Revista Ciencia & Desarrollo 2015; 20: 98-102 / ISSN 2304-8891

APLICACIÓN DEL ALGORITMO DE COLONIA DE HORMIGAS AL PROBLEMA DEL AGENTE VIAJERO

APPLICATION OF ANT COLONY ALGORITHM TO THE TRAVELIING SALESMAN PROBLEM

[1]Hugo Euler Tito Chura, [2]Carlos Alberto Silva Delgado, [2]Edith Elizabeth Alfaro Gonzales, [3]Evelyn Fajardo Espinoza

RESUMEN

ACO (algoritmo de colonia de hormigas) es una metaheurística inspirada en el comportamiento de las colonias de hormigas para solucionar problemas de optimización combinatoria, por medio de la utilización de agentes computacionales simples que trabajan de manera cooperativa y se comunican mediante rastros de feromonas artificiales. En este trabajo se presenta un modelo para resolver el Problema clásico de optimización "*Problema del Agente viajero*" (TSP, Travelling Salesman Problem).

Palabras Clave: algoritmo de colonia de hormigas, agente viajero (Travelling Sales Problem), optimización.

ABSTRACT

ACO (ant colony algorithm) is a metaheuristic inspired by the behavior of ant colonies to solve combinatorial optimization problems, through the use of simple computational agents that work cooperatively and communicate via artificial pheromone trails. This paper presents a model for solving the classic optimization problem "travelling salesman problem" (TSP Travelling Salesman Problem).

Keywords: ant colony algorithm, traveling salesman (Travelling Sales Problem) optimization.

INTRODUCCIÓN.

La existencia de una gran cantidad y variedad de Problemas de Optimización Combinatoria incluidos en la clase NP-completos que necesitan ser resueltos de forma eficiente, impulsó el desarrollo de procedimientos para encontrar buenas soluciones, aunque no fueran óptimas. Estos métodos, en los que la rapidez del proceso es tan importante como la calidad de la solución obtenida, se denominan heurísticos o aproximados.

Un método heurístico es un procedimiento para resolver un problema de optimización bien definido mediante una aproximación intuitiva, en la que la estructura del problema se utiliza de forma inteligente para obtener una buena solución (Hong & Bian, 2008). Luego, con el propósito de obtener mejores resultados que los alcanzados por los heurísticos tradicionales surgen los denominados procedimientos metaheurísticos. Los procedimientos metaheurísticos son una clase de métodos aproximados que están diseñados para resolver problemas difíciles de optimización combinatoria, en los que los heurísticos clásicos no son efectivos. Los metaheurísticos proporcionan un marco general para crear nuevos algoritmos híbridos combinando diferentes conceptos derivados de la inteligencia artificial, la evolución biológica y los mecanismos estadísticos (Wang & Yang, 2009).

Se han desarrollado varias metaheurísticas para solucionar problemas de optimización, entre ellas se encuentran: Búsqueda Tabú (Yang, *et al.*, 2009), simulated annealing (Shakouri *et al.*, 2009), Algoritmos Genéticos (Yu *et al.*, 2011), Optimización basada en Enjambre de Partículas o Particle Swarm Optimization (PSO) (Zhang *et al.*, 2007), (Duan & Ying, 2009) y Optimización basada en Colonias de Hormigas (ACO) (Bi *et al.*, 2011) (Wang & Zhu, 2008), de esta última incluida en la categoría de los algoritmos bioinspirados o de vida artificial e inteligencia colectiva (Dorigo & Stützle, 2004), trataremos en el presente trabajo.

Estado del arte

Colonias de hormigas.

En el quehacer de las hormigas se distingue la búsqueda de alimentos, trazando el camino más corto entre el hormiguero y el emplazamiento de alimentos. El eje de esa búsqueda es el depósito de *feromona* como *rastro* que orienta el recorrido. Las hormigas prefieren seguir la mayor concen-

[1] Magíster en Ciencias Computación e Informática, Ingeniero Estadístico. Docente de la Escuela de Ingeniería de Sistemas e Informática de la Universidad Nacional de Moquegua. Moquegua-Perú.

