

# Inteligencia de enjambres

Diego Milone  
Inteligencia Computacional  
Departamento de Informática

FICH-UNL

# Colonia de hormigas: introducción

Diego Milone  
Inteligencia Computacional  
Departamento de Informática

FICH-UNL

## La inspiración biológica

- Hay  $10^{16}$  hormigas en la tierra (y  $6 \times 10^9$  humanos)

## La inspiración biológica

- Hay  $10^{16}$  hormigas en la tierra (y  $6 \times 10^9$  humanos)
- Igual peso total

## La inspiración biológica

- Hay  $10^{16}$  hormigas en la tierra (y  $6 \times 10^9$  humanos)
- Igual peso total
- 30 millones por colonia

## La inspiración biológica

- Feromonas: la búsqueda del camino más corto a la comida

# La inspiración biológica

- Feromonas: la búsqueda del camino más corto a la comida
  1. Comportamiento inicial aleatorio

## La inspiración biológica

- Feromonas: la búsqueda del camino más corto a la comida
  1. Comportamiento inicial aleatorio
  2. Cuando encuentran una fuente de comida se organizan y comienzan a seguir el mismo camino



## La inspiración biológica

- Feromonas: la búsqueda del camino más corto a la comida
  1. Comportamiento inicial aleatorio
  2. Cuando encuentran una fuente de comida se organizan y comienzan a seguir el mismo camino
    - 2.1 Mecanismo de reclutamiento: mayormente por feromonas, liberadas al regresar (algunas las liberan en proporción a la cantidad de alimento encontrado)

## La inspiración biológica

- Feromonas: las búsqueda del camino más corto a la comida
  1. Comportamiento inicial aleatorio
  2. Cuando encuentran una fuente de comida se organizan y comienzan a seguir el mismo camino
    - 2.1 Mecanismo de reclutamiento: mayormente por feromonas, liberadas al regresar (algunas las liberan en proporción a la cantidad de alimento encontrado)
    - 2.2 Si otras encuentran feromonas siguen el rastro (con más probabilidad, tratando de alejarse del hormiguero si no llevan comida)

## La inspiración biológica

- Feromonas: la búsqueda del camino más corto a la comida
  1. Comportamiento inicial aleatorio
  2. Cuando encuentran una fuente de comida se organizan y comienzan a seguir el mismo camino
    - 2.1 Mecanismo de reclutamiento: mayormente por feromonas, liberadas al regresar (algunas las liberan en proporción a la cantidad de alimento encontrado)
    - 2.2 Si otras encuentran feromonas siguen el rastro (con más probabilidad, tratando de alejarse del hormiguero si no llevan comida)
    - 2.3 El camino se refuerza al ser seguido por más y más hormigas

## La inspiración biológica

- Feromonas: las búsqueda del camino más corto a la comida
  1. Comportamiento inicial aleatorio
  2. Cuando encuentran una fuente de comida se organizan y comienzan a seguir el mismo camino
    - 2.1 Mecanismo de reclutamiento: mayormente por feromonas, liberadas al regresar (algunas las liberan en proporción a la cantidad de alimento encontrado)
    - 2.2 Si otras encuentran feromonas siguen el rastro (con más probabilidad, tratando de alejarse del hormiguero si no llevan comida)
    - 2.3 El camino se refuerza al ser seguido por más y más hormigas
- Notar:
  - Comunicación indirecta
  - Modificación del entorno físico: estigmergía

# Experimento del puente binario

# Colonia de hormigas: algoritmos

Diego Milone  
Inteligencia Computacional  
Departamento de Informática

FICH-UNL

## Colonia de hormigas

### Definiciones:

- $G = (V, E)$ : grafo con vértices  $V$  y matriz de conexiones  $E$
- $\sigma_{ij}$ : feromonas en la conexión entre  $i$  y  $j$
- $k = 1, 2, \dots, N$ : hormigas
- $\mathcal{N}_i$ : nodos disponibles a partir del nodo  $i$
- $\mathbf{p}^k(t)$ : camino de la hormiga  $k$

