



Universidad de Granada



# Algoritmos Bioinspirados Basados en Inteligencia Colectiva

**Manuel Lozano**

***Email: [lozano@decsai.ugr.es](mailto:lozano@decsai.ugr.es)***

***Técnicas de Soft Computing  
para  
Aprendizaje y Optimización***



**Departamento de Ciencias de la  
Computación e Inteligencia Artificial**

# Introducción: Swarm Intelligence

## Swarm Intelligence (Inteligencia Colectiva)

La inteligencia colectiva emergente de un grupo de agentes simples”

“The emergent collective intelligence of groups of simple agents”

## ABs basados en Inteligencia Colectiva

“Algoritmos o mecanismos distribuidos de resolución de problemas inspirados en el comportamiento colectivo de colonias de insectos sociales u otras sociedades de animales”.

(Bonabeau, Dorigo, Theraulaz, 1999)

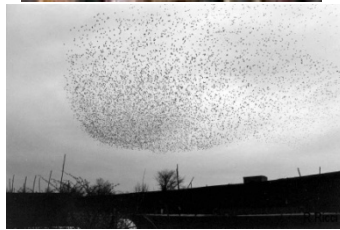
*E. Bonabeau, M. Dorigo, G. Theraulaz  
Swarm Intelligence. From Nature to Artificial  
Systems.  
Oxford University Press, 1999.*



# Swarm Intelligence: Inspiración Biológica



**flocks of birds  
(bandadas de aves)**



**sociedades de insectos  
(bees, wasps, ants, termites)  
(abejas, avispas, hormigas, termitas)**



**schools of fish  
(bancos de peces)**



**herds of mammals  
(manadas de mamíferos)**

# De las Sociedades de Insectos a los ABs basados en Int. Colectiva

---

## *Comportamiento emergente*

- 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 (éstos son no inteligentes, no fiables, 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)



# De las Sociedades de Insectos a los ABs basados en Int. Colectiva

---

## Abejas



- **Cooperación de la colmena**
- **Regulan la temperatura de la colmena**
- **Eficiencia vía especialización: división de la labor en la colonia**
- **Comunicación: Las fuentes de comida son explotadas de acuerdo a la calidad y distancia desde la colmena**

# De las Sociedades de Insectos a los ABs basados en Int. Colectiva

---

## Termitas

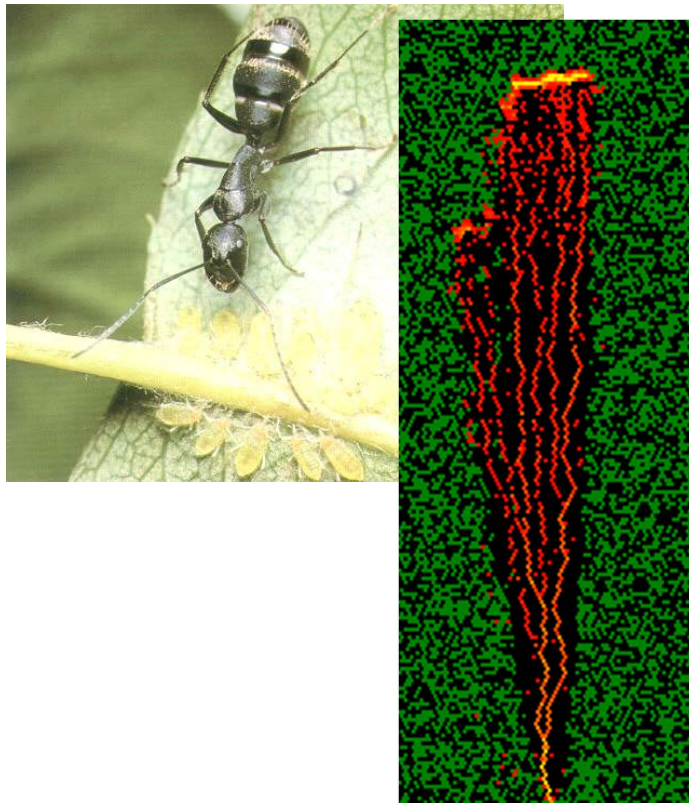


- Nido con forma de cono con paredes externas y conductos de ventilación
- Camaras de camadas en el centro de la colmena
- Rejillas de ventilación en espirales
- Columnas de soporte

# De las Sociedades de Insectos a los ABs basados en Int. Colectiva

---

## Hormigas



- Organizan autopistas “hacia y desde” la comida por medio de rastros de feromona (pheromone)



# De las Sociedades de Insectos a los ABs basados en Int. Colectiva

---

## Características de un Enjambre

- Compuesto de agentes *simples* (Self-Organized)
- Descentralizado
  - No hay un único supervisor
- No hay un plan global (emergente)
- Robusto
  - Las actuaciones se completan aunque un individuo falle
- Flexible
  - Puede responder a cambios externos
  - Percepción del entorno (sentidos)
  - No existe un modelo explícito de entorno/abilidad para cambiarlo



# Modelos de ABs Basados en IC

---

- ¿Cómo puede ser coordinado el esfuerzo individual para alcanzar una tema común?

## Swarm Inspired Methods

- **Particle Swarm Optimization – PSO**

**Optimización basada en nubes (cúmulos) de partículas**

- Conjunto de técnicas inspiradas en el comportamiento de las bandadas de aves o bancos de peces

- **Ant Colony Optimization – ACO**

**Optimización basada en colonias de hormigas**

- Conjunto de técnicas inspiradas por las actividades de una colonia de hormigas

# Particle Swarm Optimization

---

*Kennedy, J. and Eberhart, R. (1995). "Particle Swarm Optimization", Proc. 1995 IEEE Intl. Conf. on Neural Networks, pp. 1942-1948, IEEE Press*

## PARTICLE SWARM OPTIMIZATION (PSO)

**Es una técnica de optimización inspirada en el comportamiento social de bandadas de aves o peces.**

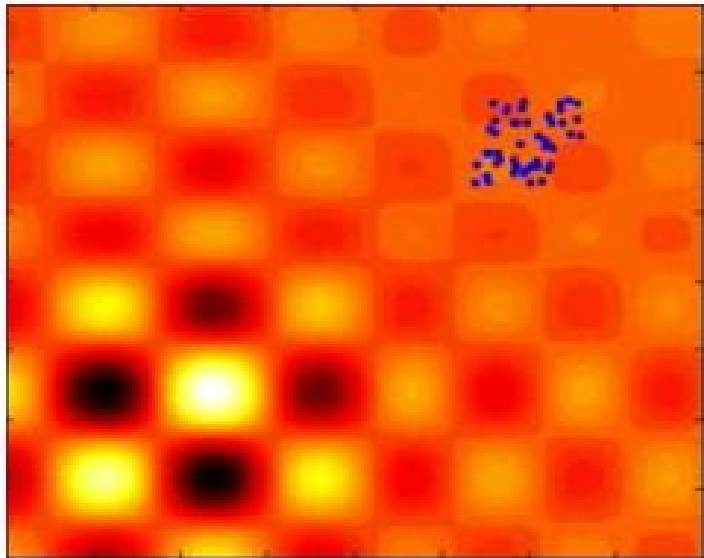


<http://www.swarmintelligence.org/>

# Particle Swarm Optimization

---

- PSO simula el comportamiento de las bandadas de aves.
- Cada solución (partícula) es un “ave” en el espacio de búsqueda que está siempre en continuo movimiento y que nunca muere.



