



VNiVERSiDAD
D SALAMANCA

Hormigas Artificiales

Sistemas Inteligentes

María Luisa Pérez-Delgado
Escuela Politécnica Superior de Zamora

7 / Noviembre / 2019



INTRODUCCIÓN

Los sistemas basados en colonias de hormigas constituyen una técnica heurística de solución de problemas surgida en los años 90 de la tesis doctoral de Marco Dorigo, [9], [6].

Esta técnica intenta imitar el comportamiento de las hormigas naturales para resolver problemas de optimización.

El primer algoritmo propuesto, denominada *Ant-System*, se aplicó inicialmente para resolver el TSP. El algoritmo se aplicó posteriormente a otros problemas de optimización, como el problema de Asignación Cuadrática, [11], [12]; problemas de Rutas de Vehículos, [2], [10]; o el problema de Coloreado de Grafos, [4].



ALGUNAS APLICACIONES DE LOS ALGORITMOS DE COLONIAS DE HORMIGAS

- Asignación Cuadrática
- Secuenciamiento de Tareas
- Coloreado de Grafos
- Rutas de Vehículos
- Ordenación Secuencial
- Problemas de Agrupamiento (Clustering)

COLONIAS DE HORMIGAS NATURALES (I/II)

Las hormigas son animales prácticamente ciegos, que se comunican entre sí mediante una sustancia química denominada **feromona**, que depositan en el suelo al desplazarse, [5].

Cuando una hormiga encuentra un camino hacia una fuente de alimento, deposita feromona en el trayecto. La cantidad de feromona depositada depende de la longitud del camino y de la calidad del alimento encontrado. La feromona se evapora con el tiempo, haciendo menos deseables los caminos que son elegidos por menos hormigas.

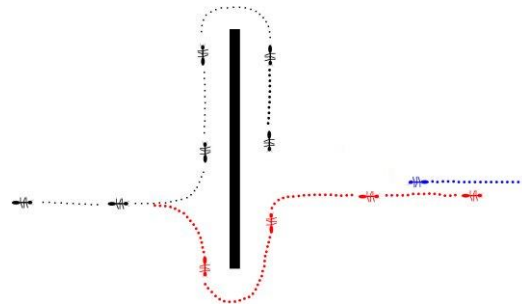


Figure 1: movimiento de las hormigas para sortear un obstáculo



COLONIAS DE HORMIGAS NATURALES (II/II)

Si una hormiga no detecta la presencia de feromona, se mueve aleatoriamente; pero si percibe dicha sustancia, decidirá con alta probabilidad moverse por los trayectos con más cantidad, lo que a su vez provocará un aumento de la feromona depositada en esa zona. De este proceso emerge un comportamiento denominado *autocatalítico*: cuanto mayor sea el número de hormigas que sigan cierto trayecto, más atractivo se hará para las hormigas.

Los caminos seleccionados por menos hormigas van perdiendo concentración de feromona con el tiempo, con lo que se hacen cada vez menos deseables.

