

La metaheurística de algoritmos de colonias de hormigas

Claudia M. Hernández
 Docente del Departamento de Telecomunicaciones
 Universidad del Cauca - Facultad de Ingeniería Electrónica y Telecomunicaciones
 claudiah@unicauca.edu.co

Resumen

Los algoritmos de colonias de hormigas constituyen una aproximación hacia la solución de problemas de tipo NP que ha sido explorada en los últimos años. En este artículo se presentan sus fundamentos, sus variantes y algunas aplicaciones en los diferentes campos del conocimiento.

Palabras clave: problemas de optimización combinatoria, metaheurística, algoritmos de colonias de hormigas, solución óptima.

I. Introducción

El estudio de las diferentes disciplinas del conocimiento conduce hacia la búsqueda de soluciones a problemas particulares, analizados previamente mediante diversas estrategias, con el fin de encontrar soluciones de buena calidad en tiempos razonables. Cada campo del conocimiento tiene asociado un conjunto de problemas que deben ser resueltos para brindar soluciones a determinados requerimientos.

Un problema de optimización es aquel para el cual existen muchas posibles soluciones y un método de comparación entre ellas. Si el conjunto de soluciones es contable, el problema se considera de

optimización combinatoria [1]. A su vez, se han clasificado los problemas combinatorios en las clases P y NP. La clase P es el subconjunto de problemas cuyos algoritmos tienen un tiempo de ejecución que crece de forma polinómica con el tamaño del problema, estos pueden resolverse eficientemente.

Sin embargo, para la mayoría de problemas analizados con intereses prácticos o científicos, no se conoce un algoritmo exacto con complejidad polinómica que lo resuelva, estos problemas se definen como NP, lo que quiere decir que no pueden resolverse en un tiempo polinomial.

Los problemas NP se pueden resolver de varias formas, una de ellas es empleando métodos heurísticos, los cuales exploran

el espacio de soluciones y obtienen una solución rápida que no necesariamente es la óptima. Por otro lado se tienen los métodos aproximados, que brindan una forma sencilla y aproximada de encontrar una solución con baja complejidad y baja calidad.

En las últimas décadas, se han buscado técnicas que brinden una solución de alta calidad y baja complejidad a dichos problemas [2], entre estas se destaca la metaheurística, un proceso de generación iterativa que guía a una heurística subordinada para estructurar la información y encontrar eficientemente soluciones cercanas a las óptimas [3], [4].

En Colombia, algunos grupos de investigación trabajan con algoritmos de colonias de hormigas, entre ellas la Universidad del Valle, que emplea los algoritmos de colonias de hormigas para solucionar problemas del área de sistemas de control. En la Escuela Colombiana de Ingeniería Julio Garavito se han desarrollado escenarios de aprendizaje de los algoritmos de colonias de hormigas. En la Universidad del Cauca, se están utilizando los algoritmos de colonias de hormigas para la etapa de detección de señal en un sistema MIMO 2x2 y 4x4, trabajando con el esquema de multiplexación espacial.

El artículo está organizado como se describe a continuación, en la segunda sección se analizan las metaheurísticas, en la tercera el algoritmo de colonias de hormigas original, en la cuarta se presentan los tipos de algoritmos de colonias de hormigas. En la sección cinco se muestran las aplicaciones de los algoritmos de colonias de hormigas y finalmente la sección seis incorpora las conclusiones.

2. Metaheurísticas

Las metaheurísticas son estrategias inteligentes para diseñar o mejorar procedimientos heurísticos muy generales con un alto rendimiento. El término metaheurística fue propuesto por Fred Glover en 1986 [5]. Desde entonces se han presentado

múltiples pautas de diseño de procedimientos para resolver problemas de asignación de recursos, planificadores, enrutamiento vehicular [6] y se han obtenido resultados satisfactorios.

