

Tipo de artículo: Artículo original
Temática: Soluciones Informáticas
Recibido: 03/05/2020 | Aceptado: 14/07/2020 | Publicado: 01/08/2020

Algoritmo de las hormigas para la elección de rutas de distribución

Ant system algorithm for the selection of delivery routes

Maybel Díaz Capote ^{1*}

¹Facultad 3, Universidad de las Ciencias Informáticas, Cuba, Carretera a San Antonio de los Baños, Km 2½, localidad Torrens, Boyeros, La Habana, C.P. 19370. mdcapote@uci.cu

* Autor para correspondencia: mdcapote@uci.cu

Resumen

Este trabajo ofrece una perspectiva general de la metaheurística de optimización basada en colonias de hormigas. Inicialmente, se analizaron tres de los algoritmos más destacados de esta metaheurística, los cuales son: Sistema de Hormigas, Sistema por Colonias de Hormigas y Sistema de Hormigas Max-Min. Estos algoritmos han sido ampliamente utilizados en la selección de rutas de distribución cercanas a las óptimas en problemas de distribución de mercancías. Además, el Sistema de Hormigas Max-Min ha sido reconocido como uno de los algoritmos más competitivos en este contexto. Como objetivo del trabajo se propone un modo para su implementación y se explican los detalles relacionados a esta. Para la validación de la implementación propuesta se seleccionaron ejemplos de Problemas del Viajante, tomados de la biblioteca de estándares de comparación TSPLIB, específicamente los problemas Berlin52 y ST70. En la investigación se evidencia la necesidad de escoger cuidadosamente los parámetros para aplicar el algoritmo, ya que la eficacia de este depende en gran medida de esta elección. En adición, la implementación de este procedimiento constituye una herramienta de alto valor añadido para la toma de decisiones, ya sea con fines docentes o profesionales. Su comprobación ante instancias de Problemas del Viajante de complejidad reconocida así lo demuestra.

Palabras clave: metaheurística, optimización por colonias de hormigas, rutas de distribución, Sistema de Hormigas Max-Min, TSP

Abstract

This paper provides an overview of the metaheuristic optimization based on ant colonies. At first, three of the more successfully algorithms are analyzed, namely the algorithms Ant System, Ant Colony System and Max-Min Ant System. They have been widely used in the selection of near-optimal solutions to the problems of delivering goods. In addition, the Max-Min Ant System has been recognized as one of the most competitive algorithm in this context. The paper aims to provide and give details on the way of implementing this algorithm. For proving the implementation approach some instances of the Travelling Salesman Problem, taken from the TSPLIB benchmark library, were selected, specifically the problems Berlin52 and ST70. The paper also describes the need to carefully select parameters for implementing the algorithm, since its effectiveness depends largely on them. The implementation of

this procedure is also a high value added tool for decision making, for teaching or professional purposes. Tests conducted through some cases of the Travelling Salesman Problem benchmark library demonstrate this fact.

Keywords: *ant colony optimization, delivery routes, Max-Min Ant System, metaheuristic, TSP*

Introducción

Con el desarrollo tecnológico adquirido en los últimos años, la cantidad de mercancías a transportar, el almacenamiento y el manejo de información entre el fabricante y el consumidor se han incrementado considerablemente. La administración eficiente de estos procesos es objeto de estudio de la logística empresarial.

Para aquellas organizaciones que brindan servicios de transporte de mercancías, el movimiento de productos desde el almacén hasta el cliente involucra grandes desafíos. La respuesta a las demandas del cliente y la necesidad de aumentar la utilidad, es un reto para los profesionales del transporte. La presión de reducir los costos de distribución y de minimizar los tiempos de entrega aumenta significativamente (Feitó, 2009). Por esta razón es de vital importancia para las empresas, la utilización de métodos que permitan la realización de una correcta selección de las rutas de distribución. Esta investigación está encaminada a analizar los algoritmos Sistema de Hormigas (AS, por sus siglas en inglés), Sistema por Colonias de Hormigas (ACS, por sus siglas en inglés) y Sistema de Hormigas Max-Min (MMAS, por sus siglas en inglés). Estos algoritmos han sido ampliamente utilizados en la selección de rutas de distribución cercanas a las óptimas en problemas de distribución de mercancías. El objetivo del trabajo es proponer y dar detalles sobre la forma de implementación del algoritmo más competitivo entre los mencionados, con el fin de apoyar la selección de rutas de distribución.

