

Bloque I: Optimización Combinatoria

Tema 3-Parte II: Metaheurísticos. Ant Colony Optimization (ACO)

Metaheurísticos

Entre los metaheurísticos más aplicados ^{1 2}, se encuentran:

- Simulated Annealing, (templado simulado).
- Tabu Search.
- Variable Neighborhood Search, (búsqueda por entornos variables).
- Genetic algorithms, (algoritmos genéticos).
- Artificial neural networks, (redes neuronales artificiales).
- **Swarm intelligence**, (inteligencia colectiva).

¹A. Gogna, A. Tayal. *Metaheuristics: review and application*. Journal of Experimental & Theoretical Artificial intelligence (2013) 25, 503-526.

²M. Gendreau, J-Y. Potvin (Eds.). *Handbook of Metaheuristics (Second ed.)*. International Series in Operations Research and Management Science, Springer (2010).

Insectos sociales

Las colonias de insectos sociales son:

- Flexibles: La colonia responde a perturbaciones internas o a cambios externos
- Robustos: Las tareas se llevan a cabo aunque fallen los individuos
- Descentralizadas: No existe un líder o control central en la colonia. No hay un supervisor.
- Auto-organizado: los caminos hacia las soluciones emergentes, no están predefinidos.

Lo que podemos aprender de los insectos sociales lo podemos aplicar al campo del diseño de Sistemas Inteligentes.

Swarm Intelligence: *Inteligencia colectiva emergente de un grupo de agentes simples.*

Swarm Intelligence

[Bonabeau, Dorigo, y Theraulaz]: Algoritmos ó mecanismos distribuídos de resolución de problemas inspirados en el comportamiento colectivo de colonias de insectos sociales, u otras sociedades de animales.

- Es una aproximación desde abajo hacia arriba para controlar y optimizar.
- Usa técnicas robustas, descentralizadas y auto-organizativas.
- Inspirado en cómo actúan los insectos sociales.

Colonias de insectos: comportamiento

- Las colonias de insectos llevan a cabo actuaciones de nivel complejo de forma inteligente, flexible y fiable, actuaciones que no serían factibles si tuviesen que ser realizadas por un insecto de forma individual, (individualmente, los insectos son no inteligentes, no fiables, y simples).
- Los insectos siguen reglas simples, y utilizan comunicación local simple
- La estructura global (nido) emerge desde las acciones de los insectos (las cuales son no fiables atendidas individualmente).

Swarm Intelligence: Metaheurísticas

Dentro de la inteligencia colectiva tenemos:

- Ant Colony Optimization (ACO): Optimización mediante colonias de hormigas. Resuelven problemas que se pueden representar como rutas/caminos entre nodos de un grafo



Swarm Intelligence: Metaheurísticas

Dentro de la inteligencia colectiva tenemos:

- Ant Colony Optimization (ACO): Optimización mediante colonias de hormigas. Resuelven problemas que se pueden representar como rutas/caminos entre nodos de un grafo



- Particle Swarm Optimization (PSO): Optimización basada en enjambres de partículas. Resuelven problemas de optimización de variables continuas



Ant Colony Optimization (ACO)

Ant Colony Optimization (ACO)

Introducido por Dorigo¹ en su tesis, se basa en el mecanismo por el que las hormigas (que son ciegas, y con un cerebro muy básico) encuentran el camino **más corto** hasta la fuente de comida, regresando nuevamente al hormiguero.

ACO: Hormigas naturales y artificiales

- Los algoritmos ACO reproducen el comportamiento de las **hormigas reales** en una **colonia artificial de hormigas** para resolver problemas complejos de camino mínimo.
- La aparición de buenas soluciones es una propiedad emergente de la interacción cooperativa entre las hormigas.

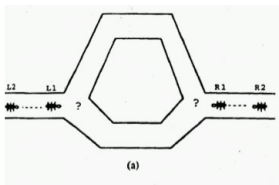
¹M. Dorigo (1992). *Optimization, Learning, and Natural Algorithms (in Italian)*. PhD thesis Politecnico di Milano, Milan.

Ant Colony Optimization (ACO)

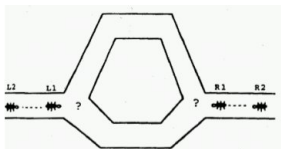
ACO: Hormigas naturales

- El mecanismo de comunicación de las hormigas se basa en que depositan una sustancia química: **feromona**, que todas pueden detectar.
- Cuando aparece un obstáculo, aquellas hormigas que lo tienen delante no pueden seguir el rastro de feromona.
- Tienen que elegir entre ir hacia la derecha o hacia la izquierda.
- Es de esperar que la mitad de las hormigas vayan hacia la derecha y la otra mitad hacia la izquierda, y lo mismo pasará en el otro lado del obstáculo.
- Aquellas hormigas que elijan el, por casualidad, el camino más corto, podrán reconstruir más rápidamente el rastro de feromona perdido (vuelven antes al nido)
- Las hormigas de este camino empiezan a emitir su feromona con anterioridad, y por lo tanto la probabilidad posterior de elegir este camino se incrementa

