"Un Acercamiento a los Algoritmos AS y ACO resolviendo el TSP."

Juan Adolfo Montesino Guerra*, Juan Martin Carpio Valadez, Héctor José Puga Soberanes, Lucero de Montserrat Ortiz-Aguilar.
Tecnologico Nacional de México, Instituto Tecnológico de León, Departamento de Investigación y Estudios de Posgrado, León, Guanajuato, México.
*m08240762@itleon.edu.mx

Recibido: 28 de octubre, 2017
Aceptado: 10 de diciembre, 2017

RESUMEN

Este trabajo explora los metodos de busqueda llamados "Sistema de Hormigas" y "Optimización por Colonia de Hormigas" (AS y ACO, por sus siglas en inglés respectivamente) con los que encontramos soluciones al Problema del Vendedor Viajero (TSP, por su siglas en inglés). El ACO presenta resultados estadisticamente significativos por encima de los obtenidos por AS para resolver el problema de TSP. Las pruebas realizadas se hicieron con bases de datos encontradas en estado del arte obtenidas de TSPlib, y se efectuaron usando las llamadas a función como condición de paro, ademas de usar configuraciones en los algoritmos que se mencionan a detalle en este trabajo. En la experimentación se evaluaron 9 instancias del problema TSP en los cuales el ACO obtuvo mejores resultados en todas ellas. En los resultados producto de dichas experimentaciones, se comenta que la superioridad del ACO sobre el AS es dada por el conocimiento global que se comparten las "hormigas" dando así el significado de colonia a esta aproximacion de busqueda.

Palabras claves: Metaheuristicas, ACO, AS, TSP, TSPlib.

ABSTRACT

This paper explores the search methods called "Ant System" and "Ant Colony Optimization" (AS and ACO respectively) with which we find solutions to the Travel Salesman Problem (TSP). The ACO presents statistically significant results over those obtained by AS to solve the TSP problem. The tests were done with state-of-the-art databases obtained from TSPlib, and were performed using function calls as a stop condition, in addition to using configurations in the algorithms that are mentioned in detail in this work. In the experimentation, 9 instances of the TSP problem were evaluated in which the ACO obtained better results in all of them. In the results of these experiments, it is commented that the superiority of the ACO on the AS is given by the global knowledge that the ants share, thus giving the colony meaning to this search approximation.

Key words: Metaheuristics, ACO, AS, TSP, TSPlib.

1. INTRODUCCIÓN

El ingenio de adaptar comportamientos observados en la naturaleza para la busqueda de soluciones ha mostrado resultados alentadores, se han propuestos diversos tipos de algoritmos inspirados en la naturaleza que pueden ayudar a encontrar soluciones a problemas de optimizacion combinatoria. Los avances en las tecnologías de la información que se han dado en los últimos años han permitido proporcionar a los usuarios una manera más fácil y rápida de poder realizar cálculos que tomarían varios días o semanas si se llevarán a cabo de forma manual, además de reducir los errores que se pueden cometer al realizar dichas operaciones de forma manual. Actualmente las empresas buscan optimizar sus procesos, procedimientos o estrategias y uno de ellos es la elección de una ruta de distribución o recorrido que sea el de menor distancia posible, en el cual se busca tener cubierta su red de clientes, centros de ventas, etc. Las empresas necesitan minimizar el costo de recorrido de distancias, ya que se ve reflejado en tiempo y por supuesto de manera económica dado el mantenimiento del medio que se use para hacer dicho recorrido, esto es parte critica para el funcionamiento de la misma. Dadas las características de este tipo de problema, se relaciona con el Traveling salesman problema (TSP) o el problema del agente viajero (Cook, et.a., 2012).

