

# Un Algoritmo ACS con Selección Dinámica de Movimiento y Operador 2-Opt

**Alejandro Pavez Salazar, Héctor Acevedo Almonacid**

Universidad Técnica Federico Santa María

Departamento de Informática

*{kirk, ric}@inf.utfsm.cl*

## Abstract in English

The Ant Colony Optimization (ACO) (Dorigo y Di Caro, 1999) metaheuristic is one of the new paradigms to allow combinational problems resolution of type NP. This article present the obtained results after applied an algorithm variant of the metaheuristic called Ant Colony System (ACS) (Dorigo y Stützle, 1999; Pavez *et al.*, 2000) complemented by one eliminatory crossing operator (2-Opt), working with the world wide knew Traveling Salesman Problem (TSP) (Reinelt, 1994). There are new solutions tours with better results until now obtained.

## Resumen en Castellano

La metaheurística Ant Colony Optimization (ACO) (Dorigo y Di Caro, 1999) es uno de los nuevos paradigmas que permiten la resolución de problemas combinatorios del tipo NP. En este artículo se presentan los resultados obtenidos al aplicar un algoritmo variante de la metaheurística ACO, llamado Ant Colony System (ACS) (Dorigo y Stützle, 1999; Pavez *et al.*, 2000) complementado con un operador de eliminación de cruces (2-Opt), en torno al mundialmente conocido Traveling Salesman Problem (TSP) (Reinelt, 1994). Se presentan nuevos tours solución con mejores resultados que los logrados hasta hoy y se discuten las oportunidades de desarrollo futuro.

**Keywords:** TSP, ACS, 2-Opt, Simulated Annealing, Metaheurística.

## 1. Introducción

La importancia del TSP dentro del mundo de los problemas NP radica en que es utilizado como conjunto de pruebas para los nuevos algoritmos asociados a la resolución de esta clase de problemas. Su naturaleza NP-Completo hace posible que se pueda resolver una gran familia de problemas equivalentes mediante transformaciones particulares para cada caso (Moscato *et al.*, 1996). Además, como el TSP es posible de utilizar en una gran cantidad de aplicaciones en el mundo real, se ha convertido en uno de los problemas más estudiados por la comunidad científica mundial. Las características de la nueva metaheurística ACO, aplicadas al TSP (Dorigo y

Gambardella, 1997a, b; Dorigo y Stützle, 1999), pueden ayudar a encontrar soluciones satisfactorias para este problema.

En una primera parte, como punto de partida se presentará el problema del TSP, para luego presentar la analogía de la Colonia de Hormigas y su representación a través de Hormigas Artificiales. En una segunda parte, en primer término, se presenta la modificación propuesta para seguir con características del Hardware y el Software utilizado. A continuación se presenta la descripción de las pruebas realizadas y los resultados obtenidos. Una última parte resume las conclusiones y algunas oportunidades de desarrollo futuro.

### 1.1 El problema del vendedor viajero<sup>1</sup>

En una apretada síntesis, el TSP (Reinelt, 1994) consiste en que una persona debe visitar un conjunto de  $n$  ciudades, comenzando en una ciudad específica (elegida arbitrariamente como origen), debiendo regresar a ella luego de haber visitado todas las demás sin haber visitado dos veces una misma ciudad. El objetivo es minimizar la longitud de la secuencia de visita a las ciudades, bajo el criterio de minimización (maximización) de la función objetivo. Un ejemplo de un tour es el que se muestra a continuación en el figura 1.