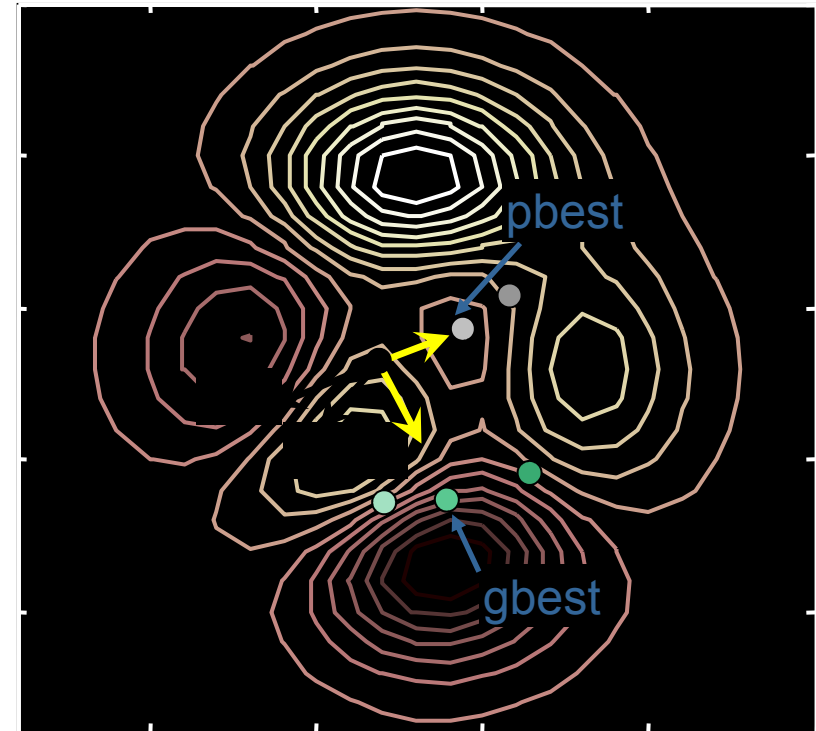
# Particle Swarm Optimization

---

- La población se compone de varias partículas (nube de partículas = *particle swarm*) que se mueven (“vuelan”) por el espacio de búsqueda durante la ejecución del algoritmo.
- Se mantiene la **posición** y **velocidad** de las partículas.
- En PSO, los individuos se comunican directa o indirectamente con otros vía las direcciones de búsqueda.

# Particle Swarm Optimization

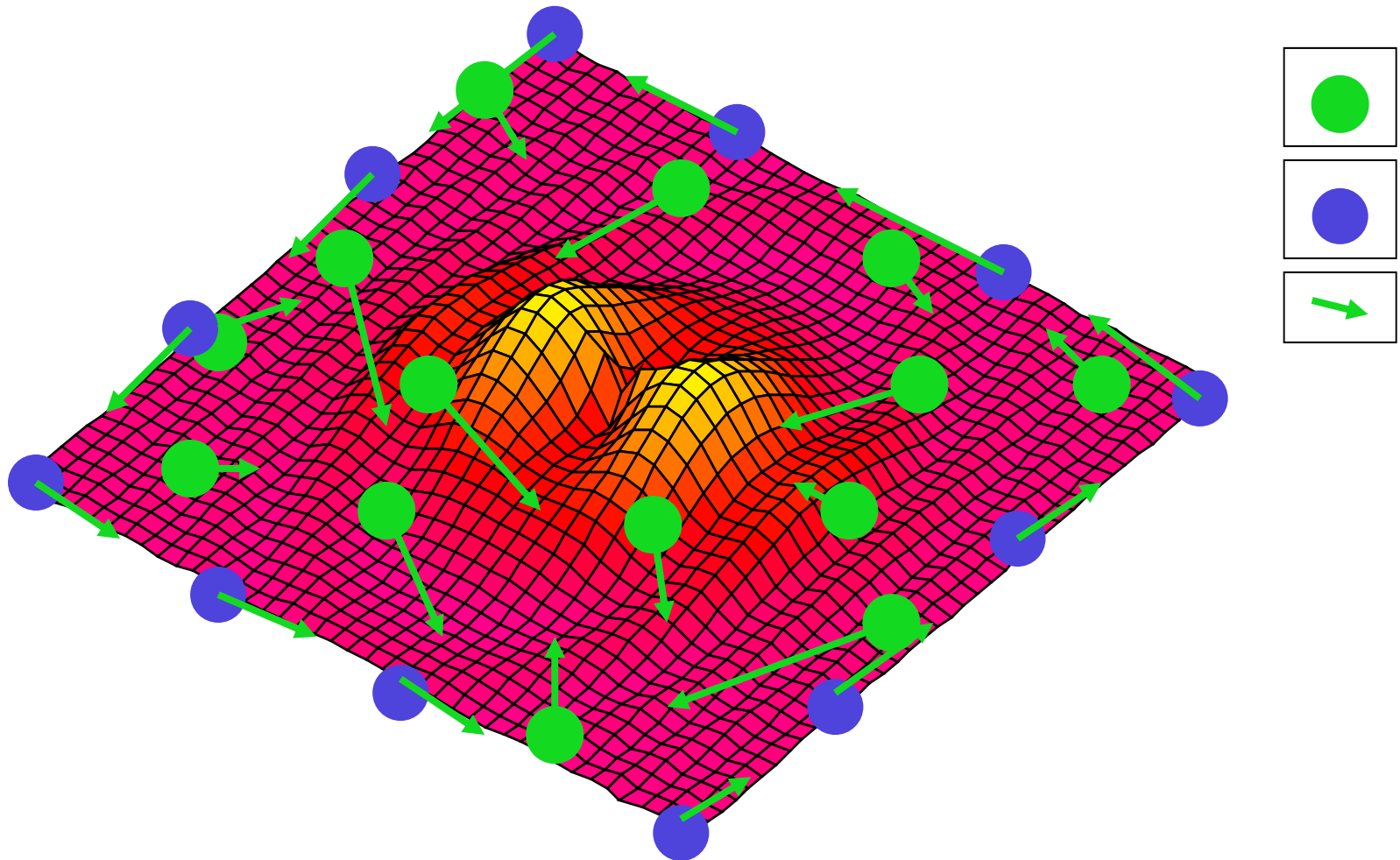
- El movimiento de cada partícula  $p$  depende de:
  - *Su mejor posición desde que comenzó el algoritmo ( $pBest$ ),*
  - *La mejor posición de las partículas de su entorno ( $lBest$ ) o de toda la nube ( $gBest$ ) desde que comenzó el algoritmo.*
- En cada iteración, se cambia la **posición** y **velocidad** de las partículas para acercarlas a las posiciones  $pBest$  y  $lBest/gBest$ .