El Problema TSP, responde a la siguiente pregunta: Dada una lista de ciudades y las distancias entre cada par de ellas, ¿cuál es la ruta más corta posible que visita cada ciudad exactamente una vez y regresa a la ciudad de origen? La "forma de visitar todas las ciudades" es simplemente el orden en el que se visitan las ciudades; el orden se llama un recorrido o circuito a través de las ciudades. Este es un problema NP-duro dentro en la optimización combinatoria, muy importante en la investigación de operaciones y en la ciencia de la computación (Cook, et.al., 2012) (L. S., et.al., 1973) (Rinnooy. et.al., 1985).

En la literatura existe un gran número de aproximaciones para resolverlo tanto con técnicas exactas como con técnicas Heurísticas y Metaheurísticas. Dentro de los métodos exactos se incluyen los planos de corte, la relajación de problemas lineales (Dantzig et.al., 1954), ramificación y acotamiento (Padberg et.al., 1987) ramificación y corte (Crowder et.al., 1980), (P. M. W. et.al., 1980), (Martin et.al., 1991) y programación dinámica (Ernest et.al., 1961), (Ergun et.al., 2006), eficientes cuando el problema es de tamaño pequeño. Los métodos exactos están diseñados de tal manera que se garantiza que van a encontrar la solución óptima en una cantidad finita de tiempo. Sin embargo, para problemas de optimización de gran tamaño (por ejemplo, problemas de tipo de complejidad NP-duros o de optimización global) esta cantidad finita de tiempo puede aumentar de forma exponencial con respecto a las dimensiones del problema y volverse intratable (no puede ser resuelto al optimo en tiempo polinomial o toma mucho tiempo para resolver). La técnica heurística es simplemente usar el conocimiento sobre el problema para acelerar la solución. Las técnicas heurísticas pueden resolver de forma muy eficiente un recorrido de TSP como el algoritmo de Dijkstra (D. E. W, 1959) esto proporciona una solución muy rápida (convergencia muy rápida), sin embargo, puede fácilmente quedar atrapado en mínimos locales. En este caso al ser un algoritmo de tipo "greedy" al igual que los métodos exactos, al incrementar el número de nodos, el tiempo de cómputo y su procesamiento se vuelve intratable, por este método.

Muchos algoritmos heurísticos son de diseño muy específicos y dependientes del problema y por lo tanto no se pueden utilizar fácilmente para resolver otros problemas (fueron pensadas/diseñadas para un problema específico). Por otro lado, una metaheurística es un método algorítmico de alto nivel con independencia del problema que ofrece una serie de pautas o estrategias para desarrollar algoritmos de optimización heurística. Así, el término "metaheurística" también se utiliza para referirse a una implementación específica a un problema de un algoritmo de optimización heurística de acuerdo con las directrices expresadas en dicho marco, los algoritmos metaheurísticos suelen encontrar soluciones buenas "en una cantidad razonable" de tiempo por lo general mientras intentan evitar quedar atrapadas en mínimos locales. En el momento de abordar los problemas de gran tamaño han sido utilizados, métodos probabilísticos como los tratados en (Rinnooy. et.al., 1985b), (Johnson. et al., 1987), (Croes. et al., 1958), cadenas de Markov (Martin. et al., 1991), metaheurísticas como Búsqueda Tabú (Johnson. et al., 1987), (Fiechter. et al., 1990), Redes Neuronales (Jerzey. et al., 1999), Recocido Simulado y Algoritmos Genéticos (Kirkpatrick. et al., 1983), (Bonomi. et al., 1984), (Kirkpatrick. et al., 1985), (Golden. et al., 1986), (Lam. et al., 1988), (Pardalos. et al., 1995) y (Lo. et al., 1998).

Es importante mencionar que en este trabajo se busca identificar aspectos importantes de los enfoques metaheurísticos de AS y ACO, que si bien se han utilizado dichos enfoques extensamente en la actualidad, pueden pasar inadvertidos y pueden ser aun temas para debate y/o investigación dentro del enfoque. Algunos de estos aspectos son la naturaleza sinérgica implícita en la interacción de los diferentes esquemas que conforman a una Metaheurística, así como el rendimiento de la función de aptitud.