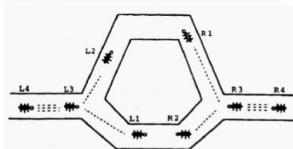
Hormigas naturales: Obstáculos



Hormigas naturales: Obstáculos

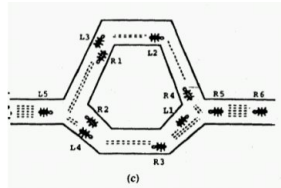
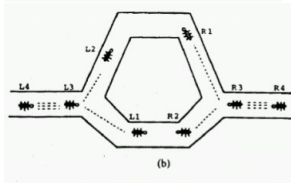
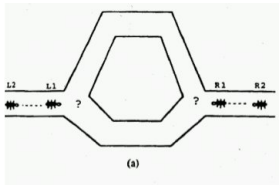


(a)

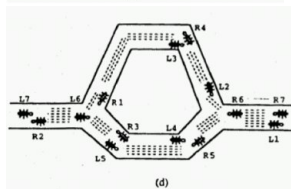
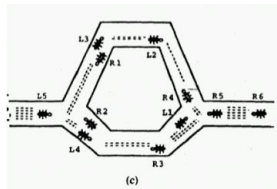
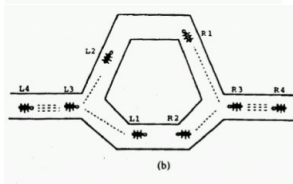
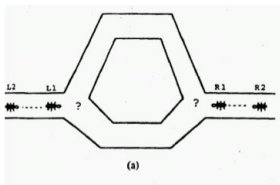


(b)

Hormigas naturales: Obstáculos



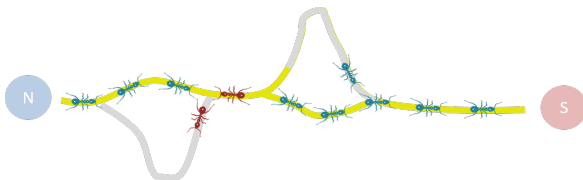
Hormigas naturales: Obstáculos



Ant Colony Optimization (ACO)

ACO: Hormigas naturales

- Las hormigas eligen el camino de forma **probabilística**, dando mayor probabilidad a los caminos con una alta concentración de feromona.
- Las bifurcaciones más prometedoras (más cercanas a la comida), van **acumulando** feromona al ser recorridas por más hormigas.
- Las bifurcaciones menos prometedoras **pierden** feromona por evaporación, al disminuir progresivamente el número de hormigas que las visitan.
- La acción continuada de la colonia da lugar a un rastro de feromona que permite a las hormigas encontrar el camino más corto desde el hormiguero a la comida.



También son capaces de adaptarse a cambios producidos por el entorno, por ejemplo, son capaces de encontrar un nuevo camino más corto.

Ant Colony Optimization (ACO)

La mayoría de las ideas ACO provienen de las hormigas reales:

- Uso de una colonia de individuos que cooperan.
- Un rastro de feromonas como vía de comunicación.
- Una secuencia de movimientos locales para encontrar el camino más corto.
- Una estrategia de decisión estocástica utilizando información local.

ACO: Hormigas artificiales.

Cada hormiga artificial: un agente que imita a la hormiga natural. Es un mecanismo probabilístico de construcción de soluciones del problema.

Ant Colony Optimization (ACO)

ACO: Hormigas artificiales.

- Una hormiga artificial puede encontrar una solución, (las reales, el camino entre nido y comida).
- Tanto las hormigas reales como las artificiales cambian su entorno. Las reales depositan **sustancias químicas**, las artificiales cambian **información numérica** del estado que visitan.
- Otras hormigas leen dicha información numérica al visitar ese estado, que es la única vía de información entre hormigas artificiales (explotación).
- Se introduce un mecanismo (fórmula) de disipación de la información, para que la hormiga *olvide* la historia pasada, y pueda buscar nuevas soluciones (exploración).

Ant Colony Optimization (ACO)

ACO: Hormigas artificiales.

- Cada hormiga se mueve hacia un estado vecino y así va construyendo una solución.
- Los movimientos se realizan mediante una estrategia estocástica local, basada en:
 - La memoria de la hormiga \Rightarrow información acerca del pasado de la hormiga.
 - La cantidad de feromona del tramo.
- La memoria es fundamental para la construcción de soluciones viables \Rightarrow lista tabú.
- La feromona codifica el conocimiento acumulado por todas las hormigas desde el inicio del proceso

Ant Colony Optimization (ACO)

- La feromona es utilizada a modo de memoria a largo plazo que influye en las decisiones de las hormigas.
- Cuándo y cuánta feromona depositar **depende del problema**.
 - Paso a paso: A medida que se va construyendo el camino
 - Retardado: Después de que se ha construido todo el camino
- Cuánta: Se deposita feromona proporcionalmente a la calidad de la solución

Ant Colony Optimization (ACO)

- La feromona es utilizada a modo de memoria a largo plazo que influye en las decisiones de las hormigas.
- Cuándo y cuánta feromona depositar **depende del problema**.
 - Paso a paso: A medida que se va construyendo el camino
 - Retardado: Después de que se ha construido todo el camino
- Cuánta: Se deposita feromona proporcionalmente a la calidad de la solución

Consecuencia

Más hormigas visitan un camino \Rightarrow más feromona tiene \Rightarrow más interesante es ese camino para todas.