El comportamiento experimental de la mayoría de las metaheurísticas es bueno, referido a la calidad de la solución de un problema específico y al tiempo necesario para resolverlo, por lo tanto se consideran como una alternativa factible para problemas difíciles de resolver y para encontrar una solución de calidad en un tiempo razonable [6].

Se pueden definir dos tipos de metaheurísticas, las trayectoriales y las poblacionales. Las primeras manejan una sola solución y realizan la búsqueda de la solución siguiendo una trayectoria en el espacio de soluciones a través de operaciones de movimientos, entre éstas se destacan: búsqueda tabú, búsqueda local iterativa, métodos multiarranque, recocido simulado [6].

Las metaheurísticas poblacionales, realizan un proceso de búsqueda manteniendo simultáneamente un conjunto de soluciones, las más utilizadas son: algoritmos evolutivos, algoritmos de estimación de la distribución, algoritmos de colonias de hormigas y algoritmos sociales jerárquicos [5], [6], [9].

Los algoritmos de colonias de hormigas - ACH son muy efectivos para resolver problemas de optimización, las hormigas son insectos sociales que viven en colonias, se caracterizan por buscar la sobrevivencia colectiva por encima de la individual, otro aspecto interesante de las hormigas que ha sido investigado, es cómo pueden encontrar los caminos más cortos entre el alimento y el hormiguero [7].

Las investigaciones han demostrado que los algoritmos basados en colonias de hormigas son muy efectivos para resolver un amplio rango de problemas que se consideran NP Hard [8], [9], [10]. ACH, siendo una metaheurística, han sido empleados en la solución de los problemas mencionados anteriormente, y además se empleó para resolver el problema del viajante del comercio, un problema representativo de la familia de los

problemas dinámicos de optimización combinatoria, y uno de los más estudiados [11].

3. Algoritmos de colonias de hormigas

Los algoritmos de colonias de hormigas están inspirados en el comportamiento de las hormigas reales y sus interacciones en la búsqueda de comida. Algunas especies de hormigas “*Lasius Niger* o *Iridomyrmex humilis*” depositan en su camino una sustancia que puede olerse, la feromona. Sin rastro de feromona las hormigas se mueven de forma aleatoria, pero si existe algún rastro, las hormigas siguen el camino marcado con la mayor concentración de ésta [8].

En un ACH se tiene un conjunto de agentes computacionales que trabajan en forma conjunta y se comunican mediante rastros de feromona artificial, que es evaporada de la misma forma.

Para resolver un problema particular empleando ACH es importante seguir una serie de pasos que permitan encontrar una solución, estos son: modelar el problema en forma de árbol, tener un conjunto finito de posibles soluciones, asignar significado a los rastros de feromona, definir la información heurística del problema a tenerse en cuenta, escoger un tipo de ACH y fijar sus parámetros de configuración. En la figura 1 se presenta un resumen de estos pasos [12].