ALGORITMOS BASADOS EN COLONIAS DE HORMIGAS ARTIFICIALES

UNIVERSIDAD
DE SALAMANCA

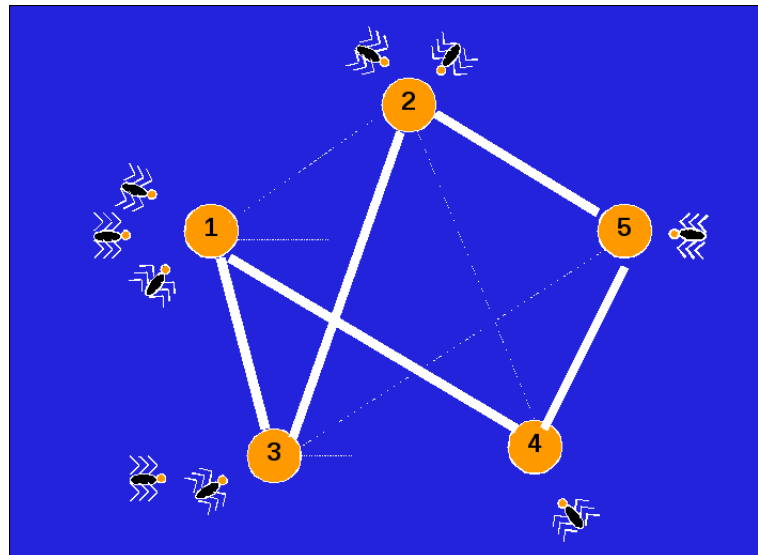
En 1991 Dorigo et al. propusieron el algoritmo *Ant System* (AS), basado en el comportamiento de las hormigas naturales, [6], [7]. El primer problema al que se aplicó esta técnica fue el TSP.

Describiremos a continuación el esquema general de los algoritmos de hormigas, aplicados a la resolución del TSP. En apartados posteriores se darán detalles de la aplicación de algoritmos concretos.

EL TSP – definición del problema

Consideremos un grafo TSP que incluye n ciudades, conectadas mediante arcos o aristas, para las que existe una medida de distancia, d , que asocia un coste d_{ij} a cada conexión (i, j) .

El objetivo del problema es encontrar un recorrido cerrado que pase una y sólo una vez por cada una de las ciudades del problema, minimizando el coste total.





EL TSP resuelto mediante hormigas

La matriz D (**matriz de distancias**) almacena las distancias asociadas a las conexiones entre ciudades del grafo TSP.

Se considera un conjunto de m hormigas que operan en paralelo.

Se asocia un rastro de feromona, $\tau_{ij}(t)$, a cada conexión (i, j) del grafo. Inicialmente todos los rastros se igualan al valor inicial τ_0 (algunos algoritmos consideran valores aleatorios). La feromona se actualiza a medida que el algoritmo avanza, posibilitando la comunicación entre las hormigas.

Cada hormiga debe definir un tour cerrado que visite una vez cada ciudad del problema. Para ello, parte desde una ciudad seleccionada aleatoriamente y determina las restantes paradas teniendo en cuenta tanto el coste como la feromona asociada a las conexiones del grafo (las hormigas prefieren seleccionar conexiones con *bajo coste y gran cantidad de feromona*).



TSP Y HORMIGAS – lista tabú

Para que una hormiga visite una y solo una vez cada ciudad, se asocia a cada hormiga una estructura de datos denominada **lista tabú**, que almacena las ciudades ya visitadas por la hormiga correspondiente.

- Inicialmente la lista tabú de todas las hormigas está vacía.
- Cada vez que una hormiga visita una ciudad, se añade a su lista tabú.
- Cuando haya completado el recorrido, todas las ciudades formarán parte de dicha lista.
- La lista se vacía cuando la hormiga vuelve a buscar un nuevo camino.



TSP Y HORMIGAS – vecindario factible

La hormiga puede elegir como destino en su desplazamiento elementos de su vecindad factible, que depende de la ciudad en la que se encuentre en cada momento.

El *vecindario factible* para la hormiga k , actualmente situada sobre la ciudad i , N_i^k , es el conjunto de ciudades aún no visitadas por la hormiga k -ésima y accesibles desde la ciudad i .



TSP Y HORMIGAS – regla de transición de estados

Permite seleccionar la siguiente parada para una hormiga.

Es diferente para cada algoritmo de hormigas, siendo función de la **feromona** y de la **visibilidad** asociadas a las conexiones del grafo.

Para el TSP la visibilidad de la conexión (i, j) , η_{ij} , es la inversa de la distancia asociada a dicha conexión. Por tanto, en este caso es un valor determinado a priori.

La regla de transición de estados considera dos parámetros, α y β , que determinan la influencia relativa del rastro de feromona y la visibilidad asociadas a las conexiones del grafo, respectivamente.