Esquema del metaheurístico

- ❶ Fijar parámetros. Inicializar rastros de feromona.
- ❷ Secuenciación de Actividades.
 - **Construir soluciones de hormigas.** Las hormigas se mueven aplicando un criterio probabilístico que hace uso de los rastros de feromona e información heurística.
 - **Actualizar feromonas.** Los rastros de feromona se modifican, bien aumentando (si las hormigas depositan feromona en los arcos visitados, ó en los nodos), bien disminuyendo (evaporación de feromona).
 - **Aplicar búsqueda local (opcional).** Acciones centralizadas que no pueden ser ejecutadas por una sólo hormiga. Por ejemplo: activar un procedimiento de optimización local, ó decidir si las hormigas que han encontrado el mejor camino hasta el momento depositen feromona adicional para sesgar el proceso de búsqueda, etc.

Ant Colony Optimization (ACO)

En el TSP, tenemos el grafo $G(V, E)$, y los costes c_{ij} de cada $(i, j) \in E$.

ACO aplicado al TSP: Construcción de soluciones

- Tenemos m hormigas ($m \leq n$), que construyen colectivamente el tour.
- Las feromonas τ_{ij} se suelen inicializar con el mismo valor τ_0 , esto es

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

Usualmente, $\tau_0 = m/C^{nn}$, donde C^{nn} es el coste (longitud) de la solución encontrada por un heurístico voraz (*hill climbing*).

- Definimos los valores heurísticos:

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

Dan información (heurística) de las aristas, de forma que las aristas de **menor coste** (ó longitud) c_{ij} tengan **mayor probabilidad** de selección.

- Cada hormiga k mantiene una lista de nodos visitados (lista tabú).

Ant Colony Optimization (ACO)

ACO aplicado al TSP: Construcción de soluciones

- **Regla probabilística de transición:** es la probabilidad con que la hormiga k , situada en la ciudad i , decide moverse a la ciudad j :

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{s \in N_k(i)} [\tau_{is}]^\alpha [\eta_{is}]^\beta}, & j \in N_k(i) \\ 0, & \text{otherwise} \end{cases}$$

donde

- τ_{ij} es la feromona de la arista (i, j) .
- $\eta_{ij} = \frac{1}{c_{ij}}$.
- $N_k(i)$ es el conjunto de nodos vecinos al nodo i , que aún no han sido visitados por la hormiga k .
- α y β Parámetros. Representan la influencia de la feromona y del coste, respectivamente.

Sobre estas probabilidades se establecerá la **política de decisión**: Criterio para la elección del nuevo vértice.

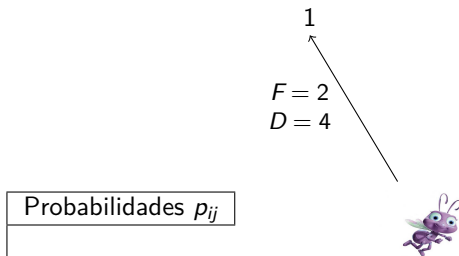
Ant Colony Optimization (ACO)

F : feromona de cada arista, D : coste (ó longitud) de cada arista, $\alpha = 1 = \beta$.



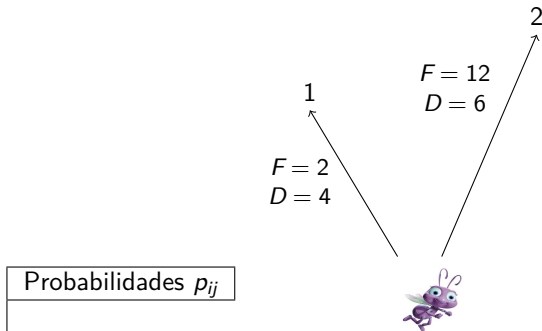
Ant Colony Optimization (ACO)

F : feromona de cada arista, D : coste (ó longitud) de cada arista, $\alpha = 1 = \beta$.



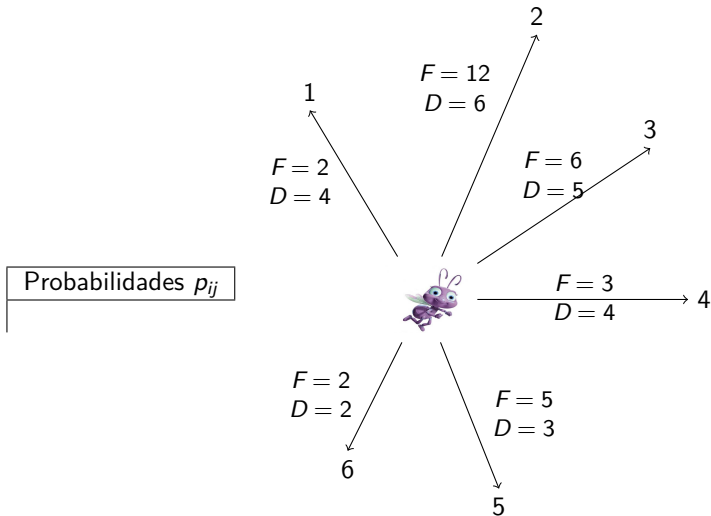
Ant Colony Optimization (ACO)

F : feromona de cada arista, D : coste (ó longitud) de cada arista, $\alpha = 1 = \beta$.