Materiales y métodos

Uno de los problemas más comunes al diseñar las rutas de distribución es el de buscar el camino mínimo, bajo la restricción de visitar exactamente una vez todos los puntos de distribución de un recorrido. Este problema es conocido también como el Problema del Viajante (TSP, por sus siglas en inglés). Si en adición, la distancia entre un nodo i y un nodo j es similar a la distancia entre el nodo j y el nodo i , entonces se dice que el problema es un TSP Simétrico (Rave, 2011). Una formulación equivalente en términos de la teoría de grafos es la de encontrar el ciclo hamiltoniano de menor coste en un grafo completo con arcos ponderados. El TSP simétrico está entre los problemas denominados NP-hard (Davendra, 2010). Por esta razón, los científicos han desarrollado toda una gama de procedimientos basados en reglas heurísticas, que si bien no siempre ofrecen un resultado óptimo, sí permiten lograr buenas soluciones de una manera mucho más rápida (Feitó, 2009). Con el surgimiento de las llamadas metaheurísticas se resuelve la

contradicción de los métodos de optimización y los heurísticos, ya que arriban a soluciones cercanas a la óptima en tiempos aceptables. Muchos de estos algoritmos se basan en el comportamiento de entes de la naturaleza, procesos físicos, entre otros, y fundamentalmente poseen un elemento de experiencia pues van aprendiendo a lo largo de su ejecución. Un ejemplo de estos algoritmos, llamados bioinspirados, es la Optimización basada en Colonia de Hormigas (ACO, por sus siglas en inglés)(Feitó, 2009). La ACO está inspirada en el comportamiento de las hormigas reales, el cual se distingue por la búsqueda de alimentos, trazando el camino más corto entre el hormiguero y la fuente de comida.

El eje de esa búsqueda está basado en la modificación del ambiente local depositando una sustancia química llamada feromona. Esta sustancia sirve como rastro y orienta el recorrido sin necesidad del sentido de la vista(Barcos, *et al.*, 2002). Estos insectos tienden a seguir la mayor concentración de feromona. En principio, ellas siguen rutas aleatorias. Por este motivo, si no existe inicialmente ningún rastro de feromona en el medio, cuando una hormiga llega a una bifurcación, elige al azar una de las vías posibles. La feromona se evapora a medida que pasa el tiempo, así que el rastro de un camino más corto tendrá más concentración de feromona que otro más largo. De esta manera, la probabilidad de que una hormiga escoja el camino más corto aumenta progresivamente. Al final, el recorrido de la colonia converge al más corto de todos los caminos posibles(Feitó, 2009). Finalmente, el proceso se caracteriza por una retroalimentación positiva, donde la probabilidad con la que una hormiga escoge un camino aumenta con el número de hormigas que previamente hayan elegido el mismo camino(Arango-Serna, *et al.*, 2009).

Trabajos relacionados

Para aplicar ACO, el problema de optimización es transformado en un problema para descubrir la mejor ruta en un grafo ponderado. Las hormigas artificiales aceleradamente construyen soluciones por movimientos en el grafo. El proceso de construcción de la solución es probabilístico(Rashid, *et al.*, 2012). Para guiar los movimientos en el grafo las hormigas se basan en dos tipos de información que se encuentra en cada arista o tramo del grafo:

- información heurística: mide la preferencia heurística de moverse desde un nodo a otro. Las hormigas no modifican esta información durante la ejecución del algoritmo.
- información de los rastros de feromona artificiales: mide la “deseabilidad aprendida” del movimiento de un nodo a otro. Esta información se modifica durante la ejecución del algoritmo(Mocholi-Agües, 2011).

El primer algoritmo de optimización basado en colonia de hormigas, propuesto por Colorni, Dorigo y Maniezzo, es el AS y fue aplicado por primera vez al TSP (De-La-Fuente-García, *et al.*, 2011). En AS, cada hormiga genera una ruta

completa seleccionando las ciudades de acuerdo a una regla probabilística de transición entre estados (las hormigas prefieren moverse hacia ciudades más cercanas y con un alto contenido de feromonas en su camino). Esta regla de transición presenta unos parámetros configurables que determinan la importancia relativa de la feromona (rastros) frente a la longitud del trayecto. Posteriormente se aplica una regla de actualización global de feromonas: se evapora una fracción de la feromona de los caminos no recorridos y cada hormiga deposita una cierta cantidad de esta sustancia en los caminos pertenecientes a su ruta, en proporción a lo corta que ésta resulte. Finalmente se itera el mismo proceso (Wu, *et al.*, 2012). Años más tarde, los mismos autores presentaron una mejora para este algoritmo híbrido. Esta mejora consiste en la introducción en la fase inicial de una lista de candidatos para la selección de los clientes más prometedores. Esta lista de candidatos es obtenida ordenando el conjunto de las localizaciones de menor a mayor distancia. Coloni, Dorigo y Maniezzo proponen el algoritmo ACS como una mejora al algoritmo AS. Por lo tanto, ACS logra mejores rendimientos debido a la introducción de nuevos mecanismos basados en ideas no incluidas en el algoritmo original (Muñoz, *et al.*, 2008). ACS difiere de AS en algunos aspectos como:

- la regla de actualización global es únicamente aplicada a los caminos pertenecientes a la mejor ruta.
- mientras las hormigas construyen una solución se aplica una regla de actualización local de feromona.