# Particle Swarm Optimization

---

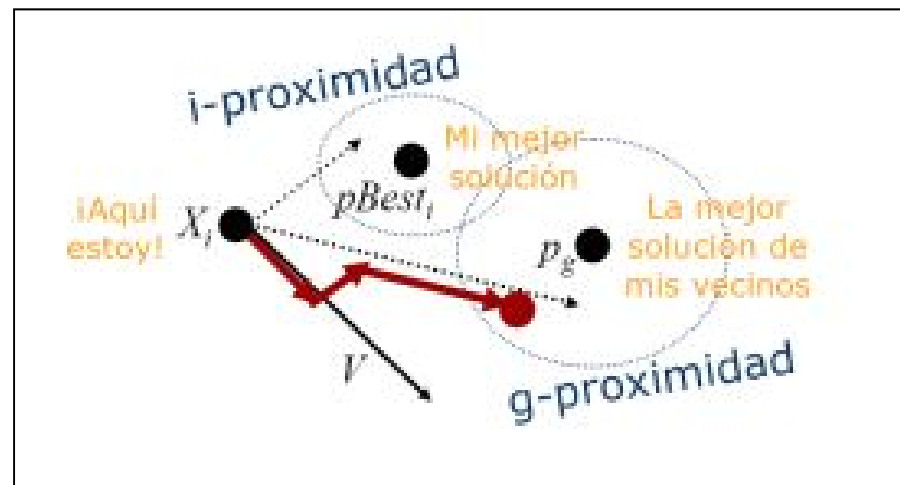


# Particle Swarm Optimization

$$v_{id} = v_{id} + \underbrace{\phi_1 \cdot \text{rnd}() \cdot (pBest_{id} - x_{id})}_{\text{COGNITIVO}} + \underbrace{\phi_2 \cdot \text{rnd}() \cdot (g_{id} - x_{id})}_{\text{SOCIAL}}$$

$$x_{id} = x_{id} + v_{id}$$

- $x_i$  es la partícula actual,  $v_i$  su velocidad.
- $\phi_1, \phi_2$  son ratios de aprendizaje (pesos) que controlan los componentes cognitivo y social,



# PSO Local

---

$t = 0;$

Para  $i=1$  hasta Número\_partículas  
    inicializar  $X_i$  y  $V_i$ ;

Mientras (no se cumpla la condición de parada) hacer  
     $t \leftarrow t + 1$

    Para  $i=1$  hasta Número\_partículas  
        evaluar  $X_i$ ;

        Si  $F(X_i)$  es mejor que  $F(pBest_i)$  entonces  
             $pBest_i \leftarrow X_i$ ;  $F(pBest_i) \leftarrow F(X_i)$

    Para  $i=1$  hasta Número\_partículas

        Escoger  $lBest_i$ , la partícula con mejor fitness del entorno de  $X_i$   
        Calcular  $V_i$ , la velocidad de  $X_i$ , de acuerdo a  $pBest_i$  y  $lBest_i$   
        Calcular la nueva posición  $X_i$ , de acuerdo a  $X_i$  y  $V_i$

Devolver la mejor solución encontrada

# PSO Global

---

$t = 0;$

Para  $i=1$  hasta Número\_partículas  
    inicializar  $X_i$  y  $V_i$ ;

Mientras (no se cumpla la condición de parada) hacer  
     $t \leftarrow t + 1$

    Para  $i=1$  hasta Número\_partículas  
        evaluar  $X_i$ ;

        Si  $F(X_i)$  es mejor que  $F(pBest_i)$  entonces  
             $pBest_i \leftarrow X_i$ ;  $F(pBest_i) \leftarrow F(X_i)$

**Si  $F(pBest)$  es mejor que  $F(gBest)$  entonces**  
             **$gBest \leftarrow pBest_i$ ;  $F(gBest_i) \leftarrow F(pBest_i)$**

    Para  $i=1$  hasta Número\_partículas

        Calcular  $V_i$ , la velocidad de  $X_i$ , de acuerdo a  $pBest_i$  y  **$gBest_i$**

        Calcular la nueva posición  $X_i$ , de acuerdo a  $X_i$  y  $V_i$

Devolver la mejor solución encontrada

# Ant Colony Optimization



**La OPTIMIZACIÓN BASADA EN COLONIAS DE HORMIGAS está basada en la simulación del comportamiento de las hormigas cuando recogen comida**

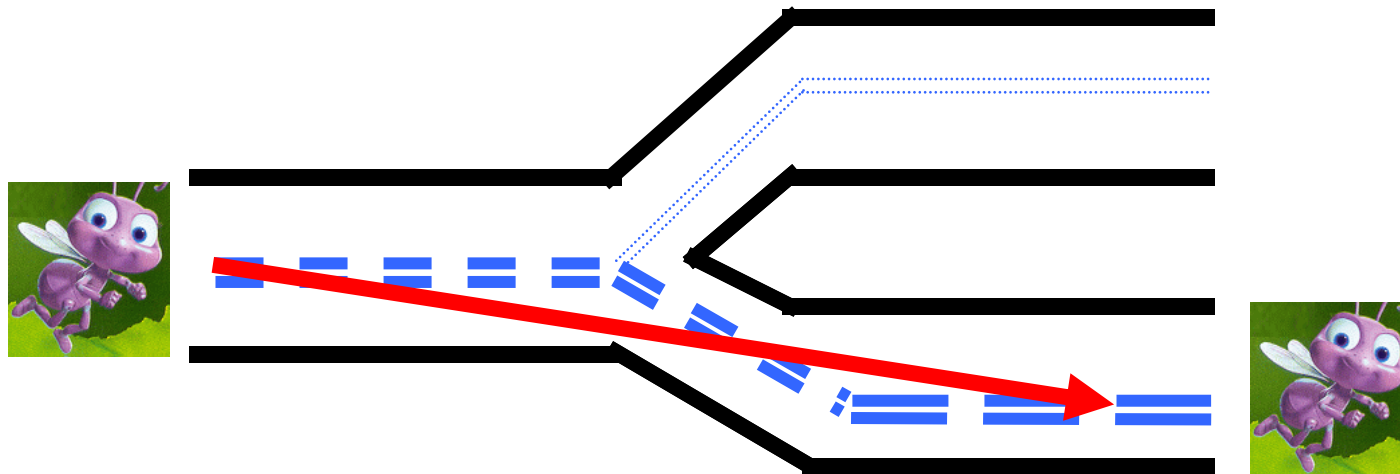
Social insects, following simple, individual rules, accomplish complex colony activities through: flexibility, robustness and self-organization