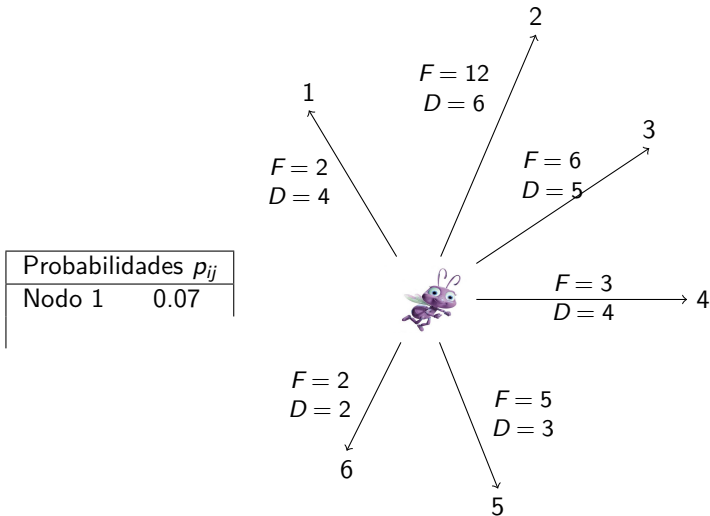
Ant Colony Optimization (ACO)

F : feromona de cada arista, D : coste (ó longitud) de cada arista, $\alpha = 1 = \beta$.



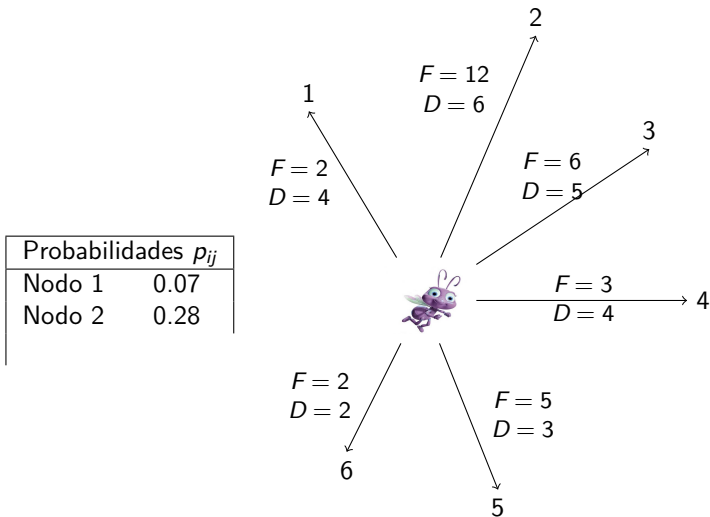
Ant Colony Optimization (ACO)

F : feromona de cada arista, D : coste (ó longitud) de cada arista, $\alpha = 1 = \beta$.



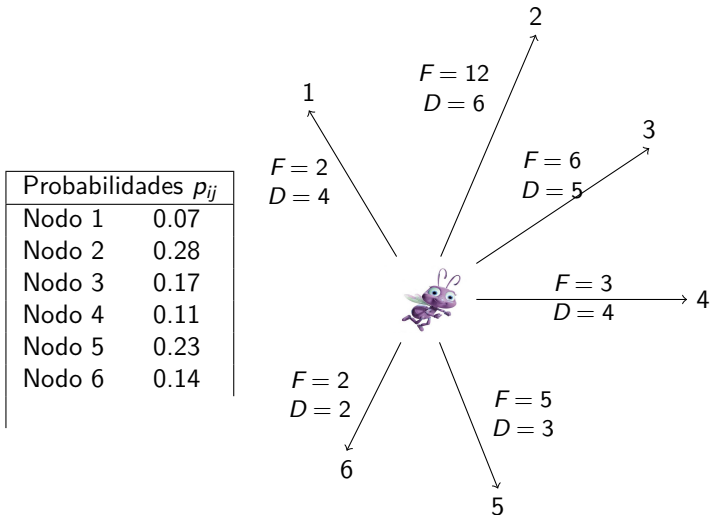
Ant Colony Optimization (ACO)

F : feromona de cada arista, D : coste (ó longitud) de cada arista, $\alpha = 1 = \beta$.



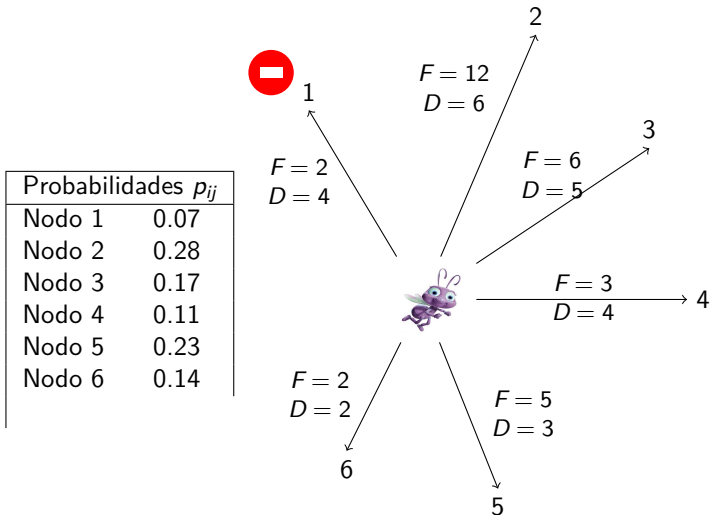
Ant Colony Optimization (ACO)

F : feromona de cada arista, D : coste (ó longitud) de cada arista, $\alpha = 1 = \beta$.