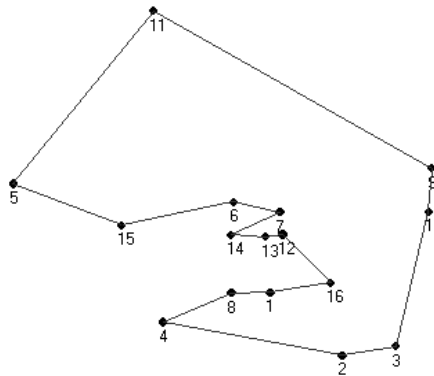


Fig. 1 - Un ejemplo de tour TSP Euclidiano (Ulisses16)

El problema puede ser representado a través de una matriz  $n \times n$  (simétrica ó asimétrica) de distancias  $d$  donde  $d_{ij}$  es la distancia entre la ciudad  $i$  y la ciudad  $j$ . El valor de  $d_{ii}$  es siempre 0. Si se tiene  $n$  ciudades, existen a lo más  $(n-1)!$  circuitos diferentes (y  $(n-1)! / 2$  en el caso del TSP simétrico).

El problema es del tipo NP-Completo, es decir, es un problema representativo de la clase NP cuyo tiempo de resolución crece en forma exponencial ante incrementos lineales del número de elementos que intervienen en el problema. En otras palabras, no se ha encontrado un algoritmo que lo resuelva en tiempo polinomial.

Una representación del TSP asimétrico desde el punto de vista de la 'Programación Lineal' es la siguiente:

<sup>1</sup> En ingles TSP, acrónimo de Traveling Salesman Problem

$$\begin{aligned}
& \min \sum_i \sum_{i \neq j} d_{ij} x_{ij} \\
& \text{talque } \sum_i x_{ij} = 1 \quad \forall j \\
& \sum_j x_{ij} = 1 \quad \forall i \\
& \sum_{i \in S} \sum_{j \notin S} x_{ij} \geq 1 \quad \forall \text{ subconjunto apropiado } S, |S| \geq 2 \\
& \text{donde } x_{ij} = \begin{cases} 1 & \text{si el tour pasa desde } i \text{ a } j \\ 0 & \text{sino} \end{cases} \\
& \text{y } d_{ij} \text{ es la distancia desde } i \text{ a } j
\end{aligned}$$

En este trabajo se hará uso de problemas TSP simétricos, es decir:  $d_{ij} = d_{ji}$ .

## 1.2 La analogía de la colonia de hormigas

La Metaheurística ACO (Dorigo y Di Caro, 1999) se basa en el comportamiento de una colonia de Hormigas (Dorigo *et al.*, 1996). Las hormigas reales son capaces de encontrar el camino más corto entre una fuente de comida y su nido - por ejemplo - sin usar mecanismos visuales, sino sólo explotando el rastro de Feromona<sup>2</sup>.

Una forma en que las hormigas explotan la feromona para encontrar el camino más corto entre dos puntos es el que se muestra en la figura 2.

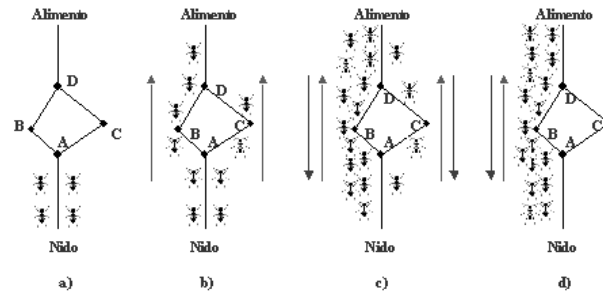


Fig. 2 – Las hormigas y el camino más corto

En a) las hormigas llegan a un punto en el cual deben decidirse por uno de los dos caminos a seguir. En b) se realiza una elección de camino, la que puede ser aleatoria, al haber bajos niveles de feromona, o guiada, al haber una diferencia notable entre la cantidad de feromona depositada en cada camino. En c), dado que la velocidad de una hormiga se considera aproximadamente constante, es claro sostener que las hormigas que eligieron el camino más corto se demorarán menos que las otras en llegar hasta el otro extremo, lo cual genera una mayor acumulación de feromona en el camino más circulado. En d), la feromona acumulada en mayor cantidad en el

<sup>2</sup> Feromona (Pheromone), sustancia olorosa producida por un animal que afecta a la conducta de otros animales. Para este trabajo sólo se considerará el uso de la feromona como medio de compartir información relevante entre los agente, reconociendo que el comportamiento de las Hormigas reales posee una mayor complejidad y un mayor número de características dignas de estudio.

camino más corto y más circulado guía a las hormigas a la fuente de alimento de la forma más rápida.