## Colonia de hormigas

### Definiciones:

- $G = (V, E)$ : grafo con vértices  $V$  y matriz de conexiones  $E$
- $\sigma_{ij}$ : feromonas en la conexión entre  $i$  y  $j$
- $k = 1, 2, \dots, N$ : hormigas
- $N_i$ : nodos disponibles a partir del nodo  $i$
- $\mathbf{p}^k(t)$ : camino de la hormiga  $k$

### Notar:

- $t \rightarrow t + 1$ : se incrementa una vez que *todas* las hormigas encuentran el alimento y vuelven al origen
- $N_i^k$ : en algunos casos el entorno se limita a los nodos que no haya visitado previamente la hormiga  $k$



## Algoritmo 1: Colonia de hormigas simple (sACO)

1.  $t = 0$ , Inicializar feromonas con valores pequeños al azar ( $\sigma_{ij}(t) \leftarrow U(0, \sigma_0)$ )
2. Ubicar  $N$  hormigas en el nodo origen

# Algoritmo 1: Colonia de hormigas simple (sACO)

1.  $t = 0$ , Inicializar feromonas con valores pequeños al azar ( $\sigma_{ij}(t) \leftarrow U(0, \sigma_0)$ )
2. Ubicar  $N$  hormigas en el nodo origen
3. **repetir**
  - 3.1 **para** cada hormiga  $k = 1, 2, \dots, N$ 
    - 3.1.1  $\mathbf{p}^k(t) = \emptyset$

# Algoritmo 1: Colonia de hormigas simple (sACO)

1.  $t = 0$ , Inicializar feromonas con valores pequeños al azar ( $\sigma_{ij}(t) \leftarrow U(0, \sigma_0)$ )
2. Ubicar  $N$  hormigas en el nodo origen
3. **repetir**
  - 3.1 **para** cada hormiga  $k = 1, 2, \dots, N$ 
    - 3.1.1  $\mathbf{p}^k(t) = \emptyset$
    - 3.1.2 **repetir**
      - seleccionar el próximo nodo según la *probabilidad*

$$p_{ij}^k(t) = \begin{cases} \frac{\sigma_{ij}^\alpha(t)}{\sum_{u \in \mathcal{N}_i} \sigma_{iu}^\alpha(t)} & \text{si } j \in \mathcal{N}_i \\ 0 & \text{en otro caso.} \end{cases}$$

# Algoritmo 1: Colonia de hormigas simple (sACO)

1.  $t = 0$ , Inicializar feromonas con valores pequeños al azar ( $\sigma_{ij}(t) \leftarrow U(0, \sigma_0)$ )
2. Ubicar  $N$  hormigas en el nodo origen
3. **repetir**
  - 3.1 **para** cada hormiga  $k = 1, 2, \dots, N$ 
    - 3.1.1  $\mathbf{p}^k(t) = \emptyset$
    - 3.1.2 **repetir**
      - seleccionar el próximo nodo según la *probabilidad*  $p_{ij}^k(t) = \dots$
      - agregar un paso  $(i, j)$  al camino  $\mathbf{p}^k(t)$
    - hasta** *alcanzar el destino*
    - 3.1.3 eliminar los ciclos en  $\mathbf{p}^k(t)$
    - 3.1.4 calcular la longitud del camino encontrado  $f(\mathbf{p}^k(t))$