2. MARCO TEÓRICO

2.1 Problema Del Agente Viajero

Este problema es de compleja formulación matemática, la propuesta por Dantzig, Fulkerson y Johnson (Fulkerson. et al., 1954) es una de las más aceptadas por la comunidad académica y consiste en:

Dado un conjunto de n ciudades $V = \{1,2,3, \dots n\}$, y un conjunto de arcos $A = \{(i,j) | i,j \in V\}$ uniendo cada una de las ciudades, donde A es el espacio de búsqueda. Si C_{ij} es la distancia para ir de la ciudad i a la ciudad j donde $(C_{ij} = C_{ji})$ en el caso simétrico y X_{ij} la variable de decisión del problema, la formulación es:

$$x_{ij} = \begin{cases} 1, \textit{Si el arco } (i,j) \textit{es usado para hacer el tour} \\ 0, \textit{En caso contrario} \end{cases}$$

El modelo matemático asume la siguiente forma:

$$Min\ Z(x) = \sum_{(i,j)\,\in\,A} c_{ij}x_{ij} \tag{1}$$

$$\sum_{\{i\,:\,(i,j)\,\in\,A\}} x_{ij} = 1 \;\; \forall\, j \;\in V \tag{2}$$

$$\sum_{\{j\,:\,(i,j)\,\in\,A\}} x_{ij} = 1 \;\; \forall\, i \in V \tag{3}$$

$$\sum_{\{(i,j)\,\in\,A\,:\,i\,\in\,U,j\,\in\,(V-U)\,\}} x_{ij} = 1 \quad 2 \leq |U| \leq |V| - 2 \tag{4}$$

Donde $\boldsymbol{A}$ es el espacio de búsqueda, $\boldsymbol{C_{ij}}$ es el peso de la arista o la distancia asociada a $\boldsymbol{X_{ij}}$, la ecuación (1) corresponde al cálculo de la función objetivo. La restricción (2) indica que se puede llegar a cada ciudad desde una única ciudad anterior. La restricción (3) indica que desde la ciudad *i* se puede pasar a una única ciudad (de la ciudad *i* se puede salir por un único camino). La restricción (4) evita que se generen subtours.

Aparentemente es un problema sencillo de resolver, pero si se considera un ejemplo con n=12 ciudades. El número total de ciclos/rutas donde dado este grafo, cada nodo es una ciudad y las aristas son la distancia que las conecta entre si, y se recorren todos los nodos sin pasar dos veces por uno mismo, se tendrian 12! = 479.001.600 ciclos diferentes. Si se resuelven cinco ciclos hamiltonianos en 1 segundo, se necesitarían 95.800.320 segundos o 1.108.8 días más de 3 años para encontrar la solución óptima.

El problema tiene considerables aplicaciones prácticas, aparte de las más evidentes en áreas de logística de transporte, que cualquier negocio de reparto, pequeño o grande, conoce. Por ejemplo, en robótica, permite resolver problemas de fabricación para minimizar el número de desplazamientos al realizar una serie de perforaciones en una plancha o en un circuito impreso. También puede ser utilizado en ruteamiento de vehículos, manufactura flexible, etc.

2.2 Sistema de hormigas (Ant System)

El algoritmo AS fue introducido por Dorigo en 1992 (Dorigo. et al., 2004). Es un algoritmo de búsqueda inspirado por el comportamiento de hormigas reales. Donde cada hormiga construye una ruta desde el hormiguero hasta la comida siguiendo estocasticamente (aleatorio) cantidades de niveles de feromonas, la intensidad de las feromonas situadas sesgarán la trayectoria de hormigas sucesoras.

El algoritmo de Hormigas fue desarrollado como una heurística de proposito general que puede ser usado para solucionar diferentes problemas de optimizacion combinatoria. Esta heurística tiene las siguientes caracteristicas (Dorigo. et al., 1996):

- Versatil. Puede ser aplicada a diferentes versiones del mismo problema.
- Robusta. Puede ser aplicada, con cambios mínimos, a otros problemas de optimizacion combinatoria.
- Enfoque basado en poblaciones. Permite la explotación con retro-alimentación positiva como un mecanismo de búsqueda.