[2] Ingeniero en Informática y Sistemas. Docente de la Escuela Profesional de Ingeniería de Sistemas e Informática de la Universidad Nacional de Moquegua. Moquegua-Perú.

[3] Administradora de Empresas. Universidad Nacional Jorge Basadre Grohmann. Tacna-Perú.

tración de feromona, la misma que se consigue por el recorrido más corto hecho por hormigas (que por esta razón, pueden repetirlo en forma más seguida). En principio siguen rutas aleatorias, pero las que han utilizado las de menor longitud pueden regresar más rápidamente (pues transitan a una velocidad uniforme), y repetir el camino, dejando una mayor densidad de huellas de feromona. Las depositadas en las otras rutas se van evaporando, y dejan de tener interés en los nuevos recorridos (Dorigo & Stützle, 2004). Ello se ilustra en las siguientes figuras (Figura 1 y Figura 2), adaptada de las imágenes clásicas para este tema.

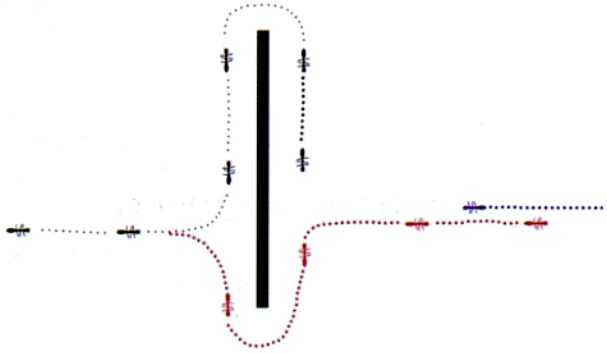

Figura 1. Hormigas buscando el camino corto

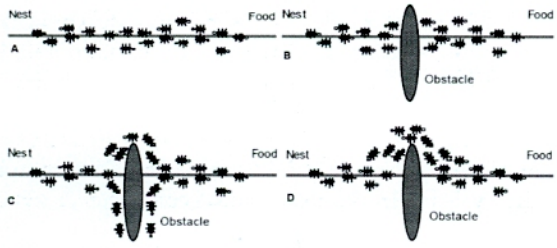

Figura 2. Hormigas siguiendo las feromonas

Algoritmo basado en Colonia de hormigas.

Consiste en la simulación de una colonia de hormigas artificiales que trabajan en grupo y se comunican a través de rastros de feromona artificial. Cada hormiga artificial construye una solución al problema recorriendo un grafo de construcción. Cada arista o tramo del grafo tiene dos tipos de información que guían el movimiento de la hormiga (Dorigo & Stützle, 2004):

- La *preferencia* de moverse desde el nodo *r* al nodo *s* en la arista *rs*. Esta preferencia se designa con η_{rs}. El valor no se modifica en el proceso y es función de la distancia entre los nodos.
- La *deseabilidad* aprendida del movimiento de *r* a *s*. Simula la feromona o rastro depositado, y se va modificando. Se designa con τ_{rs}.

Optimización con Colonia de hormigas.

Consiste en la simulación de la reacción que exhibe una colonia de hormigas. La sustancia virtual es denominada *rastro* como similitud a la feromona. El algoritmo sigue la estructura computacional básica siguiente (Dorigo & Stützle, 2004):

Inicio de rutina o ciclo de rastros
Do While (hasta que el criterio no sea satisfecho)
    Do Until (cada hormiga completa un viaje)
        Mecanismo de decisión de la hormiga
        Actualización del rastro local
    End Do
    Actualización del ciclo global de rastros
End Do

La *decisión* está basada en la intensidad de rastro presente en cada recorrido entre dos puntos adyacentes (Dorigo & Stützle, 2004).

- El que tiene más rastros, tiene la mayor probabilidad de ser elegido.
- Si no hay rastros, la probabilidad de elección es cero.
- Si todas las rutas tienen igual cantidad de rastros, la decisión es aleatoria.