# Algoritmo 1: Colonia de hormigas simple (sACO)

1.  $t = 0$ , Inicializar feromonas con valores pequeños al azar ( $\sigma_{ij}(t) \leftarrow U(0, \sigma_0)$ )
2. Ubicar  $N$  hormigas en el nodo origen
3. **repetir**
  - 3.1 **para** cada hormiga  $k = 1, 2, \dots, N$ 
    - 3.1.1  $\mathbf{p}^k(t) = \emptyset$
    - 3.1.2 **repetir**
      - seleccionar el próximo nodo según la *probabilidad*  $p_{ij}^k(t) = \dots$
      - agregar un paso  $(i, j)$  al camino  $\mathbf{p}^k(t)$
    - hasta** *alcanzar el destino*
    - 3.1.3 eliminar los ciclos en  $\mathbf{p}^k(t)$
    - 3.1.4 calcular la longitud del camino encontrado  $f(\mathbf{p}^k(t))$
  - 3.2 **para** cada conexión  $(i, j)$ 
    - reducir por evaporación la cantidad de feromonas:  $\sigma_{ij}(t) \leftarrow (1 - \rho)\sigma_{ij}(t)$

# Algoritmo 1: Colonia de hormigas simple (sACO)

1.  $t = 0$ , Inicializar feromonas con valores pequeños al azar ( $\sigma_{ij}(t) \leftarrow U(0, \sigma_0)$ )
2. Ubicar  $N$  hormigas en el nodo origen
3. **repetir**
  - 3.1 **para** cada hormiga  $k = 1, 2, \dots, N$ 
    - 3.1.1  $\mathbf{p}^k(t) = \emptyset$
    - 3.1.2 **repetir**
      - seleccionar el próximo nodo según la *probabilidad*  $p_{ij}^k(t) = \dots$
      - agregar un paso  $(i, j)$  al camino  $\mathbf{p}^k(t)$
    - hasta** *alcanzar el destino*
    - 3.1.3 eliminar los ciclos en  $\mathbf{p}^k(t)$
    - 3.1.4 calcular la longitud del camino encontrado  $f(\mathbf{p}^k(t))$
  - 3.2 **para** cada conexión  $(i, j)$ 
    - reducir por evaporación la cantidad de feromonas:  $\sigma_{ij}(t) \leftarrow (1 - \rho)\sigma_{ij}(t)$
  - 3.3 **para** cada conexión  $(i, j)$ 
    - depositar feromonas proporcionalmente a la bondad de la solución

$$\sigma_{ij}(t+1) = \sigma_{ij}(t) + \sum_{\forall k/(i,j) \in \mathbf{p}^k(t)} 1/f(\mathbf{p}^k(t))$$

# Algoritmo 1: Colonia de hormigas simple (sACO)

1.  $t = 0$ , Inicializar feromonas con valores pequeños al azar ( $\sigma_{ij}(t) \leftarrow U(0, \sigma_0)$ )
2. Ubicar  $N$  hormigas en el nodo origen
3. **repetir**
  - 3.1 **para** cada hormiga  $k = 1, 2, \dots, N$ 
    - 3.1.1  $\mathbf{p}^k(t) = \emptyset$
    - 3.1.2 **repetir**
      - seleccionar el próximo nodo según la *probabilidad*  $p_{ij}^k(t) = \dots$
      - agregar un paso  $(i, j)$  al camino  $\mathbf{p}^k(t)$
    - hasta** *alcanzar el destino*
    - 3.1.3 eliminar los ciclos en  $\mathbf{p}^k(t)$
    - 3.1.4 calcular la longitud del camino encontrado  $f(\mathbf{p}^k(t))$
  - 3.2 **para** cada conexión  $(i, j)$ 
    - reducir por evaporación la cantidad de feromonas:  $\sigma_{ij}(t) \leftarrow (1 - \rho)\sigma_{ij}(t)$
  - 3.3 **para** cada conexión  $(i, j)$ 
    - depositar feromonas proporcionalmente a la bondad de la solución

$$\sigma_{ij}(t+1) = \sigma_{ij}(t) + \sum_{\forall k/(i,j) \in \mathbf{p}^k(t)} 1/f(\mathbf{p}^k(t))$$

3.4  $t \leftarrow t + 1$

**hasta** que *todas* las hormigas sigan el mismo camino

4. **devolver** el camino más corto ( $\downarrow f(\mathbf{p}^k(t))$ )