El algoritmo ACO esta inspirado en el comportamiento real de las hormigas. Estos insectos son capaces de encontrar la ruta mas corta entre su colonia y una fuente de alimento. Esto se debe a que las hormigas pueden “transmitir” información entre ellas, gracias a un rastro de feromona que cada una de ellas deja al desplazarse. Cuando una hormiga descubre una fuente de alimento retorna a la colonia, siguiendo el rastro de feromona y reforzando el deposito de esta. La concentración de mayor de esta sustancia en este camino atrae a otras hormigas de la colonia, las cuales en su recolección de alimento siguen el mismo camino y refuerzan la feromona sobre este. Si existen varios caminos hacia la fuente de alimento las hormigas seleccionan el camino que va a ser recorrido basándose en la concentración de feromona sobre los caminos existentes (Dorigo. et al., 1996).

Las hormigas que recorren caminos mas cortas hacia las fuentes de alimento, regresan mas rápido que aquellas que seleccionaron caminos mas largos. De este modo el camino mas corto atraerá poseerá una mayor concentración de feromona, atrayendo a un numero mayor de hormigas (Dorigo. et al., 1996).

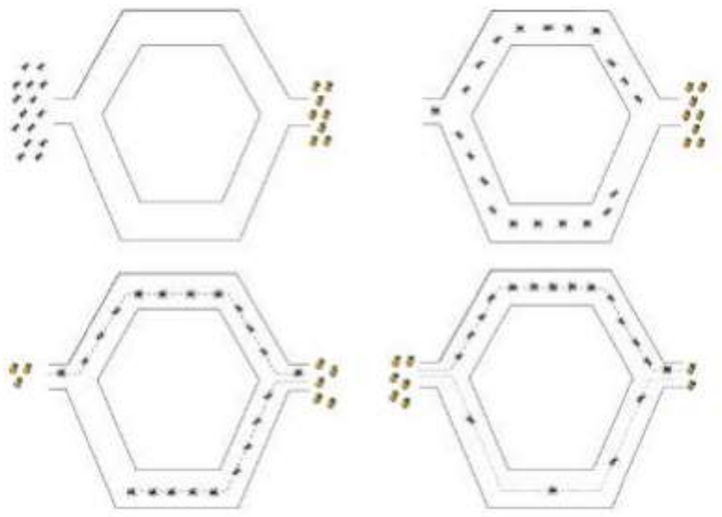

Figura 1. El comportamiento de la colonia termina por obtener el camino más corto entre dos puntos. Tomada de (Cordon. et al., 2001).

2.2.1 AS aplicado a TSP

El funcionamiento del AS con respecto del TSP se muestra a continuación (Xuan. et al., 2007). Dado un conjunto finito de ciudades y la distancia entre cada par de ciudades, el TSP tiene como objetivo encontrar una ruta a través de todas las ciudades visitandolas exactamente una sola vez y regresar a la ciudad inicial (ciclo de Hamilton) tratando que la distancia total recorrida sea la mínima. Se asume que las m hormigas artificiales viajan por todas las n ciudades. Las operaciones de AS para el TSP esta dado por:

Paso 1: Se selecciona aleatoriamente la ciudad de inicia para cada hormiga. El nivel de feromonas inicial entre cualesquiera dos ciudades se ajusta para que sea una constante pequeña positiva entre [0,1]. Se ajusta el contador de ciclos a cero.