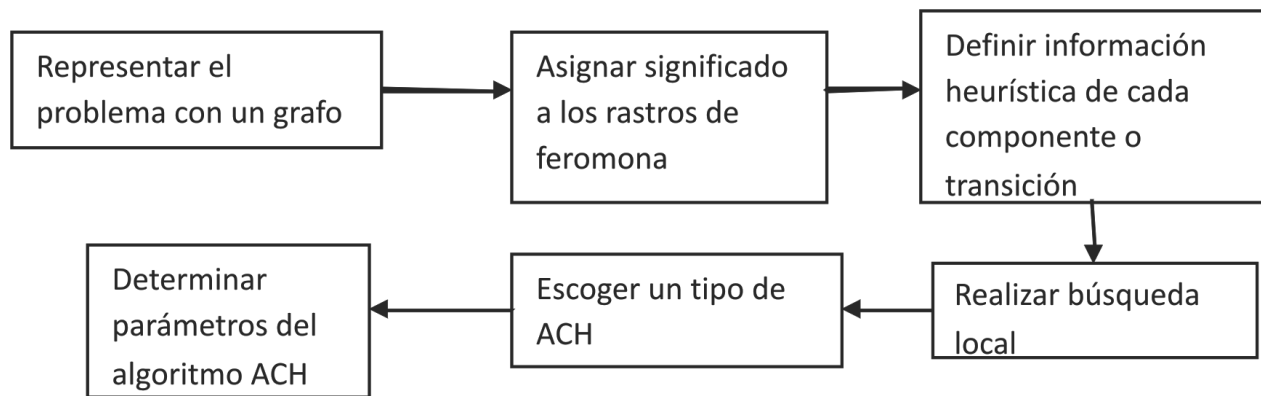


Figura 1. Pasos para analizar un problema con ACH

A medida que las hormigas transitan por el mismo camino, mayor es la concentración de feromona, dado que las hormigas regresan rápidamente al hormiguero, por ser el camino más corto. Después de un tiempo, toda la colonia de hormigas se mueve por el mismo camino.

Pero las hormigas pueden quedar estancadas en una ruta que no sea la más corta, porque inicialmente se ha depositado una gran cantidad de feromona en el camino equivocado, esta característica lleva a la colonia a un estancamiento, dado que no pueden encontrar el camino más corto, dado que la feromona no se evapora rápidamente. Este comportamiento natural debe corregirse en el algoritmo computacional para garantizar que se encuentre la mejor solución a un problema específico.

De forma general, un algoritmo de colonia de hormigas busca la solución a un problema en 3 pasos [11], estos se representan en el pseudocódigo de la figura 2.

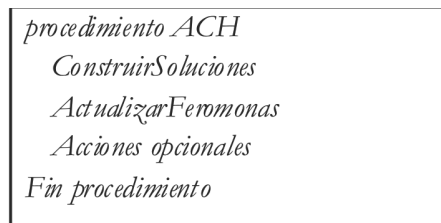


Figura 2. Pseudocódigo de un ACH

Construir soluciones: gestiona la colonia de hormigas que se está moviendo entre los nodos del grafo del problema, para realizar los movimientos las hormigas toman decisiones probabilísticas, usando los trayectos de feromona y la información

heurística asociada y construyen incrementalmente las soluciones. Cuando una hormiga ha construido una solución o mientras la construye, evalúa la solución parcial que será usada por el procedimiento de actualización de feromona para decidir cuanta feromona utilizar.

Actualizar feromona: proceso que modifica los trayectos de feromona, las hormigas depositan feromonas en los nodos o en las conexiones que utilizan. Mediante la evaporación de feromona se disminuye la concentración de feromona con el paso del tiempo, esta constituye una forma de evitar el estancamiento del algoritmo en una solución equivocada. Entre mayor sea la cantidad de feromona que existe en un nodo o en una conexión, mayor será la probabilidad de que dicho nodo o conexión sea empleado por otras hormigas.

Acciones opcionales: procedimientos empleados para realizar acciones centralizadas que requieren de un conocimiento global, que no pueden ser desarrolladas por las hormigas individuales, por ejemplo la decisión de depositar cantidades adicionales de feromonas en algunas conexiones o nodos, este procedimiento no tiene uno similar en el ambiente natural.

En la figura 3 se presenta el diagrama de flujo asociado con el funcionamiento de un algoritmo de optimización basado en colonias de hormigas [7], [8], [9], [10].

El primer sistema de hormigas fue planteado por Marco Dorigo en 1992, “*ANT System*”, de él se mantiene el procedimiento de construcción de soluciones y la evaporación de feromona, las principales diferencias con los algoritmos propuestos posteriormente, radican en la forma de actualización de la feromona y en algunos aspectos de la gestión de los trayectos de feromona. Entre los principales tipos de algoritmos de optimización basada en colonias de hormigas se pueden nombrar: “*Elitist AS*”, “*Rank-based AS*”, “*MAX - MIN AS*” y “*Ant Colony System (ACS)*” [7], [9], [10].