Esta metaheurística trabaja de la siguiente manera:

“Cada hormiga genera un tour completo escogiendo un destino (ciudad) de acuerdo a una regla de transición de estados probabilística. Las hormigas prefieren moverse a ciudades que están conectadas a caminos de bajo costo con una alta cantidad de feromona. Una vez que todas las hormigas han completado su tour, se aplica una regla de actualización de la feromona global. A continuación se evapora una fracción de la feromona, mientras que por otra parte algunas hormigas depositan una cantidad proporcional de ella sobre los caminos que formaron parte de su tour (de acuerdo al costo total de cada uno), lo cual provoca un incremento de feromona a través del tiempo. Luego, el proceso comienza a iterar. A través de este mecanismo, la colonia genera una convergencia común, dirigida hacia el camino más corto.”

Cabe señalar que la actualización de feromona tiene como propósito la coordinación entre los distintos tours realizados en forma concurrente por cada una de las hormigas. Ello lleva a entender dicha coordinación como una manera de compartir información de calidad, la cual sirve, más que al objetivo particular de cada hormiga, al objetivo global de la colonia: encontrar el tour más corto.

### 1.3 Las hormigas artificiales

De acuerdo al comportamiento descrito anteriormente, es posible definir el concepto de Hormiga Artificial. Un ejemplo de pseudocódigo para una implementación de la metaheurística ACO, se presenta a continuación:

```
procedimiento ACO()
  establecer_feromona_inicial();
  mientras (el criterio de término no esté satisfecho)
    crear_las_hormigas();
    para (cada hormiga)
      {mover_hormiga();} hasta (completar tour);
    fin para
    actualizar_feromona();
    destruir_hormigas();
  fin mientras
fin procedimiento
```

```
procedimiento mover_hormiga()
  para (todo el vecindario factible)
    calcular_probabilidades_de_movimiento();
  fin para
  ciudad_seleccionada:=seleccionar_el_movimiento();
  si (RAND  $\geq$  probabilidad(ciudad_seleccionada))
```

```

    ciudad_seleccionada:=RAND(vecindario_factible);
fin si
    llevar_hormiga_a_ciudad(ciudad_seleccionada);
fin procedimiento

```

Para este estudio se utilizó una implementación del algoritmo Ant Colony System (ACS) (Dorigo y Stützle, 1999) con la variante de selección dinámica de movimiento presentada en (Pavez *et al.*, 2000). En él, cada una de las  $n$  hormigas construye una solución (tour) del TSP. Inicialmente, cada una de las hormigas es colocada en alguna ciudad (en forma aleatoria o secuencial). Una vez hecho esto, a cada hormiga se le aplica una regla de elección probabilística de los caminos a seguir. En particular, la probabilidad con que la hormiga  $k$ -ésima (actualmente en la ciudad  $i$ ) decide ir a la ciudad  $j$  en la  $t$ -ésima iteración es:

$$p_{ij}^k(t) = \frac{[t_{ij}(t)]^a \cdot [h_{ij}]^b}{\sum_{l \in N_i^k} [t_{il}(t)]^a \cdot [h_{il}]^b} \quad \text{si } j \in N_i^k \quad (1)$$