## Algoritmo 2: Sistema de hormigas (AS)

1.  $t = 0, \sigma_{ij}(t) \leftarrow U(0, \sigma_0)$
2. Ubicar  $N$  hormigas en el nodo origen



## Algoritmo 2: Sistema de hormigas (AS)

1.  $t = 0, \sigma_{ij}(t) \leftarrow U(0, \sigma_0)$
2. Ubicar  $N$  hormigas en el nodo origen
3. **repetir**
  - 3.1 **para** cada hormiga  $k = 1, 2, \dots, N$ 
    - 3.1.1  $\mathbf{p}^k(t) = \emptyset$

## Algoritmo 2: Sistema de hormigas (AS)

1.  $t = 0, \sigma_{ij}(t) \leftarrow U(0, \sigma_0)$
2. Ubicar  $N$  hormigas en el nodo origen

### 3. repetir

3.1 para cada hormiga  $k = 1, 2, \dots, N$

3.1.1  $\mathbf{p}^k(t) = \emptyset$

3.1.2 repetir

- seleccionar el próximo nodo según la *probabilidad*

$$p_{ij}^k(t) = \begin{cases} \frac{\sigma_{ij}^\alpha(t) \eta_{ij}^\beta(t)}{\sum_{\forall u \in \mathcal{N}_i^k} \sigma_{iu}^\alpha(t) \eta_{ij}^\beta(t)} & \text{si } j \in \mathcal{N}_i^k \\ 0 & \text{en otro caso.} \end{cases}$$

- Deseo de moverse inverso al costo entre nodos:  $\eta_{ij} = 1/d_{ij}$ .
- Lista de nodos vecinos con tabú:  $\mathcal{N}_i^k$

## Algoritmo 2: Sistema de hormigas (AS)

1.  $t = 0, \sigma_{ij}(t) \leftarrow U(0, \sigma_0)$
2. Ubicar  $N$  hormigas en el nodo origen
3. **repetir**
  - 3.1 **para** cada hormiga  $k = 1, 2, \dots, N$ 
    - 3.1.1  $\mathbf{p}^k(t) = \emptyset$
    - 3.1.2 **repetir**
      - seleccionar el próximo nodo según la *probabilidad*  $p_{ij}^k(t) = \dots$
      - agregar un paso  $(i, j)$  al camino  $\mathbf{p}^k(t)$
    - hasta** *alcanzar el destino*
    - 3.1.3 calcular la longitud del camino encontrado  $f(\mathbf{p}^k(t))$

## Algoritmo 2: Sistema de hormigas (AS)

1.  $t = 0, \sigma_{ij}(t) \leftarrow U(0, \sigma_0)$
2. Ubicar  $N$  hormigas en el nodo origen
3. **repetir**
  - 3.1 **para** cada hormiga  $k = 1, 2, \dots, N$ 
    - 3.1.1  $\mathbf{p}^k(t) = \emptyset$
    - 3.1.2 **repetir**
      - seleccionar el próximo nodo según la *probabilidad*  $p_{ij}^k(t) = \dots$
      - agregar un paso  $(i, j)$  al camino  $\mathbf{p}^k(t)$
    - hasta** *alcanzar el destino*
    - 3.1.3 calcular la longitud del camino encontrado  $f(\mathbf{p}^k(t))$
  - 3.2 **para** cada conexión  $(i, j)$ 
    - reducir por evaporación la cantidad de feromonas:  $\sigma_{ij}(t) \leftarrow (1 - \rho)\sigma_{ij}(t)$