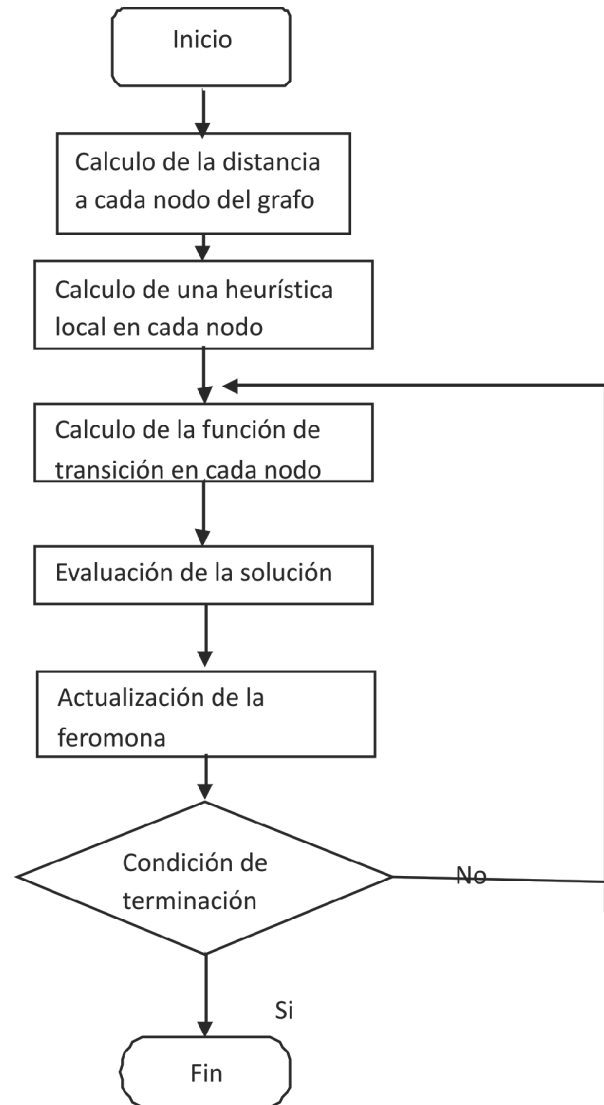


Figura 3. Diagrama de flujo

En pruebas realizadas a los algoritmos en la solución del problema del viajante del comercio, se ha encontrado que todas las variantes presentan mejor desempeño que el algoritmo original, y entre ellas se clasifican como los mejores: el sistema de “Max-Min”, “Ant Colony System” y “Rank-based AS” [11].

4. Tipos de OCH

A. *Ant System (AS)*

Fue el primer sistema, representa el comportamiento de las hormigas de forma casi

exacta, en este sistema se basan las variantes que surgieron posteriormente [9], [10], [11].

Construcción de la solución

Para que una hormiga construya una solución a un problema específico, debe recorrer una ruta en el grafo de búsqueda, conformada por un conjunto de nodos a través de los cuales se mueve de acuerdo a decisiones probabilísticas. Cuando una hormiga se encuentra en un nodo del grafo, para decidir hacia donde dirigirse sigue una regla de transición dada por:

$$\Delta\rho_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha n_{ij}^\beta}{\sum_{l \in N_i^k} n_{il}^\beta \tau_{il}^\alpha} & \text{si } j \text{ pertenece a } K \\ 0 & \text{de otra forma} \end{cases} \quad (1)$$

α y β son los parámetros que controlan la importancia relativa de la feromona vs la información heurística dadas por

$$n_{ij} = \frac{1}{d_{ij}} \quad (2)$$

d_{ij} es la longitud del arco que une los nodos (i,j) del grafo

N_i^k representa los nodos a los que se puede mover la hormiga, si α es cero los nodos más cercanos tienen mayor probabilidad de ser escogidos, si α es mayor que 1 todas las hormigas siguen el mismo camino y construyen el mismo tour. Si β es cero solo se tienen en cuenta los rastros de feromona para la elección, esto puede generar un estancamiento rápido del algoritmo, porque las hormigas siempre siguen las mismas soluciones, es necesario establecer un equilibrio entre la información heurística y la información de los rastros de feromona [11]. Finalmente, una hormiga se mueve al nodo que tiene el valor más alto de $\Delta\rho_{ij}^k$.