# Actuación hormigas naturales

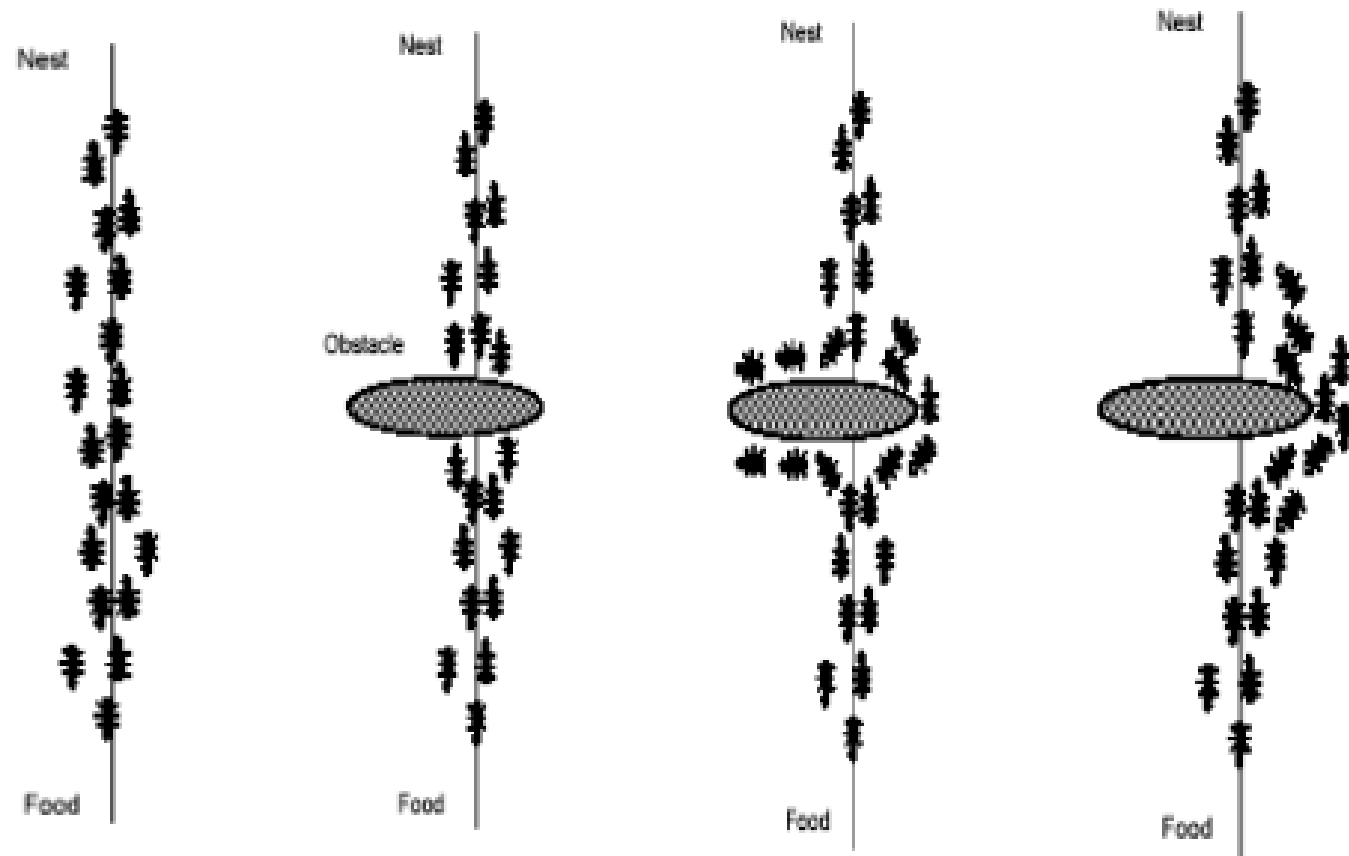
- Cada vez que una hormiga llega a una intersección, decide el camino a seguir de un modo probabilístico