## Algoritmo 2: Sistema de hormigas (AS)

1.  $t = 0, \sigma_{ij}(t) \leftarrow U(0, \sigma_0)$
2. Ubicar  $N$  hormigas en el nodo origen
3. **repetir**
  - 3.1 **para** cada hormiga  $k = 1, 2, \dots, N$ 
    - 3.1.1  $\mathbf{p}^k(t) = \emptyset$
    - 3.1.2 **repetir**
      - seleccionar el próximo nodo según la *probabilidad*  $p_{ij}^k(t) = \dots$
      - agregar un paso  $(i, j)$  al camino  $\mathbf{p}^k(t)$
    - hasta** *alcanzar el destino*
    - 3.1.3 calcular la longitud del camino encontrado  $f(\mathbf{p}^k(t))$
  - 3.2 **para** cada conexión  $(i, j)$ 
    - reducir por evaporación la cantidad de feromonas:  $\sigma_{ij}(t) \leftarrow (1 - \rho)\sigma_{ij}(t)$
    - depositar feromonas proporcionalmente a la bondad de la solución

$$\Delta\sigma_{ij}^k(t) = \begin{cases} Q/f(\mathbf{p}^k(t)) & \text{global.} \\ Q & \text{uniforme.} \\ Q/d_{ij} & \text{local.} \end{cases}$$

## Algoritmo 2: Sistema de hormigas (AS)

1.  $t = 0, \sigma_{ij}(t) \leftarrow U(0, \sigma_0)$
2. Ubicar  $N$  hormigas en el nodo origen
3. **repetir**
  - 3.1 **para** cada hormiga  $k = 1, 2, \dots, N$ 
    - 3.1.1  $\mathbf{p}^k(t) = \emptyset$
    - 3.1.2 **repetir**
      - seleccionar el próximo nodo según la *probabilidad*  $p_{ij}^k(t) = \dots$
      - agregar un paso  $(i, j)$  al camino  $\mathbf{p}^k(t)$
    - hasta** *alcanzar el destino*
    - 3.1.3 calcular la longitud del camino encontrado  $f(\mathbf{p}^k(t))$
  - 3.2 **para** cada conexión  $(i, j)$ 
    - reducir por evaporación la cantidad de feromonas:  $\sigma_{ij}(t) \leftarrow (1 - \rho)\sigma_{ij}(t)$
    - depositar feromonas proporcionalmente a la bondad de la solución

$$\Delta\sigma_{ij}^k(t) = \begin{cases} Q/f(\mathbf{p}^k(t)) & \text{global.} \\ Q & \text{uniforme.} \\ Q/d_{ij} & \text{local.} \end{cases}$$

$$\sigma_{ij}(t+1) = \sigma_{ij}(t) + \sum_{\forall k/(i,j) \in \mathbf{p}^k(t)} \Delta\sigma_{ij}^k(t)$$

## Algoritmo 2: Sistema de hormigas (AS)

1.  $t = 0, \sigma_{ij}(t) \leftarrow U(0, \sigma_0)$
2. Ubicar  $N$  hormigas en el nodo origen
3. **repetir**
  - 3.1 **para** cada hormiga  $k = 1, 2, \dots, N$ 
    - 3.1.1  $\mathbf{p}^k(t) = \emptyset$
    - 3.1.2 **repetir**
      - seleccionar el próximo nodo según la *probabilidad*  $p_{ij}^k(t) = \dots$
      - agregar un paso  $(i, j)$  al camino  $\mathbf{p}^k(t)$**hasta** *alcanzar el destino*
    - 3.1.3 calcular la longitud del camino encontrado  $f(\mathbf{p}^k(t))$
  - 3.2 **para** cada conexión  $(i, j)$ 
    - reducir por evaporación la cantidad de feromonas:  $\sigma_{ij}(t) \leftarrow (1 - \rho)\sigma_{ij}(t)$
    - depositar feromonas proporcionalmente a la bondad de la solución

$$\Delta\sigma_{ij}^k(t) = \begin{cases} Q/f(\mathbf{p}^k(t)) & \text{global.} \\ Q & \text{uniforme.} \\ Q/d_{ij} & \text{local.} \end{cases}$$

$$\sigma_{ij}(t+1) = \sigma_{ij}(t) + \sum_{\forall k/(i,j) \in \mathbf{p}^k(t)} \Delta\sigma_{ij}^k(t)$$