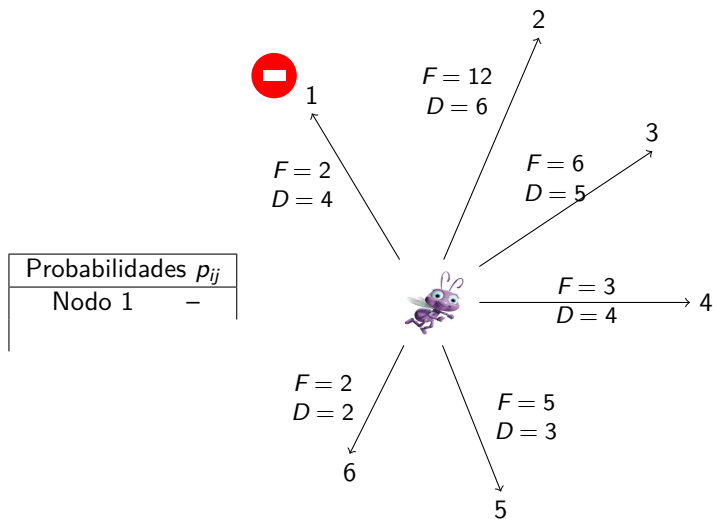
Ant Colony Optimization (ACO)

F : feromona de cada arista, D : coste (ó longitud) de cada arista, $\alpha = 1 = \beta$.



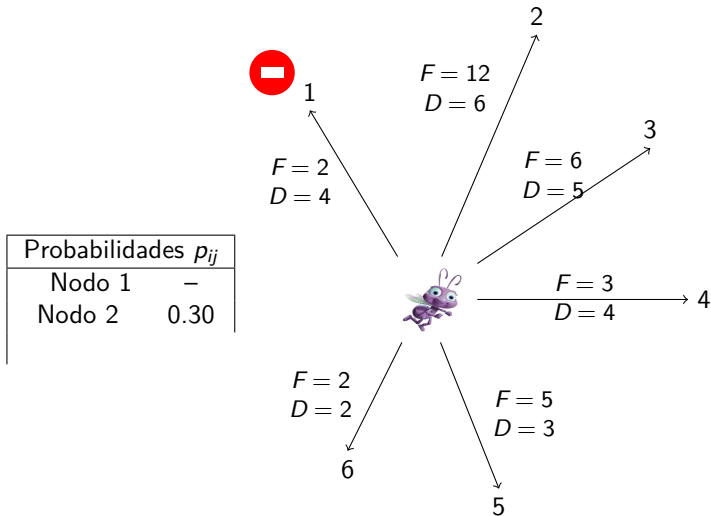
Ant Colony Optimization (ACO)

F : feromona de cada arista, D : coste (ó longitud) de cada arista, $\alpha = 1 = \beta$.



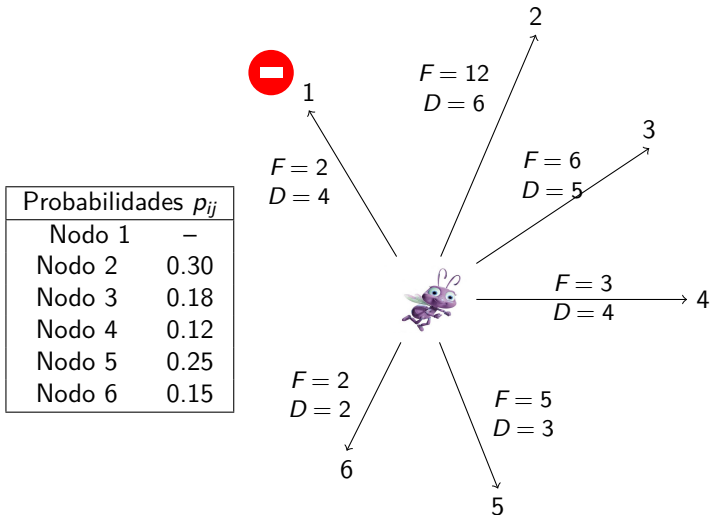
Ant Colony Optimization (ACO)

F : feromona de cada arista, D : coste (ó longitud) de cada arista, $\alpha = 1 = \beta$.



Ant Colony Optimization (ACO)

F : feromona de cada arista, D : coste (ó longitud) de cada arista, $\alpha = 1 = \beta$.



Ant Colony Optimization (ACO)

Parámetros

- α y β controlan la influencia de la feromona y el coste.
- α alto \Rightarrow las hormigas prefieren tramos con más feromona.
- β alto \Rightarrow las hormigas prefieren tramos poco costosos.
- Si predomina la feromona se pueden generar ciclos subóptimos.
- Si predomina el coste, se pueden generar soluciones triviales \Rightarrow métodos greedy.
- Hay que buscar un compromiso entre los dos.

Los valores de α y β dependen del problema, y se identifican experimentalmente. Los mejores valores testados en la literatura han sido: $\alpha = 1$, $\beta \in [2, 5]$.

