

BIOINFORMÁTICA

2013 - 2014



PARTE I. INTRODUCCIÓN

- Tema 1. Computación Basada en Modelos Naturales

PARTE II. MODELOS BASADOS EN ADAPTACIÓN SOCIAL (Swarm Intelligence)

- Tema 2. Introducción a los Modelos Basados en Adaptación Social
- **Tema 3. Optimización Basada en Colonias de Hormigas**
- Tema 4. Optimización Basada en Nubes de Partículas (Particle Swarm)

PARTE III. COMPUTACIÓN EVOLUTIVA

- Tema 5. Introducción a la Computación Evolutiva
- Tema 6. Algoritmos Genéticos I. Conceptos Básicos
- Tema 7. Algoritmos Genéticos II. Diversidad y Convergencia
- Tema 8. Algoritmos Genéticos III. Múltiples Soluciones en Problemas Multimodales
- Tema 9. Estrategias de Evolución y Programación Evolutiva
- Tema 10. Algoritmos Basados en Evolución Diferencial (Differential Evolution – DE)
- Tema 11. Modelos de Evolución Basados en Estimación de Distribuciones (EDA)
- Tema 12. Algoritmos Evolutivos para Problemas Multiobjetivo
- Tema 13. Programación Genética
- Tema 14. Modelos Evolutivos de Aprendizaje

PARTE IV. OTROS MODELOS DE COMPUTACIÓN BIOINSPIRADOS

- Tema 15. Sistemas Inmunológicos Artificiales
- Tema 16. Otros Modelos de Computación Natural/Bioinspirados

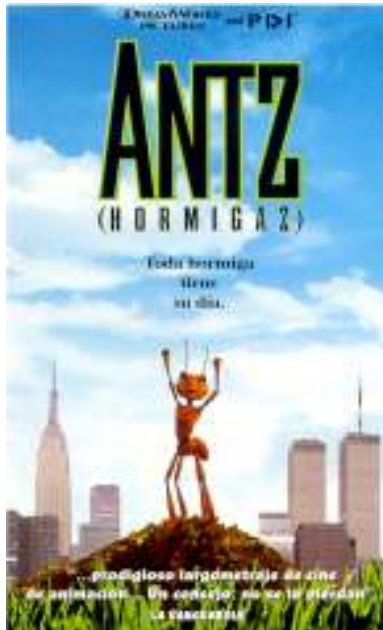
BIOINFORMÁTICA

TEMA 3: ALGORITMOS DE OPTIMIZACIÓN BASADOS EN COLONIAS DE HORMIGAS

- 1. LAS COLONIAS DE HORMIGAS NATURALES**
- 2. ALGORITMOS DE OPTIMIZACIÓN BASADOS EN COLONIAS DE HORMIGAS**
- 3. APLICACIONES**
- 4. COMENTARIOS FINALES**
- 5. BIBLIOGRAFÍA Y RECURSOS**

1. LAS COLONIAS DE HORMIGAS NATURALES

- Las hormigas son insectos sociales que viven en colonias y que tienen un comportamiento dirigido al desarrollo de la colonia como un todo mas que a un desarrollo individual



“Antz (Hormiga Z)”

© DreamWorks Pictures. 1998

Recordad...

¡SED LA BOLA!

1. LAS COLONIAS DE HORMIGAS NATURALES (2)

- Una característica interesante del comportamiento de las colonias de hormigas es cómo pueden encontrar los caminos más cortos entre el hormiguero y la comida
- Sobre todo porque... ¡¡LAS HORMIGAS SON CIEGAS!!

Entonces...

¿Cómo lo hacen?



1. LAS COLONIAS DE HORMIGAS NATURALES (3)

- En su recorrido, depositan una sustancia llamada feromona que todas pueden oler (*estimergia*)
- Este rastro permite a las hormigas volver a su hormiguero desde la comida

“Bichos. Una aventura en miniatura”

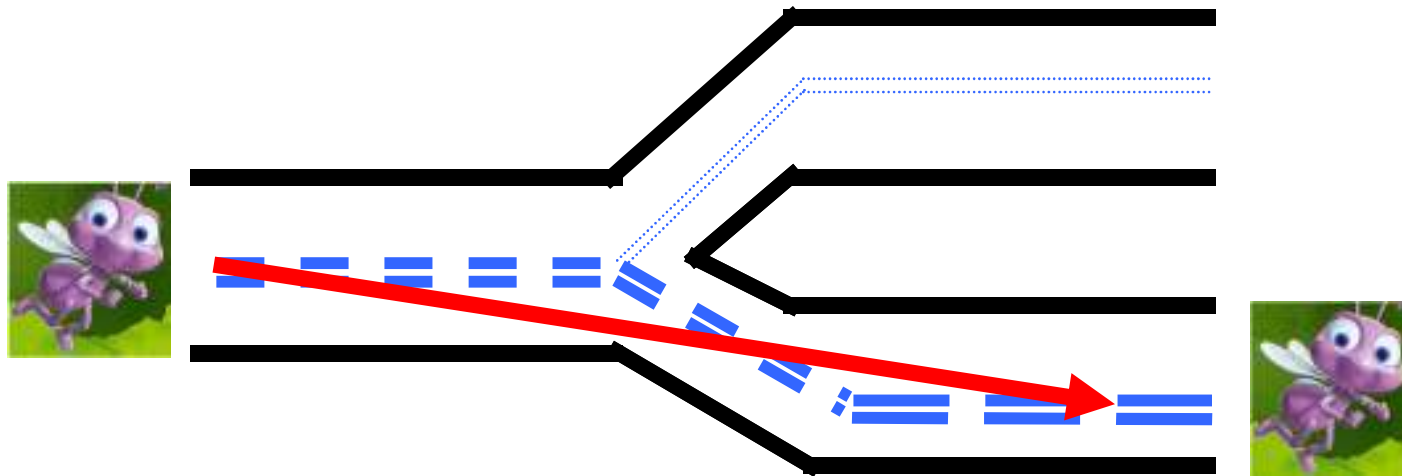
© Disney-Pixar. 1999

¡He perdido el rastro,
he perdido el rastro!



1. LAS COLONIAS DE HORMIGAS NATURALES (4)

- Cada vez que una hormiga llega a una intersección, decide el camino a seguir de un modo probabilístico

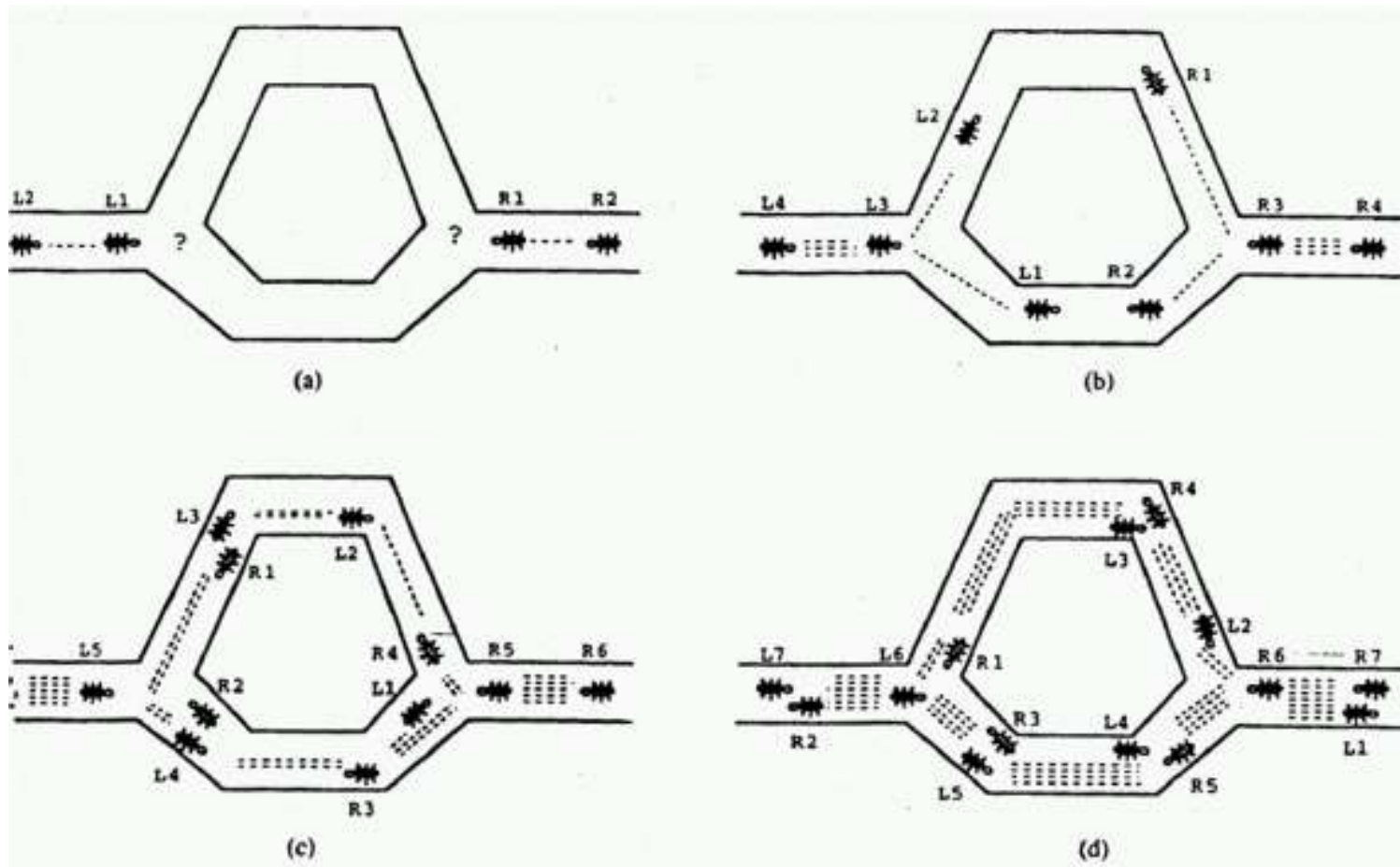


- Las hormigas eligen con mayor probabilidad los caminos con un alto rastro de feromona

1. LAS COLONIAS DE HORMIGAS NATURALES (5)

- Las bifurcaciones más prometedoras (más cercanas a la comida) van acumulando feromona al ser recorridas por más hormigas (*reclutamiento de masas*)
- Las menos prometedoras pierden feromona por *evaporación* al ser visitadas por menos hormigas cada vez. Aún así, *la gran perduración de los rastros hace que la evaporación influya poco*
- La acción continuada de la colonia da lugar a un rastro de feromona que permite a las hormigas encontrar un camino cada vez más corto desde el hormiguero a la comida

1. LAS COLONIAS DE HORMIGAS NATURALES (6)

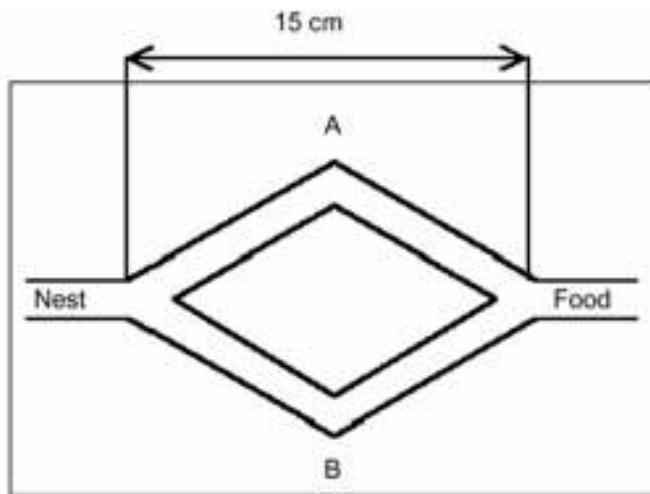


1. LAS COLONIAS DE HORMIGAS NATURALES (7)

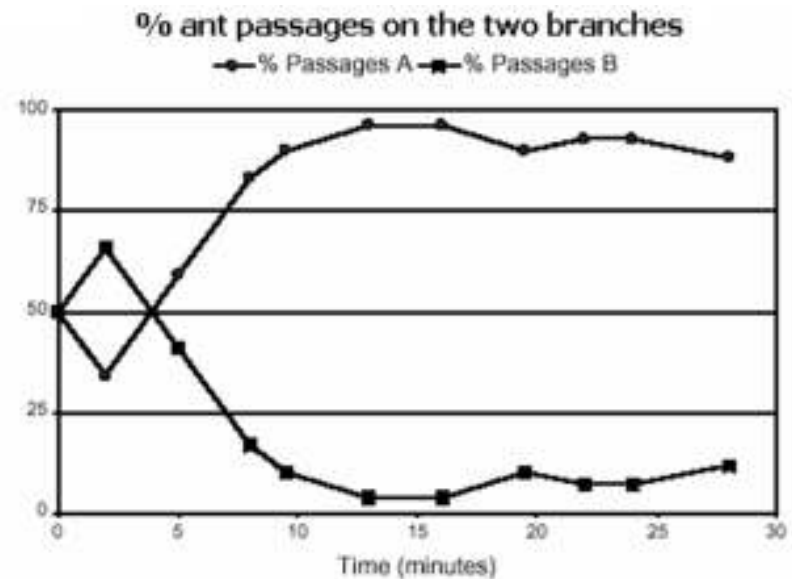
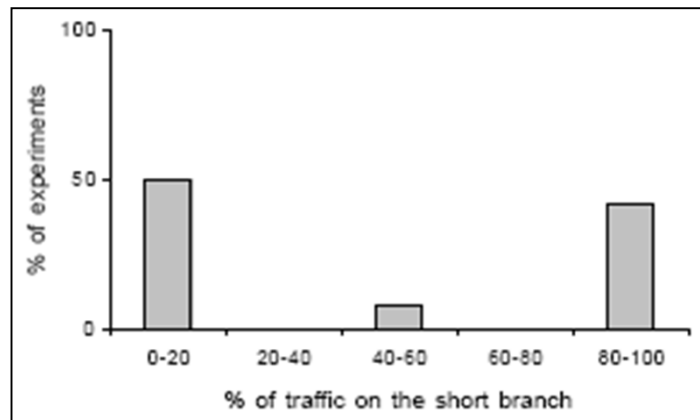
EXPERIMENTOS DEL DOBLE PUENTE