- 3.3  $t \leftarrow t + 1$   
**hasta** que *todas* las hormigas sigan el mismo camino

4. **devolver** el mejor camino

# Enjambre de partículas

Diego Milone  
Inteligencia Computacional  
Departamento de Informática

FICH-UNL



## La inspiración biológica

- Bandadas de pájaros
- Cada individuo vuela/navega el espacio ajustando su posición en base a su experiencia y a la información de los individuos de su entorno

## La inspiración biológica

- Bandadas de pájaros
- Cada individuo vuela/navega el espacio ajustando su posición en base a su experiencia y a la información de los individuos de su entorno
- Emular el éxito de los vecinos y propio
  - $\mathbf{x}_k(t)$ : posición de la partícula  $k$ , en el tiempo  $t$  (espacio  $\mathbb{R}^N$ )
  - $\mathbf{v}_k(t)$ : velocidad de la partícula  $k$ , en el tiempo  $t$

## La inspiración biológica

- Bandadas de pájaros
- Cada individuo vuela/navega el espacio ajustando su posición en base a su experiencia y a la información de los individuos de su entorno
- Emular el éxito de los vecinos y propio
  - $\mathbf{x}_k(t)$ : posición de la partícula  $k$ , en el tiempo  $t$  (espacio  $\mathbb{R}^N$ )
  - $\mathbf{v}_k(t)$ : velocidad de la partícula  $k$ , en el tiempo  $t$
  - Regla básica:  $\mathbf{x}_k(t + 1) = \mathbf{x}_k(t) + \mathbf{v}_k(t)$

## Enjambre de partículas

### Definiciones:

- $\mathbf{y}_k$ : mejor posición *personal* de la partícula  $k$  (la mejor que visitó desde  $t = 0$  hasta la actualidad)
- $f(\mathbf{x})$ : función de *error* a minimizar

## Enjambre de partículas

### Definiciones:

- $\mathbf{y}_k$ : mejor posición *personal* de la partícula  $k$  (la mejor que visitó desde  $t = 0$  hasta la actualidad)
- $f(\mathbf{x})$ : función de *error* a minimizar

### Mejor global (gEP):

- Topología estrella: usa información de todo el enjambre
- $\hat{\mathbf{y}}$ : mejor posición visitada por todo el enjambre

## Algoritmo 3: Enjambre del mejor global (gEP)

1. Inicializar  $\mathbf{x}_{ki}(0) \leftarrow U(\mathbf{x}_i^{min}, \mathbf{x}_i^{max})$

## Algoritmo 3: Enjambre del mejor global (gEP)

1. Inicializar  $\mathbf{x}_{ki}(0) \leftarrow U(\mathbf{x}_i^{min}, \mathbf{x}_i^{max})$
2. **repetir**
  - 2.1 **para** cada partícula  $k = 1, 2, \dots, N$ 
    - **si**  $f(\mathbf{x}_k(t)) < f(\mathbf{y}_k) \rightarrow \mathbf{y}_k = \mathbf{x}_k(t)$

## Algoritmo 3: Enjambre del mejor global (gEP)

1. Inicializar  $\mathbf{x}_{ki}(0) \leftarrow U(\mathbf{x}_i^{min}, \mathbf{x}_i^{max})$
2. **repetir**
  - 2.1 **para** cada partícula  $k = 1, 2, \dots, N$ 
    - **si**  $f(\mathbf{x}_k(t)) < f(\mathbf{y}_k) \rightarrow \mathbf{y}_k = \mathbf{x}_k(t)$
    - **si**  $f(\mathbf{y}_k) < f(\hat{\mathbf{y}}) \rightarrow \hat{\mathbf{y}} = \mathbf{y}_k$