Cada hormiga elige una ruta usando un mecanismo de decisión. Algunos algoritmos usan una actualización local en la prueba, reduciendo la cantidad de rastro en el recorrido elegido por la hormiga (la cual tendrá menor probabilidad de elegir la misma trayectoria que las anteriores).

La hormiga continúa los recorridos del viaje entre puntos llegando a cada punto hasta haber visitados todos, y regresa a su punto de origen. Cuando regresa al comienzo, la hormiga ha completado un viaje.

Al completar el viaje, se analiza cuán bien resuelve el problema. La intensidad de rastro en cada recorrido en el viaje es ajustada a través de un proceso de actualización global. Los recorridos que mejor resuelven el problema reciben más rastros. De esta manera, cuando todas la hormigas han completado un viaje y todos los viajes son analizados, así como son actualizados los rastros, el ciclo de optimización se ha completo.

Un nuevo ciclo comienza y se repite el proceso. Casi todas las hormigas seguirán el mismo viaje en cada ciclo y convergerá la solución. Se compara la mejor solución en el último ciclo para obtener la solución global (Dorigo & Stützle, 2004).

Colonia de hormigas en el Problema del Agente Viajero.

En la propuesta hecha por Dorigo & Stützle (2004), Milano en 1992, del problema del agente viajero (Travelling Salesman Problem), cuyo objetivo es buscar la distancia recorrida más corta de su viaje, que puede realizar partiendo de su ciudad natal (nodo), a toda las otras ciudades (sin repetir los caminos recorridos), para finalmente regresar a su destino (ciudad natal).

Entonces, podemos representar en un grafo el problema del agente viajero para 4 ciudades de la siguiente forma. (Ver figura 3).

El número de caminos existentes a ser recorridos de acuerdo al número de nodos se visualiza en la tabla 1.

Siendo n el número de ciudades, i alguna ciudad, $b_i(t)$ el número de hormigas en i en el tiempo t, el total de hormigas m resulta:

$$m = \sum_{i=1}^{n} b_i(t)$$

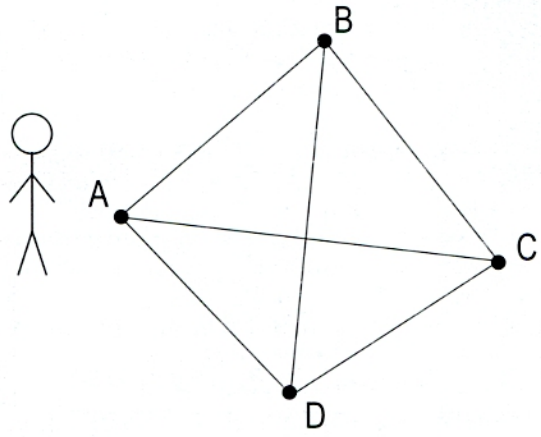

Figura 3. Grafo de 4 ciudades.

Tabla 1. Número de caminos de acuerdo al número de nodos

Nodos	Número de Caminos	
3	3!	6
4	4!	24
5	5!	120
6	6!	720
7	7!	5,040
8	8!	40,320
9	9!	36,2880
10	10!	3'628,800
11	11!	39'916,800
12	12!	479'001,600
13	13!	6,227'020,800

Cada hormiga tiene las siguientes características:

- Selecciona cuál ciudad será la siguiente a ser visitada. Esta selección está basada en una probabilidad que es función de la distancia a la ciudad (o visibilidad) y la cantidad de rastros presente en el tramo que conecta las ciudades (Figura 4).
- Tiene memoria sobre las ciudades que ha visitado (según una lista tabú) y las que no ha visitado.
- Cuando ha completado el viaje, deja una sustancia o rastro en cada tramo.
- Del *rastro*:
  - ✓ La *intensidad del rastro* en la ruta en cada *tramo* que conecta i con j en el tiempo t, se designa con $\tau_{ij}(t)$. En el tiempo 0, $\tau_{ij}(0)$ es tan pequeño como una constante pequeña positiva τ_0.
  - ✓ En el tiempo t, cada hormiga elige una ruta y viaja a la siguiente ciudad en el tiempo t+1.
  - ✓ En cada *iteración* hay m movimientos provocados por las m hormigas (cada una haciendo un movimiento) en el intervalo (t,t+1).
  - ✓ Un *ciclo* es definido como n iteraciones, y significa que las m hormigas han completado un viaje.
  - ✓ Al final de cada iteración, el algoritmo implementa una *actualización local*: reduce el nivel de rastro en los tramos seleccionados por la colonia en la iteración precedente. Cuando una hormiga viaja a la ciudad **j** desde la ciudad **i**, la regla de actualización ajusta la intensidad del rastro del tramo conectado por:

$$\tau_{ij}(t) \leftarrow (1-\varphi)\tau_{ij}(t) + \varphi\tau_0$$

Donde φ es un parámetro ajustable entre 0 y 1 que representa la persistencia del rastro. Además:

$$\tau_0 = \frac{1}{nL_{nn}}$$

Donde L_{nn} es la longitud calculada con el algoritmo heurístico el vecino más cercano (se comienza aleatoriamente en una ciudad y se selecciona la siguiente basada en la distancia más corta hasta visitar todas las ciudades).

Al final de cada ciclo la intensidad del rastro es ajustada usando una *actualización global*. El primer paso es calcular la cantidad de rastro dejada en cada tramo, como:

$$\Delta\tau_{ij}^{k} = \frac{1}{L_k}$$

Donde **k** representa alguna hormiga (de 1 a m). Y $\mathbf{L_k}$ es la longitud del ciclo elegida por una hormiga **k**.

El valor total de actualización del rastro para cada tramo es la suma de las cantidades dejadas por cada hormiga que haya seleccionado dicho tramo **ij**, y será:

$$\Delta\tau_{ij}^{k} = \sum_{i=1}^{m} \Delta\tau_{ij}^{k}$$

El valor de la actualización global está determinado por la siguiente expresión:

$$\tau_{ij}(t+n) = (1-\rho)\tau_{ij}(t) + \rho\Delta\tau_{ij}$$

Donde ϱ es una constante entre 0 y 1, tal que **(1-ϱ)** representa la evaporación del rastro entre **t** y **t+n** (el tiempo requerido para completar un ciclo).

- De la *visibilidad*. El sistema admite que las hormigas posean una cantidad de visibilidad η_{ij}, asociada con cada tramo, en un valor que depende de la distancia del mismo (los cercanos tienen mayor preferencia que los lejanos).

$$\eta_{ij} = \frac{1}{d_{ij}}$$

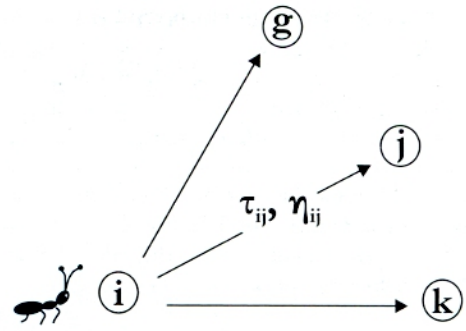

Figura 4. Parámetros de decisión

- *Probabilidad de preferencia.* La combinación de la visibilidad con la intensidad del rastro, resultan en:

$$a_{ij}(t) = \frac{\left[\tau_{ij}(t)\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}}{\sum_{l \in permitidas}^{n}\left[\tau_{il}(t)\right]^{\alpha}\left[\eta_{il}\right]^{\beta}}$$

Se trata de la proporción respecto a las ciudades vecinas permitidas desde la ciudad **i**. Los parámetros α y β son constantes usadas para controlar la importancia de los valores de los rastros locales y visibles.