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

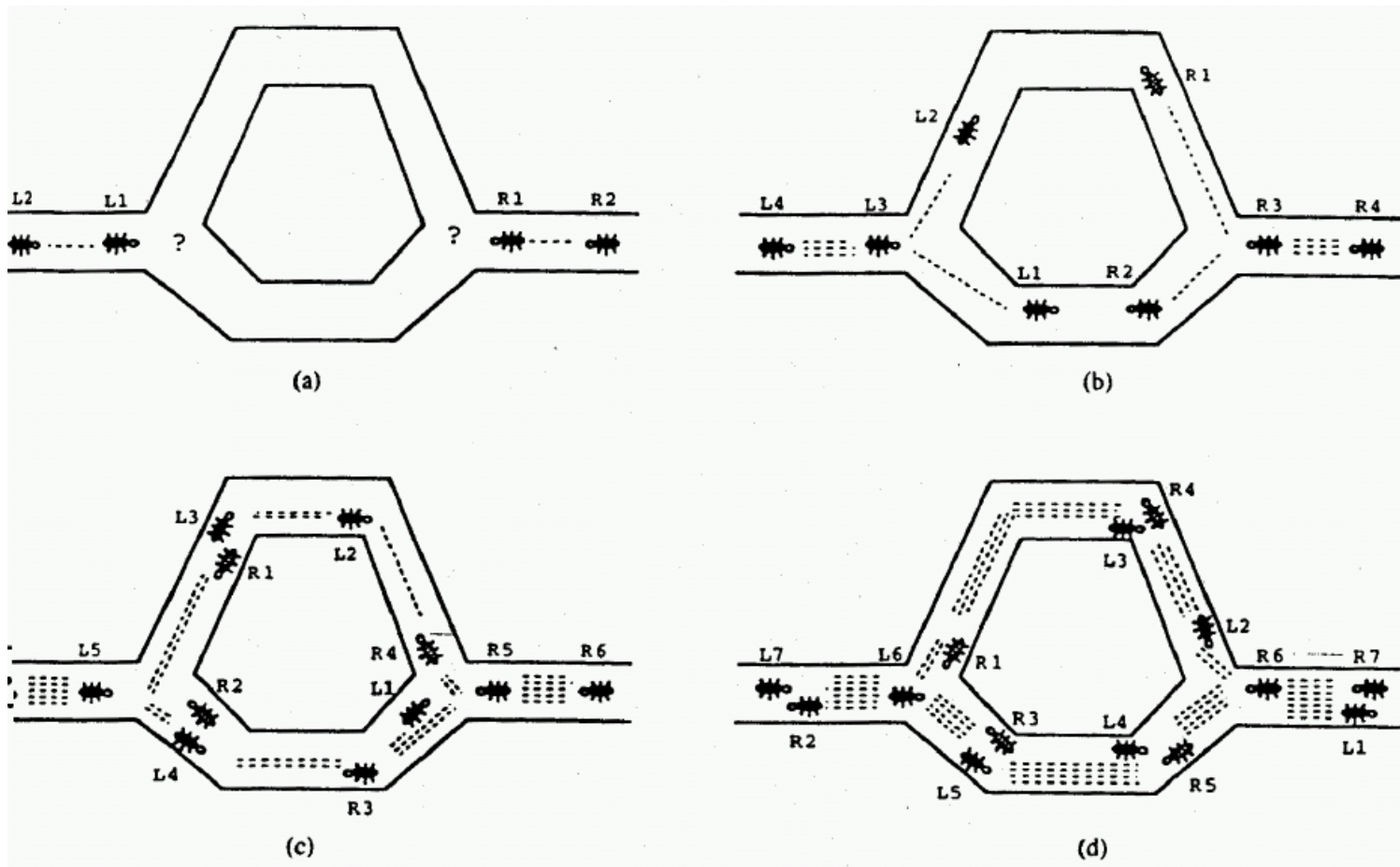
# Actuación hormigas naturales

---



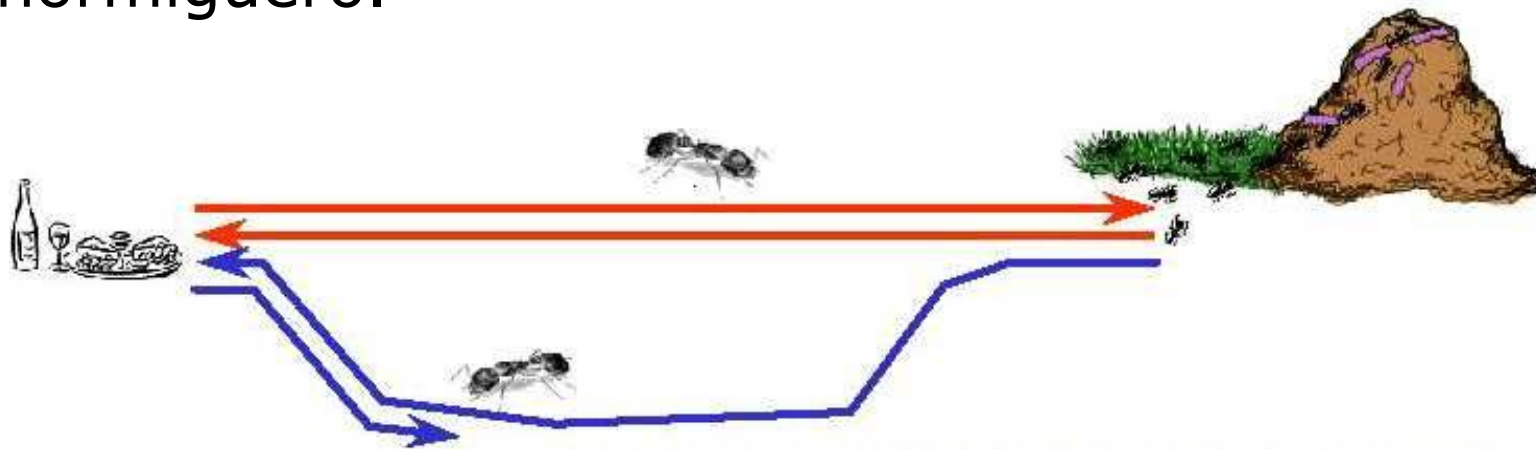
Ant Algorithms – (P.Koumoutsakos – based on notes L. Gamberdella ([www.idsia.ch](http://www.idsia.ch)))

# Actuación hormigas naturales



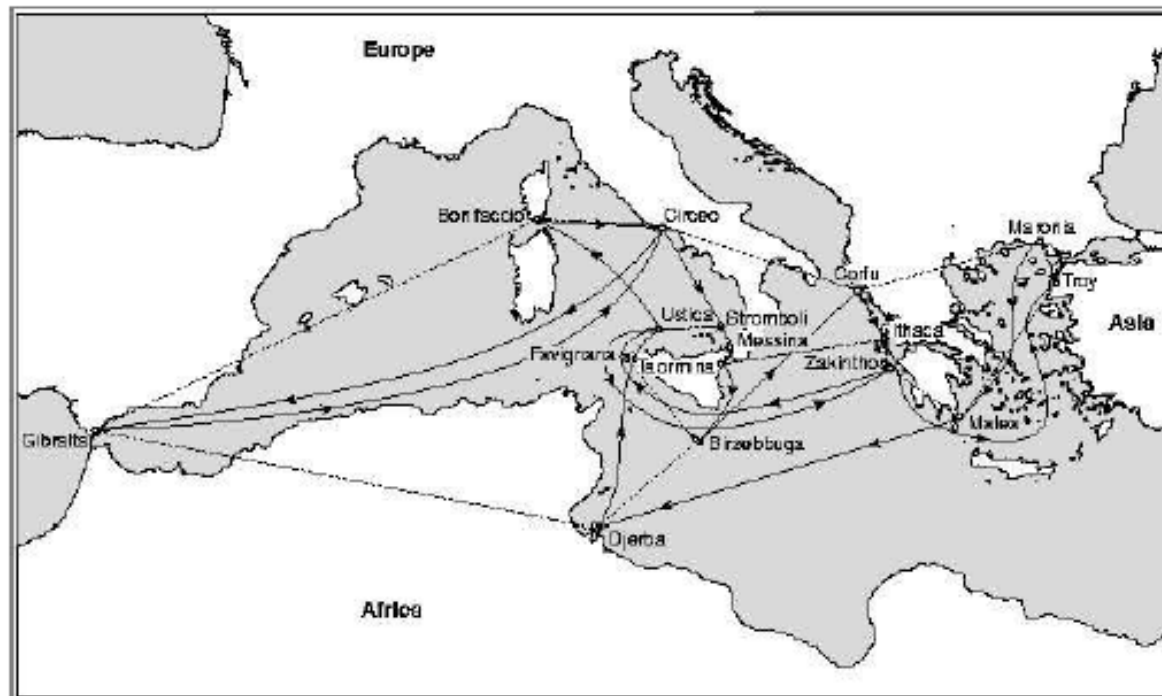
# Ant Colony Optimization

- La analogía más cercana a ACO son los **problemas de rutas en grafos**
- Mientras las hormigas buscan comida, depositan rastros de feromona que atraen a otras hormigas. Desarrollan caminos mínimos entre la comida y el hormiguero.