- Deneubourg realizó un experimento de laboratorio con un tipo concreto de hormigas que depositan feromona al ir del hormiguero a la comida y al volver
- Usaron dos tipos de circuitos (**puentes**). En el primero, las dos ramas del puente tenían la misma longitud. En el segundo, una rama era el doble de larga que la otra
- Después, unieron dos puentes cruzados del segundo tipo

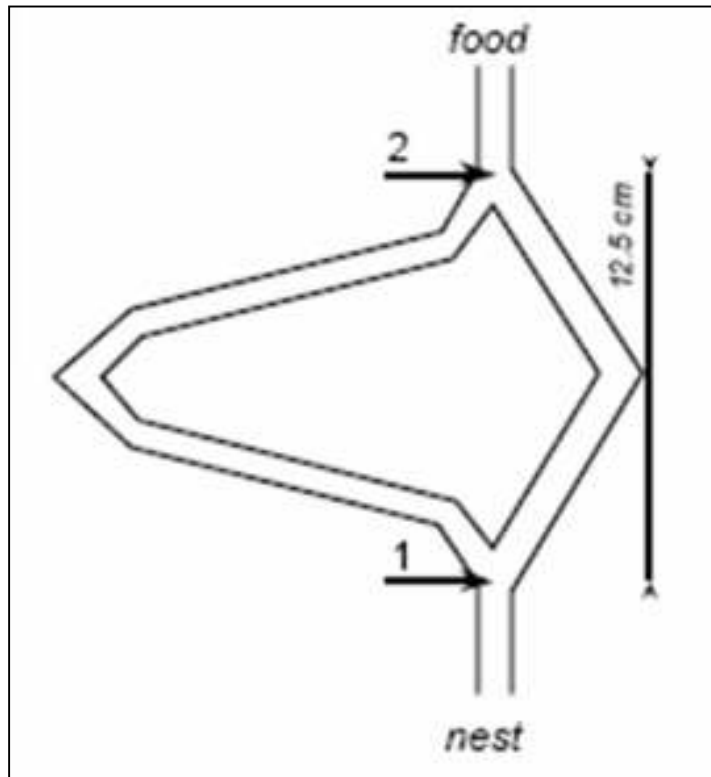
1. LAS COLONIAS DE HORMIGAS NATURALES (8)



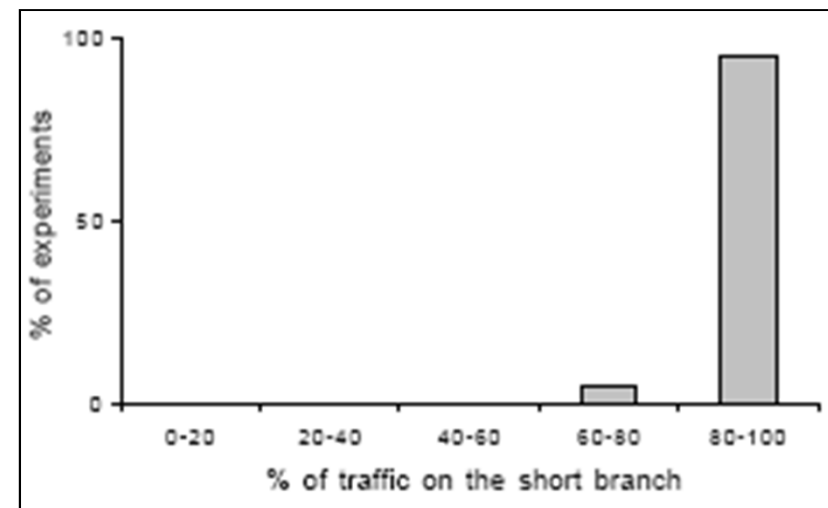
- En el primer puente, las hormigas terminaban por converger a una sola rama (cualquiera de las dos)



1. LAS COLONIAS DE HORMIGAS NATURALES (9)

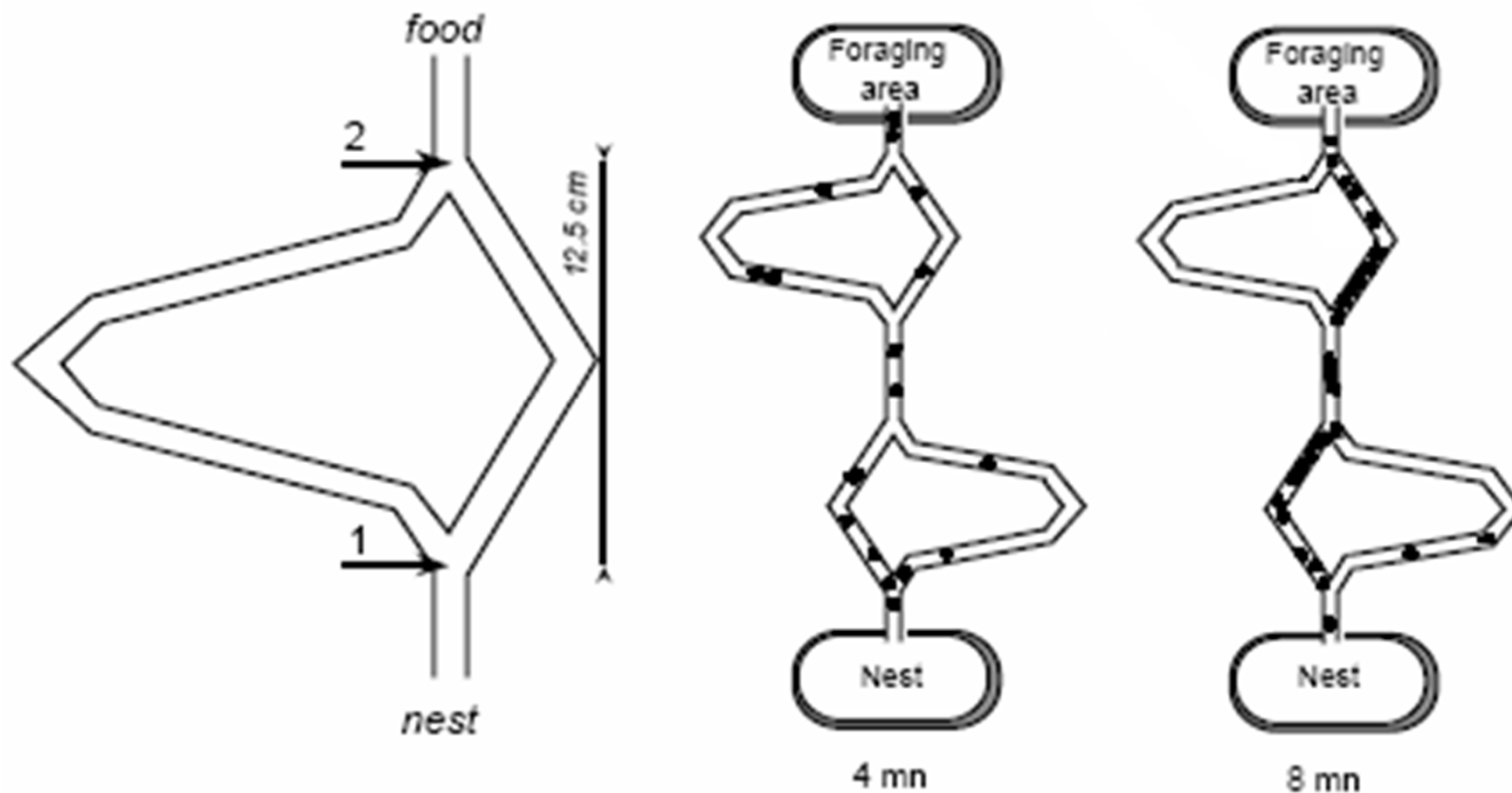


- En el segundo, las hormigas convergían a la rama más corta



1. LAS COLONIAS DE HORMIGAS NATURALES (10)

- En el circuito con dos puentes dobles cruzados, las hormigas consiguen encontrar el camino más corto



1. LAS COLONIAS DE HORMIGAS NATURALES (11)

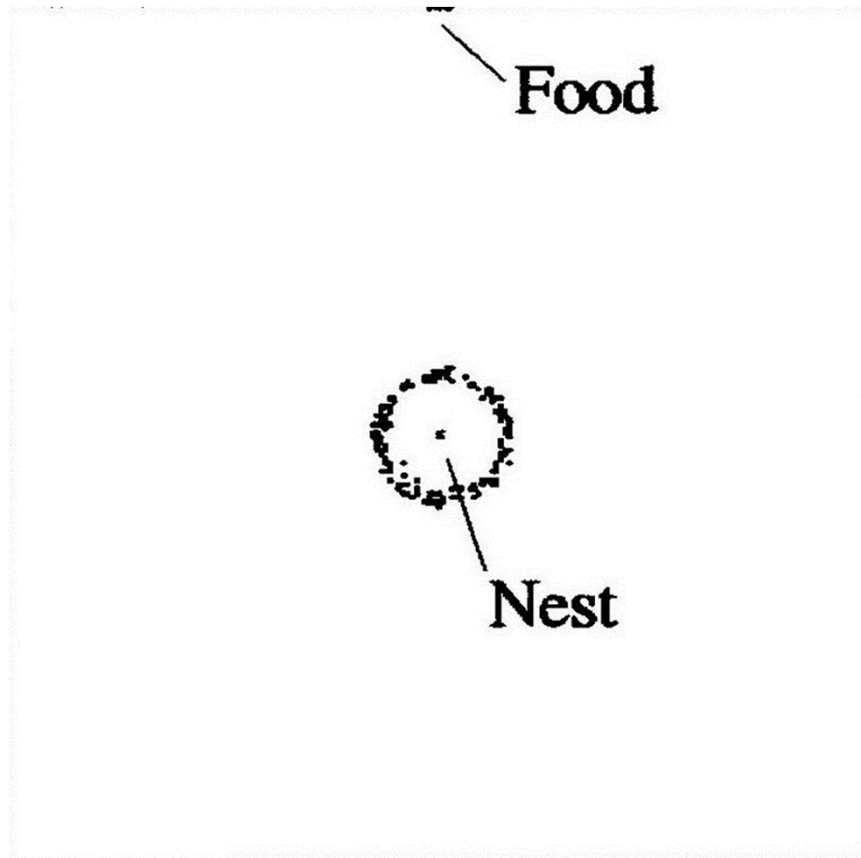
- Como resultado de estos experimentos, Deneubourg y su equipo diseñaron un **modelo estocástico** del proceso de decisión de las hormigas naturales:

$$p_{i,a} = \frac{[k + \tau_{i,a}]^\alpha}{[k + \tau_{i,a}]^\alpha + [k + \tau_{i,a'}]^\alpha}$$

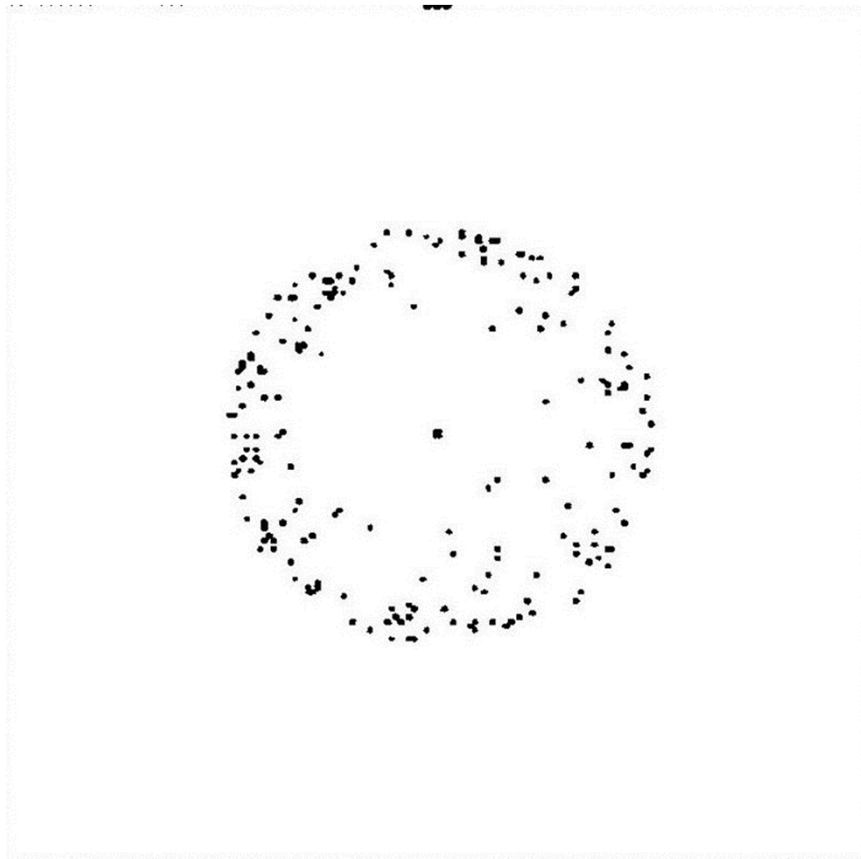
donde:

- $p_{i,a}$ es la probabilidad de escoger la rama a estando en el punto de decisión i , y
- $\tau_{i,a}$ es la concentración de feromona en la rama a