Ant Colony Optimization (ACO)

ACO aplicado al TSP: Actualización de la feromona

La actualización de la feromona se realiza de un solo golpe, al final de cada ciclo. Para cada arista (i, j) , el valor de su feromona τ_{ij} se actualiza como sigue:

$$\tau_{ij} \leftarrow \underbrace{(1 - \rho)\tau_{ij}}_{\text{evaporación}} + \underbrace{\sum_{k=1}^m \Delta\tau_{ij}^k}_{\text{aporte de todas las hormigas}}$$

- $\rho \in (0, 1]$: es el *factor* de evaporación (común a todas las aristas).
- $\Delta\tau_{ij}^k$ es la cantidad de feromona dejada en la arista (i, j) por la hormiga k :

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{C(S_k)}, & \text{si } i, j \in S_k, \text{ (circuito de la hormiga } k) \\ 0, & \text{otherwise} \end{cases}$$

donde Q es una constante (usualmente, $Q = 1$), S_k es el circuito generado por la hormiga k , y $C(S_k)$ es su coste (ó longitud).

Ant Colony Optimization (ACO)

ACO aplicado al TSP: Actualización de la feromona

Cuando todas las hormigas han acabado los recorridos, se actualiza la feromona de toda la red.

En la actualización de la feromona:

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

- **Disipación:** Si una arista (i, j) es poco visitada, el sumando $\sum_{k=1}^m \Delta\tau_{ij}^k$ es pequeño (ó cero, si no es visitada), por lo tanto pierde feromona.
- Cuanto más visitada sea dicha arista, mayor es el aporte de feromona (valor de dicho sumando).
- Por tanto: se acumula más feromona en las aristas de los mejores tours (tours menos costosos).

Ant Colony Optimization (ACO)

ACO aplicado al TSP: Terminación

Cuando se alcance un número prefijado de tours, ó bien hasta llegar al estancamiento (*stagnation*): todas las hormigas hacen el mismo tour.

Ant Colony Optimization (ACO)

ACO aplicado al TSP: Terminación

Cuando se alcance un número prefijado de tours, ó bien hasta llegar al estancamiento (*stagnation*): todas las hormigas hacen el mismo tour.

ACO aplicado al TSP: Método Básico

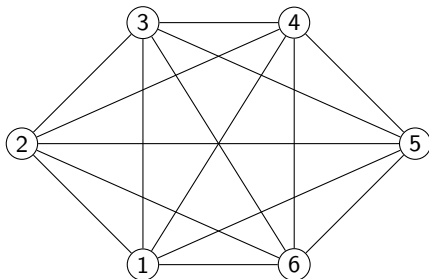
- ❶ Fijar parámetros, inicializar valores de feromona.
- ❷ Colocar cada hormiga aleatoriamente en un vértice.
- ❸ **Repeat** until **Stopping criteria**:
 - ❶ **For** cada hormiga k **do**:
 - Calcular probabilidades de transición p_{ij}^k .
 - Aplicar **política de decisión** para trasladarse al siguiente vértice hasta finalizar el tour.
 - ❷ [Opcional]: Aplicar búsqueda local.
 - ❸ **Actualizar** feromonas.
- ❹ **Output**: Mejor solución encontrada.

Nota: La política de decisión es el criterio para elegir el nuevo vértice (no necesariamente tiene que ser elegir el de mayor probabilidad).

Ant Colony Optimization (ACO)

Ejemplo de aplicación al TSP

Veremos **una iteración** del algoritmo para un caso sencillo del TSP, con $n = 6$ nodos, y $m = 6$ hormigas, colocadas en los nodos $1, \dots, 6$, respectivamente.



Ant Colony Optimization (ACO)

Matriz de distancias (d_{ij}) , y correspondiente heurística (η_{ij}) (con $\eta_{ij} = \frac{1}{d_{ij}}$):

Matriz de distancias

-	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	$\sqrt{5}$	$\sqrt{2}$	-

Ant Colony Optimization (ACO)

Matriz de distancias (d_{ij}) , y correspondiente heurística (η_{ij}) (con $\eta_{ij} = \frac{1}{d_{ij}}$):

Matriz de distancias

-	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	$\sqrt{5}$	$\sqrt{2}$	-

Matriz de valor heurístico

-	1	0.447	0.447	0.5	0.707
1	-	0.707	0.5	0.447	0.447
0.447	0.707	-	0.707	0.447	0.333
0.447	0.5	0.707	-	1	0.447
0.5	0.447	0.447	1	-	0.707
0.707	0.447	0.333	0.447	0.707	-

Ant Colony Optimization (ACO)

Ejemplo de aplicación al TSP

Los parámetros y política de decisión son:

- Inicializamos la feromona a $\tau_0 = 10$, (en todas las aristas).
- Parámetros $\alpha = 1 = \beta$.
- **Política de decisión:** para la elección de la ciudad de destino, generamos un número aleatorio uniforme en $(0, 1)$, $u \in \mathbb{U}(0, 1)$, tal que: si la hormiga k parte de la ciudad i , irá a la primera ciudad j no visitada verificando:

$$\sum_{\substack{1 \leq s \leq j \\ s \in N_k(i)}} p_{is}^k > u$$

(con $N_k(i)$: nodos vecinos al i todavía no visitados por la hormiga k)

Ant Colony Optimization (ACO)

Se trata de hacer **una iteración**. El proceso constructivo de las tres primeras hormigas es:

hormiga 1

	1	2	3	4	5	6	Unifo	Solución
1	-	0.322	0.144	0.144	0.161	0.227	0.00	(1 2 - - -)

Ant Colony Optimization (ACO)

Se trata de hacer **una iteración**. El proceso constructivo de las tres primeras hormigas es:

hormiga 1

	1	2	3	4	5	6	Unifo	Solución
1	-	0.322	0.144	0.144	0.161	0.227	0.00	(1 2 - - -)
2	-	-	0.336	0.237	0.212	0.212	0.031	(1 2 3 - -)