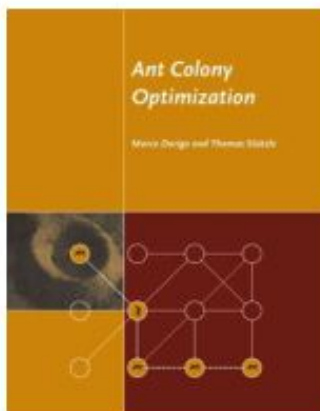
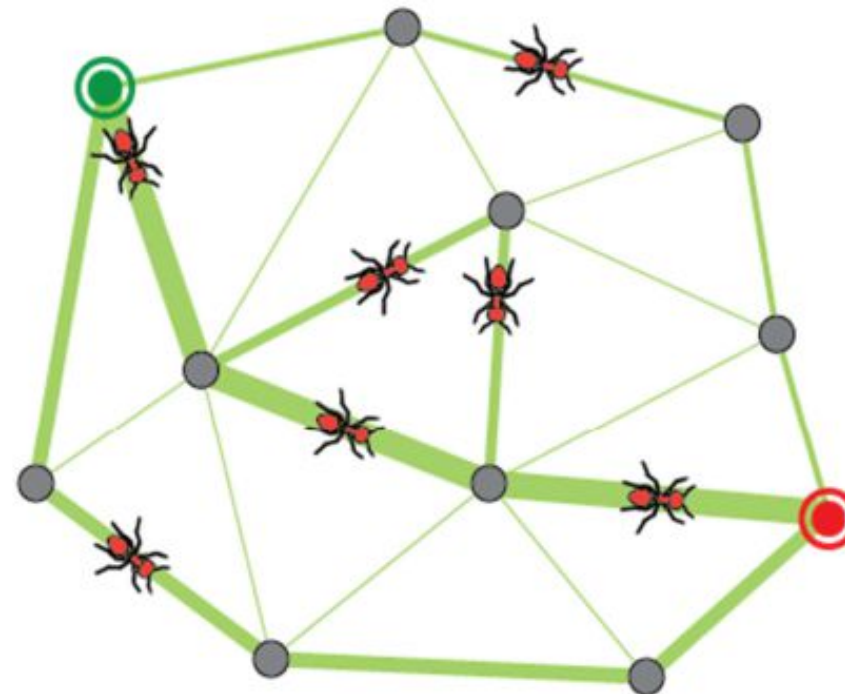
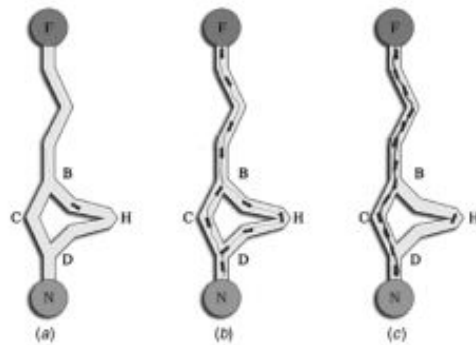
# Ant Colony Optimization

- El Problema del Viajante de Comercio (TSP) es uno de los problemas de optimización combinatoria más conocido y problema sobre el que se comenzó a ensayar satisfactoriamente el comportamiento de los algoritmos ACO.





# Ant Colony Optimization

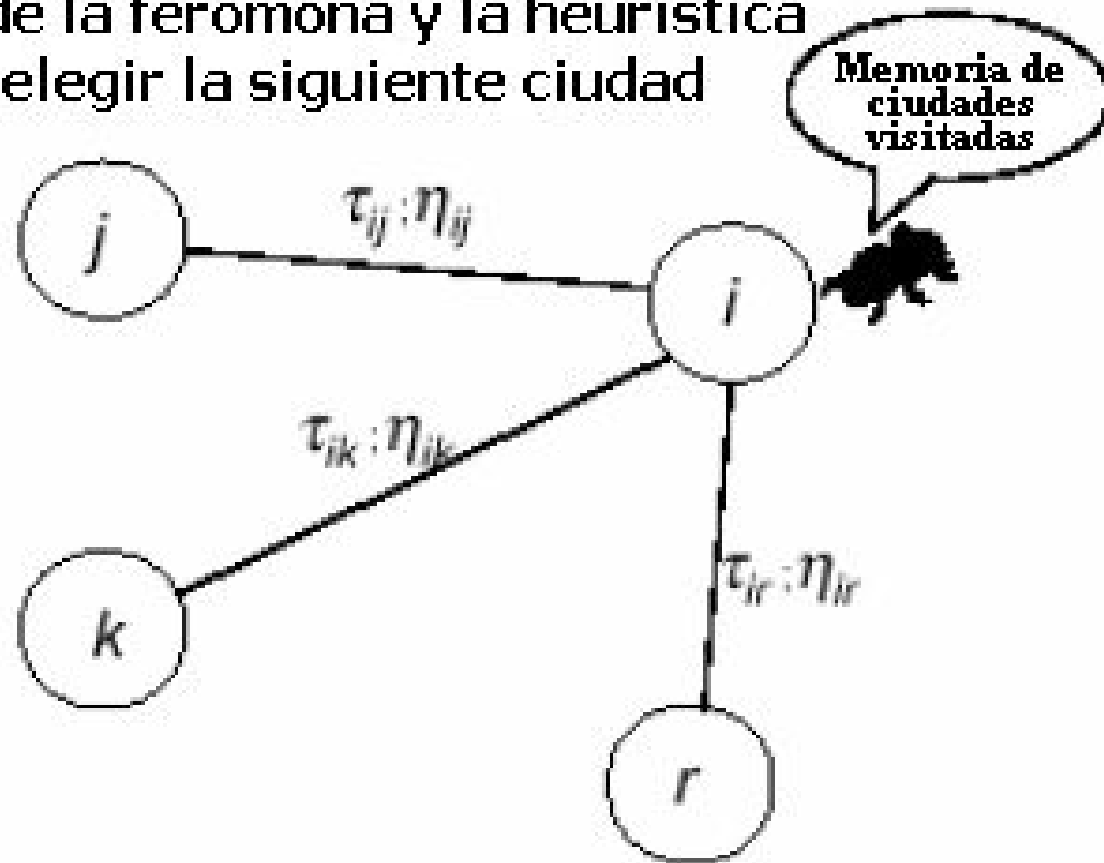


M. DORIGO, T. STÜTZLE, **Ant Colony Optimization**  
The MIT Press, 2004.

# Ant Colony Optimization

---

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



# Ant Colony Optimization

---

- 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  $s$ :

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

- $\tau_{rs}$  es la feromona del arco  $a_{rs}$
- $\eta_{rs}$  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  $k$
- $\alpha$  y  $\beta$  son pesos que establecen un equilibrio entre la importancia de la información memorística y heurística

# Ant Colony Optimization

---

- Un ejemplo de regla de actualización de feromona sería:

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

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

- $C(S_k)$  es el coste de la solución generada por la hormiga  $k$ , es decir, la longitud del circuito  $S_k$
- El valor del aporte anterior es para problemas de minimización como el TSP. Si el problema es de maximización,  $\Delta \tau_{rs}^k = C(S_k)$
- $m$  es el número de hormigas