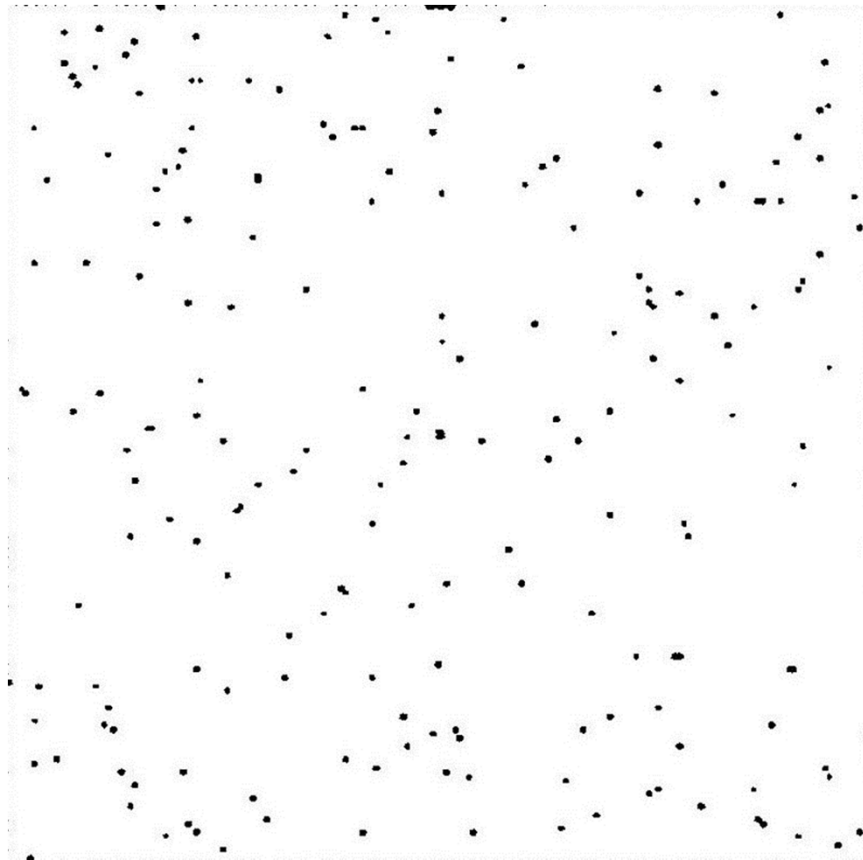
1. LAS COLONIAS DE HORMIGAS NATURALES (12)



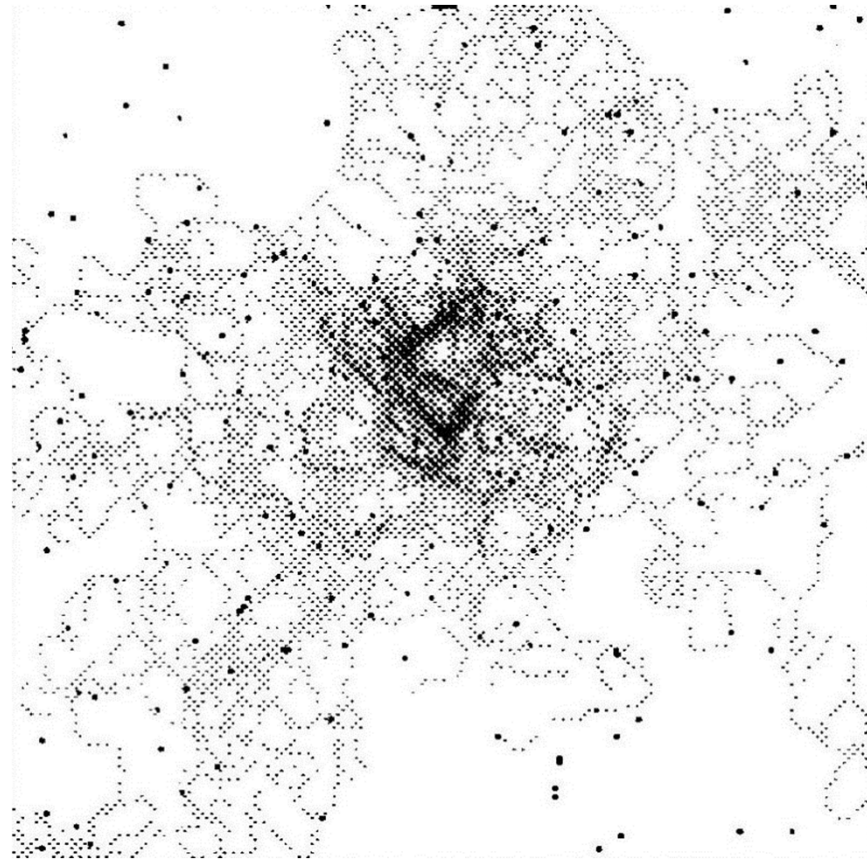
1. LAS COLONIAS DE HORMIGAS NATURALES (12)



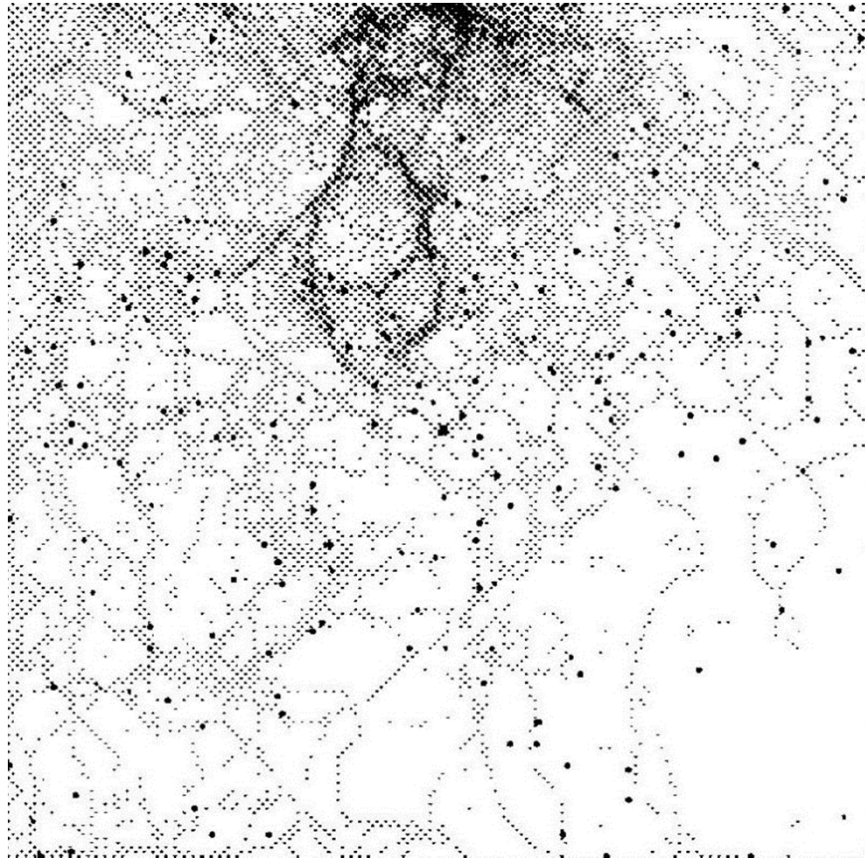
1. LAS COLONIAS DE HORMIGAS NATURALES (12)



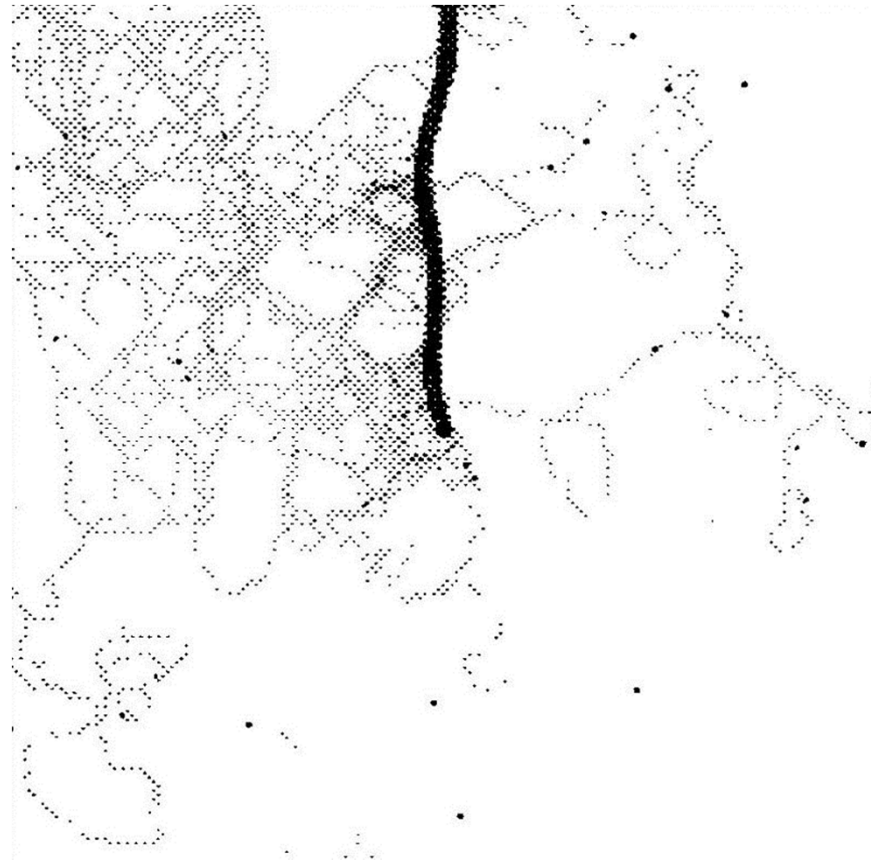
1. LAS COLONIAS DE HORMIGAS NATURALES (12)



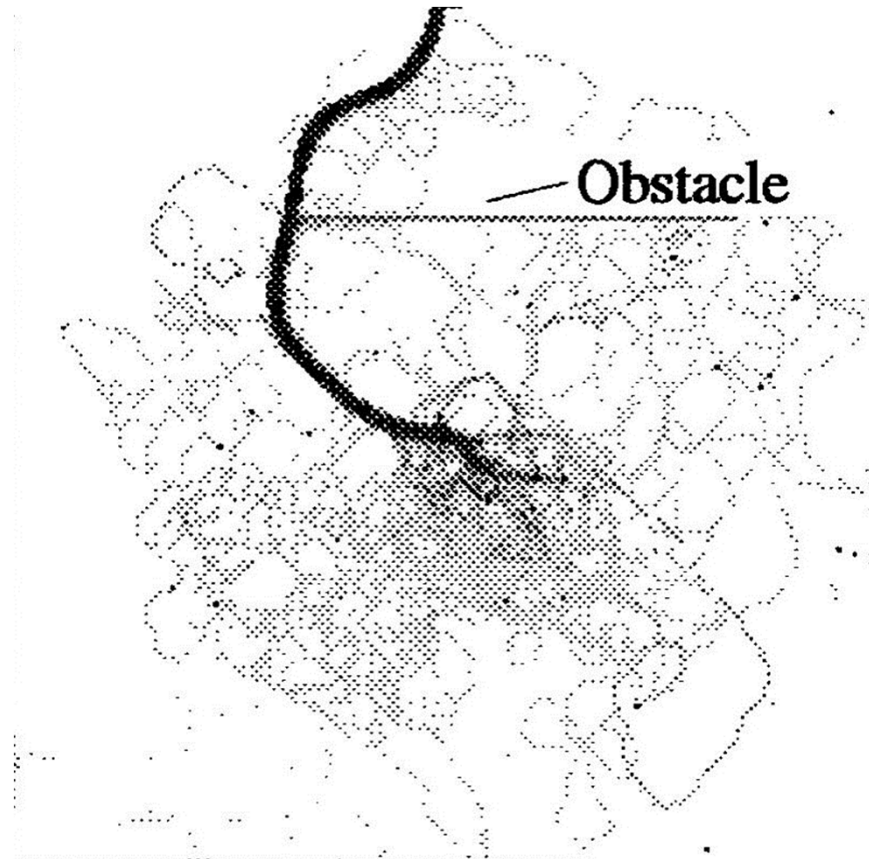
1. LAS COLONIAS DE HORMIGAS NATURALES (12)



1. LAS COLONIAS DE HORMIGAS NATURALES (12)

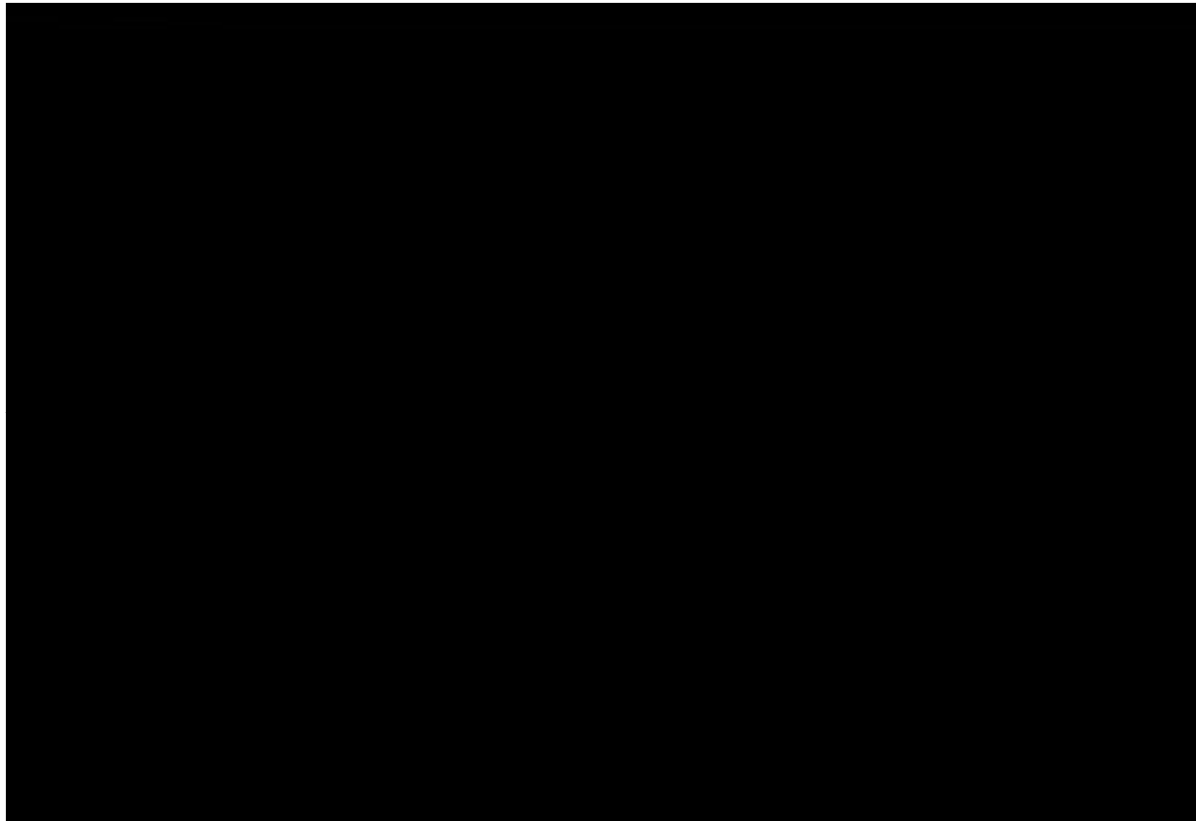


1. LAS COLONIAS DE HORMIGAS NATURALES (12)



1. LAS COLONIAS DE HORMIGAS NATURALES (13)

Experimento con Hormigas reales. Como encuentran el camino mínimo (159 segundos)