La construcción de la solución se puede realizar de forma secuencial o paralela. El desempeño de este algoritmo decrece drásticamente a medida que se incrementa el tamaño de las instancias de prueba.

Actualización de la feromona

Una vez que las hormigas completan el recorrido, es necesario depositar feromona para marcar los trayectos recorridos, esto se realiza de acuerdo a la

expresión:

$$\tau_{ij} = \tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k \quad \forall (i,j) \in L \quad (3)$$

m el número de hormigas

$\Delta\tau_{ij}$ es la cantidad de feromonas por unidad de longitud depositada en el lado (i,j) por la hormiga k .

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_Q} & \text{si la hormiga } k \text{ usa el borde } (i,j) \text{ en su ruta} \\ 0 & \text{de otra forma} \end{cases} \quad (4)$$

Q es una constante y L_Q es la longitud de la ruta de la hormiga.

Evaporación de la feromona

El algoritmo simula el efecto natural de evaporación de la feromona, de acuerdo a la expresión:

$$\tau_{ij} = (1 - \rho)\tau_{ij} \quad \forall (i,j) \in L \quad 0 < \rho \leq 1 \quad (5)$$

ρ es la tasa de evaporación de la feromona, se utiliza para evitar la acumulación ilimitada de trayectos de feromona y permitir que el algoritmo olvide malas decisiones tomadas anteriormente, eliminando parte de la feromona depositada en los trayectos a medida que transcurre el tiempo.

Acciones Adicionales

En esta primera versión del algoritmo no se implementan tareas distintas a las del ambiente natural, por lo tanto no se tienen acciones adicionales.

A continuación se presentan algunas variantes del sistema de colonias de hormigas original.

B. Sistema Max-Min

Propuesto por Stutzle y Hoos[10], quienes realizaron cambios al original:

- Solamente la mejor hormiga puede actualizar los trayectos de feromona.
- Los valores mínimos y máximos de la feromona se limitan para evitar que las hormigas se estancuen en una solución que no sea la mejor.
- Los trayectos de feromona inician en el límite mínimo, para beneficiar la exploración de rutas al comienzo de la

búsqueda.

- Los trayectos de feromona se reinician cada vez que la búsqueda se estanca o cuando no se ha generado una mejor ruta sobre un número de iteraciones consecutivas.

La construcción de la solución se mantiene igual, a continuación se presentan los pasos que tienen modificaciones con respecto al algoritmo original.

Actualización

La actualización de la feromona se realiza así:

$$\tau_{ij} = \tau_{ij} + \Delta\tau_{ij}^{best} \quad \forall (i, j) \in L \quad (6)$$

$$\Delta\tau_{ij}^{best} = \frac{1}{C^{best}} \quad (7)$$

La hormiga que se le permite adicionar feromona puede ser la que ha construido la mejor ruta hasta el momento $\Delta\tau_{ij}^{best} = 1/C^{best}$, o la mejor iteración $\Delta\tau_{ij}^{best} = 1/C^{ib}$ es la longitud de la mejor iteración, generalmente se usan alternadamente cada una de las reglas de actualización.

Los límites de los valores de feromona se escogen buscando que el algoritmo no se estanque, por tanto el valor máximo se estima como:

$$\tau_{max} = \frac{1}{\rho} * \frac{1}{C^{bs*}} \quad (8)$$

Cada vez que se encuentra una mejor ruta, el valor de τ_{max} se actualiza, para el valor mínimo se emplea

$$\tau_{min} = \frac{\tau_{max}}{a} \quad (9)$$

Acciones adicionales:

Consisten en encontrar la mejor ruta hasta el momento o la mejor iteración, para depositar feromona.