Stützle y Hoos propusieron el algoritmo MMAS como una mejora del AS inicial. Sus características principales son que únicamente la mejor hormiga actualiza el rastro de feromona y que el valor de la feromona está acotado, tanto superior como inferiormente. Estas cotas son típicamente obtenidas de forma empírica y ajustadas según el problema específico (Socha, *et al.*, 2002). Para el caso del algoritmo MMAS es posible determinar las expresiones que definen dichos intervalos. Este algoritmo presenta muy buenos resultados en la búsqueda de las mejores soluciones. Hoy en día es uno de los algoritmos más competitivos para la solución del TSP (Stützle y Hoos, 2000). Por las novedades que incorpora y por los resultados que con él se han alcanzado se propone su aplicación en la selección de rutas de distribución.

Procedimiento de optimización mediante el algoritmo MMAS

En el MMAS las hormigas construyen las soluciones de manera probabilística, guiándose por un rastro de feromona artificial y por una información calculada a priori de manera heurística. Sea α y β la importancia concedida a la feromona y a la heurística respectivamente, se sigue para el TSP la regla probabilística mostrada en la ecuación (1):

$$p_{ij}^k = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}, \quad \text{con } j \in N_i^k \quad (1)$$

donde, p_{ij}^k es la probabilidad con la que, en una iteración t , la hormiga k , situada actualmente en la ciudad i , elige a la ciudad j como próxima parada; N_i^k es el conjunto de ciudades no visitadas por la hormiga k ; τ_{ij} es la cantidad de feromona acumulada sobre el arco (i, j) , en la iteración t ; η_{ij} es la información heurística. Para un problema de distribución de mercancía la información heurística debe elegirse como el inverso de la distancia existente entre las ciudades i y j . Cuando todas las hormigas han construido una solución debe actualizarse la feromona en cada arco según el procedimiento general. La manera en que ésta actualización ocurre distingue a cada tipo de algoritmo. En MMAS, sea ρ el coeficiente de evaporación de la feromona definido en el intervalo $(0, 1]$, se sigue la ecuación (2) para la actualización de dicha sustancia:

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \Delta\tau_{ij}^{best}, \quad (2)$$

$$\Delta\tau_{ij}^{best} = \begin{cases} \frac{1}{L^{best}} & : \text{si el arco } (i, j) \in S^{best} \\ 0 & : \text{en caso contrario} \end{cases}$$

donde, S^{best} es la mejor solución encontrada en la iteración; L^{best} es la longitud de la solución S^{best} . Vale destacar que para la actualización, la elección de S^{best} como la mejor solución encontrada en la iteración actual (s^{ib}), constituye un rasgo distintivo del algoritmo MMAS. Así se garantiza que las aristas que frecuentemente se hallen en esta mejor solución refuercen con un valor más alto su cantidad de feromona. Otros como ACS típicamente utilizan la mejor solución global encontrada (s^{gb}). Cuando únicamente se utiliza esta última, la búsqueda puede concentrarse demasiado rápido alrededor de la solución y la exploración de posibles mejores soluciones queda limitada, con el consiguiente peligro de quedar atrapado con soluciones de pobre calidad. Aun así, una elección juiciosa entre la hormiga de la mejor iteración y la mejor hormiga global a la hora de actualizar la feromona puede controlar la forma en que el historial de búsqueda es aprovechado (Stützle y Hoos, 2000).

Independientemente de la elección (s^{gb}) o (s^{ib}) puede ocurrir un estancamiento en la búsqueda de soluciones. El estancamiento define aquella situación en la que todas las hormigas construyen una y otra vez la misma solución, cuestión que debe ser evitada sobre todo al inicio de una corrida. Esto pudiera suceder si para un nodo dado el rastro

de feromona es significativamente más grande en una arista que en cualquier otra. Por tal motivo, el algoritmo MMAS obliga a que el nivel de feromona permanezca acotado en un intervalo $[\tau_{min}, \tau_{max}]$.