La regla de transición de estado aplicada por las hormigas indica que éstas prefieren moverse a ciudades que están próximas a la actual y conectadas a ella con arcos o aristas de feromona alta.



TSP Y HORMIGAS – actualización de la feromona

Una vez que todas las hormigas han determinado un camino, se aplica una **regla de actualización de la feromona** de las conexiones del grafo.

Dependiendo del algoritmo de hormigas considerado, dicha actualización puede ser *local* o *global*.

Diferentes algoritmos consideran diferentes hormigas para actualizar la feromona.

La regla de actualización utiliza un parámetro denominado **factor de evaporación de la feromona**, ρ , que determina la velocidad a la que se evapora la feromona asociada a las conexiones del grafo.



TSP Y HORMIGAS – esquema del algoritmo

Parámetros: $D, \alpha, \beta, \tau_0, t_{max}$

INICIO

$t = 0$

Hacer $\tau_{ij}(t) = \tau_0 \forall (i, j) \in E$

Mientras ($t < t_{max}$)

Elegir la ciudad de partida de cada hormiga (aleatoria).

Determinar las sucesivas paradas de cada hormiga, según la regla probabilística.

Aplicar la actualización local de la feromona (si procede).

Aplicar la actualización global de la feromona.

$t = t + 1$

fin-mientras

FIN

Antes de aplicar el algoritmo de hormigas, se determina el valor de ρ , α , β , la feromona inicial (suele ser una constante positiva pequeña, τ_0), el número máximo de iteraciones que realizará el algoritmo en busca de la solución, t_{max} , y el número de hormigas que operan en paralelo, m .

El proceso se repite hasta que la solución converja o hasta que se hayan realizado el número máximo de iteraciones prefijadas.



TSP Y HORMIGAS – implementación secuencial o paralela

El esquema de construcción de la solución se puede implementar de forma secuencial o paralela.

Si se implementa de forma secuencial, cada hormiga construye un tour completo antes de que la siguiente hormiga comience a construir otro.

Si se implementa de forma paralela, en cada paso de construcción todas las hormigas se mueven desde la ciudad en la que están hasta la siguiente parada.

Para algunos algoritmos la implementación elegida influye en el comportamiento del algoritmo.



VARIANTES DE LA SOLUCIÓN BASADA EN HORMIGAS

Se han propuestos diversas variantes y mejoras sobre el modelo de hormigas artificiales. Entre los algoritmos aplicables a problemas de optimización combinatoria NP-completos se encuentran:

- Ant System (sistema de hormigas)
- Ant Colony System (sistema de colonia de hormigas)
- Max-Min Ant System (sistema de hormigas max-min)
- Rank-based Ant System (sistema de hormigas con ordenación)
- Best-worst Ant System (sistema de la mejor-peor hormiga)
- Elitist Ant System (sistema de hormigas elitista)



ALGORITMO ANT SYSTEM - AS

Desarrollado por Dorigo, Maniezzo y Colorni en 1991 [9], fue el primer algoritmo basado en el uso de colonias de hormigas artificiales. Su interés radica en este hecho, más que en sus resultados, que son peores que los obtenidos por otros modelos que describiremos.

Se desarrollaron tres variantes, denominadas *Ant-quantity*, *Ant-density* y *Ant-cycle*, que se diferencian en la forma en que se actualiza la feromona. En *Ant-cycle* la feromona se deposita una vez que la solución está completa. Esta versión es la que generó mejores resultados, y es la que se suele tomar por defecto para Ant-System.