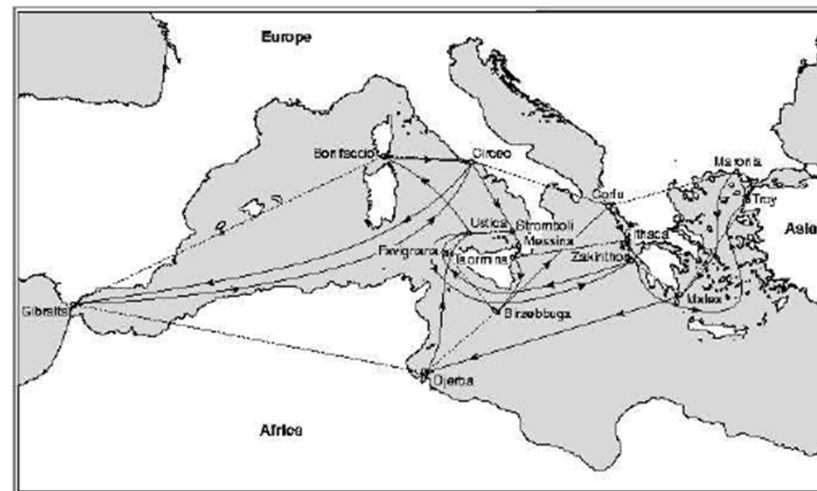


2. ALGORITMOS DE OPTIMIZACIÓN BASADOS EN COLONIAS DE HORMIGAS

- **De las hormigas naturales a la OCH**
- **La hormiga artificial**
- **Actualización de feromona**
- **El Sistema de Hormigas**
- **Ejemplo de aplicación del SH al TSP**
- **Otros algoritmos de OCH**
- **Estudio comparativo de la OCH en el TSP**
- **Algoritmos de OCH con búsqueda local**
- **La metaheurística OCH**

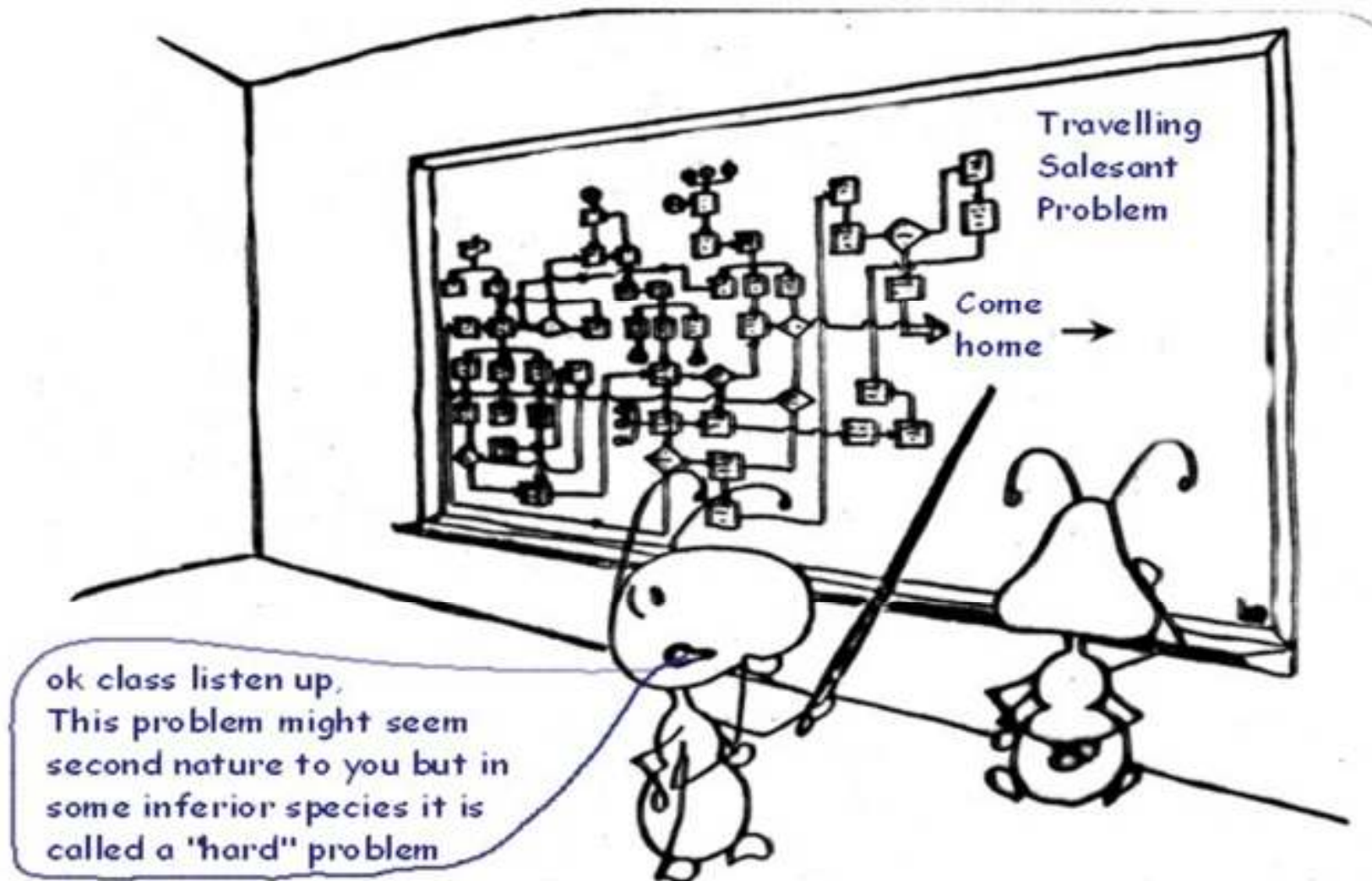
DE LAS HORMIGAS NATURALES A LA OCH

- El Problema del Viajante de Comercio (TSP) es uno de los problemas de optimización combinatoria más conocido
- Se dispone de un conjunto $N = \{1, \dots, n\}$ de ciudades, que han de ser visitadas una sola vez, volviendo a la ciudad de origen, y recorriendo la menor distancia posible



- **Las hormigas naturales son capaces de resolver problemas de camino mínimo como el TSP**

DE LAS HORMIGAS NATURALES A LA OCH

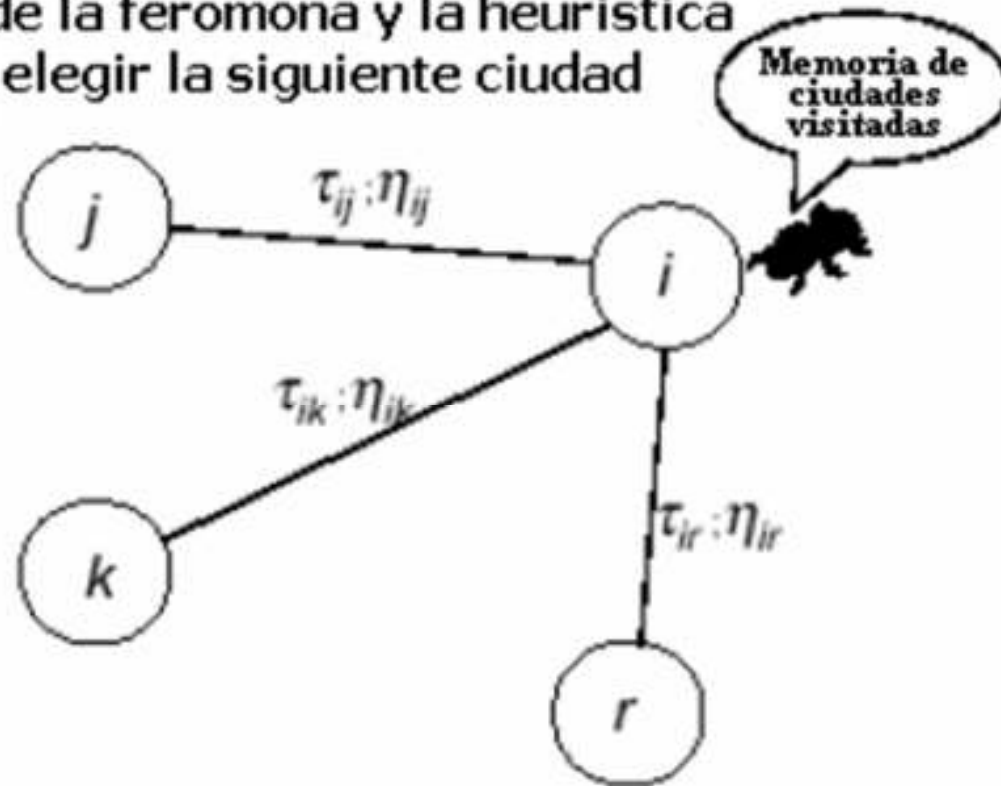


DE LAS HORMIGAS NATURALES A LA OCH (2)

- Los algoritmos de OCH reproducen el comportamiento de las hormigas reales en una colonia artificial de hormigas para resolver problemas complejos de camino mínimo
- Cada hormiga artificial es un mecanismo probabilístico de construcción de soluciones al problema (un agente que imita a la hormiga natural) que usa:
 - Unos rastros de feromona (artificiales) τ que cambian con el tiempo para reflejar la experiencia adquirida por los agentes en la resolución del problema
 - Información heurística η sobre la instancia concreta del problema

DE LAS HORMIGAS NATURALES A LA OCH (3)

Uso de la feromona y la heurística para elegir la siguiente ciudad



2. ALGORITMOS DE OPTIMIZACIÓN BASADOS EN COLONIAS DE HORMIGAS

- De las hormigas naturales a la OCH
- **La hormiga artificial**
- **Actualización de feromona**
- **El Sistema de Hormigas**
- **Ejemplo de aplicación del SH al TSP**
- **Otros algoritmos de OCH**
- **Estudio comparativo de la OCH en el TSP**
- **Algoritmos de OCH con búsqueda local**
- **La metaheurística OCH**

LA HORMIGA ARTIFICIAL

La hormiga artificial es un agente que:

- Recuerda los nodos que ha recorrido, utilizando para ello una lista de nodos visitados (L). Al finalizar, esta lista contiene la solución construida por la hormiga
- En cada paso, estando en la ciudad r elige hacia qué ciudad s moverse de entre las vecinas de r que no hayan sido visitados aún ($J(r) = \{s \mid \exists a_{rs} \text{ y } s \notin L\}$), según una regla probabilística de transición

La decisión tomada es función de la preferencia heurística $\eta_{rs}=1/d_{rs}$ y la feromona τ_{rs}

LA HORMIGA ARTIFICIAL (2)

- La regla probabilística de transición más habitual define la probabilidad con la que la hormiga k , situada en la ciudad r , decide moverse hacia la ciudad s :

$$p_k(r, s) = \begin{cases} \frac{[\tau_{rs}]^\alpha \cdot [\eta_{rs}]^\beta}{\sum_{u \in J_k(r)} [\tau_{ru}]^\alpha \cdot [\eta_{ru}]^\beta}, & \text{si } s \in J_k(r) \\ 0, & \text{en otro caso} \end{cases}$$

- τ_{rs} es la feromona del arco a_{rs}
- η_{rs} es la inf. heurística del arco a_{rs}
- $J_k(r)$ es el conjunto de nodos alcanzables desde r no visitados aún por la hormiga k
- α y β son pesos que establecen un equilibrio entre la importancia de la información memorística y heurística

2. ALGORITMOS DE OPTIMIZACIÓN BASADOS EN COLONIAS DE HORMIGAS

- De las hormigas naturales a la OCH
- La hormiga artificial
- **Actualización de feromona**
- **El Sistema de Hormigas**
- **Ejemplo de aplicación del SH al TSP**
- **Otros algoritmos de OCH**
- **Estudio comparativo de la OCH en el TSP**
- **Algoritmos de OCH con búsqueda local**
- **La metaheurística OCH**

ACTUALIZACIÓN DE FEROMONA

1. Se usa una **retroalimentación positiva** para reforzar en el futuro los componentes de las buenas soluciones mediante un aporte adicional de feromona

Cuanto mejor sea la solución, más feromona se aporta

2. Se usa la **evaporación de feromona** para evitar un incremento ilimitado de los rastros de feromona y para permitir olvidar las malas decisiones tomadas

La evaporación es la misma para todos los rastros, eliminándose un porcentaje de su valor actual: $0 \leq \rho \leq 1$

Es un **mecanismo de evaporación más activo** que el natural, lo que evita la perduración de los rastros de feromona y, por tanto, el estancamiento en óptimos locales

ACTUALIZACIÓN DE FEROMONA (2)

- Un ejemplo de regla de actualización de feromona sería:

$$\tau_{rs}(t) = (1 - \rho) \cdot \tau_{rs}(t-1) + \sum_{k=1}^m \Delta \tau_{rs}^k$$

$$- \Delta \tau_{rs}^k = \begin{cases} \frac{1}{C(S_k)}, & \text{si la hormiga } k \text{ ha visitado el arco } a_{rs} \\ 0, & \text{en otro caso} \end{cases}$$