Límites para el valor de rastro de feromona

Con el algoritmo MMAS los niveles de feromona permanecen acotados dentro de un intervalo $[\tau_{min}, \tau_{max}]$. Estos límites se imponen con el objetivo de influir en la probabilidad de escoger la próxima componente de una solución, cuestión que depende directamente del rastro de feromona y de la información heurística, ver ecuación (1). Si se limita la influencia del rastro de feromona, se puede evitar que las diferencias relativas entre rastros de feromonas sean excesivamente grandes durante la corrida del algoritmo. Para realizar el cálculo de los límites de la concentración de feromona, se siguen las ecuaciones (3) y (4):

$$\tau_{max} = \frac{1}{\rho} \frac{1}{L^{best}}, \quad (3)$$

$$\tau_{min} = \frac{\tau_{max}(1 - \sqrt[n]{P_{best}})}{(\text{avg} - 1)\sqrt[n]{P_{best}}}, \quad (4)$$

siendo, $\text{avg} = \frac{n}{2}$, n la cantidad de vértices del grafo y P_{best} la probabilidad con la que se encuentra la mejor solución. La ecuación (3), equivalente a la expresión propuesta en (Stützle y Hoos, 2000), constituye la cota superior del valor máximo de feromona que puede acumularse en una arista.

Proposición 1. Para cualquier arista τ_{ij} se cumple que:

$$\lim_{t \rightarrow \infty} \tau_{ij}(t) = \tau_{ij} \leq \frac{1}{\rho L^{best}}. \quad (5)$$

Prueba. De la ecuación (2) es posible notar que la máxima cantidad de feromona que puede depositarse en una arista durante una iteración es $\Delta\tau_{ij}^{best} = \frac{1}{L^{best}}$.

Así para la iteración 1 se tiene:

$$\tau_{ij}(1) = (1 - \rho)\tau_{ij}(0) + \Delta\tau_{ij}^{best}. \quad (6)$$

Para la iteración 2 se tiene:

$$\tau_{ij}(2) = (1 - \rho)\tau_{ij}(1) + \Delta\tau_{ij}^{best}. \quad (7)$$

Escribiendo (7) en función de la ecuación (6) queda:

$$\tau_{ij}(2) = (1 - \rho)^2 \tau_{ij}(0) + (1 - \rho) \Delta \tau_{ij}^{best} + \Delta \tau_{ij}^{best}. \quad (8)$$

A partir de este comportamiento es posible arribar a la siguiente expresión general que describe la cantidad de feromona depositada en la arista (i, j) hasta la iteración t.

$$\tau_{ij}(t) = (1 - \rho)^t \tau_{ij}(0) + \Delta \tau_{ij}^{best} \sum_{k=0}^{t-1} (1 - \rho)^k. \quad (9)$$

El valor máximo que se puede acumular será:

$$\tau_{ij}^{max}(t) = \lim_{t \rightarrow \infty} \tau_{ij}(t),$$

$$\tau_{ij}^{max}(t) = \lim_{t \rightarrow \infty} \left[(1 - \rho)^t \tau_{ij}(0) + \Delta \tau_{ij}^{best} \sum_{k=0}^{t-1} (1 - \rho)^k \right].$$

Dado que $\rho \in (0, 1]$ el primer sumando en el límite resulta 0. Por igual razón la sumatoria en el segundo describe una serie geométrica absolutamente convergente cuya suma es $\frac{1}{\rho} \Delta \tau_{ij}^{best}$. Por tanto el valor del límite será:

$$\tau_{ij}^{max}(t) = \lim_{t \rightarrow \infty} \tau_{ij}(t) = \frac{1}{\rho \Delta \tau_{ij}^{best}}. \quad \square$$

La determinación de la expresión (4) se realiza mediante suposiciones y observaciones empíricas que pueden ser estudiadas en (Stützle y Hoos, 2000). Tras la actualización de la feromona se comienza una nueva iteración. El algoritmo MMAS converge cuando para cada vértice del grafo, cada arista de la solución tiene asociada la mayor cantidad de feromona τ_{max} , mientras que las restantes tienen el valor de feromona τ_{min} . Si el MMAS converge, la solución que se construye eligiendo siempre la arista con mayor rastro de feromona corresponderá típicamente con la mejor solución encontrada por el algoritmo (Stützle y Hoos, 2000).

Detalles de implementación del algoritmo

Valor inicial de τ_{max}

Para estimar el valor de τ_{max} es necesario conocer de antemano alguna ruta óptima. De ahí que uno de los detalles principales sea su valor inicial, ver ecuación (5). En el algoritmo MMAS, después de la primera iteración todos los rastros de feromona toman el valor τ_{max} . Esto se logra estableciendo el valor inicial $\tau_{ij}(0)$ igual a algún valor grande arbitrario en todas las aristas del grafo. Así, luego de la primera iteración los rastros de feromona se verán

forzados a permanecer dentro del intervalo definido $[\tau_{min}, \tau_{max}]$; en particular se les asignará el valor $\tau_{max}(1)$. Este tipo de inicialización se elige para favorecer la exploración de soluciones durante las primeras iteraciones del algoritmo. Vale destacar que para el cálculo de τ_{max} se utilizará siempre el valor L^{best} que corresponda a la mejor hormiga global en vez de la mejor hormiga de la iteración en curso. Cada vez que una solución mejor es encontrada el valor τ_{max} se actualiza, lo cual conlleva a un cambio dinámico de los valores de τ_{max} .