La probabilidad p^{k}_{ij} de que la hormiga **k** elija ir de **i** a **j** en el tiempo **t** es:

$$p^{k}_{ij}(t) = \frac{a_{ij}(t)}{\sum_{l \in permitidas}^{n} a_{il}(t)}$$

Si la hormiga no está permitida a viajar a la ciudad **j**, entonces $p^{k}_{ij}=0$.

Aplicación del modelo ACO.

Siguiendo las recomendaciones de otros trabajos (Gómez & Barán, 2004) para el problema TSP, el rango de valores de los parámetros del modelo ACO considerados se presenta en la tabla 2, dado que para estos valores de los parámetros se ha mostrado experimentalmente la obtención de buenos resultados en variados estudios de problemas TSP (con $\alpha = 1$).

Se consideran m = 200 hormigas, igual tasa de evaporación para la actualización global y local ($\gamma = \varrho$), y el valor inicial de feromona τ_0 fue determinado considerando el número de ciudades (n) como el número de secuencias parciales a unir. Para la implementación del modelo se utilizó el

Tabla 2. Parámetros del Modelo ACO

Parámetro	Valor
m	200
β	[1-2-3-4-5]
ϱ	[0,1, 0,2, 0,3]
τ_0	1/(n,L)
q_0	[0,9 - 0,95 - 0,98]

Tabla 3. Resultados mejor solución

		$\varrho = 0,1$	$\varrho = 0,2$	$\varrho = 0,3$	*Mínimo*
$q_{0=0,90}$	$\beta = 1$	13469	15050	15666	13469
	$\beta = 2$	12128	12187	14636	12128
	$\beta = 3$	10552	12156	13408	10552
	$\beta = 4$	9660	11087	13049	9660
	$\beta = 5$	**7680**	9468	12719	**7680**
$q_{0=0,95}$	$\beta = 1$	13614	14081	15132	13614
	$\beta = 2$	12760	13262	14178	12760
	$\beta = 3$	10133	12046	12918	10133
	$\beta = 4$	10297	10870	11986	10297
	$\beta = 5$	9522	10812	12434	9522
$q_{0=0,98}$	$\beta = 1$	13670	15086	15498	13670
	$\beta = 2$	11516	12448	13763	11516
	$\beta = 3$	10752	11467	12408	10752
	$\beta = 4$	9926	10523	11500	9926
	$\beta = 5$	9059	10094	13041	9059

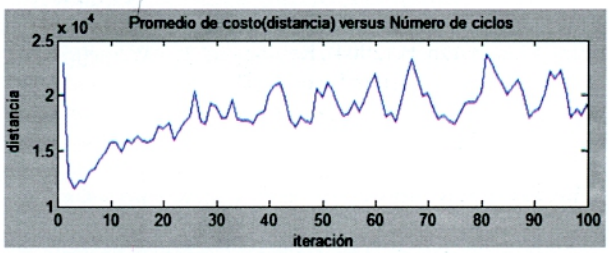

Figura 5. Promedio de costo vs Número de ciclos

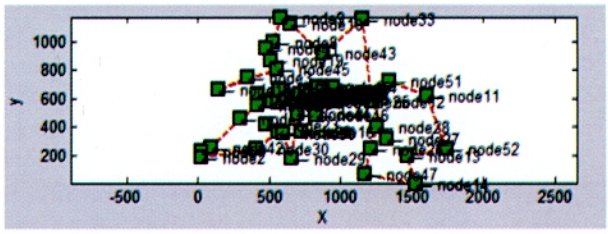

Figura 6. Nodos a ser recorridos.

MatLab con la data de Berlin52 , tomado de TSPLIB.

Se realizaron 5 réplicas para cada combinación de valores de los parámetros ϱ, β y q_0. Los resultados mínimos obtenidos para la suma de las distancias entre secuencias parciales se presentan en la tabla 3, donde podemos apreciar que la mejor solución es 7680.