- $C(S_k)$ es el coste de la solución generada por la hormiga k , es decir, la longitud del circuito S_k
- m es el número de hormigas
- Los arcos visitados por hormigas en la iteración actual (**arcos prometedores**) reciben un aporte extra de feromona y los no visitados por ninguna hormiga (**poco prometedores**) la pierden

2. ALGORITMOS DE OPTIMIZACIÓN BASADOS EN COLONIAS DE HORMIGAS

- De las hormigas naturales a la OCH
- La hormiga artificial
- Actualización de feromona
- **El Sistema de Hormigas**
- **Ejemplo de aplicación del SH al TSP**
- **Otros algoritmos de OCH**
- **Estudio comparativo de la OCH en el TSP**
- **Algoritmos de OCH con búsqueda local**
- **La metaheurística OCH**

EL SISTEMA DE HORMIGAS

- El algoritmo anterior fue el primero de OCH que se propuso, denominado Sistema de Hormigas (*Ant System*) (SH)

M. Dorigo, V. Maniezzo, A. Coloni, *The Ant System: Optimization by a Colony of Cooperating Agents*. IEEE Trans. On Systems, Man and Cybernetics-Part B, Vol. 26, 1996, 1-13

- Obtenía buenos resultados en el TSP pero no eran lo suficientemente competitivos con respecto a los de los mejores algoritmos (state-of-the-art)
- Fue la base para el desarrollo posterior de la OCH, al proponerse muchas mejoras

EL SISTEMA DE HORMIGAS (2)

ALGORITMO SISTEMA DE HORMIGAS

- Inicialización de parámetros (p.e., asignación de la cantidad inicial a los rastros de feromona $\tau[i][j] \leftarrow \tau_0$)
- *Para* $It=1$ *hasta* Número_de_Iteraciones *hacer*
 1. *Para* $k=1$ *hasta* m (Número_de_Hormigas) *hacer*
 - $L[k][1] \leftarrow$ nodo inicial
- /* Construcción de soluciones por las hormigas */*
- 2. *Para* $i=2$ *hasta* Número_de_Nodos *hacer*
 - Para* $k=1$ *hasta* Número_de_Hormigas *hacer*
 - $L[k][i] \leftarrow$ Regla_transición ($L[k], \tau, \eta$)

EL SISTEMA DE HORMIGAS (3)

/ Actualización de feromona */*

3. Para $k=1$ hasta Número_de_Hormigas hacer
 Coste[k] \leftarrow C(L[k])
 Mejor_Actual \leftarrow Mejor (L[k])

4. Para $i=1$ hasta Número_de_Nodos hacer
 Para $j=1$ hasta Número_de_Nodos hacer
 Actualización_feromona($\tau[i][j]$, L, C(L))

5. Si C(Mejor_Actual) es mejor que C(Mejor_Global)
 entonces Mejor_Global \leftarrow Mejor_Actual

▪ *DEVOLVER Mejor_Global*

2. ALGORITMOS DE OPTIMIZACIÓN BASADOS EN COLONIAS DE HORMIGAS

- De las hormigas naturales a la OCH
- La hormiga artificial
- Actualización de feromona
- El Sistema de Hormigas
- **Ejemplo de aplicación del SH al TSP**
- **Otros algoritmos de OCH**
- **Estudio comparativo de la OCH en el TSP**
- **Algoritmos de OCH con búsqueda local**
- **La metaheurística OCH**

EJEMPLO DE APLICACIÓN DEL SH AL TSP

A modo de ejemplo, vamos a describir una iteración del SH sobre un caso sencillo del TSP de tamaño $n=6$ con las siguientes matrices de distancias D y heurística η :

$D =$

∞	1	$\sqrt{5}$	$\sqrt{5}$	2	$\sqrt{2}$
1	∞	$\sqrt{2}$	2	$\sqrt{5}$	$\sqrt{5}$
$\sqrt{5}$	$\sqrt{2}$	∞	$\sqrt{2}$	$\sqrt{5}$	3
$\sqrt{5}$	2	$\sqrt{2}$	∞	1	$\sqrt{5}$
2	$\sqrt{5}$	$\sqrt{5}$	1	∞	$\sqrt{2}$
$\sqrt{2}$	$\sqrt{5}$	3	$\sqrt{5}$	$\sqrt{2}$	∞

$\eta =$

-	1.000	0.447	0.447	.500	0.707
1.000	-	0.707	0.500	0.447	0.447
0.447	0.707	-	0.707	0.447	0.333
0.447	0.500	0.707	-	1.000	0.447
0.500	0.447	0.447	1.000	-	0.707
0.707	0.447	0.333	0.447	0.707	-

Fuente: J.M. Moreno-Vega, J.A. Moreno-Pérez, *Heurísticas en Optimización*, Consejería de Educación, Cultura y Deportes. Gobierno de Canarias, 1999.

EJEMPLO DE APLICACIÓN DEL SH AL TSP (2)

Suponiendo una población de $m=n=6$ hormigas, cada una partiendo de una ciudad distinta, y la matriz de feromona inicializada a $\tau_0=10$, el proceso constructivo de las tres primeras podría ser el siguiente:

Probabilidades de transición							Uniforme	Solución
	1	2	3	4	5	6		
1	-	0.322	0.144	0.144	0.161	0.227	0.000	(1 2 - - -)
2	-	-	0.336	0.237	0.212	0.212	0.031	(1 2 3 - -)
3	-	-	-	0.475	0.300	0.225	0.673	(1 2 3 5 -)
5	-	-	-	0.585	-	0.415	0.842	(1 2 3 5 6 -)
6	-	-	-	1.000			-	(1 2 3 5 6 4)

EJEMPLO DE APLICACIÓN DEL SH AL TSP (3)

Probabilidades de transición								
	1	2	3	4	5	6	Uniforme	Solución
2	0.322	-	0.227	0.161	0.144	0.144	0.279	(2 3 - - -)
3	0.231	-	-	0.365	0.231	0.172	0.671	(2 3 5 - -)
5	0.227	-	-	0.453	-	0.320	0.931	(2 3 5 6 -)
6	0.612	-	-	0.388	-	-	0.873	(2 3 5 6 4 -)
4	1.000	-	-	-	-	-	-	(2 3 5 6 4 1)

Probabilidades de transición								
	1	2	3	4	5	6	Uniforme	Solución
3	0.169	0.267	-	0.267	0.169	0.126	0.372	(3 2 - - -)
2	0.417	-	-	0.208	0.186	0.186	0.415	(3 2 1 - -)
1	-	-	-	0.267	0.309	0.434	0.321	(3 2 1 5 -)
5	-	-	-	0.585	-	0.415	0.474	(3 2 1 5 4 -)
4	-	-	-	-	-	1.000	-	(3 2 1 5 4 6)

EJEMPLO DE APLICACIÓN DEL SH AL TSP (4)

Una vez construidas las 6 soluciones, se aplica la actualización de feromona. Considerando que cada hormiga k aporta $100/C(S_k)$ en los arcos visitados, tendríamos los siguientes aportes:

Hormiga	$C(S_k)$	<i>Aporte</i>	Solución	Arcos afectados
1	10.53	9.49	(1 2 3 5 6 4)	{ $a_{12}, a_{23}, a_{35}, a_{56}, a_{64}, a_{41}$ }
2	10.53	9.49	(2 3 5 6 4 1)	{ $a_{23}, a_{35}, a_{56}, a_{64}, a_{41}, a_{12}$ }
3	9.05	11.04	(3 2 1 5 4 6)	{ $a_{32}, a_{21}, a_{15}, a_{54}, a_{46}, a_{63}$ }
4	11.12	8.99	(4 2 6 1 3 5)	{ $a_{42}, a_{26}, a_{61}, a_{13}, a_{35}, a_{54}$ }
5	10.88	9.19	(5 1 6 2 4 3)	{ $a_{51}, a_{16}, a_{62}, a_{24}, a_{43}, a_{35}$ }
6	9.47	10.55	(6 5 3 4 2 1)	{ $a_{65}, a_{53}, a_{34}, a_{42}, a_{21}, a_{16}$ }

APLICACIÓN DEL SH AL TSP (5)

Aplicando el mecanismo de actualización de feromona, se evapora la feromona y se realizan los aportes comentados:

$$\tau_{rs}(t) = (1 - \rho) \cdot \tau_{rs}(t-1) + \sum_{k=1}^m \Delta \tau_{rs}^k$$

Considerando $\rho=0.2$, la matriz de feromona quedaría:

$\tau =$

0.00	45.58	13.99	23.99	25.23	33.73
-	0.00	35.03	33.73	8.00	23.18
-	-	0.00	24.74	52.72	16.04
-	-	-	0.00	35.03	35.03
-	-	-	-	0.00	79.54
-	-	-	-	-	0.00

2. ALGORITMOS DE OPTIMIZACIÓN BASADOS EN COLONIAS DE HORMIGAS

- De las hormigas naturales a la OCH
- La hormiga artificial
- Actualización de feromona
- El Sistema de Hormigas
- Ejemplo de aplicación del SH al TSP
- **Otros algoritmos de OCH**
- **Estudio comparativo de la OCH en el TSP**
- **Algoritmos de OCH con búsqueda local**
- **La metaheurística OCH**

OTROS ALGORITMOS DE OCH: EL SISTEMA DE HORMIGAS ELITISTA

- En el mismo trabajo en el que propusieron el SH, Dorigo y sus colaboradores introdujeron el **Sistema de Hormigas Elitista** (Elitist Ant System) (SHE) para solucionar el problema de la lentitud de convergencia del SH
- La única diferencia entre ambos es la **regla de actualización** que aplica un **refuerzo adicional de los buenos arcos**:

$$\tau_{rs}(t) = (1 - \rho) \cdot \tau_{rs}(t-1) + \sum_{k=1}^m \Delta \tau_{rs}^k + e \cdot \Delta \tau_{rs}^{mejor_global}$$

donde e es el número de hormigas elitistas consideradas y *mejor_global* es la mejor solución obtenida hasta el momento.

OTROS ALGORITMOS DE OCH: EL SISTEMA DE COLONIA DE HORMIGAS

M. Dorigo, L.M. Gambardella, *Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem*. IEEE Trans. on Evolutionary Computation, Vol. 1, n° 1, 1997, 53-66

- El Sistema de Colonia de Hormigas (Ant Colony System) (SCH) extiende a su predecesor, el SH, en tres aspectos:
 - La **regla de transición** establece un equilibrio entre la exploración de nuevos arcos y la explotación de la información acumulada
 - Para la **actualización (global) de feromona** sólo se considera la hormiga que generó la mejor solución hasta ahora. Sólo se evapora feromona en los arcos que componen ésta
 - Se añade una nueva **actualización (local) de feromona** basada en que cada hormiga modifica automáticamente la feromona de cada arco que visita para diversificar la búsqueda

OTROS ALGORITMOS DE OCH: EL SISTEMA DE COLONIA DE HORMIGAS (2)

- La regla de transición de SCH (regla proporcional pseudo-aleatoria) es:

$$s = \begin{cases} \arg \max_{u \in J_k(r)} \{[\tau_{ru}]^\alpha \cdot [\eta_{ru}]^\beta\}, & \text{si } q \leq q_0 \\ S, & \text{en otro caso} \end{cases}$$

- s es la ciudad escogida por la hormiga k en su siguiente movimiento
- q es un uniforme en $[0,1]$
- $q_0 \in [0,1]$ es la probabilidad con la que se escoge determinísticamente el arco más prometedor (**explotación**)
- S es una ciudad aleatoria seleccionada según la regla de transición del SH (**exploración dirigida**)

OTROS ALGORITMOS DE OCH: EL SISTEMA DE COLONIA DE HORMIGAS (3)

- La actualización global de feromona se realiza mediante la operación:

$$\tau_{rs}(t) = (1 - \rho) \cdot \tau_{rs}(t-1) + \rho \cdot \frac{1}{C(S_{mejor-global})}$$

que sólo se aplica sobre los arcos $\tau_{rs} \in S_{mejor-global}$

- Sólo la hormiga que generó la mejor solución hasta el momento modifica los niveles de feromona
- El aporte de feromona es función de la calidad de esta solución ($C(S_{mejor-global})$)

OTROS ALGORITMOS DE OCH: EL SISTEMA DE COLONIA DE HORMIGAS (4)

- Cada vez que una hormiga recorre un arco, aplica la actualización local de feromona:

$$\tau_{rs}(t) = (1 - \varphi) \cdot \tau_{rs}(t-1) + \varphi \cdot \tau_0$$

- Con esta operación, la feromona asociada a un arco disminuye cada vez que lo visita una hormiga
- Los arcos ya visitados van siendo menos prometedores según los recorren más hormigas en la iteración actual, lo que favorece la exploración de arcos no visitados
- Así, las hormigas tienden a no converger a soluciones parecidas en la iteración actual

OTROS ALGORITMOS DE OCH: EL SISTEMA DE COLONIA DE HORMIGAS (5)

ALGORITMO SISTEMA DE COLONIA DE HORMIGAS

- Inicialización de parámetros (p.e., $\tau_{rs} \leftarrow \tau_0$)
- *Para It=1 hasta Número_de_Iteraciones hacer*
 1. *Para k=1 hasta Número_de_Hormigas hacer*
 $L[k][1] \leftarrow$ nodo inicial
 - /* Construcción de soluciones por las hormigas */*
 2. *Para i=2 hasta Número_de_Nodos hacer*
Para k=1 hasta Número_de_Hormigas hacer
 $L[k][i] \leftarrow$ Regla_transición ($L[k], \tau, \eta$)
Actualización_local_feromona ($L[k][i-1], L[k][i]$)

OTROS ALGORITMOS DE OCH: EL SISTEMA DE COLONIA DE HORMIGAS (6)

/ Actualización global de feromona */*

3. *Para k=1 hasta Número_de_Hormigas hacer*

Coste[k] ← C(L[k])

Mejor_Actual ← Mejor (L[k])