Elección de las constantes

Para la implementación de la solución se escogieron los siguientes valores: $\alpha = 1$, $\beta = 2$, $m = n$, siendo m el número de hormigas, n las ciudades y $\rho = 0,02$ como factor de evaporación de feromonas. Este último valor resulta, en una convergencia lenta para el algoritmo MMAS. La selección de estos valores fue realizada a partir de los resultados de Stützle y Hoos (Stützle y Hoos, 2000). La actualización de la feromona es realizada únicamente por la mejor hormiga de la iteración en curso y sus límites son elegidos en virtud de las ecuaciones (3) y (4) con $P_{best} = 0,05$.

Pseudocódigos

En general, todos los algoritmos de optimización basados en colonias de hormigas siguen un esquema algorítmico específico que ha sido descrito ampliamente en (Dorigo, *et al.*, 1996; Maniezzo, *et al.*, 2004; Stützle y Hoos, 2000). La implementación realizada en esta investigación del algoritmo MMAS se describe a continuación. Consta de una función principal denominada **LeastCostHamiltonianPath** que será descrita al igual que la llamada **GlobalUpdate**.

El algoritmo comienza con una etapa de preprocesamiento, en la que se crean tantas hormigas como nodos tenga el grafo. Así, se hace corresponder cada hormiga con un nodo siguiendo algún criterio y finalmente cada hormiga, marca su nodo inicial como visitado. Luego se entra a un bucle donde: mientras no se exceda de una cantidad de ciclos definida con anterioridad y el algoritmo no haya convergido, se ejecutará el bloque principal. Primero, se borran todas las listas *tabu* de las hormigas, donde estas mantienen una memoria de los nodos ya visitados. Luego cada hormiga realiza su búsqueda de solución de manera probabilística, guiadas por la ecuación (1). A continuación se escoge la mejor hormiga de la iteración, la que ha encontrado el camino más corto. Si en ese instante se detecta una hormiga mejor que la mejor hormiga global hallada hasta ese momento, ésta se actualiza y con ella el límite máximo de feromona. Con la mejor hormiga de la iteración se actualizan además los rastros de feromonas de todo el grafo. El ciclo continúa hasta tanto no se cumpla alguna condición de parada: se alcance la cantidad máxima de ciclos permisibles o el algoritmo converja de manera natural. Finalmente, el algoritmo retorna el camino encontrado de la mejor hormiga global.

El algoritmo 2 describe el proceso de actualización global. Este comienza recuperando el camino de la mejor hormiga de la iteración en curso. Se actualiza luego el límite inferior para el rastro de feromona y se entra en un ciclo por todas las aristas del grafo. Para cada una de las aristas:

- se elige el máximo entre su valor de feromona reducido por el factor ρ y el valor mínimo establecido.
- si el mejor camino hallado contiene la arista analizada se refuerza su valor de feromona con el inverso de la longitud del mejor camino.
- se elige el mínimo entre el valor total de feromona de la arista y el máximo establecido. Así se mantiene acotado superiormente dicho valor en virtud del algoritmo. Finalmente se establece el nuevo valor.

Algoritmo 1. Algoritmo propuesto para la llamada principal del MMAS	Algoritmo 2. Algoritmo de actualización global propuesto para el MMAS
<pre> 1. procedureLeastCostHamiltonianPath() 2. begin 3. it<- 0 4. bestAnt<- 0 5. Preprocess() 6. while it< COUNTCYCLES and 7. notHasConverged(globalBestAnt) do 8. ClearAllTabuLists() 9. for all ant in antList 10. FindSolution() 11. end for 12. bestAnt<- AntShortestPath() 13. GlobalUpdate(bestAnt) 14. it <- it + 1 15. end while 16. if it = COUNTCYCLES 17. return emptyList // didn't converge 18. end if 19. return globalBestAnt.GetTabu() 20. end </pre>	<pre> 1. procedureGlobalUpdate(bestAnt) 2. begin 3. bestPath<- bestAnt.GetTabu() 4. n <- graph.VertexesLength() 5. avg<- n / 2; 6. minPheromone<- Ecuación (4) 7. for all edge in graph.edges() 8. pheromone = Max((1 - rho) * 9. edge.GetPheromone(), minPheromone) 10. if bestPath.contains(edge) 11. pheromone <- pheromone 12. + 1/bestPath.length() 13. end if 14. aux <- Min(pheromone,maxPheromone) 15. edge.SetPheromone(aux) 16. endfor 17. end </pre>