AS – regla de transición de estados

$$p_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(t) \cdot \eta_{ij}^\beta}{\sum_{u \in N_i^k} [\tau_{iu}^\alpha(t) \cdot \eta_{iu}^\beta]} & \text{si } j \in N_i^k \\ 0 & \text{en otro caso} \end{cases} \quad (1)$$

Cada hormiga, k , determina la siguiente parada de su camino aplicando la **regla de transición de estados** 1, denominada *regla proporcional aleatoria*. Dicha expresión determina la probabilidad con que la hormiga k , actualmente situada en la ciudad i , elige moverse a la ciudad j . La probabilidad de elegir una conexión (i, j) concreta se incrementa con el valor del rastro de feromona actualmente asociado, $\tau_{ij}(t)$, y el valor de la información heurística, η_{ij} .

AS - actualización de la feromona

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k(t), \forall(i, j) \quad (2)$$

$\Delta\tau_{ij}^k(t)$: cantidad de feromona depositada por la hormiga k sobre la conexión (i, j) en la iteración t .

1- se evapora parte de la huella asociada a todas las conexiones del grafo (evita un crecimiento ilimitado de ésta, y representar el fenómeno observado en las colonias de hormigas naturales).

2- cada hormiga k deposita una cantidad de feromona en las conexiones del tour que ha construido, T_k , proporcional a la longitud de dicho tour, L_k
 \Rightarrow las conexiones pertenecientes a muchas soluciones ven incrementada en mayor medida su feromona.

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{1}{L_k(t)} & \text{si } (i, j) \in T_k(t) \\ 0 & \text{en otro caso} \end{cases} \quad (3)$$

siendo $0 < \rho \leq 1$, el **factor de evaporación** de la feromona.



ALGORITMO ELITIST ANT-SYSTEM - EAS

El algoritmo fue propuesto por Dorigo, [6], [7], para mejorar la lentitud en la convergencia del algoritmo AS. La única diferencia con respecto a dicho modelo reside en la regla de actualización, que en este caso *aplica un refuerzo adicional de las conexiones buenas*.

Este algoritmo considera la misma regla de selección de las paradas del camino que AS, 1.

Cuando cada hormiga k ha definido su tour, S_k , el más corto se selecciona como la mejor solución de la iteración actual del algoritmo. Si la mejor solución de la iteración es más corta que la obtenida en las iteraciones previas, se selecciona como la mejor solución hasta el momento, T_{best} .

EAS – actualización de la feromona

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k + e\Delta\tau_{ij}^b \quad (4)$$

1- se reduce la feromona de todas las conexiones en un factor constante.

2- cada hormiga k deposita feromona sobre las conexiones del tour que ha definido en la iteración actual, S_k .

3- se deposita feromona en las conexiones que pertenecen a la mejor solución encontrada desde el comienzo del algoritmo, T_{best} , proporcional al número de hormigas elitistas, e , consideradas por el algoritmo.

$\Delta\tau_{ij}^k$ viene dado por la expresión 3, y $\Delta\tau_{ij}^b$ por 5 (realmente las expresiones 3 y 5 tienen la misma forma).

$$\Delta\tau_{ij}^b = \begin{cases} \frac{1}{L_{best}} & \text{si } (i, j) \in T_{best} \\ 0 & \text{en otro caso} \end{cases} \quad (5)$$

El uso de la estrategia elitista con un valor adecuado del parámetro e permite a AS tanto encontrar mejores tours como encontrarlos en un número menor de iteraciones.



ALGORITMO RANK-BASED ANT SYSTEM – RAS

Es otra extensión del modelo propuesto por Bullnheimer, Hartl y Strauss en 1997, [1], [2], [3].

Se diferencia del algoritmo AS en el mecanismo de actualización de la feromona. En este caso no depositan feromona todas las hormigas, sino sólo las que han encontrado las mejores soluciones en la iteración actual, así como la hormiga que ha encontrado la mejor solución desde el comienzo del algoritmo. Además, el aporte de cada hormiga es proporcional al orden de la hormiga y la calidad de su solución.



RAS – parámetros iniciales

Número de hormigas que actualizan la huella, $w \leq m$. El valor inicial para la huella, τ_0 se calcula según:

$$\tau_0 = \frac{0.5w(w-1)}{\rho L_{nn}} \quad (6)$$

donde L_{nn} es el coste de una solución de vecino más próximo para el problema.