Paso 2: Se calcula la probabilidad de transición de la ciudad r a la ciudad s para la k-ésima hormiga como:

$$P_k(r,s) = \left\{ \frac{[\tau(r,s)]^{\alpha} * [\eta(r,s)]^{\beta}}{\sum_{s \in J_k(r)} [\tau(r,s)]^{\alpha} * [\eta(r,s)]^{\beta}} \right\} \quad (5)$$

donde r es la ciudad actual, s es la ciudad siguiente, $\tau(r,s)$ es el nivel de feromona entre la ciudad r y s, $\eta(r,s) = \frac{1}{\delta(r,s)}$ la inversa de la distancia $\delta(r,s)$ entre la ciudad r y s. $J_k(r)$ es el conjunto de ciudades que quedan por recibir la visita de la hormiga k-ésima situada en la ciudad r, α es el parámetro de importancia de la feromona y β es el parámetro que determina la importancia la distancia. Se selecciona la siguiente ciudad s a visitar por la k-ésima hormiga con la probabilidad de $P_k(r,s)$. Se repite el paso 2 hasta que cada hormiga haya recorrido todas las ciudades.

Paso 3: Se actualizan los niveles de feromonas entre las ciudades:

$$\tau(r,s) \leftarrow (1-\rho) * \tau(r,s) \quad + \quad \sum_{k=1}^{m} \Delta\, \tau_k(r,s) \quad (6)$$

Donde:
$\rho =$ Coeficiente de evaporación
$m =$ total de hormigas

$$\vartriangle \tau_k(r,s) = \begin{cases} \frac{1}{L_k} \\ 0 \end{cases} \tag{7}$$

$\vartriangle \tau_k(r,s)$ es el nivel de feromona establecida entre las ciudades r y s por la k-ésima hormiga, L_k es la longitud recorrida por cada k-ésima hormiga, y $0 < \rho < 1$.

Paso 4: Incrementar el contador del ciclo. Mover las hormigas a la selección original y continuar el paso 2 hasta el paso 4 hasta que el número máximo de ciclos se ha alcanzado, donde esto nos indica que las hormigas tomaron el mismo camino.

En el ACO para el cálculo de la probabilidad de las ciudades se usó (5) en el caso de exploración y (8) en el caso de explotación. Para la actualización de la tabla de feromona local se usó la función (9) y (10) para la actualización de la feromona global.

$$s = \begin{cases} argmax_{u \epsilon j_k}\{[\tau(r,s)][\eta[(r,s)]^{\beta}]\} \quad si \quad q \leq q_0 \quad explotacion \\ S \quad exploracion \end{cases} \tag{8}$$

Donde:
q es un número aleatorio entre [0,1]
q_0 es un parámetro entre [0,1].

$$\tau(r,s) \leftarrow (1-\rho)\tau(r,s) + \rho\tau_0 \tag{9}$$

Donde:
τ_0 es el valor inicial de la feromona.

$$\tau(r,s) \leftarrow (1-\rho)\tau(r,s) + \rho \vartriangle \tau_k(r,s) \tag{10}$$

2.2.2 Algoritmos

El algoritmo AS se muestra en (Algoritmo 1) y el ACO en (Algoritmo 2), los cuales fueron implementados para el desarrollo de los experimentos de este documento.

Algoritmo 1 Algoritmo de Sistema de Hormigas

Inicializar parámetros
mientras condición **hacer**
  mientras estado ≠ estadoFin **hacer**
    para Cada - Arco - Posible - Movimiento **hacer**
      Calcular - Probabilidad - Elección
    fin para
    Siguiente - Posición = Política - Decisión
    Lista - Posiciones = +Siguiente - Posición
  fin mientras
  Realizar - Evaporación
  Deposito - Feromona(Lista - Posiciones)
fin mientras

3. METODOLOGIA

A través del algoritmo AS y ACO, se realizaron pruebas con 9 instancias del problema TSP simétrico (KroA100, KroA150, KroA200, KroB100, KroB150, KroB200, KroC100, KroD100 y KroE100) obtenidas de Ruprecht Karls Universitat Heidelberg (Reinelt. et al., 1991), en las mismas el numero que aparece en el nombre de cada una representa el numero de puntos (ciudades) en las que hay que encontrar la ruta mas corta entre ellas; Todas estas instancias fueron evaluadas con cada uno de los algoritmos. Cada algoritmo fue ejecutado 33 veces con

100,000 llamadas a función y en cada ejecución se obtuvo el $fitness$ de la mejor hormiga, de dicho conjunto de resultados se obtuvo la mediana como representante estadístico. Se crearon 20 hormigas para cada uno de los algoritmos.