donde  $h_{ij} = 1/d_{ij}$  es un valor disponible a priori definido por la distancia  $d_{ij}$ ,  $\alpha$  y  $\beta$  son dos parámetros los cuales determinan la influencia relativa del camino de feromona y la distancia respectiva, y  $N_i^k$  es el vecindario factible para la hormiga  $k$ , esto es, el conjunto de ciudades a las cuales la hormiga  $k$  aún no ha visitado. El rol de los parámetros  $\alpha$  y  $\beta$  es el siguiente. Si  $\alpha=0$ , las ciudades cercanas tienen más probabilidad de ser seleccionadas: esto corresponde a un algoritmo clásico voraz estocástico<sup>3</sup> (Moscato *et al.*, 1996) (con múltiples puntos de inicio desde que las hormigas están distribuidas en las ciudades). Si  $\beta=0$ , sólo trabajará la amplificación de feromona: este método podría dirigirnos a una rápida situación de estancamiento con la correspondiente generación de tours los cuales, en general, son fuertemente subóptimos. La búsqueda de estancamiento se define como la situación donde todas las hormigas siguen la misma ruta y construyen un óptimo local. Por lo tanto, debe existir una mezcla entre la influencia de la información heurística y los caminos de feromona.

La variante de selección dinámica de movimiento consiste en la aplicación del concepto de temperatura - definido en los algoritmos de tipo *simulated annealing* (Moscato *et al.*, 1996) - dentro de la cadena de decisión del próximo movimiento. Una vez calculadas las probabilidades tradicionales de movimiento, señaladas anteriormente, cada Hormiga tiene la opción aleatoria de moverse bajo el criterio de exploración temprana y explotación tardía. Esto se define como sigue:

$$Rand \geq p_{ij}^k(t) \Rightarrow j \equiv Rand(N_i^k) \quad (2)$$

donde se establece que si la probabilidad asociada a la ciudad  $j$  (Ver ecuación 1), seleccionada como el próximo movimiento, es inferior a un número aleatorio, la nueva ciudad destino será asignada en forma aleatoria desde el vecindario factible  $N_i^k$ .

---

<sup>3</sup> En ingles: *stochastic greedy*

Claramente, puede observarse que a medida que el algoritmo genere probabilidades mayores de movimiento, tendientes a uno, debido al mejor tour encontrado, la probabilidad de asignación de un nuevo movimiento aleatorio tenderá a cero (este es análogo al concepto de temperatura).

Después que todas las hormigas han construido su tour, los caminos de feromona son actualizados. Esto se hace primero disminuyendo el nivel de feromona en todos los arcos por un factor constante, para luego permitir a cada hormiga agregar feromona a los arcos que fueron visitados:

$$t_{ij}(t+1) = (1 - r) \cdot t_{ij}(t) + r \cdot \Delta t_{ij}^{gb}(t) \quad (3)$$

donde  $0 < r \leq 1$  es el factor de evaporación de la feromona. El parámetro  $r$  es utilizado para evitar la acumulación ilimitada de feromona en los caminos y le permite al algoritmo “olvidar” las malas decisiones previamente tomadas. Ello provoca que si un arco no es elegido por las hormigas, su feromona asociada decrezca exponencialmente.  $\Delta t_{ij}^{gb}(t)$  es la cantidad de feromona que la mejor hormiga global  $gb$  coloca en el arco visitado. Esto se define como sigue:

$$\Delta t_{ij}^{gb}(t) = \begin{cases} 1/L^{gb}(t) & \text{si arco}(i, j) \text{ es usado por la hormiga } gb \\ 0 & \text{en otro caso} \end{cases} \quad (4)$$

donde  $L^{gb}(t)$  es el largo del tour de la mejor hormiga global  $gb$ . Dado por la ecuación anterior, el mejor tour de una hormiga es el que más feromona recibió por los arcos a través del tour. En general, los arcos que son usados por muchas hormigas y los cuales están contenidos en tours cortos podrían recibir más feromona y por consiguiente son más probables a ser seleccionados en futuras iteraciones del algoritmo.

## 2. La modificación propuesta: El operador 2-Opt

El objetivo del operador es aplicar el criterio de eliminación de cruces (Kernighan y Lin, 1973) dentro del tour generado por las iteraciones del algoritmo ACS, con el objeto de obtener un tour de menor costo. Debido a que no es posible saber de antemano el número de cruces a eliminar, es necesario iterar hasta que no sea posible realizar una nueva reducción.