Ant Colony Optimization (ACO)

Se trata de hacer **una iteración**. El proceso constructivo de las tres primeras hormigas es:

hormiga 1

	1	2	3	4	5	6	Unifo	Solución
1	-	0.322	0.144	0.144	0.161	0.227	0.00	(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 -)

Ant Colony Optimization (ACO)

Se trata de hacer **una iteración**. El proceso constructivo de las tres primeras hormigas es:

hormiga 1

	1	2	3	4	5	6	Unifo	Solución
1	-	0.322	0.144	0.144	0.161	0.227	0.00	(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)

Ant Colony Optimization (ACO)

Se trata de hacer **una iteración**. El proceso constructivo de las tres primeras hormigas es:

hormiga 1

	1	2	3	4	5	6	Unifo	Solución
1	-	0.322	0.144	0.144	0.161	0.227	0.00	(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	-	-	-	(1 2 3 5 6 4)

Ant Colony Optimization (ACO)

hormiga 2

	1	2	3	4	5	6	Unifo	Solución
2	0.322	-	0.227	0.161	0.144	0.144	0.479	(2 3 - - - -)

Ant Colony Optimization (ACO)

hormiga 2

	1	2	3	4	5	6	Unifo	Solución
2	0.322	-	0.227	0.161	0.144	0.144	0.479	(2 3 - - -)
3	0.231	-	-	0.365	0.231	0.172	0.671	(2 3 5 - -)

Ant Colony Optimization (ACO)

hormiga 2

	1	2	3	4	5	6	Unifo	Solución
2	0.322	-	0.227	0.161	0.144	0.144	0.479	(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 -)

Ant Colony Optimization (ACO)

hormiga 2

	1	2	3	4	5	6	Unifo	Solución
2	0.322	-	0.227	0.161	0.144	0.144	0.479	(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 -)

Ant Colony Optimization (ACO)

hormiga 2

	1	2	3	4	5	6	Unifo	Solución
2	0.322	-	0.227	0.161	0.144	0.144	0.479	(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	-	-	-	-	-	-	(2 3 5 6 4 1)

Ant Colony Optimization (ACO)

hormiga 2

	1	2	3	4	5	6	Unifo	Solución
2	0.322	-	0.227	0.161	0.144	0.144	0.479	(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	-	-	-	-	-	-	(2 3 5 6 4 1)

hormiga 3

	1	2	3	4	5	6	Unifo	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	-	(3 2 1 5 4 6)

Ant Colony Optimization (ACO)

Una vez construidas las 6 soluciones, se aplica la actualización de feromona. Considerando que **cada hormiga k** deposita la cantidad de feromona $\Delta\tau_{ij}^k = 100/C(S_k)$ en los arcos visitados, (donde $C(S_k)$ es la longitud (coste) de la solución S_k), tendríamos los siguientes aportes:

Hormiga	$C(S_k)$	Aporte	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)	$\{a_{65}, a_{53}, a_{34}, a_{42}, a_{21}, a_{16}\}$

Ant Colony Optimization (ACO)

Aplicando el mecanismo de actualización de la feromona, donde se evapora la feromona y se realizan los aportes, mediante la expresión:

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} + \Delta\tau_{ij} \quad \text{con} \quad \Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k$$

(donde $\Delta\tau_{ij}$ es el aporte de **todas** las hormigas a la arista (i, j)), y tomando $\rho = 0.5$, la matriz de feromonas, después de la primera iteración, quedaría:

Matriz de feromonas (τ_{ij}) actualizada

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

Ant Colony Optimization (ACO): Variantes

ACO: Variantes

El algoritmo anterior fue el primer ACO que se propuso (1991), denominado Sistema de Hormigas: **Ant System, (AS)**.

Existen diversas variantes, que suponen mejoras sobre el AS:

- Sistema de Hormigas Elitista: Elitist Ant System (EAS, 1991, 1996). Introducido por Dorigo y sus colaboradores^{1 2} para solucionar el problema de la lentitud de la convergencia del AS.
- Sistemas de hormigas max-min: Max-Min Ant System (MMAS, 1997).
- Sistemas de colonias de hormigas: Ant Colony System (ACS, 1996, 1997).
- Sistemas de hormigas basadas en rankings: Rank-Based AS, (ASrank, 1997, 1999).

Y otros más en la literatura: Sistemas de hormigas mejor-peor (BWAS), con búsqueda local, y aplicando estudio comparativo.

¹M. Dorigo, V. Maniezzo, A. Coloni (1991). Technical Report 91-016. Politecnico di Milano.

²M. Dorigo, V. Maniezzo, A. Coloni (1996). *The Ant System: Optimization by a colony of cooperative agents*. IEEE Trans. on Systems, Man and Cybernetics-part B, 26, 1-13.

Ant Colony Optimization (ACO): Variantes

Variante ACO: Elitist Ant System (EAS)

La única diferencia entre EAS, y el AS que hemos visto, es la regla de actualización de las feromonas.