Los parámetros de entrada para el AS fueron, obtenidos de (Dorigo. et al., 2004):

- τ = 0.3, α = 0.7, β = 0.7, ρ = 0.8

Los parámetros de entrada para el ACO fueron, obtenidos de (Dorigo. et al., 2004):

- τ = 0.3, α = 0.7, β = 0.7, ρ = 0.8, q_0 = 0.9

Algoritmo 2 Algoritmo de Colonia de Hormigas

```
Inicializar parámetros
Inicializar Feromona arcos
mientras no se cumple criterio de paro hacer
    mientras estado ≠ estadoFin hacer
        si esultimaposicionenlistatabu entonces
            Insertar ciudad origen
        si no
            para cada hormiga hacer
                Elegir próxima ciudad
                Mover hormiga
                Insertar ciudad en tabú
            fin para
        fin si
        Actualizar Feromona local
    Fin mientras
    para cada hormiga hacer
        Ver longitud de ruta
        Actualizar feromona global
    fin para
fin mientras
```

4. RESULTADOS.

En la **Tabla 1** se muestra la mediana de los $fitness$ de cada algoritmo en cada una de las instancias de prueba, además del resultado óptimo de cada instancia conocido en el estado del arte (Reinelt. et al., 1991) con fines de comparación de rendimiento.

Tabla 1. Medianas de los mejores Fitness de cada conjunto de experimentos.

Instancia TSP	AS	ACO	Optimo conocido
kroA100	36150	26393	21282
kroA150	47757	35868	26524
kroA200	56791	42633	29368
kroB100	37068	26554	22141
kroB150	46938	34953	26130
kroB200	56763	42371	29437
kroC100	35543	25005	20749
kroD100	35338	25593	21294
kroE100	37160	26618	22068

Para comparar dichos resultados de manera estadística se utilizó la prueba de suma de rangos con signo de Wilcoxon y verificar si los resultados del ACO con los resultados del AS provienen de poblaciones con medianas iguales. Los resultados se muestran en la **Tabla 2**.

Al Aplicar la prueba de suma de rangos con signo de Wilcoxon (ver **Tabla 2**), bajo las hipótesis:

$h_0 =$ No existen diferencias en las medianas vs. $h_a =$ Existen diferencias entre las medianas.

Donde se tomó un valor de significancia $\alpha = 0.05$ y 9 grados de libertad, encontramos que existe evidencia suficiente para rechazar h_0 . Por lo tanto, tenemos que efectivamente los datos provienen de poblaciones con medianas diferentes, es decir, el algoritmo ACO mejora los resultados obtenidos por el AS.

Tabla 2. Prueba de suma de rangos con signo de Wilcoxon entre ACO y resultados del AS

Instancia	ACO	AS	Jerarquía	Signo
1	26393	36150	2	-
2	35868	47757	6	-
3	42633	56791	8	-
4	26554	37068	3	-
5	34953	46938	4.5	-
6	42371	56763	5.5	-
7	25005	35543	4	-
8	25593	35338	1	-
9	26618	37160	5.5	-
$W^- =$	39.5	W =	0	
$W^+ =$	0	$W_0 =$	6	

5. DISCUSION

Los algoritmos de AS y ACO implementados en los experimentos tienen un costo computacional (número de operaciones) elevado, ya que tienen embebido un mecanismo de construcción de la propuesta de solución guiado con información del problema (las feromonas) y debido a ello logran obtener mejores resultados que en su mayoría de los algoritmos de la familia de los evolutivos. La clave para la aplicación de ACO a un problema es identificar una representación apropiada para el problema (a ser representado como un grafo recorrido por muchas hormigas artificiales) y una adecuada heurística que defina la distancia entre cualquier par de nodos del grafo. Entonces la interacción probabilística entre las hormigas artificiales medidas por el rastro de feromona depositada en los bordes del grafo generará buenas soluciones al problema.