El operador funciona de la manera como se indica a continuación:

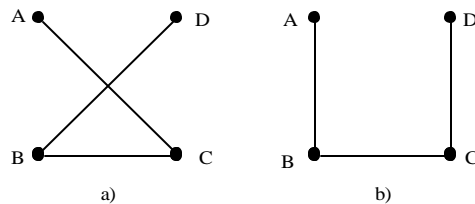


Fig. 1 - El operador 2-Opt

En la figura 3 se presenta el criterio de aplicación del operador 2-Opt. En ella se refleja una situación en la cual es necesario ir desde el punto A al punto D de forma tal hacerlo a través de los

puntos B y C. En a) se tiene que una situación con un cruce entre los caminos AC y BD, mientras que en b) la situación es la ideal, ya que la sucesión ABCD realiza el viaje desde A hasta D sin ningún cruce de caminos. Bajo este criterio de rompimiento de cruce, transformando un subtour caracterizado por la situación a) por la situación b) se presenta como la funcionalidad del operador propuesto.

Dado que se ha observado que, como resultado de las iteraciones del algoritmo ACS, el tour resultado puede contar con un par de cruces debido a errores en las aproximaciones numéricas tanto en el cálculo de costos como en el cálculo de las probabilidades de movimiento, esto debido a las limitaciones propias a la máquina de cómputo y el nivel exactitud numérica disponible, se ha visto necesario aplicar este operador como apoyo a la eliminación de los cruces existentes y la generación de una mejor solución.

### 3. La experimentación en la práctica

A continuación, en el punto 3.1 se presentará las características del hardware y software usado, en el punto 3.2 se presenta la descripción de pruebas realizadas, y en el punto 3.3 se presentan los resultados obtenidos.

#### 3.1 Características del hardware y software usado

La implementación y ejecución de este algoritmo fueron realizadas en un equipo de tipo Intel Pentium III 550MHz con sistema operativo Windows 98 versión 4.10 2222 A y 64MB RAM.

El software desarrollado permite, bajo una interfaz gráfica, establecer los parámetros asociados a los factores de evaporación  $\rho$ ,  $\alpha$ ,  $\beta$  y el archivo del problema a resolver.

#### 3.2 Descripción de las pruebas

Como esquema de trabajo se decidió realizar las pruebas del algoritmo implementado utilizando tours probados y generales. Los tours se obtuvieron de la biblioteca TSPLIB<sup>4</sup>, de los que se seleccionaron, los de tipo Euclidiano, es decir, aquellos en los cuales el costo asociado a realizar el viaje, desde la ciudad  $i$  hacia la ciudad  $j$ , se define como la distancia asociada al ciclo euclidiano entre las ciudades del tour.

El parámetro  $\alpha$  tomó los valores [0.5, 1, 1.5, 2, 2.5, 3, 3.5]. mientras que el parámetro  $\beta$  se mantuvo constante en el valor 2. Ello debido a que la ecuación de probabilidad de movimiento (Ver ecuación 1) normaliza los valores calculados a través de su término inferior. Además, las hormigas se distribuyeron tanto en forma aleatoria como secuencial. Los valores del factor de evaporación  $\rho$  fueron [0.3, 0.5, 0.7]. El número de hormigas considerado, en general, fue menor o igual al número de ciudades asociadas al problema. El número de iteraciones para cada corrida, no superó las 200 unidades. Los tours seleccionados fueron 'Ulisses16', 'Ulisses22', 'Eil51', 'Eil101' y 'Ch150'.

#### 3.3 Los resultados obtenidos

---

<sup>4</sup> <http://www.iwr.uni-heidelberg.de/iwr/comopt/software/TSPLIB95/>

Los resultados obtenidos utilizando los tours seleccionados se presentan en la siguiente tabla:

Tabla 1 Detalle de los resultados obtenidos.