4. *Si C(Mejor_Actual) es mejor que C(Mejor_Global)
entonces Mejor_Global ← Mejor_Actual*

5. *Para i=1 hasta Número_de_Nodos hacer*

*Actualización_global_feromona (Mejor_Global[k][i],
Mejor_Global [k][i+1], C(Mejor_Solución))*

▪ *DEVOLVER Mejor_Global*

OTROS ALGORITMOS DE OCH: EL SISTEMA DE HORMIGAS MAX-MIN

T. Stützle, H.H. Hoos, MAX-MIN Ant System. Future Generation Computer Systems, Vol. 16, n° 8, 2000, 889-914

- El Sistema de Hormigas Max-Min (Max-Min Ant System) (SHMM) es una nueva extensión del SH con una mayor explotación de las mejores soluciones y un mecanismo adicional para evitar el estancamiento de la búsqueda
- Mantiene la regla de transición del SH y cambia:
 - El mecanismo de actualización es más agresivo al evaporar todos los rastros y aportar sólo en los de la mejor solución
 - Define unos topes mínimo y máximo para los rastros de feromona
 - Reinicializa la búsqueda cuando se estanca

OTROS ALGORITMOS DE OCH: EL SISTEMA DE HORMIGAS MAX-MIN (2)

- En la actualización de feromona sólo se considera la mejor solución, ya sea de la iteración actual o la global:

$$\tau_{rs}(t) = (1 - \rho) \cdot \tau_{rs}(t-1) + \Delta \tau_{rs}^{mejor}$$

donde *mejor* es $S_{mejor-global}$ o $S_{mejor-actual}$

- Se evaporan todos los rastros de feromona

OTROS ALGORITMOS DE OCH: EL SISTEMA DE HORMIGAS MAX-MIN (3)

- Se establecen unos límites en los valores posibles de feromona en los rastros:

$$\tau_{\min} \leq \tau_{rs}(t) \leq \tau_{\max}$$

Los límites se calculan de forma heurística

- Todos los rastros de feromona se inicializan al máximo valor τ_{\max} , en lugar de a un valor pequeño τ_0
- Así, al aplicar la regla de actualización los arcos de las buenas soluciones mantienen valores altos mientras que los de las malas reducen el valor de sus rastros
- También da lugar a una mayor exploración al comienzo de la ejecución del algoritmo

OTROS ALGORITMOS DE OCH: EL SISTEMA DE HORMIGAS MAX-MIN (4)

- Además, cuando se detecta que la búsqueda está estancada, se aplica una **reinicialización**, volviendo a poner todos los rastros de feromona a τ_{\max}
- La combinación de todo lo anterior:
 - establece un buen balance exploración-explotación y
 - reduce la posibilidad de estancamiento de la búsqueda
- El SHMM es hoy en día el algoritmo de OCH más competitivo

OTROS ALGORITMOS DE OCH: EL SISTEMA DE HORMIGAS MEJOR-PEOR

O. Cordón, F. Herrera, Ll. Moreno, *Integración de Conceptos de Computación Evolutiva en un Nuevo Modelo de Colonia de Hormigas*. Actas de la Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA'99), 1999, pp. 98-104.

O. Cordón, I. Fernández de Viana, F. Herrera, Ll. Moreno, *A New ACO Model Integrating Evolutionary Computation Concepts: The Best-Worst Ant System*. Actas de ANTS'2000 - From Ant Colonies to Artificial Ants: Second International Workshop on Ant Algorithms, 2000, pp. 22-29.

- El Sistema de Hormigas Mejor-Peor (Best-Worst Ant System) (SHMP) es otra extensión del SH basada en la incorporación de componentes de Computación Evolutiva para mejorar el equilibrio intensificación-diversificación
- Mantiene la regla de transición del SH y cambia:
 - El mecanismo de actualización es más explotativo al evaporar todos los rastros, reforzar positivamente sólo los de la mejor solución global y negativamente los de la peor solución actual
 - Aplica una mutación de los rastros de feromona para diversificar
 - Reinicializa la búsqueda cuando se estanca

OTROS ALGORITMOS DE OCH: EL SISTEMA DE HORMIGAS MEJOR-PEOR (2)

- La actualización de feromona de la mejor y la peor hormiga se realiza en dos pasos:

1. Se evaporan todos los rastros de feromona y se aporta en los de la mejor solución global:

$$\tau_{rs}(t) = (1 - \rho) \cdot \tau_{rs}(t-1) + \Delta \tau_{rs}^{mejor_global}$$

2. Se realiza una evaporación adicional de los rastros de feromona de la peor solución de la iteración actual que no estén contenidos en la mejor global:

$$\tau_{rs}(t) \leftarrow (1 - \rho) \cdot \tau_{rs}(t), \quad \forall a_{rs} \in S_{peor-actual} \quad y \quad a_{rs} \notin S_{mejor-global}$$

- El refuerzo negativo de $S_{peor-actual}$ hace que la regla de actualización tenga un comportamiento más intensificativo

OTROS ALGORITMOS DE OCH: EL SISTEMA DE HORMIGAS MEJOR-PEOR (3)

- El SHMP considera la búsqueda estancada si durante un número consecutivo de iteraciones (un porcentaje del total) no se consigue mejorar la mejor solución global obtenida
- En ese caso, se aplica la **reinicialización**, volviendo a poner todos los rastros de feromona a τ_0

OTROS ALGORITMOS DE OCH: EL SISTEMA DE HORMIGAS MEJOR-PEOR (4)

- Para conseguir diversidad en el proceso de búsqueda se mutan los valores de los rastros de feromona
- La **mutación** se aplica en cada rastro de feromona con probabilidad P_m :

$$\tau'_{rs}(t) = \tau_{rs}(t) + N(0, \tau_{umbral}) \quad ; \quad \tau_{umbral} = \frac{\sum_{a_{rs} \in S_{mejor-global}} \tau_{rs}(t)}{n}$$

A cada rastro mutado se le añade un valor normal de media 0 en $[-\tau_{umbral}, \tau_{umbral}]$. τ_{umbral} corresponde a la media de los rastros de feromona de $S_{mejor-global}$

OTROS ALGORITMOS DE OCH: EL SISTEMA DE HORMIGAS MEJOR-PEOR (5)

- Esta función de mutación se caracteriza por:
 1. La fuerza de la mutación aumenta con las iteraciones: primero, τ_{umbral} es cercano a τ_0 y la mutación es pequeña. Luego, según crecen los rastros de $S_{mejor-global}$ va siendo más grande
 2. Al reinicializar, vuelve a su rango inicial

- El SMPH consigue un buen balance diversificación-intensificación combinando:
 - La intensificación que introduce la regla de actualización de feromona con la mejor y la peor hormiga
 - La diversificación de la mutación de rastros de feromona y la reinicialización

2. ALGORITMOS DE OPTIMIZACIÓN BASADOS EN COLONIAS DE HORMIGAS

- De las hormigas naturales a la OCH
- La hormiga artificial
- Actualización de feromona
- El Sistema de Hormigas
- Ejemplo de aplicación del SH al TSP
- Otros algoritmos de OCH
- **Estudio comparativo de la OCH en el TSP**
- **Algoritmos de OCH con búsqueda local**
- **La metaheurística OCH**

ESTUDIO COMPARATIVO DE LA OCH EN EL TSP

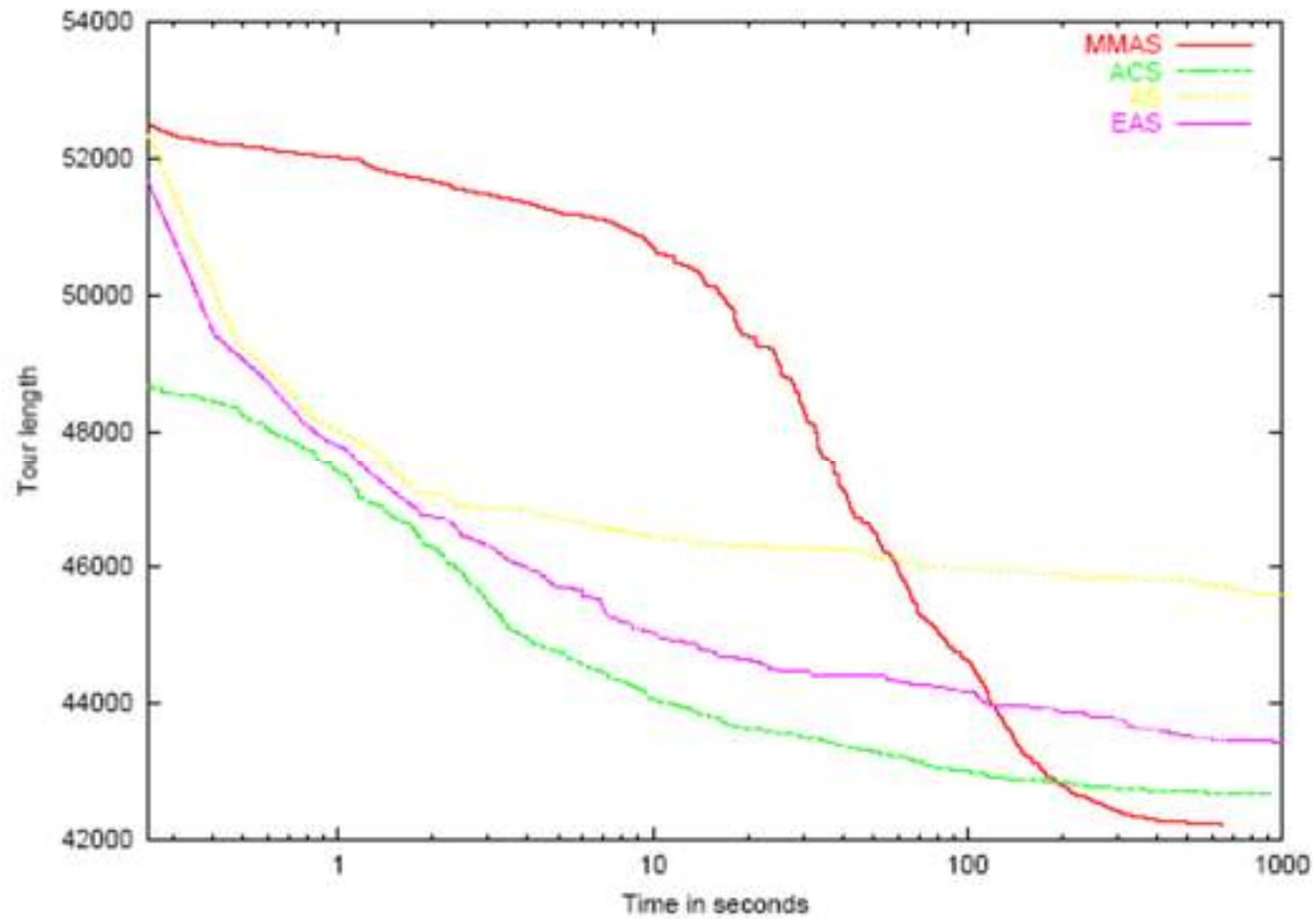
Ejemplo de valores de parámetros para la OCH en el TSP:

- Número de hormigas: $m = 10$
- Nivel inicial de feromona: $\tau_0 = 1/(n \cdot C(S_{\text{greedy}}))$
(S_{greedy} = solución obtenida por un algoritmo greedy)
- Regla de transición: $\alpha=1, \beta=2, q_0=0.98$ (SCH)
- Regla de actualización de feromona: $\rho=0.1$
- Regla de actualización local (SCH): $\varphi=0.1$
- Criterio de parada: 10000·n iteraciones para TSP simétrico y 20000·n para el TSP asimétrico

ESTUDIO COMPARATIVO DE LA OCH EN EL TSP (2)

Instance	<i>opt</i>	<i>MMAS</i>	ACS	<i>AS_e</i>	AS
eil51.tsp	426	427.6	428.1	428.3	437.3
kroA100.tsp	21282	21320.3	21420.0	21522.83	22471.4
d198.tsp	15780	15972.5	16054.0	16205.0	16705.6
lin318.tsp	42029	42220.2	42570.0	43422.8	45535.2
ry48p.atsp	14422	14553.2	14565.4	14685.2	15296.4
ft70.atsp	38673	39040.2	39099.0	39261.8	39596.3
kro124p.atsp	36230	36773.5	36857.0	37510.2	38733.1
ftv170.atsp	2755	2828.8	2826.5	2952.4	3154.5

ESTUDIO COMPARATIVO DE LA OCH EN EL TSP (3)



2. ALGORITMOS DE OPTIMIZACIÓN BASADOS EN COLONIAS DE HORMIGAS

- De las hormigas naturales a la OCH
- La hormiga artificial
- Actualización de feromona
- El Sistema de Hormigas
- Ejemplo de aplicación del SH al TSP
- Otros algoritmos de OCH
- Estudio comparativo de la OCH en el TSP
- **Algoritmos de OCH con búsqueda local**
- **La metaheurística OCH**

ALGORITMOS DE OCH CON BÚSQUEDA LOCAL

- Existe la posibilidad de hibridar los algoritmos de OCH con técnicas de búsqueda local para mejorar su eficacia
- La hibridación consiste en aplicar una búsqueda local sobre las soluciones construidas por todas las hormigas en cada iteración antes de actualizar la feromona
- El aumento en la eficacia reduce la eficiencia. Es habitual emplear la búsqueda local junto con las llamadas Listas de Candidatos, que consisten en estudiar sólo las ciudades candidatas más prometedoras en cada paso de la hormiga
- Al usar búsqueda local se obtienen muy buenos resultados con muchas menos hormigas que en la OCH básica (lo que implica mayor rapidez al hacer menos búsquedas locales)

ALGORITMOS DE OCH CON BÚSQUEDA LOCAL (2)