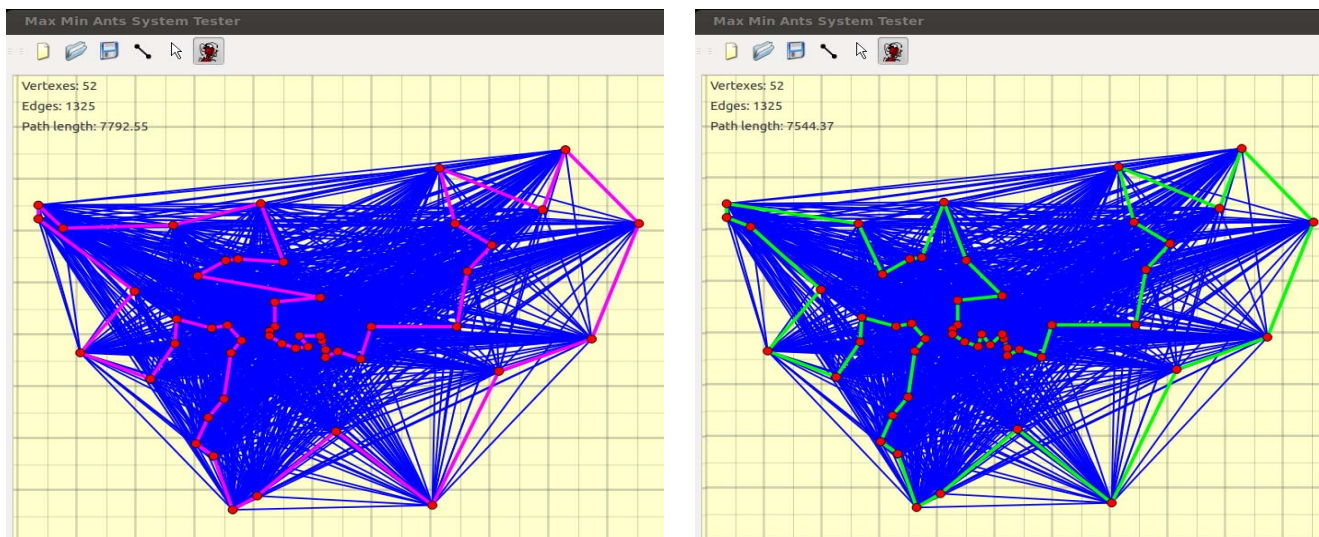
Resultados y discusión

El algoritmo propuesto se implementó en un sistema para darle solución a problemas de selección de rutas de distribución. Para validar la implementación se seleccionaron ejemplos de problemas TSP, tomados de la biblioteca TSPLIB. Dicha biblioteca está accesible en <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/>. Estos ejemplos han sido utilizados en otros estudios y provienen en parte de aplicaciones prácticas del problema TSP. El sistema se desarrolló utilizando el *framework* multiplataforma Qt, ampliamente utilizado por los desarrolladores e ideal para desarrollar interfaces gráficas de usuario. La aplicación desarrollada, permite obtener una ruta cercana a la óptima,

mostrando además la longitud recorrida. A continuación se muestran las salidas del algoritmo para las instancias seleccionadas.

La Figura 1(a) muestra el juego de datos correspondiente a la instancia de problema TSP Berlin52. Este ejemplo describe la ciudad de Berlín con 52 vértices y 1325 aristas. La salida del algoritmo implementado para este caso presenta una longitud total de 7792,55 unidades. La mejor ruta conocida para resolver el TSP en este juego de datos se resalta en la Figura 1(b). Presenta una longitud total de 7544,37 unidades. La solución alcanzada presenta una desviación porcentual de 3,2%, lo cual brinda un margen de error aceptable por debajo del 5%. Es válido notar que la mayor diferencia entre las dos soluciones radica en la zona noroeste de Berlín. El resto de las aristas que conforman la solución es muy similar en ambas salidas. Nótese cómo la solución alcanzada describe el mismo patrón que la solución óptima.