TSP	Valor actual	Valor encontrado	Tour encontrado
Ulisses 16	74.108735958 153	74.096450656 9666 74.001336099 1673 73.999826312 7488 73.987618045 1750	1-8-4-2-3-16-13-12-6-7-10-9-11-5-15-14 1-8-4-2-3-16-12-13-14-6-7-10-9-11-5-15 1-8-4-2-3-16-13-12-7-6-10-9-11-5-15-14 1-3-2-4-8-15-5-11-9-10-7-6-14-13-12-16
Ulisses 22	75.665149471 3561	75.543191685 6301 75.454485295 7371 75.309701206 1126	1-14-13-12-7-6-15-5-11-9-10-19-21-20-16-3-2-17-18-4-22-8 1-14-13-12-7-6-15-5-11-9-10-19-21-20-16-3-2-17-4-18-22-8 1-14-13-12-7-6-15-5-11-9-10-19-20-21-16-3-2-17-4-18-22-8
Eil51	429.98331198 3384	429.11793919 9826	1-22-8-26-31-28-3-36-35-20-2-29-21-16-50-34-30-9-49-10-39-33-45-15-44-42-19-40-41-13-25-14-24-43-7-23-48-6-27-51-46-12-47-18-4-17-37-5-38-11-32
Eil101	642.30953579 0602	641.14363452 0988	1-69-27-101-53-28-26-12-80-68-29-24-54-55-25-4-39-67-23-56-75-41-22-74-72-73-21-40-58-13-94-95-97-87-2-57-15-43-42-14-44-38-86-16-61-85-91-100-98-37-92-59-93-99-96-6-89-52-18-83-60-5-84-17-45-8-46-47-36-49-64-63-90-32-10-62-11-19-48-82-7-88-31-70-30-20-66-65-71-35-34-78-81-9-51-33-79-3-77-76-50
Ch150	6532.2809331 4576	6530.9027218 5219	1-98-103-82-95-107-5-100-143-97-146-26-75-18-142-85-65-132-137-50-55-58-141-83-56-90-46-92-54-138-134-131-32-23-38-67-43-109-51-20-25-110-81-29-86-135-70-108-102-114-99-19-2-37-6-28-9-



			42-120-47-139-40-53-118-24- 12-116-101-41-57-39-127-69- 36-61-11-148-130-17-66-60- 140-117-129-27-31-123-74- 13-106-91-119-68-128-45-71- 44-64-112-136-145-144-147- 49-72-80-14-122-77-133-15- 78-21-150-115-4-104-22-125- 149-62-3-113-10-94-88-121- 79-59-16-111-105-33-126-52- 93-124-35-96-89-8-7-84-30- 63-48-73-76-34-87
--	--	--	---

donde **TSP** indica el nombre del tour, **Valor actual** indica el mejor valor registrado la fecha en la biblioteca TSPLIB, **Valor encontrado** indica el o los mejores valores encontrados a través de la implementación realizada para este estudio, **Tour encontrado** detalla el tour asociado al Valor encontrado.

#### 4. Conclusiones y trabajo futuro

La metaheurística ACO, a través de la variante ACS con selección dinámica de movimiento, ha demostrado ser una buena opción para la resolución de problemas de tipo combinatorios. La rápida convergencia, junto con la calidad de las soluciones generadas, la presentan como una opción efectiva para la resolución de problemas pequeños del tipo NP.

El operador propuesto se presenta como un opción para complementar la metaheurística ACO eliminando los ciclos que no son visibles a las hormigas artificiales. Además, como resultado se ha mejorado algunos de los resultados obtenidos con anterioridad (Pavez *et al.*, 2000), junto con proponer nuevos óptimos. Con respecto al comportamiento observado durante el desarrollo de las pruebas, se pudo apreciar un nivel notable de mejora al mejor tour encontrado en la gran mayoría de los resultados generados a través de las iteraciones realizadas por el algoritmo ACS.

Una de las posibles aplicaciones futuras consiste en la utilización de la implementación actual a pruebas, problemas y máquinas mucho más grandes, con lo cual se podrían generar resultados aún más satisfactorios.