## Algoritmo 3: Enjambre del mejor global (gEP)

1. Inicializar  $\mathbf{x}_{ki}(0) \leftarrow U(\mathbf{x}_i^{min}, \mathbf{x}_i^{max})$
2. **repetir**
  - 2.1 **para** cada partícula  $k = 1, 2, \dots, N$ 
    - **si**  $f(\mathbf{x}_k(t)) < f(\mathbf{y}_k) \rightarrow \mathbf{y}_k = \mathbf{x}_k(t)$
    - **si**  $f(\mathbf{y}_k) < f(\hat{\mathbf{y}}) \rightarrow \hat{\mathbf{y}} = \mathbf{y}_k$
  - 2.2 **para** cada partícula  $k = 1, 2, \dots, N$ 
    - $v_{ki}(t + 1) = v_{ki}(t) + c_1 r_{1i} [y_{ki} - x_{ki}(t)]$

## Algoritmo 3: Enjambre del mejor global (gEP)

1. Inicializar  $\mathbf{x}_{ki}(0) \leftarrow U(\mathbf{x}_i^{min}, \mathbf{x}_i^{max})$
2. **repetir**
  - 2.1 **para** cada partícula  $k = 1, 2, \dots, N$ 
    - **si**  $f(\mathbf{x}_k(t)) < f(\mathbf{y}_k) \rightarrow \mathbf{y}_k = \mathbf{x}_k(t)$
    - **si**  $f(\mathbf{y}_k) < f(\hat{\mathbf{y}}) \rightarrow \hat{\mathbf{y}} = \mathbf{y}_k$
  - 2.2 **para** cada partícula  $k = 1, 2, \dots, N$ 
    - $$v_{ki}(t + 1) = v_{ki}(t) + c_1 r_{1i} [y_{ki} - x_{ki}(t)] + c_2 r_{2i} [\hat{y}_i - x_{ki}(t)]$$

## Algoritmo 3: Enjambre del mejor global (gEP)

1. Inicializar  $\mathbf{x}_{ki}(0) \leftarrow U(\mathbf{x}_i^{min}, \mathbf{x}_i^{max})$

2. **repetir**

2.1 **para** cada partícula  $k = 1, 2, \dots, N$

- **si**  $f(\mathbf{x}_k(t)) < f(\mathbf{y}_k) \rightarrow \mathbf{y}_k = \mathbf{x}_k(t)$
- **si**  $f(\mathbf{y}_k) < f(\hat{\mathbf{y}}) \rightarrow \hat{\mathbf{y}} = \mathbf{y}_k$

2.2 **para** cada partícula  $k = 1, 2, \dots, N$

- $$v_{ki}(t+1) = v_{ki}(t) + c_1 r_{1i} [y_{ki} - x_{ki}(t)]$$

$$+ c_2 r_{2i} [\hat{y}_i - x_{ki}(t)]$$

$$r_{1i}, r_{2i} \leftarrow U(0, 1)$$

## Algoritmo 3: Enjambre del mejor global (gEP)

1. Inicializar  $\mathbf{x}_{ki}(0) \leftarrow U(\mathbf{x}_i^{min}, \mathbf{x}_i^{max})$
2. **repetir**
  - 2.1 **para** cada partícula  $k = 1, 2, \dots, N$ 
    - **si**  $f(\mathbf{x}_k(t)) < f(\mathbf{y}_k) \rightarrow \mathbf{y}_k = \mathbf{x}_k(t)$
    - **si**  $f(\mathbf{y}_k) < f(\hat{\mathbf{y}}) \rightarrow \hat{\mathbf{y}} = \mathbf{y}_k$
  - 2.2 **para** cada partícula  $k = 1, 2, \dots, N$ 
    - $$\begin{aligned} v_{ki}(t+1) = & v_{ki}(t) + c_1 r_{1i} [y_{ki} - x_{ki}(t)] \\ & + c_2 r_{2i} [\hat{y}_i - x_{ki}(t)] \\ & r_{1i}, r_{2i} \leftarrow U(0, 1) \end{aligned}$$
    - $\mathbf{x}_k(t+1) = \mathbf{x}_k(t) + \mathbf{v}_k(t+1)$
- hasta** cumplir con la condición de finalización
3. **devolver** la mejor partícula encontrada