- Además, se suelen considerar optimizaciones en el cálculo de la función objetivo dentro de la búsqueda local para acelerar la ejecución
- Por ejemplo, las búsquedas locales habitualmente usadas en el TSP son la 2-opt y la 3-opt (que intercambian el orden de 2 y 3 dos ciudades, es decir, de 4 y 8 arcos, respectivamente)
- Así, en la 2-opt, la diferencia de costes entre la solución actual S y la vecina S' se puede calcular a partir de los cuatro arcos cambiados como:

$$\begin{aligned} F(S') = & F(S) - D(S[i-1],S[i]) - D(S[i],S[i+1]) \\ & - D(S[j-1],S[j]) - D(S[j],S[j+1]) \\ & + D(S[i-1],S[j]) + D(S[j],S[i+1]) \\ & + D(S[j-1],S[i]) + D(S[i],S[j+1]) \end{aligned}$$

ALGORITMOS DE OCH CON BÚSQUEDA LOCAL (3)

- El algoritmo básico de OCH con búsqueda local es:

Mientras (no se dé la condición de parada) *hacer*

- Construcción probabilística de las soluciones preliminares mediante la colonia de hormigas
 - Refinamiento de dichas soluciones mediante la búsqueda local
 - Actualización global de feromona
- De este modo, Los algoritmos de OCH con búsqueda local son algoritmos híbridos con una generación probabilística de soluciones iniciales para una técnica clásica de búsqueda local

ALGORITMOS DE OCH CON BÚSQUEDA LOCAL (4)

- Así, pueden encuadrarse en los algoritmos de **Búsqueda Local Múltiarranque**, aunque éstos generan aleatoriamente las soluciones iniciales sin usar información heurística
- También son similares a los algoritmos **GRASP**, aunque con otra filosofía de generación probabilística de soluciones
- La principal diferencia con los dos algoritmos anteriores es el mecanismo de cooperación global de las soluciones generadas que hace que las ejecuciones no sean independientes entre sí

ALGORITMOS DE OCH CON BÚSQUEDA LOCAL (5)

EJEMPLO: SCH CON BÚSQUEDA LOCAL PARA EL TSP

Caso TSP	SCH (media)	SCH (mejor)	Otros (media)	Otros (mejor)
d198	15781.7	15780	15780	15780
lin318	42029	42029	42029	42029
att532	27718.2	27693	27693.7	27686
rat783	8837.9	8818	8807.3	8806

Fuente: Bonabeau, Dorigo, Theraulaz, *Swarm Intelligence*, Oxford, 1999

- Nuevos valores de parámetros: $q_0 = 0.9$ y $cl=20$
- Algoritmo de búsqueda local: 3-opt (búsqueda local del mejor)
- Otros = mejores resultados obtenidos en el Primer Concurso Internacional de Optimización Evolutiva (ICEO)
- Resultados medios de 10 ejecuciones

ALGORITMOS DE OCH CON BÚSQUEDA LOCAL (6)

EJEMPLO 2: COMPARATIVA DE OCH CON BL PARA EL TSP

<i>Parámetro</i>	<i>Valor</i>
Número de hormigas	m = 15
Número de ejecuciones	10
Tiempo máximo	$N_{it} = 900$ a 3600
Regla de transición	$\alpha = 1 ; \beta = 2$
Cte. evaporación global	$\rho = 0.2$
<i>SHE</i>	
Hormigas elitistas	m (15)
<i>SCH</i>	
Cte. evaporación local	$q_0 = 0.8 \quad \varphi = 0.2$
<i>SMPH</i>	
Prob. de mutación	$P_m = 0.3, 0.25, 0.2$
Iteraciones sin mejora	0.3, 0.2, 0.3
<i>Búsqueda local</i>	
Número de vecinos	40
Regla de selección	primero mejor
Algoritmo	2-opt

ALGORITMOS DE OCH CON BÚSQUEDA LOCAL (7)

Problema	Eil51 ($C_{opt} = 426$, $n = 51$)					Berlín52 ($C_{opt} = 7542$), $n = 52$)				
Modelo	<i>Mejor</i>	<i>Media</i>	<i>Dev.</i>	<i>Error</i>	<i>#R</i>	<i>Mejor</i>	<i>Media</i>	<i>Dev.</i>	<i>Error</i>	<i>#R</i>
SHE	426	426.2	0.44	0.04	-	7542	7542	0	0	-
SCH	426	426.7	0.48	0.48	-	7542	7542	0	0	-
SHE + Re	426	426	426.6	0.26	0	7542	7542	0	0	0
SCH + Re	426	426.5	0.53	0.12	2.4	7542	7542	0	0	0
SMPH	426	426	0	0	0	7542	7542	0	0	0
Problema	Brazil58 ($C_{opt} = 25395$, $n = 58$)					Kroal100 ($C_{opt} = 21282$), $n = 100$)				
Modelo	<i>Mejor</i>	<i>Media</i>	<i>Dev.</i>	<i>Error</i>	<i>#R</i>	<i>Mejor</i>	<i>Media</i>	<i>Dev.</i>	<i>Error</i>	<i>#R</i>
SHE	25395	25395	0	0	-	21282	21282	0	0	-
SCH	25395	25395	0	0	-	21282	21282	0	0	-
SHE + Re	25395	25395	0	0	0	21282	21282	0	0	0
SCH + Re	25395	25395	0	0	0	21282	21282	0	0	0
SMPH	25395	25395	0	0	0	21282	21282	0	0	0
Problema	Gr120 ($C_{opt} = 6942$, $n = 120$)					D198 ($C_{opt} = 15780$), $n = 198$)				
Modelo	<i>Mejor</i>	<i>Media</i>	<i>Dev.</i>	<i>Error</i>	<i>#R</i>	<i>Mejor</i>	<i>Media</i>	<i>Dev.</i>	<i>Error</i>	<i>#R</i>
SHE	6942	6942	0	0	-	15780	15781	0.70	$\cong 0$	-
SCH	6942	6946.1	5.49	0.06	-	15780	15784.9	5.67	0.03	-
SHE + Re	6942	6942	0	0	0	15780	15781	0.70	$\cong 0$	0
SCH + Re	6942	6943.8	3.79	0.03	1.1	15780	15782.9	4.31	0.02	2.3
SMPH	6942	6942	0	0	0.5	15780	15780.4	0.51	$\cong 0$	1.7

ALGORITMOS DE OCH CON BÚSQUEDA LOCAL (8)

Problema	Lin318 ($C_{opt} = 42029$, $n = 318$)					Pcb442 ($C_{opt} = 50788$), $n = 442$)				
Modelo	<i>Mejor</i>	<i>Media</i>	<i>Dev.</i>	<i>Error</i>	<i>#R</i>	<i>Mejor</i>	<i>Media</i>	<i>Dev.</i>	<i>Error</i>	<i>#R</i>
SHE	42029	42123.8	67.28	0.22	-	51015	51104.6	82.44	0.63	-
SCH	42029	42230	148.48	0.48	-	50919	51048	75.29	0.53	-
SHE + Re	42029	42105	53.79	0.18	3	50980	51066.8	71.52	0.56	3.2
SCH + Re	42029	42182.4	118.12	0.36	5	50860	51147.5	173.11	0.72	8
SMPH	42029	42084.1	98.60	0.13	2	50788	50888.8	87.83	0.21	6.9
Problema	Att532 ($C_{opt} = 27686$, $n = 532$)					Rat783 ($C_{opt} = 8806$), $n = 783$)				
Modelo	<i>Mejor</i>	<i>Media</i>	<i>Dev.</i>	<i>Error</i>	<i>#R</i>	<i>Mejor</i>	<i>Media</i>	<i>Dev.</i>	<i>Error</i>	<i>#R</i>
SHE	27745	27823	70.67	0.49	-	8860	8878.6	17.06	0.81	-
SCH	27705	27810.3	64.44	0.45	-	8857	8892.7	20.93	0.97	-
SHE + Re	27793	27825.6	41.59	0.50	3.6	8843	8878.4	32.25	0.81	3.8
SCH + Re	27745	27835	57.56	0.54	7	8875	8899.5	22.33	1.05	7.6
SMPH	27686	27711	12.16	0.09	7.7	8816	8833.4	15.37	0.31	9.5
Problema	U1060 ($C_{opt} = 224094$, $n = 1060$)					D1291 ($C_{opt} = 50801$), $n = 1291$)				
Modelo	<i>Mejor</i>	<i>Media</i>	<i>Dev.</i>	<i>Error</i>	<i>#R</i>	<i>Mejor</i>	<i>Media</i>	<i>Dev.</i>	<i>Error</i>	<i>#R</i>
SHE	231644	232145.8	297.91	3.46	-	51210	51347.2	138.15	1.06	-
SCH	225675	226387.8	668.9	1.01	-	51901	51953.6	60.01	2.21	-
SHE + Re	230360	230806.4	390.8	2.90	2.9	51073	51236.7	119.56	0.85	4.5
SCH + Re	225243	226501	1059.57	1.06	2	51828	51986.8	105.80	2.28	5
SMPH	225310	225721.1	476.88	0.72	5.3	50890	50986.3	74.87	0.36	17.2

ALGORITMOS DE OCH CON BÚSQUEDA LOCAL (9)

- Por otro lado, el MMAS con la búsqueda local 3-opt:
 - obtiene el óptimo para los casos de hasta 500 ciudades en pocos segundos/minutos
 - en los casos grandes (se ha probado en instancias de hasta 3038 ciudades), se obtienen soluciones con un error de un 0.25% con respecto al óptimo en tiempo razonable (menos de 1 hora)
- El rendimiento es cercano al de los algoritmos state-of-the-art para el TSP

2. ALGORITMOS DE OPTIMIZACIÓN BASADOS EN COLONIAS DE HORMIGAS

- De las hormigas naturales a la OCH
- La hormiga artificial
- Actualización de feromona
- El Sistema de Hormigas
- Ejemplo de aplicación del SH al TSP
- Otros algoritmos de OCH
- Estudio comparativo de la OCH en el TSP
- Algoritmos de OCH con búsqueda local
- **La metaheurística OCH**

LA METAHEURÍSTICA DE OPTIMIZACIÓN BASADA EN COLONIAS DE HORMIGAS

- A posteriori, cuando la mayoría de los algoritmos de OCH estaban ya propuestos, Dorigo y Di Caro propusieron un marco de trabajo general que define la Metaheurística OCH:

M. Dorigo, G. Di Caro, Ant Algorithms for Discrete Optimization. Artificial Life, Vol. 5, nº 2, 1999, 137-172

- Instanciando el algoritmo general con componentes concretas (reglas de transición, actualización, ...), se pueden obtener distintas variantes de algoritmos OCH
- Los distintos algoritmos se pueden implementar de forma **secuencial** o **paralela** para ser aplicados respectivamente a **problemas estáticos** o **dinámicos**

LA METAHEURÍSTICA DE OPTIMIZACIÓN BASADA EN COLONIAS DE HORMIGAS (2)

- En general, para aplicar la OCH a un problema, es necesario que pueda ser representado en forma de grafo con pesos
- Cada arco a_{rs} del grafo contendrá dos tipos de información:
 - Información heurística: **preferencia heurística** del arco, η_{rs} , dependiente del caso concreto del problema. **Las hormigas no la modifican durante la ejecución del algoritmo, aunque puede variar a lo largo del tiempo en problemas dinámicos**
 - Información memorística: medida de la **“deseabilidad”** del arco, τ_{rs} , representada por la cantidad de **feromona** depositada en él y **modificada durante el algoritmo**

LA METAHEURÍSTICA DE OPTIMIZACIÓN BASADA EN COLONIAS DE HORMIGAS (3)

ESTRUCTURA GENERAL DE UN ALGORITMO DE OCH

1. Inicialización de parámetros (p.e., asignación de una cantidad inicial de feromona τ_0 a cada arco)
2. *Planificación_de_actividades*
 - generación_y_activación_de_hormigas
 - evaporación_de_feromona
 - acciones_del_demonio (**OPCIONAL**)

El constructor **Planificación_de_actividades** define la planificación de las tres tareas, ya sea secuencialmente (problemas estáticos) o en paralelo (problemas dinámicos)

LA METAHEURÍSTICA DE OPTIMIZACIÓN BASADA EN COLONIAS DE HORMIGAS (4)

ESTRUCTURA GENERAL DE UN ALGORITMO DE OCH

1. Inicialización de parámetros (p.e., asignación de una cantidad inicial de feromona τ_0 a cada arco)
2. *Planificación_de_actividades*
 - generación_y_activación_de_hormigas
 - evaporación_de_feromona
 - acciones_del_demonio (**OPCIONAL**)

El **demonio** es un elemento opcional que realiza una serie de funciones adicionales centralizadas sin contrapunto natural (aplicar búsqueda local, ...)