RAS – actualización de la feromona

Para realizar esta actualización, se ordenan las hormigas por longitud de tour creciente (las longitudes ordenadas de forma creciente serán: L'_1, L'_2, \dots, L'_m , correspondiendo a los tours T'_1, T'_2, \dots, T'_m). La feromona la actualizan la mejor hormiga desde el comienzo del algoritmo y las $w - 1$ mejores hormigas de la iteración actual.

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \sum_{r=1}^{w-1} (w - r)\Delta\tau_{ij}^r + w\Delta\tau_{ij}^b. \quad (7)$$

$\Delta\tau_{ij}^r$ viene dado por 8 y $\Delta\tau_{ij}^b$ por 5.

$$\Delta\tau_{ij}^r(t) = \begin{cases} \frac{1}{L'_r(t)} & \text{si } (i, j) \in T'_r(t) \\ 0 & \text{en otro caso} \end{cases} \quad (8)$$

Según muestra la expresión 7, en primer lugar, se reduce la feromona de todas las conexiones en un factor constante. A continuación, las hormigas seleccionadas depositan feromona sobre las conexiones del tour que han definido. La cantidad de feromona que deposita una hormiga está ponderada de acuerdo con el rango de la hormiga. El mejor tour hasta el momento contribuye más a la actualización de la feromona, aplicando peso w . El peso asociado a la contribución de la r -ésima mejor hormiga de la iteración viene dado por $\max\{0, w - r\}$; es decir, cada hormiga deposita una cantidad de feromona que decrece con su rango.

Por tanto, las hormigas depositan una cantidad de feromona que decrece con su rango; además, la mejor hormiga hasta el momento siempre deposita la mayor cantidad de feromona en cada iteración.



ALGORITMO ANT COLONY SYSTEM - ACS

Fue propuesto por Dorigo y Gambardella, [7], [8]. Incluye tres diferencias básicas respecto a Ant-System:

- La regla de transición establece un equilibrio entre la exploración de nuevas soluciones y la explotación de la información acumulada.
- La feromona sólo la actualiza la hormiga que generó la mejor solución hasta el momento y sólo afecta a las conexiones de dicha solución. Esta *actualización es global*.
- Se añade una *actualización local*, por la cual cada hormiga modifica la feromona de cada conexión que recorre, para diversificar la búsqueda.

La implementación del mecanismo de construcción de la solución (paralelo o secuencial), sí que incluye en el comportamiento del algoritmo.

ACS – regla de transición de estados

Se denominada **regla proporcional pseudoaleatoria**.

Para la hormiga k , actualmente situada en la ciudad i , la siguiente parada del camino, j , se elige de forma aleatoria mediante la siguiente distribución de probabilidad:

Si $q \leq q_0$:

$$p_{ij}^k = \begin{cases} 1 & \text{si } j = \arg \max_{u \in N_i^k} \left\{ \tau_{iu}^\alpha \cdot \eta_{iu}^\beta \right\} \\ 0 & \text{en otro caso} \end{cases} \quad (9)$$

Si $q > q_0$:

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha \cdot \eta_{ij}^\beta}{\sum_{u \in N_i^k} [\tau_{iu}^\alpha \cdot \eta_{iu}^\beta]} & \text{si } j \in N_i^k \\ 0 & \text{en otro caso} \end{cases} \quad (10)$$

$q_0 \in [0, 1]$ es un parámetro y q un valor aleatorio uniformemente distribuido en el intervalo $[0, 1]$.

Con probabilidad q_0 las hormigas realizan el mejor movimiento posible teniendo en cuenta la información proporcionada por la feromona. En este caso se realiza una *explotación de la información aprendida* y almacenada en la feromona. Con probabilidad $(1 - q_0)$ se realiza una exploración sesgada de las conexiones. Dando diferentes valores a q_0 se puede potenciar la *exploración de nuevas soluciones* o la explotación del conocimiento existente, concentrando la búsqueda entorno a la mejor solución encontrada hasta el momento.