Es posible percibir desde ya la complementación de lo logrado mediante aplicación de técnicas de paralelización, en donde las distintas instancias de los grupos de agentes podrán trabajar comunicando la mejor solución obtenida localmente, cada cierto número de intervalos. En el caso de la presente aplicación, el tener permanentemente la mejor solución global encontrada, podría facilitar el uso de una técnica de evaporación de feromona global utilizando aquella solución global.

Además, es necesario realizar un estudio asociado al control de parámetros (Michalewicz *et al.*, 1999), lo cual podría ayudar a determinar patrones o guías con respecto a ellos, incluso la autoparametrización, para la obtención de resultados más cercanos a los óptimos buscados.

Por otro lado, es posible desarrollar operadores 3-Opt o superiores que contemplen rotación de subciclos (Michalewicz y Tao, 1998), ya que dentro de las soluciones generadas por el algoritmo ACS existen subciclos que al ser rotados, en forma manual, presentan una mejora considerable.

### Referencias

- [1] Dorigo, M., V. Maniezzo y A. Coloni (1996), The Ant System: Optimization by a Colony of Cooperating Agents, *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, **Vol. 26** N°1, pp.29-41.
- [2] Dorigo, M., A. Coloni, F.Maffioli, V. Maniezzo, G. Righini y M. Trubian (1996), Heuristics from Nature for Hard Combinatorial Problems, *International Transactions in Operational Research*, **Vol. 3** N°1, pp. 1-21.
- [3] Dorigo, M. y L.M. Gambardella (1997a), Ant Colonies for the Traveling Salesman Problem, *BioSystems*, **N° 43**, pp.73-81.
- [4] Dorigo, M. y L.M. Gambardella (1997b), Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem, *IEEE Transactions on Evolutionary Computation*, **Vol. 1** N°1, pp. 53-66.
- [5] Dorigo, M. y G. Di Caro (1999), The Ant Colony Optimization Meta-Heuristic, *New Ideas in Optimization*, McGraw-Hill, UK.
- [6] Dorigo, M., G. Di Caro y L. M. Gambardella (1999), Ant Algorithms for Discrete Optimization. Artificial Life, *Tech.Rep.IRIDIA/98-10*, Université Libre de Bruxelles, Belgium.
- [7] Dorigo, M. y T. Stützle (1999). ACO Algorithms for the Traveling Salesman Problem, *Evolutionary Algorithms in Engineering and Computer Science: Recent Advances in Genetic Algorithms, Evolution Strategies, Evolutionary Programming, Genetic Programming and Industrial Applications*, John Wiley & Sons., EEUU.
- [8] Kernighan, B.W. y S. Lin (1973), An Effective Heuristic Algorithm for the Traveling Salesman Problem, *Operation Research*, **Nº. 21**, pp. 498-516.
- [9] Moscato, P., A. Días, F. Glover, H.M. Ghaziri, J. L. González, M. Laguna y F. T. Tseng (1996), *Optimización Heurística y Redes Neuronales*. Editorial Paraninfo, Argentina.
- [10] Michalewicz, Z. y G. Tao (1998), Inver-over Operator for the TSP, *Lectures Notes in Computer Science*, pp. 803-812.
- [11] Michalewicz, Z., A. E. Eiben y R. Hinterding (1999), Parameter Control on Evolutionary Algorithms, *IEEE Transactions on Evolutionary Computation*, **Vol 3** N°2, pp. 124-141.
- [12] Pavez, A., B. Arenas y R. Vidal (2000), Algoritmo ACO aplicado al TSP: Resumen de una experiencia práctica, *Workshop Advances & Trends in Artificial Intelligence for Problems Solving*, Jornadas Chilenas de Computación 2000, Universidad de Santiago de Chile, Chile.
- [13] Reinelt, G. (1994), *The Traveling Salesman. Computational Solutions for TSP Applications*, Springer-Verlag, Computer Science Editorial I, EEUU.