3. APLICACIONES DE LA OCH

- Aplicación de la OCH al Enrutamiento en Redes de Telecomunicaciones: AntNet
- Planificación de rutas para el transporte de mercancías: Ant@ptima
- Aplicación de la OCH al Problema de Asignación Cuadrática
- Otras Aplicaciones

APLICACIÓN AL ENRUTAMIENTO DE PAQUETES EN REDES DE TELECOMUNICACIONES: ANTNET

G. Di Caro, M. Dorigo, *AntNet: Distributed Stimergetic Control for Communication Networks*. Journal of Artificial Intelligence Research, Vol. 9, 1998, 317-365

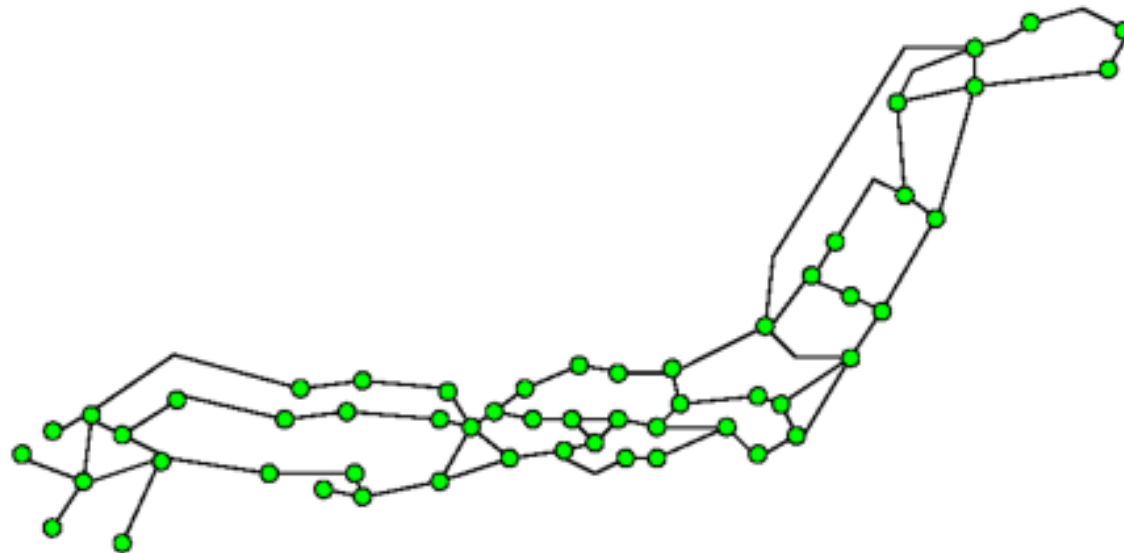
- El **enrutamiento** es la tarea consistente en determinar el camino que seguirán los paquetes en una red de telecomunicaciones cuando llegan a un nodo para alcanzar su nodo destino de la forma más rápida posible
- **AntNet** es un algoritmo de hormigas adaptativo y distribuido para enrutamiento de paquetes en redes

APLICACIÓN AL ENRUTAMIENTO DE PAQUETES EN REDES DE TELECOMUNICACIONES: ANTNET (2)

- Las redes se modelan mediante un grafo dirigido con N nodos de procesamiento/destino
- Los arcos del grafo están caracterizados por el ancho de banda (bits/segundo) y el retardo de transmisión (segundos) del enlace físico
- Se consideran dos tipos de paquetes: enrutamiento y datos. Los de enrutamiento tienen una mayor prioridad

APLICACIÓN AL ENRUTAMIENTO DE PAQUETES EN REDES DE TELECOMUNICACIONES: ANTNET (3)

- Una de las redes consideradas, la NNTnet de Japón:



APLICACIÓN AL ENRUTAMIENTO DE PAQUETES EN REDES DE TELECOMUNICACIONES: ANTNET (4)

- Las hormigas (paquetes de enrutamiento) se lanzan asíncronamente a la red hacia nodos destino aleatorios
- Cada hormiga busca un camino de coste mínimo entre su nodo de partida y su nodo destino
- Se mueve paso a paso por la red (grafo). En cada nodo intermedio, lanza la regla de transición para decidir a qué nodo se dirige
- Para ello, considera la feromona (almacenado en los nodos y función del tiempo consumido en el envío de los paquetes) y la preferencia heurística (dependiente del estado actual) de los enlaces de la red

APLICACIÓN AL ENRUTAMIENTO DE PAQUETES EN REDES DE TELECOMUNICACIONES: ANTNET (5)

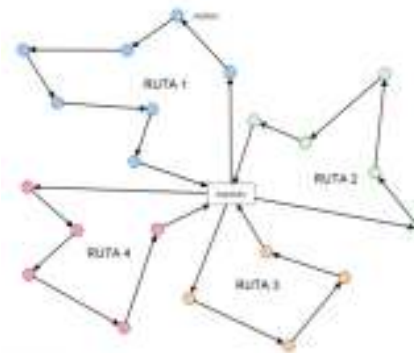
- El estado de la red varía con el tiempo (caída de enlaces, congestión, ...). El algoritmo maneja adecuadamente este hecho gracias a su naturaleza distribuida y su capacidad de adaptación
- Cuando la hormiga llega al nodo destino, vuelve sobre sus pasos y actualiza las tablas de enrutamiento de los nodos de acuerdo al tiempo que tardó en hacer el camino (refuerzo positivo o negativo)
- En un estudio experimental en el que se comparó su funcionamiento contra el de seis algoritmos de enrutamiento diferentes, AntNet proporcionó el mejor comportamiento

Planificación de Rutas para Transporte de Mercancías



<http://www.antoptima.com>

- Hoy en día es difícil encontrar empresas que gestionen las **operaciones de logística** sin la ayuda del ordenador
- El problema típico es diseñar las rutas más adecuadas de transporte/recogida de productos entre un almacén central y unos destinos dispersos geográficamente



- Su resolución de forma adecuada puede suponer **ahorros muy significativos para la empresa**

Planificación de Rutas para Transporte de Mercancías



<http://www.antoptima.com>

- Esta tarea se lleva a cabo empleando una flota de vehículos pertenecientes o no a la empresa



- Un **sistema de planificación de vehículos** debe proporcionar un conjunto de **rutas de reparto** a los conductores
- Las mercancías deben ser entregadas cuándo y donde se requieran, con el mínimo coste posible y verificando todas las restricciones legales y políticas de la empresa
- Los **algoritmos de hormigas (AntRoute)** son una herramienta muy potente para la planificación de rutas

Planificación de Rutas para Transporte de Mercancías



Ant@ptima

<http://www.antoptima.com>

-
- **AntRoute** planifica diariamente las rutas de reparto desde el almacén central de **Migros**, una gran cadena suiza con 600 supermercados, localizado en **Suhr (AG)**, a toda **Suiza**
 - Migros dispone de una **flota de entre 150 y 200 vehículos** con tres tamaños: camiones (capacidad de 17 palés), trailers (35 palés) y unidades tractoras (33 palés)
 - Esto provoca restricciones de acceso a los almacenes de los supermercados, restricciones de uso de ciertas carreteras, ...
 - Los repartos tienen de realizarse a horas específicas, todos ellos en un solo día (**productos perecederos**) y el último tiene que hacerse lo más lejos posible del almacén (**servicios extra**)

Planificación de Rutas para Transporte de Mercancías



Ant@ptima

<http://www.antoptima.com>

- Por ejemplo, en un reparto de 52000 palés a 6800 clientes en un periodo de 20 días, AntRoute obtuvo el diseño diario de rutas en menos de 5 minutos en un PC estándar
- Los expertos de la empresa necesitaron tres horas...
- Las soluciones de AntRoute fueron de mucha mejor calidad en cuanto al número de rutas necesario, la distancia total recorrida y al aprovechamiento de los vehículos:

	Human Planner	AR-RegTW	AR-Free
Total number of tours	2056	1807	1614
Total km	147271	143983	126258
Average truck loading	76.91%	87.35%	97.81%

OTRAS APLICACIONES DE LA OCH

- Los algoritmos de OCH se han aplicado a otros muchos problemas:
 - Secuenciación de Tareas
 - Coloreo de Grafos
 - Enrutamiento de Vehículos
 - Ordenación Secuencial
 - “Pooling” de vehículos
 - Líneas de producción de coches
 - Problemas de Agrupamiento (Clustering)
 - Aprendizaje de Reglas Clásicas y Difusas
 - Bioinformática: plegado de proteínas 2D
 - etc.

4. COMENTARIOS FINALES



- La OCH es una metaheurística bioinspirada que permite diseñar algoritmos:
 - Sencillos de entender
 - Rápidos
 - Con buen rendimiento

para problemas de optimización que se puedan representar en forma de grafo con pesos

4. COMENTARIOS FINALES (2):

ASPECTOS PRÁCTICOS DEL DISEÑO DE UN ALGORITMO DE OCH PARA UN PROBLEMA

DEFINICIÓN DE LOS RASTROS DE FEROMONA

- Tarea totalmente dependiente del problema a resolver
- Depende directamente de la representación de soluciones considerada y de su significado
- Una mala elección provoca un mal rendimiento del algoritmo de OCH

4. COMENTARIOS FINALES (3):

ASPECTOS PRÁCTICOS DEL DISEÑO DE UN ALGORITMO DE OCH PARA UN PROBLEMA

BALANCE EXPLORACIÓN-EXPLOTACIÓN

- El algoritmo de OCH tendrá el mejor rendimiento si se consigue un balance adecuado para el problema
- En OCH, son los rastros de feromona los que establecen ese equilibrio definiendo una distribución de probabilidad para muestrear el espacio de búsqueda
- Componentes de OCH para intensificación:
 - Actualización de feromona por la calidad de las soluciones
 - Estrategias “elitistas”
 - Regla de transición pseudo-aleatoria del SCH

4. COMENTARIOS FINALES (4):

ASPECTOS PRÁCTICOS DEL DISEÑO DE UN ALGORITMO DE OCH PARA UN PROBLEMA

- Componentes de OCH para diversificación:
 - Construcción aleatoria de soluciones por las hormigas
 - Mecanismos para evitar el estancamiento de la búsqueda:
 - Regla de actualización local del SCH
 - Límites a los valores de feromona en el SHMM/MMAS
 - Mutación de los rastros de feromona en el SHMP
 - Reinicialización de los rastros de feromona

- También es importante para este balance el papel de los pesos α y β de la regla de transición

4. COMENTARIOS FINALES (5):

ASPECTOS PRÁCTICOS DEL DISEÑO DE UN ALGORITMO DE OCH PARA UN PROBLEMA

OCH y BÚSQUEDA LOCAL

- En muchos problemas NP-duros, la inclusión de optimizadores locales en el algoritmo de OCH consigue el mejor rendimiento
- Los dos enfoques son complementarios, convirtiendo la OCH en un algoritmo de búsqueda multiarranque
- Sin embargo, la OCH puede ser muy efectiva en aquellos problemas en los que no es sencillo aplicar una búsqueda local

4. COMENTARIOS FINALES (6):

ASPECTOS PRÁCTICOS DEL DISEÑO DE UN ALGORITMO DE OCH PARA UN PROBLEMA

USO DE LA INFORMACIÓN HEURÍSTICA

- Ayuda a explotar el conocimiento específico del problema a resolver
- Distinción entre información heurística en problemas estáticos y dinámicos. La primera es más efectiva
- En algunos problemas, no es posible usarla y el algoritmo de OCH puede obtener buenos resultados guiado sólo por los rastros de feromona (Ejemplo: QAP)
- Su importancia disminuya cuando se usa OCH con búsqueda local

4. COMENTARIOS FINALES (7):

ASPECTOS PRÁCTICOS DEL DISEÑO DE UN ALGORITMO DE OCH PARA UN PROBLEMA

NÚMERO DE HORMIGAS EN LA COLONIA

- ¿Por qué hay que usar una colonia de hormigas y no una sola?
- Hay un algoritmo que maneja una única hormiga: el Sistema de Hormigas Rápido (Fast Ant System)
- Experimentalmente se ha demostrado que el uso de varias hormigas permite obtener mejores resultados
- Aún así, los algoritmos de OCH son robustos en cuanto al número de hormigas considerado, es decir, obtiene buenos resultados independientemente de este valor

5. BIBLIOGRAFÍA Y RECURSOS

BIBLIOGRAFÍA BÁSICA

- E. BONABEAU, M. DORIGO, G. THERAULAZ, **Swarm Intelligence. From Natural to Artificial Systems**, Oxford University Press, 1999.



- M. DORIGO, T. STÜTZLE, **Ant Colony Optimization**, The MIT Press, 2004.

5. BIBLIOGRAFÍA Y RECURSOS (2)

BIBLIOGRAFÍA BÁSICA

- M. DORIGO, G. DI CARO, *Ant Algorithms for Discrete Optimization*, **Artificial Life** 5:2, 1999, pp. 137-172.
- M. DORIGO, T. STÜTZLE, *The Ant Colony Optimization Metaheuristic: Algorithms, Applications and Advances*, En: F. Glover, T. Kochenberger (Eds.), **Handbook of Metaheuristics**, pp. 251-285, 2003, Kluwer.
- O. CORDÓN, F. HERRERA, T. STÜTZLE, *A Review on the Ant Colony Optimization Metaheuristic: Basis, Models and New Trends*, *Mathware & Soft Computing* 9:2-3, 2002, pp. 141-175.

<http://iridia.ulb.ac.be/~mdorigo/ACO/ACO.html>

5. BIBLIOGRAFÍA Y RECURSOS (3)

BIBLIOGRAFÍA DE APOYO

- M. DORIGO, V. MANIEZZO, A. COLORNI, *The Ant System: Optimization by a Colony of Cooperating Agents*, **IEEE Transactions on Systems, Man, and Cybernetics** 26:1, 1996, pp. 29-41.
- M. DORIGO, L.M. GAMBARDELLA, *Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem*, **IEEE Transactions on Evolutionary Computation** 1:1, 1997, pp. 53-66.

Códigos disponibles de SH, SH MAX-MIN, SH Best Worst y SCH
<http://sci2s.ugr.es/docencia/bioinformatica/>

BIOINFORMÁTICA

2013 - 2014



PARTE I. INTRODUCCIÓN

- Tema 1. Computación Basada en Modelos Naturales

PARTE II. MODELOS BASADOS EN ADAPTACIÓN SOCIAL (Swarm Intelligence)

- Tema 2. Introducción a los Modelos Basados en Adaptación Social
- Tema 3. Optimización Basada en Colonias de Hormigas
- **Tema 4. Optimización Basada en Nubes de Partículas (Particle Swarm)**

PARTE III. COMPUTACIÓN EVOLUTIVA

- Tema 5. Introducción a la Computación Evolutiva
- Tema 6. Algoritmos Genéticos I. Conceptos Básicos
- Tema 7. Algoritmos Genéticos II. Diversidad y Convergencia
- Tema 8. Algoritmos Genéticos III. Múltiples Soluciones en Problemas Multimodales
- Tema 9. Estrategias de Evolución y Programación Evolutiva
- Tema 10. Algoritmos Basados en Evolución Diferencial (Differential Evolution – DE)
- Tema 11. Modelos de Evolución Basados en Estimación de Distribuciones (EDA)
- Tema 12. Algoritmos Evolutivos para Problemas Multiobjetivo
- Tema 13. Programación Genética
- Tema 14. Modelos Evolutivos de Aprendizaje

PARTE IV. OTROS MODELOS DE COMPUTACIÓN BIOINSPIRADOS

- Tema 15. Sistemas Inmunológicos Artificiales
- Tema 16. Otros Modelos de Computación Natural/Bioinspirados