Por tanto, se establece un compromiso entre la exploración de nuevas conexiones y la explotación de la información disponible en ese momento.



ACS – actualización de la feromona

En este algoritmo se realiza una actualización de la feromona tanto local como global.

ACTUALIZACIÓN LOCAL: Se realiza a medida que cada hormiga construye su solución.

ACTUALIZACIÓN GLOBAL: Se realiza al concluir cada iteración.

El parámetro ρ puede tomar diferente valor para ambas actualizaciones.

El esquema de actualización local hace que la implementación paralela o secuencial del algoritmo condicione su resultado.

ACS – actualización local de la feromona

Cada hormiga h realiza la **actualización local** de la huella depositando feromona en las conexiones utilizadas en la solución que ha definido, T_h :

$$\tau_{ij}(t) = \rho_L \tau_0 + (1 - \rho_L) \tau_{ij}(t) \quad \forall (i, j) \in T_h \quad (11)$$

τ_0 : valor inicial de feromona, $0 < \rho_L < 1$: persistencia local.

La feromona de cada conexión se actualiza inmediatamente después de que dicha conexión es recorrida por una hormiga, con lo que su valor disminuye, haciéndola menos deseable para las siguientes hormigas. Esto hace que las conexiones ya visitadas vayan siendo menos prometedoras a medida que las recorren más hormigas en la iteración actual, lo que favorece la exploración de conexiones no visitados, contribuyendo a que el algoritmo no se estanque.

ACS – actualización global de la feromona

La feromona es actualizada globalmente sólo por la hormiga que encontró la mejor solución. Se pueden considerar tanto la mejor hormiga de la iteración actual (*iteration-best*), como la mejor hasta el momento (*best-so-far*).

En primer lugar se evapora la feromona en todas las conexiones del grafo:

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) \quad \forall (i, j) \quad (12)$$

A continuación, se deposita feromona en las conexiones que forman parte de la ruta de la mejor hormiga, T_{best_iter} :

$$\tau_{ij}(t+1) = \tau_{ij}(t+1) + \rho\Delta\tau_{ij}^{best} \quad \forall (i, j) \in T_{best_iter} \quad (13)$$

donde el incremento es inversamente proporcional a la longitud de la solución encontrada por la mejor hormiga, L_{best_iter} :

$$\Delta\tau_{ij}^{best} = \frac{1}{L_{best_iter}} \quad (14)$$

A diferencia de las expresiones que actualizan la feromona en AS, en este caso, la actualización global pondera el incremento multiplicándolo por ρ , lo que da lugar a que la nueva feromona sea una media ponderada entre el valor previo de la feromona y la cantidad de feromona depositada.



ALGORITMO MAX-MIN ANT SYSTEM - MMAS

Es una mejora de Ant System, propuesta por Stützle y Hoos en 1997 [13], cuyo objetivo es realizar una mayor explotación de las mejores soluciones y utilizar un mecanismo de búsqueda adicional para evitar el estancamiento del proceso.

Usa la misma regla de selección que el algoritmo AS, 1, añadiendo a éste tres aspectos:

- El mecanismo de actualización es más agresivo, al evaporar todos los rastros y aportar sólo en los de la mejor solución.
- Define unos valores mínimo y máximo para la feromona: τ_{min} y τ_{max} .
- Reinicializa la búsqueda cuando se estanca.



MMAS – rango limitado para la feromona

Para evitar el estancamiento de la búsqueda, se limita el valor de la feromona al intervalo $[\tau_{min}, \tau_{max}]$, siendo $\tau_{min} > 0$. Ambos valores se fijarán para cada problema particular, siendo más crítico τ_{min} .