- En el AS, todas las hormigas tienen la misma participación en la actualización de las feromonas. En el EAS, las *mejores* hormigas contribuyen más que las demás.
- Las mejores hormigas son aquellas con el mejor tour (el tour de menor coste, ó más corto). Estas hormigas son *elitistas*.
- La regla de actualización aplica un refuerzo adicional a los buenos arcos:

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} + \sum_{k=1}^m \Delta_{ij}^k + e \Delta \tau_{ij}^{best}$$

- e es el número de hormigas elitistas, (valor típico $e = n$).
- $\Delta \tau_{ij}^{best} = \begin{cases} \frac{Q}{C_{best}}, & (i,j) \text{ está en el mejor tour } S^{best} \text{ hasta el momento} \\ 0, & \text{otherwise} \end{cases}$
- C^{best} el coste (longitud) del mejor tour S^{best} . Q es una constante, (se puede tomar $Q = 1$).

Variante ACO: Ant Colony System (ACS)

Esta nueva versión del ACO, llamada ACS¹, se diferencia del algoritmo AS original en tres aspectos principales:

- Regla de transición. Explora más la experiencia acumulada por las hormigas, utilizando una regla de elección más agresiva para pasar de una ciudad a otra.
- La actualización global de la feromona: sólo se considera la hormiga que generó la mejor solución hasta el momento.
- Se añade una nueva **actualización local** de la feromona, basado en que cada hormiga modifica automáticamente la feromona de cada arco que visita para diversificar la búsqueda.

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

Ant Colony Optimization (ACO): Variantes

Variante ACO: Ant Colony System (ACS)

Regla de transición. Es una **regla proporcional pseudoaleatoria**, basada en:

- Un parámetro del sistema $q_0 \in [0, 1]$.
- Un valor $q \in [0, 1]$, generado aleatoriamente en cada transición.

Transición del ACS. La hormiga k se encuentra en la ciudad i , y tiene que moverse a un nodo de $N_k(i)$ (nodos vecinos todavía no visitados). Para ello, aplica la siguiente regla proporcional pseudoaleatoria:

- Si $q \leq q_0$: elige moverse a la ciudad $j \in N_k(i)$ tal que:

$$j \text{ maximiza } \{ [\tau_{is}]^\alpha [\eta_{is}]^\beta : s \in N_k(i) \}$$

- Si $q > q_0$, aplica la regla de transición del AS:

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{s \in N_k(i)} [\tau_{is}]^\alpha [\eta_{is}]^\beta}, & j \in N_k(i) \\ 0, & \text{en caso contrario} \end{cases}$$

Variante ACO: Ant Colony System (ACS)

- Cuando $q \leq q_0$ la hormiga elige el tramo más eficaz
- Cuando $q > q_0$ la hormiga elige el tramo aleatoriamente, pero proporcional a su eficacia.
- Al modificar el parámetro q_0 se regula el que el sistema se guíe por las mejores soluciones según su experiencia, o explore regiones desconocidas del espacio de búsqueda
- Si q_0 grande \Rightarrow más explotación.
- Si q_0 pequeño \Rightarrow más exploración

Variante ACO: Ant Colony System (ACS)

- **Actualización global de la feromona.** Después de cada iteración (cuando cada hormiga ha completado su recorrido). Sólo una hormiga elitista (la mejor hasta el momento) puede añadir feromona en los arcos del mejor recorrido. La actualización es:

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} + \rho\Delta_{ij}^{best}, \text{ para } (i, j) \in S^{best}$$

donde $\Delta\tau_{ij}^{best} = 1/C^{best}$ (siendo C^{best} el coste (longitud) del mejor tour S^{best}).

- Se refuerzan las aristas de la (las, si hay más de una) mejor solución.

Variante ACO: Ant Colony System (ACS)

- **Actualización local de la feromona** (se ejecuta inmediatamente cada vez que una hormiga cruza un arco (i, j)):

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

- El valor de τ_0 es el mismo que el de la inicialización del algoritmo.
- $\phi \in (0, 1)$ es un parámetro (experimentalmente, el valor $\phi = 1/C^{nn}$ ha dado buenos resultados).
- Esta actualización local pretende que un arco ya elegido sea menos deseable para la siguiente hormiga, al objeto de incrementar la exploración de arcos todavía no visitados.

Ant Colony Optimization (ACO)

Algoritmos ACO: Algunas ventajas

- Un feedback positivo, que da lugar a una rápida obtención de *buenas* soluciones.
- La computación distribuída evita convergencia prematura.
- Puede adaptarse para aplicaciones dinámicas.
- Eficiente para TSP, y problemas similares (de caminos más cortos).
- La interacción colectiva de la colonia, que hace que funcione a pesar del fallo en algunas hormigas.

Ant Colony Optimization (ACO)

Algoritmos ACO: Algunas ventajas

- Un feedback positivo, que da lugar a una rápida obtención de *buenas* soluciones.
- La computación distribuída evita convergencia prematura.
- Puede adaptarse para aplicaciones dinámicas.
- Eficiente para TSP, y problemas similares (de caminos más cortos).
- La interacción colectiva de la colonia, que hace que funcione a pesar del fallo en algunas hormigas.

Algoritmos ACO: Algunos inconvenientes

- Aunque la convergencia a la solución óptima está garantizada, el tiempo de convergencia es incierto. De hecho, para algunos problemas la convergencia es más lenta respecto de otros heurísticos.
- Ausencia de un proceso centralizado que guíe al algoritmo hacia buenas soluciones.
- La distribución de probabilidad cambia en cada iteración.
- Dificultad de un análisis teórico. La investigación es fundamentalmente experimental.