Por lo tanto, los nodos a ser recorridos son:
nodo=[22, 19, 49, 15, 45, 43, 33, 34, 35, 38, 39, 36, 37, 47, 23, 4, 14, 5, 3, 24, 11, 27, 26, 25, 46, 12, 13, 51, 10, 50, 32, 42, 9, 8, 7, 40, 18, 44, 31, 48, 0, 21, 30, 17, 2, 16, 20, 41, 1, 6, 29, 28].

También tenemos resultados del número de ciclos versus el promedio de distancia en la figura 5 y los nodos a ser recorridos por el agente viajero en la figura 6.

CONCLUSIONES

En nuestro experimento, aplicado a la data Berlin52, que cuenta con 52 ciudades, obtuvimos la mejor solución de 7680 de distancia mínima, con parámetros de: $\alpha = 1$, $\beta = 5$, $\varrho = 0,1$, $q_0 = 0,90$.

La aplicación de este modelo de optimización entrega la posibilidad de generar mejoras en las distancias totales mínimas de recorrido y a su vez nos muestra los nodos por los que se deben transitar, en problemas que tengan el contexto del Agente Viajero (TSP).

REFERENCIAS BIBLIOGRÁFICAS

Bi, S., Dong, X., Ma, Y. (2011). The Design and Analysis of TSP Problem Based on Genetic Algorithm and Ant Colony Algorithm. etcs, Third International Workshop on Education Technology and Computer Science. IEEE. vol. 1, pp.324-326

Dorigo, M. & Stützle, T. (2004). Ant Colony Optimization. Massachusetts Institute of Technology. Proceedings of 4th International Workshop, ANTS 2004, Brussels, Belgium, September 5 – 8.

Duan, Y. & Ying, Y. (2009). A Particle Swarm Optimization Algorithm with Ant Search for Solving Traveling Salesman Problem. Cis. International Conference on Computational Intelligence and Security. IEEE. vol.

2, pp.137-141

Gómez, O. & Barán, B. (2004). Reasons of ACO's Success in TSP. Massachusetts Institute of Technology. Conference: Ant Colony Optimization and Swarm Intelligence, Proceedings of 4th International Workshop, ANTS 2004, Brussels, Belgium, September 5 – 8.–

Hong, Z. & Bian, F. (2008). Novel Ant Colony Optimization for Solving Traveling Salesman Problem in Congested Transportation System, paciia, 2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application. vol. 2, pp.122-125.

Li, G. & Shujing, Z. (2009). Adaptive ACO for Complicated Optimization Problems. ifita. International Forum on Information Technology and Applications. vol. 2, pp.15-18.

Shakouri, G., Shojaee, K. & Behnam, T. (2009). Investigation on the choice of the initial temperature in the Simulated Annealing: A mushy state SA for TSP. med. 17th Mediterranean Conference on Control and Automation. IEEE. pp. 1050-1055

Wang, A. & Yang, L. (2009). The Application of the Ant Colony Pheromones in Intelligent Learning. Ifita. International Forum on Information Technology and Applications. vol. 3, pp.488-491.

Wang, L & Zhu, Q. (2008). An Efficient Approach for Solving TSP: The Rapidly Convergent Ant Colony Algorithm," icnc. Fourth International Conference on Natural Computation. IEEE. vol. 4, pp.448-452.

Yang, N., Ma, X. & Li, P. (2009). An Improved Angle-Based Crossover Tabu Search for the Larger-Scale Traveling Salesman Problem. gcis. WRI Global Congress on Intelligent Systems, IEEE. vol. 1, pp.584-587.

Yu, Y., Chen, Y. & Li, T. (2011). A New Design of Genetic Algorithm for Solving TSP. cso. Fourth International Joint Conference on Computational Sciences and Optimization. IEEE. pp. 309-313

Zhang, Ch., Sun, J., Wang, Y. & Yang, Q. (2007) An Improved Discrete Particle Swarm Optimization Algorithm for TSP. wi-iatw. IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Workshops. IEEE. pp. 35-38

Correspondencia:
Hugo Euler Tito Chura: eulertito@hotmail.com

Fecha de Recepción: 08/10/2015
Fecha de Aceptación: 11/12/2015