La feromona se inicia al valor τ_{max} para todas las conexiones, lo que permite una mayor exploración al comienzo del algoritmo. Además, iniciando al valor máximo, al aplicar la regla de actualización las conexiones de las buenas soluciones mantienen valores altos y las de las malas lo reducen.

Al actualizar la feromona, se fuerzan los valores al intervalo $[\tau_{min}, \tau_{max}]$.

MMAS – actualización de la feromona

Tras cada iteración sólo añade feromona una hormiga (como en ACS), aplicando la siguiente regla de actualización de la huella a todas las conexiones (en ACS sólo se aplicaba a las visitadas por la mejor hormiga):

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^{best} \quad (15)$$

donde

$$\Delta\tau_{ij}^{best} = \begin{cases} \frac{1}{L_{best}} & \text{si } (i, j) \in L_{best} \\ 0 & \text{en otro caso} \end{cases} \quad (16)$$

donde L_{best} : longitud del mejor tour elegido.

Como en ACS, se puede elegir como mejor hormiga la que generó la mejor solución en la iteración actual, o la que ha generado la mejor solución a lo largo de todas las iteraciones realizadas.

En este algoritmo, cuando se detecta que la búsqueda está estancada, se aplica una reinicialización, volviendo a poner todos los valores de feromona a τ_{max} .



BIBLIOGRAFÍA

References

- [1] B. Bullnheimer, R. F. Hartl, and C. Strauss. The general routing problem on a mixed graph. Technical report, Institute of Management Science, University of Vienna, Austria, 1997.
- [2] B. Bullnheimer, R. F. Hartl, and C. Strauss. Applying the ant system to the vehicle routing problem. *Advances and Trends in Local Search Paradigms for Optimization*, pages 285–296, 1999.
- [3] B. Bullnheimer, R. F. Hartl, and C. Strauss. A new rank-based version of the ant system: A computational study. *Central European Journal for Operations Research and Economy*, 7(1):25–38, 1999.

- [4] D. Costa and A. Hertz. Ants can colour graphs. *Journal of the Operational Research Society*, 48:295–305, 1997.
- [5] J.-L. Deneubourg, S. Aron, S. Goss, and J.-M. Pasteels. The self-organizing exploratory pattern of the argentine ant. *Journal of Insect Behaviour*, 3:159–168, 1990.
- [6] M. Dorigo. *Optimization, Learning and Natural Algorithms*. PhD thesis, Dip. Elettronica, Politecnico di Milano, 1992.
- [7] M. Dorigo and L. M. Gambardella. Ant colony system: A cooperative learning approach to the traveling salesman problem. Technical Report IRIDIA. 96-05, Université Libre de Bruxelles, 1996.
- [8] M. Dorigo and L. M. Gambardella. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transaction on Evolutionary Computation*, 1(1):53–66, 1997.

- [9] M. Dorigo, V. Maniezzo, and A. Coloni. Ant system: An atocatalytic optimizing process. Technical report, Dipartimento di Electronica e Informazione - Politecnico di Milano, Italia, 1991.
- [10] L. M. Gambardella, E. Taillard, and G. Agazzi. *Ant Colonies for Vehicle Routing Problems*, chapter New Ideas in Optimization. D. Corne, M. Dorigo, and F. Glover, editors, 1999.
- [11] V. Maniezzo, A. Coloni, and M. Dorigo. The ant system applied to the quadratic assignment problem. Technical Report IRIDIA/94-28, Université Libre de Bruxelles, Belgium, 1994.
- [12] V. Maniezzo, A. Coloni, and M. Dorigo. The ant system applied to the quadratic assignment problem. Technical Report IRIDIA/94-28, Université Libre de Bruxelles, Belgium, 1994.
- [13] T. Stützle T. and H. Hoos. The MAX-MIN ant system and local search

for the traveling salesman problem. In T. Baeck, Z. Michalewicz, and X. Yao, editors, *Proceedings of the IEEE International Conf on Evolutionary Computation*, pages 309–314, 1997.