# Ant Colony Optimization

---

Depositar una cantidad de feromona inicial en todas las aristas

Crear m hormigas

Repetir:

- Reiniciar hormigas (borrar memoria)

- Cada hormiga: Construir solución usando feromonas y coste de las aristas

- Cada hormiga: Depositar feromonas en aristas de la solución

- Evaporar feromona en las aristas

Devolver: la mejor solución encontrada



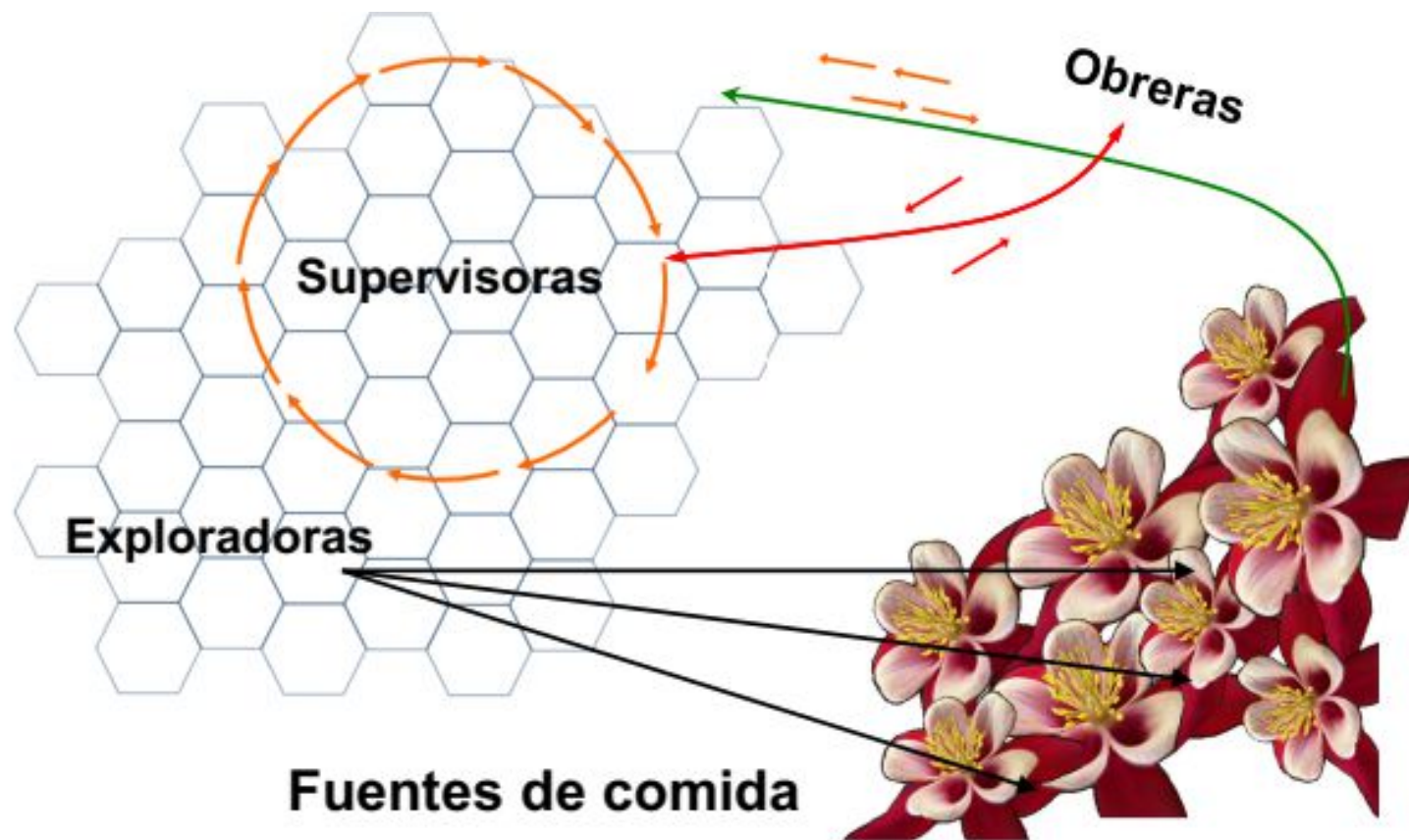
# Ant Colony Optimization

---

- ***Sistema Elitista:*** Añadir en cada paso feromona a las aristas del mejor recorrido encontrado hasta el momento, haya sido encontrado en el paso actual o no.
- ***Sistema por Ranking:*** Cada hormiga deposita feromona proporcional a la bondad de la solución encontrada.
- ***Sistema de Colonias:*** Todas las hormigas dejan la misma cantidad de feromona.
  - Las hormigas que acaban antes pueden volver a comenzar una nueva generación de camino.
  - Los caminos más cortos, tendrán más posibilidad de ser repetidos.
  - Es el mecanismo que aparece en la naturaleza.

# Artificial Bee Colony (ABC)

---



# Artificial Bee Colony (ABC)

---

The colony of artificial bees contains three groups of bees:

- **Employed bees,**
- **Onlookers and**
- **Scouts.**

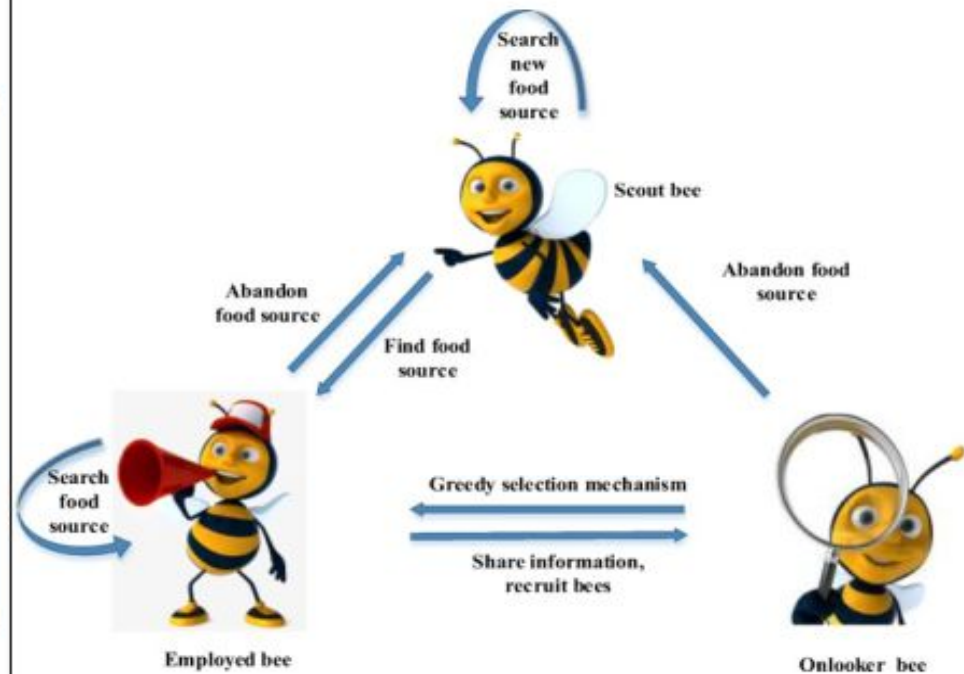
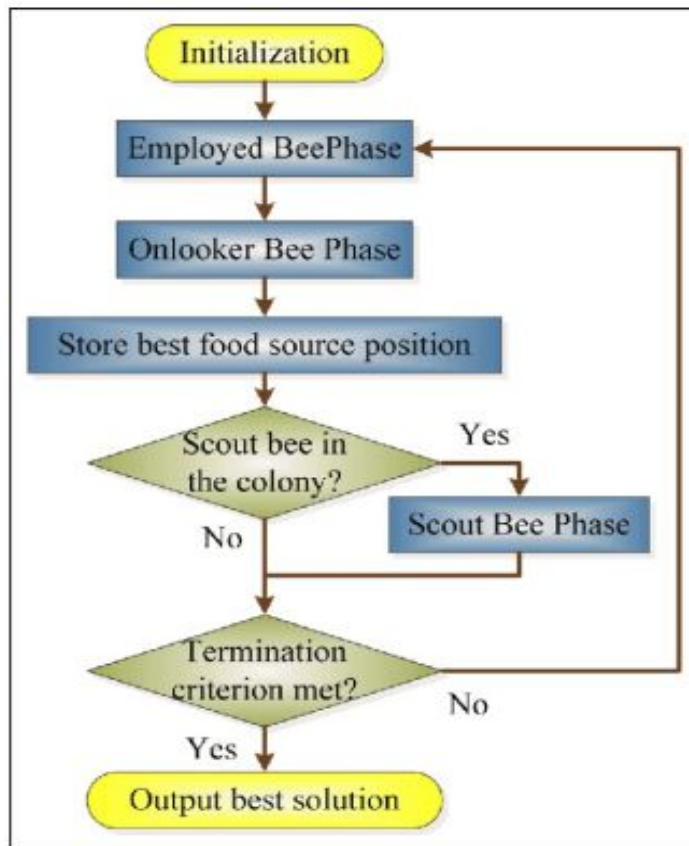
“The position of a **food source** represents a possible solution of the optimization problem”

The **nectar amount** of a food source corresponds to the quality (fitness) of the associated solution



*D. Karaboga, B. BasturkJ. A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm. Glob Optim (2007) 39:459–471.*

# Artificial Bee Colony (ABC)



# Artificial Bee Colony (ABC)



# Artificial Bee Colony (ABC)

```

Input:  $n, m, t_{max}, limit, NP, p_{ls}$ 
Output:  $S^*$ 
//Initialization phase
1 for  $i = 1$  to  $NP$  do
2    $S_i \leftarrow \text{Construct-Solution}(n, m);$ 
3   if  $U(0, 1) < p_{ls}$  then
4      $S_i \leftarrow \text{Local-Improvement}(S_i);$ 
5   end
6 end
7 while computation time  $t_{max}$  not reached do
  //Employed bees phase
  8   for  $i = 1$  to  $NP$  do
  9      $E \leftarrow \text{Generate-Neighbouring}(S_i);$ 
10    if  $U(0, 1) < p_{ls}$  then
11       $E \leftarrow \text{Local-Improvement}(E);$ 
12    end
13    if  $E$  is better than  $S_i$  then
14       $S_i \leftarrow E;$ 
15    end
16  end
```

# Artificial Bee Colony (ABC)

---

```
17 //Onlooker bees
18 for  $i=1$  to  $NP$  do
19      $j \leftarrow \text{Binary-Tournament}(1, \dots, NP)$ ;
20      $O \leftarrow \text{Generate-Neighbouring}(S_j)$ ;
21     if  $U(0, 1) < p_{ls}$  then
22          $O \leftarrow \text{Local-Improvement}(O)$ ;
23     end
24     if  $O$  is better than  $S_j$  then
25          $S_j \leftarrow O$ ;
26     end
27 end
```



# Artificial Bee Colony (ABC)

---

```
27      //Scout bees phase
28      for  $i=1$  to  $NP$  do
29          if  $S_i$  does not change for limit iterations then
30               $S_i \leftarrow \text{Construct-Solution}(n, m)$ ;
31              if  $U(0, 1) < p_{ls}$  then
32                   $S_i \leftarrow \text{Local-Improvement}(S_i)$ ;
33              end
34          end
35      //Remember the best food source found so far
36       $S^* \leftarrow \text{Best-Solution-Found}()$ ;
37  end
```

# Artificial Bee Colony (ABC)

---

1. M. Lozano, F.J. Rodríguez. Network Reconstruction from Betweenness Centrality by Artificial Bee Colony. *Swarm and Evolutionary Computation* 62 (2021) Num. art. 100851. (Q1 en 2021; Computer science, artificial intelligence). DOI: 10.1016/j.swevo.2021.100851
2. R. Goścień, M. Lozano. Artificial Bee Colony for Optimization of Cloud-Ready and Survivable Elastic Optical Networks. *Computer Communications* 128 (2018) 35-45. (Q2 en 2018; Computer science, information systems). DOI: 10.1016/j.comcom.2018.07.011
3. M. Lozano, C. García-Martínez, F.J. Rodríguez, H.M. Trujillo. Optimizing Network Attacks by Artificial Bee Colony. *Information Sciences* 377 (2017) 30-50. (Q1 en 2017; Computer science, information systems). DOI: 10.1016/j.ins.2016.10.014
4. J.A. Delgado-Osuna, M. Lozano, C. Garcia-Martinez. An Alternative Artificial Bee Colony Algorithm with Destructive-Constructive Neighbourhood Operator for the Problem of Composing Medical Crews. *Information Sciences* 326 (2016) 215-226. (Q1 en 2015; Computer science, information systems). DOI: 10.1016/j.ins.2015.07.051
5. F.J. Rodriguez, M. Lozano, C. García-Martínez, J.D. González-Barrera. An Artificial Bee Colony Algorithm for the Maximally Diverse Grouping Problem. *Information Sciences* 230:1 (2013) 183-196. (Q1 en 2013; Computer science, information systems). DOI: 10.1016/j.ins.2012.12.020

# Bibliografía Básica

---

- E. BONABEAU, M. DORIGO, G. THERAULAZ, **Swarm Intelligence. From Natural to Artificial Systems**, Oxford University Press, 1999.
- J. KENNEDY, R.C. EBERHART, Y. SHI, **Swarm Intelligence**. Morgan Kaufmann Publishers, 2001.
- M. DORIGO, T. STÜTZLE, **Ant Colony Optimization**, The MIT Press, 2004.