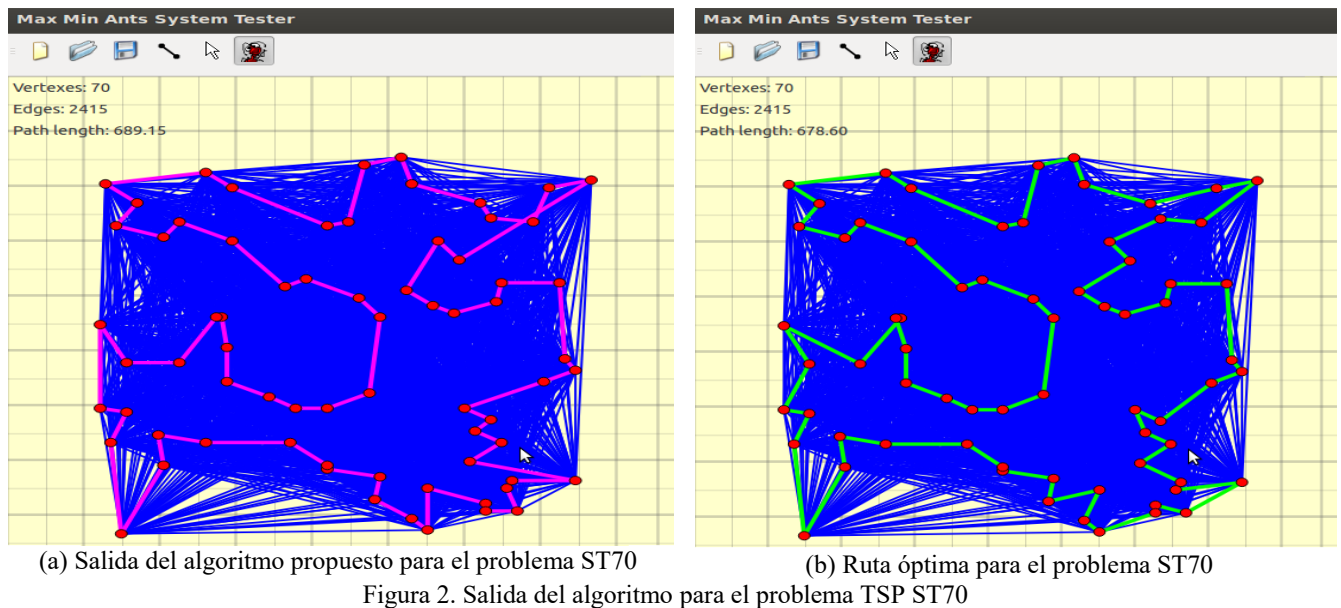
La **¡Error! No se encuentra el origen de la referencia.** Figura 2(a) muestra otro juego de datos comprobado que consta de 70 ciudades, el problema TSP Smith/Thompson ST70. Este ejemplo cuenta con 2415 aristas. La salida del algoritmo implementado para este caso presenta una longitud total de 689,15 unidades. La mejor ruta conocida para resolver este problema se resalta en la Figura 2(b). Esta presenta una longitud total de 678,60 unidades. En este caso, la solución alcanzada presenta una desviación porcentual de solo 1,5%. Nótese también cómo la solución alcanzada presenta el mismo patrón que la mejor solución conocida hasta el momento para este problema.



(a) Salida del algoritmo propuesto para el problema Berlin52

(b) Ruta óptima para el problema Berlin52

Figura 1. Salidas para el problema TSP Berlin52



En el trabajo se puede evidenciar cómo ésta variante de algoritmo basado en colonia de hormigas resulta una herramienta viable para la toma de decisiones en problemas de distribución de mercancías. En general, su aplicación a problemas prácticos, tan comunes y difíciles de resolver en las empresas, significa ahorros considerables en términos de combustibles y demás recursos. De las pruebas realizadas resalta además la importancia que tiene escoger cuidadosamente los valores de los parámetros α , β y ρ , para este tipo de algoritmo, pues de ellos depende mayormente el éxito de la corrida. Para el caso de α , si se elige igual a cero ($\alpha = 0$), la regla probabilística estaría gobernada únicamente por la heurística en cuestión y se estaría frente a una búsqueda estocástica clásica propiamente. Si se elige $\alpha > 1$, el algoritmo converge rápidamente a un estancamiento inicial indeseado. Los mejores resultados se obtuvieron para $\alpha = 1$. Para el caso de β , si se elige igual a cero ($\beta = 0$), la regla probabilística estaría gobernada únicamente por los valores de feromona. Se estaría en este caso ante un algoritmo totalmente aleatorio, que simula las colonias de hormigas naturales. Este caso queda descartado, pues en la práctica no produce buenos resultados. Los mejores resultados se obtuvieron para $\beta = 2$.

Adicionalmente, se recomienda hibridar la presente implementación con técnicas de búsqueda local, con el fin de refinar las soluciones encontradas. La hibridación consiste en aplicar una búsqueda local sobre las soluciones construidas por todas las hormigas en cada iteración antes de actualizar la feromona. Específicamente se recomienda el algoritmo 3-opt descrito en(Davendra, 2010).

