial Engineering**, 135, 628-642. <https://doi.org/10.1016/j.cie.2019.06.045>
- [4] DORIGO, M., BIRATTARI, M., and STUTZLE, T. (2006): Ant colony optimization. **IEEE Computational Intelligence Magazine**, 1(4), 28-39. <https://doi.org/10.1109/MCI.2006.329691>
- [5] DORIGO, M., and GAMBARDELLA, L. M. (1997): Ant colony system: A cooperative learning approach to the traveling salesman problem. **IEEE Transactions on Evolutionary Computation**, 1, 53-66. [HTTPS://DOI.ORG/10.1109/4235.585892](https://doi.org/10.1109/4235.585892)
- [6] DORIGO, M., MANIEZZO, V., and COLORNI, A. (1996): Ant system: Optimization by a colony of cooperating agents. **IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)**, 26, 29-41. <https://doi.org/10.1109/3477.484436>

- [7] IBRAHIM, E.-S., BIRCHELL, S., and ELFAYOUMY, S. (2012): Automatic heart volume measurement from CMR images using ant colony optimization with iterative salient isolated thresholding. **Journal of Cardiovascular Magnetic Resonance**, *14*, P286. [HTTPS://DOI.ORG/10.1186/1532-429X-14-S1-P286](https://doi.org/10.1186/1532-429X-14-S1-P286)
- [8] Jalali, M. R., Afshar, A., and Mariño, M. A. (2007): Multi-Colony Ant Algorithm for Continuous Multi-Reservoir Operation Optimization Problem. **Water Resources Management**, *21*, 1429-1447. [HTTPS://DOI.ORG/10.1007/S11269-006-9092-5](https://doi.org/10.1007/S11269-006-9092-5)
- [9] Kefi, S., Rokbani, N., and Alimi, A. M. (2017): Solving the Traveling Salesman Problem Using Ant Colony Metaheuristic, A Review. En A. Abraham, A. Haqiq, A. M. Alimi, G. Mezzour, N. Rokbani, and A. K. Muda (Eds.), **Proceedings of the 16th International Conference on Hybrid Intelligent Systems (HIS 2016)**, 421-430. Springer International Publishing. [https://doi.org/10.1007/978-3-319-52941-7\\_42](https://doi.org/10.1007/978-3-319-52941-7_42)
- [10] LI, S., CAI, S., LI, L., SUN, R., and YUAN, G. (2020): CAAS: A novel collective action-based ant system algorithm for solving TSP problem. **Soft Computing**, *24*, 9257-9278. [HTTPS://DOI.ORG/10.1007/S00500-019-04452-Y](https://doi.org/10.1007/S00500-019-04452-Y)
- [11] LIZÁRRAGA, E., CASTILLO, O., and SORIA, J. (2013): A Method to Solve the Traveling Salesman Problem Using Ant Colony Optimization Variants with Ant Set Partitioning. **Recent Advances on Hybrid Intelligent Systems**, 237-246. [https://doi.org/10.1007/978-3-642-33021-6\\_19](https://doi.org/10.1007/978-3-642-33021-6_19)
- [12] LIZARRAGA, E., CASTILLO, O., SORIA, J., and VALDEZ, F. (2014): A Fuzzy Control Design for an Autonomous Mobile Robot Using Ant Colony Optimization. En O. Castillo, P. Melin, W. Pedrycz, and J. Kacprzyk (Eds.), **Recent Advances on Hybrid Approaches for Designing Intelligent Systems**, 289-304. Springer International Publishing. [https://doi.org/10.1007/978-3-319-05170-3\\_20](https://doi.org/10.1007/978-3-319-05170-3_20)
- [13] MIRJALILI, S., SONG DONG, J., and LEWIS, A. (2020): Ant Colony Optimizer: Theory, Literature Review, and Application in AUV Path Planning. En S. Mirjalili, J. Song Dong, and A. Lewis (Eds.), **Nature-Inspired Optimizers: Theories, Literature Reviews and Applications**, 7-21. Springer International Publishing. [https://doi.org/10.1007/978-3-030-12127-3\\_2](https://doi.org/10.1007/978-3-030-12127-3_2)
- [14] PINTEA, C.-M., MATEI, O., RAMADAN, R. A., PAVONE, M., NIAZI, M., and AZAR, A. T. (2016): A Fuzzy Approach of Sensitivity for Multiple Colonies on Ant Colony Optimization. **Soft Computing Applications**, 87-95. [https://doi.org/10.1007/978-3-319-62524-9\\_8](https://doi.org/10.1007/978-3-319-62524-9_8)
- [15] PURIS, A., BELLO, R., and HERRERA, F. (2010): Analysis of the efficacy of a Two-Stage methodology for ant colony optimization: Case of study with TSP and QAP. **Expert Systems with Applications**, *37*, 5443-5453. <https://doi.org/10.1016/j.eswa.2010.02.069>
- [16] STÜTZLE, T., and HOOS, H. H. (2000): MAX-MIN Ant System. **Future Generation Computer Systems**, *16*, 889-914. [HTTPS://DOI.ORG/10.1016/S0167-739X\(00\)00043-1](https://doi.org/10.1016/S0167-739X(00)00043-1)
- [17] TIWARI, P., and CHANDE, S. V. (2019): Optimal Ant and Join Cardinality for Distributed Query Optimization Using Ant Colony Optimization Algorithm. En V. S. Rathore, M. Worring, D. K. Mishra, A. Joshi, and S. Maheshwari (Eds.), **Emerging Trends in Expert Applications and Security**, 385-392. SPRINGER. [HTTPS://DOI.ORG/10.1007/978-981-13-2285-3\\_45](https://doi.org/10.1007/978-981-13-2285-3_45)
- [18] VESCAN, A., PINTEA, C.-M., and POP, P. C. (2020): Solving the Test Case Prioritization Problem with Secure Features Using Ant Colony System. En F. Martínez Álvarez, A. Troncoso Lora, J. A. Sáez Muñoz, H. Quintián, and E. Corchado (Eds.), **International Joint Conference: 12th International Conference on Computational Intelligence in Security for Information Systems (CISIS 2019) and 10th International Conference on European Transnational Education (ICEUTE 2019)**, 67-76. Springer International Publishing. [https://doi.org/10.1007/978-3-030-20005-3\\_7](https://doi.org/10.1007/978-3-030-20005-3_7)
- [19] ZHANG, S., and ZHANG, Y. (2018): A Hybrid Genetic and Ant Colony Algorithm for Finding the Shortest Path in Dynamic Traffic Networks. **Automatic Control and Computer Sciences**, *52*, 67-76. <https://doi.org/10.3103/S014641161801008X>