C. Sistema de colonias de hormigas

Dorigo & Gambardella trabajaron una nueva variante de AS, la cual emplea una regla de transición más fuerte, actualiza la feromona solamente en la mejor ruta, y además cada vez que una hormiga usa un arco del grafo, se remueve algo de feromona del mismo para incrementar la

exploración de otros trayectos y evitar el estancamiento del algoritmo [9], [11].

Construcción de la solución

Para que una hormiga k ubicada en un nodo i del grafo decida hacia donde debe moverse se emplea una nueva regla proporcional pseudoaleatoria dada por:

$$j = \begin{cases} \operatorname{argmax}_{l \in N_i^k} \{\tau_{il}[n_{il}]^{\beta}\}, & \text{si } q \leq q_0 \\ J & \text{de otro modo} \end{cases} \quad (10)$$

q es una variable aleatoria, uniformemente distribuida entre $[0,1]$, q_0 ($0 < q_0 < 1$) es un parámetro y J es una variable aleatoria seleccionada de acuerdo a la distribución de probabilidad dada por la segunda ecuación (1) con $\alpha=1$.

Actualización

En ACS solo deposita feromona la hormiga de la mejor ruta encontrada hasta el momento R^{bs} , la actualización se realiza así:

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \rho\Delta\tau_{ij}^{bs} \quad \forall (i, j) \in R^{bs} \quad (11)$$

$$\Delta\tau_{ij}^{bs} = 1/C^{bs} \quad (12)$$

Este algoritmo es uno de los más empleados por su buen desempeño y complejidad moderada.

Acciones adicionales:

Consisten en realizar la evaporación de la feromona en las conexiones o nodos visitados por la hormiga, y en la selección de la mejor ruta explorada.

5. Aplicaciones

Los algoritmos de colonias de hormigas están siendo ampliamente estudiados y analizados en diferentes áreas del conocimiento, tienen características únicas, como la actualización de la feromona basada en la calidad de las soluciones o el uso de diferentes estrategias para construir soluciones, lo que permite realizar un procesamiento más eficiente en la búsqueda de una solución.

Entre las metaheurísticas poblacionales, ACH se destaca como una de las mejores técnicas de optimización, por ejemplo, con respecto a los algoritmos evolutivos, en estos, el conocimiento del problema está contenido en la población actual, mientras que en la ACH se almacena en la estructura de memoria que guarda los rastros de feromona [13].

Las principales aplicaciones se han dado en diferentes áreas: Asignación Cuadrática, Secuenciación de Tareas, Enrutamiento Vehicular, Ordenación Secuencial, Líneas de producción de autos, entre otras, en cada una de estas aplicaciones se han logrado buenos resultados de desempeño del algoritmo [11], [12].

En comunicaciones inalámbricas, se ha empleado ACH para detectar usuarios en un ambiente multiusuario [13], obteniendo resultados satisfactorios. Actualmente, en la Universidad del Cauca se está analizando la posibilidad de adaptar un algoritmo de detección de señal en un sistema MIMO 2x2 y 4x4, empleando este algoritmo.

6. Conclusiones

Los algoritmos de colonias de hormigas han sido empleados para resolver problemas de tipo NP y se han obtenido resultados satisfactorios, por lo tanto se espera que sea posible resolver este tipo de problemas en el área de telecomunicaciones.

A pesar de que en las pruebas de desempeño realizadas a los tipos de algoritmos de colonias de hormigas, el sistema “*Ant System*” brinda los resultados más pobres, este se constituye en un algoritmo muy importante, por ser la base de todas las variantes y porque sus principios se mantienen, lo que se modifica son las formas de actualización y evaporación de feromona.