El campo de aplicación de este algoritmo es inmenso. Puede aplicarse a un problema de distribución de mercancía o a un videojuego de realidad virtual que requiera de estos temas de inteligencia artificial basados en algoritmos bioinspirados. De igual manera y con relevancia puede ser aplicado para la resolución de problemas prácticos, como el de realizar apertura de huecos en circuitos impresos sobre placas electrónicas; típicamente el caso de una placa madre de un ordenador.

Conclusiones

Con la presente investigación se expuso cómo el algoritmo MMAS resulta una alternativa viable para la resolución de problemas de distribución, tipo TSP simétrico. La elección adecuada de los valores β , α y ρ resulta crucial para este tipo de algoritmo, pues de ellos depende mayormente su éxito. Aplicar estas técnicas a problemas tan comunes y difíciles de resolver en las empresas significa ahorros considerables para sus economías. Contribuye al uso eficiente de los recursos energéticos a la vez que reduce tiempos de entrega. La implementación de este procedimiento en el presente trabajo constituye en adición una herramienta de alto valor agregado, ya sea con fines docentes o profesionales. Su comprobación ante instancias de problemas TSP de reconocida complejidad así lo demuestra.

Referencias

- ARANGO-SERNA, M. D.; H. GIL-GOMEZ y J. ZAPATA-CORTÉS. *Logística esbelta aplicada al transporte en el sector minero*. Boletín de Ciencias de la Tierra, 2009, (25), p. 121-136.
- BARCOS, L.; V. RODRÍGUEZ; M. J. ÁLVAREZ y F. ROBUSTÉ. Algoritmo basado en la optimización mediante colonias de hormigas para la resolución del problema del transporte de carga desde varios orígenes a varios destinos. In *V Congreso de Ingeniería del Transporte. Santander*. 2002, p. 709-717.
- DAVENDRA, D. *Traveling Salesman Problem, Theory and Applications*. edited by RIJEKA. IN-TECH, 2010.
- DE-LA-FUENTE-GARCÍA, D.; J. L. MOSTERÍN; E. O. DE-OLANO-HOYOS y M. DÍAZ. Estado del arte de algoritmos basados en colonias de hormigas para la resolución del problema VRP. In *XV Congreso de Ingeniería de Organización: Cartagena, 7 a 9 de septiembre de 2011*. 2011, p. 811-824.
- DORIGO, M.; V. MANIEZZO y A. COLORNI. *Ant system: optimization by a colony of cooperating agents*. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 1996, 26(1), p. 29-41.
- FEITÓ, M. *Optimización por colonia de hormigas. Caso de estudio en Almacenes Universales S.A.* Vector, 2009, 4, p. 59-66.

- MANIEZZO, V.; L. GAMBARDELLA; MARIA; y F. DE-LUIGI. *Ant Colony Optimization*. edited by G.C. ONWUBOLU AND B.V. BABU. Springer-Verlag Berlin Heidelberg, 2004. 101-117 p.
- MOCHOLI-AGÜES, J. A. *SABACO: Extensiones a los Algoritmos de Optimización basados en Colonias de Hormigas para la Toma de Decisiones Influenciada por Emociones y el Aprendizaje de Secuencias Contextuales en Ambientes Inteligentes* 2011, p.
- MUÑOZ, M. A.; J. A. LÓPEZ y E. F. CAICEDO. *Inteligencia de enjambres: sociedades para la solución de problemas (una revisión)*. Ingeniería e Investigación, 2008, 28(2), p. 119-130.
- RASHID, M. F. F.; W. HUTABARAT y A. TIWARI. *A review on assembly sequence planning and assembly line balancing optimisation using soft computing approaches*. The International Journal of Advanced Manufacturing Technology, 2012, 59(1), p. 335-349.
- RAVE, J. I. P. *Heurística inspirada en el análisis sistémico del “Vecino más cercano”, para solucionar instancias simétricas TSP, empleando una base comparativa multicriterio* 2011, p.
- SOCHA, K.; J. KNOWLES y M. SAMPELS. *A max-min ant system for the university course timetabling problem*. Ant algorithms, 2002, 2463, p. 1-13.
- STÜTZLE, T. y H. HOOS. *MAX-MIN Ant System*. Future generation computer systems, 2000, 16(8), p. 889-914.
- WU, M. T.; T. P. HONG y C. N. LEE. *A continuous ant colony system framework for fuzzy data mining*. Soft Computing-A Fusion of Foundations, Methodologies and Applications, 2012, 16, p. 1-12.