El algoritmo colonia de hormigas ACO fue el algoritmo que obtuvo rutas más cortas en las diferentes instancias del problema. Sin duda la explotación ayuda en la obtención de mejores resultados, además de que se ve beneficiado por la información encontrada por otras hormigas en la actualización global de feromonas. Esto evidencia que el trabajo en grupos reporta mejores hallazgos que el trabajo individual por muy bien elaborado que este sea efectuado. Se muestra que el ACO es también una muy buena heurística constructiva para proporcionar soluciones de iniciales para problemas de optimización local.

6. CONCLUSIONES Y/O PROYECTOS FUTUROS.

El ACO obtiene resultados considerablemente cercanos a los óptimos conocidos, con bastante diferencia con respecto al AS en todas las instancias presentadas. Dada la evidencia estadística estos resultados pueden ser obtenidos con la frecuencia y repetitividad esperada, dando el sustento necesario con respecto a análisis de investigaciones anteriores. Conservando estas configuraciones experimentales para este algoritmo se evidencia un buen rendimiento en instancias de este tipo de problema. Debido a que el ACO fue concebido como un algoritmo hecho con el propósito de resolver el TSP, bajo restricciones específicas del problema, es posible pensar que es un método ad-hoc para resolver problemas del tipo TSP solamente. Sin embargo, recordando que el TSP es un problema que se puede modelar con una representación de grafos, bajo esta representación es posible utilizarlo en otros problemas con esta estructura de representación. En esta categoría se puede encontrar en el estado del arte problemas que pueden ser modelados como grafos, como el Coloreo de Grafos, Timetabling o Clique. Permitiendo que el ACO sea una herramienta robusta de importancia en la búsqueda de soluciones de problemas de búsqueda y optimización.

En investigaciones futuras con el enfoque de explicación para la profundidad del mismo, se buscará encontrar si esta Metaheurística es capaz de resolver con buen rendimiento diversos problemas que compartan el modelado de tipo grafo como el TSP.

7. AGRADECIMIENTO.

Agradecimientos al Tecnológico Nacional de México (TecNM), por el apoyo brindado a través del proyecto de investigación "Enfoque evolutivo para la optimización del diseño de particiones en la solución de problemas de Timetabling" Proyecto: xlfh6p(1559), al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo económico brindado en esta investigación por medio del apoyo en forma de beca con CVU: 595547 y a la División de Estudios de Posgrado e Investigación del Instituto Tecnológico de León.

8. REFERENCIAS BIBLIOGRAFICAS

COOK, W. J. (2012). In Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation. Princeton, NJ, USA: Princeton University Press.

LIN. S. AND K. B. W., (1973) "An efective heuristic algorithm for the traveling-salesman problem" Operations Research, vol. 21, no. 2, pp. 498-516.

LAWLER. E. L., L. J. K., K. A. H. G. Rinnooy, and S. D. B., eds., (1985) "The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization." New York: Wiley-Interscience.

DANTZIG G., Fulkerson R., and Johnson S., (1954) "Solution of a large-scale traveling salesman problem" Journal of the Operations Research Society of America, vol. 2, no. 4, pp. 393-410.

PADBERG M. AND RINALDI G., (1987) Optimization of a 532-city symmetric traveling salesman problem by branch and cut," Operations Research Letters, vol. 6, no. 1, pp. 1-7.

CROWDER H. AND PADBERG M. W., (1980) "Solving large-scale symmetric travelling salesman problems to optimality" Management Science, vol. 26, no. 5, pp. 495-509.

PADBERG. M. W. AND SAMAN H., (1980) "On the symmetric travelling salesman problem: A computational study" in Combinatorial Optimization (M. Padberg, ed.), 148 Bibliografía vol. 12 of Mathematical Programming Studies, pp. 78-107, Springer Berlin Heidelberg.