Existen diferentes formas de mejorar el desempeño de los algoritmos de colonias de hormigas, sin embargo esto implica incrementar la

complejidad computacional del algoritmo, por lo tanto es necesario ser cuidadoso para encontrar un punto de equilibrio entre desempeño y complejidad.

Existen múltiples campos en los que pueden aplicarse los algoritmos de colonias de hormigas, en este trabajo se analiza la factibilidad de emplearse en la detección de señal en un sistema MIMO 2x2 y 4x4, hasta el momento se ha modelado el problema en forma de grafo, se ha definido el significado de los rastros de feromona y se han establecido los tipos de algoritmos de colonias de hormigas con los cuales se trabajará.

Los ACH se constituyen en una opción viable para solucionar problemas NP de diferentes áreas, buscando garantizar encontrar una solución óptima en un tiempo eficiente computacionalmente.

7. Bibliografía

- [1] J. Pantrigo, “Resolución de Problemas de Optimización Dinámica mediante la Hibridación entre Filtros de Partículas y Metaheurísticas Poblacionales,” Tesis Doctoral, Universidad Rey Juan Carlos, 2005.
- [2] A. Khan, M. Naeem, S. Shah “Binary Ant Colony Algorithm for Symbol Detection in a Spatial Multiplexing System,” en *Unconventional Computation*, Ed Springer Berlin, 2007, pag. 115-126.
- [3] A. Duarte, “Panorámica de los procedimientos metaheurísticos,” en *Seminario Sobre Sistemas Inteligentes*, 2007, pag. 331-341.
- [4] J. Pantrigo, “Resolución de Problemas de Optimización Dinámica mediante la Hibridación entre Filtros de Partículas y Metaheurísticas Poblacionales,” Tesis Doctoral, Universidad Rey Juan Carlos, 2005.
- [5] R. Marti, “Procedimientos Metaheurísticos en Optimización Combinatoria,” *Mathematiques*, vol. 1, no. 1, 2003.[online]. Disponible: <http://www.revicien.net/>. Accedido [15 de Abril de 2009].
- [6] S. Krzysztof, M. Dorigo. “Ant colony optimization for continuous domains,” *European Journal of Operational Research*, vol. 185, no 3, pag. 1155-1173, 2008.
- [7] M. Dorigo, G. Di Caro, y J. Gambardella, “Ant algorithms for discrete optimization,” *Artificial Life*, vol. 5, no. 2, pag 137 - 172, 1999.

- [8] S. Krzysztof, G. D. Caro, "An Introduction to Ant Colony Optimization," en *Approximation Algorithms and Metaheuristics*, 2006.
- [9] M. Dorigo and T. Stutzle, "The Ant colony optimization metaheuristic: algorithms, applications and Advances," en *Handbook of Metaheuristics*, volumen 57, Ed. Springer New York, 2006, pag. 250-285.
- [10] G. Singh, S. Das, S. Gosavi, y S. Pujar, "Ant colony algorithms for steiner trees: an application to routing in sensor networks," en *Recent Developments in Biologically Inspired Computing*, Ed. Idea Group Publishing, 2005, pag. 181-206.
- [11] M. Dorigo and T. Stutzle, "*Ant colony optimization*" MIT Press, 2004.
- [12] D. Darquennes, "Implementation and Applications of Ant Colony Algorithms," M.Sc. Tesis, University of Namur, Belgium, 2005.
- [13] S. Alonso, O. Cordón , et al, "La Metaheurística de Optimización Basada en Colonias de Hormigas: Modelos y Nuevos Enfoques," en *Optimización inteligente: técnicas de inteligencia computacional para optimización*, Ed. Universidad de Málaga, pag. 261-314, 2004.
- [14] S. Hijazi, B. Natarajan, et al, "An Ant Colony Algorithm for Multiuser Detection in Wireless Communication Systems," *Genetic and Evolutionary Computation Conference*, pag. 2121–2126, Junio 2005.