MARTIN G. AND OLAF H., (1991) "Solution of large-scale symmetric travelling salesman problems" Mathematical Programming, vol. 51, no. 1-3, pp. 141-202.

BELLMAN R. E., (1961) "Dynamic programming treatment of the traveling salesman problem" J. Assoc. Comput., pp. 1-3.

OZLEM E. AND ORLIN J. B., (2006) "A dynamic programming methodology in very large scale neighborhood search applied to the traveling salesman problem" Discrete Optimization, vol. 3, pp. 78-85.

DIJKSTRA. E. W., (1959) "A note on two problems in connexion with graphs" Numerische Mathematik, vol. 1, pp. 269-271, Dec.

JOHNSON. D., (1987) "More approaches to the travelling salesman guide" nat, vol. 330, p. 525, Dec.

CROES G. A., (1958) "A method for solving traveling-salesman problems" Operations Research, vol. 6, no. 6, pp. 791-812.

MARTIN O., Otto S. W., and Felten E. W., (1991) "Large-step markov chains for the traveling salesman problem" Complex Systems, vol. 5, pp. 299-326.

FIECHTER C. N., (1990) “A parallel Tabu search algorithm for large scale traveling salesman problems” Working Paper, no. 90/1.

JERZY M., (1999) “Neural network approach to design of distributed hard real-time systems” in Proceedings of the 6th International Conference on Computational Intelligence, Theory and Applications: Fuzzy Days, (London, UK, UK), pp. 118-131, Springer-Verlag.

KIRKPATRICK. S., Gelatt. C. D., and Vecchi. M. P., (1983) “Optimization by simulated annealing” Science (New York, N.Y.), vol. 220, pp. 671-680, May.

BONOMI E. AND LUTTON J. L., (1984) “The n-city travelling salesman problem: Statistical mechanics and the metropolis algorithm” SIAM, vol. 26(4), p. 551-568.

KIRKPATRICK S., (1985) “Configuration space analysis of the traveling salesman problem” J. de Physique, vol. 46 (8), pp. 1277-1292.

GOLDEN B. L. AND S. C. C., (1986) “Using simulated annealing to solve routing and location problems" Naval Research Logistics Quarterly, vol. 33, no. 2, pp. 261-279.

LAM J. AND MARC DELOSME J., (1988) “An eficient simulated annealing schedule: derivation” tech. rep., Department of Computer Science, Yale University.

PARDALOS P. M., Pitsoulis L., Mavridou T., and Resende M. G. C., (1995) “Parallel search for combinatorial optimization: Genetic algorithms, simulated annealing, tabu search and grasp" in ‘Proceedings Of The Second International Workshop On Parallel Algorithms For Irregularly Structured Problems, Irregular 95’, pp. 317-331, Springer-Verlag.

LO C. C. and Hsu C. C., (1998) “An annealing framework with learning memory” Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, vol. 28, pp. 648-661, Sep.

ALONSO S., Cordón O., Fernández I., Herrera F. (2001) “La Metaheurísticas de optimización basada en colonias de hormigas: modelos y nuevos enfoques.” Optimización inteligente: técnicas de inteligencia computacional para optimización, 261-314.

DANTZIG G., Fulkerson R., and Johnson S., (1954) “Solution of a large-scale traveling salesman problem” Journal of the Operations Research Society of America, vol. 2, no. 4, pp. 393-410.

Dorigo, M.; THOMAS STUTZLE. (2004) “Ant Colony Optimization”. ISBN 0-262- 04219-3. Page 73.

DORIGO, M.; Maniezzo, V.; Colorni, A., (1996) “Ant system: optimization by a colony of cooperating agents” Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol.26, no.1, pp.29, 41, Feb.

XUAN F. Zha. (2007) “Artificial Intelligence and Integrated Intelligent Information Systems.” Energy Technologies and Applications. ISBN 1-59904-251-7 (ISBN). Page 19.

REINELT. G., (1991) “TSPLIB - A Traveling Salesman Problem Library” ORSA Journal on Computing, vol. 3.