## Enjambre de partículas

### Definiciones:

- $\mathbf{y}_k$ : mejor posición *personal* de la partícula  $k$  (la mejor que visitó desde  $t = 0$  hasta la actualidad)
- $f(\mathbf{x})$ : función de *error* a minimizar

### Mejor local ( $\ell$ EP):

- Topología anillo: usa información del entorno cercano a la partícula
- Selección: basada en los índices de las partículas (caso más simple)

$$\mathcal{E}_k = \{\mathbf{y}_{k-n}, \mathbf{y}_{k-n+1}, \dots, \mathbf{y}_{k-1}, \mathbf{y}_k, \mathbf{y}_{k+1}, \dots, \mathbf{y}_{k+n}\}:$$

## Enjambre de partículas

### Definiciones:

- $\mathbf{y}_k$ : mejor posición *personal* de la partícula  $k$  (la mejor que visitó desde  $t = 0$  hasta la actualidad)
- $f(\mathbf{x})$ : función de *error* a minimizar

### Mejor local ( $\ell$ EP):

- Topología anillo: usa información del entorno cercano a la partícula
- Selección: basada en los índices de las partículas (caso más simple)

$$\mathcal{E}_k = \{\mathbf{y}_{k-n}, \mathbf{y}_{k-n+1}, \dots, \mathbf{y}_{k-1}, \mathbf{y}_k, \mathbf{y}_{k+1}, \dots, \mathbf{y}_{k+n}\}:$$

- $\hat{\mathbf{y}}_k = \arg \min_{\mathbf{y}_u \in \mathcal{E}_k} \{f(\mathbf{y}_u)\}$  mejor posición visitada por el entorno de la partícula  $k$

## Algoritmo 4: Enjambre del mejor local ( $\ell$ EP)

1. Inicializar  $\mathbf{x}_{ki}(0) \leftarrow U(\mathbf{x}_i^{\min}, \mathbf{x}_i^{\max})$

## Algoritmo 4: Enjambre del mejor local ( $\ell$ EP)

1. Inicializar  $\mathbf{x}_{ki}(0) \leftarrow U(\mathbf{x}_i^{min}, \mathbf{x}_i^{max})$
2. **repetir**
  - 2.1 **para** cada partícula  $k = 1, 2, \dots, N$ 
    - **si**  $f(\mathbf{x}_k(t)) < f(\mathbf{y}_k) \rightarrow \mathbf{y}_k = \mathbf{x}_k(t)$



## Algoritmo 4: Enjambre del mejor local ( $\ell$ EP)

1. Inicializar  $\mathbf{x}_{ki}(0) \leftarrow U(\mathbf{x}_i^{min}, \mathbf{x}_i^{max})$
2. **repetir**
  - 2.1 **para** cada partícula  $k = 1, 2, \dots, N$ 
    - **si**  $f(\mathbf{x}_k(t)) < f(\mathbf{y}_k) \rightarrow \mathbf{y}_k = \mathbf{x}_k(t)$
    - **si**  $f(\mathbf{y}_k) < f(\hat{\mathbf{y}}) \rightarrow \hat{\mathbf{y}} = \mathbf{y}_k$

## Algoritmo 4: Enjambre del mejor local ( $\ell$ EP)

1. Inicializar  $\mathbf{x}_{ki}(0) \leftarrow U(\mathbf{x}_i^{min}, \mathbf{x}_i^{max})$
2. **repetir**
  - 2.1 **para** cada partícula  $k = 1, 2, \dots, N$ 
    - **si**  $f(\mathbf{x}_k(t)) < f(\mathbf{y}_k) \rightarrow \mathbf{y}_k = \mathbf{x}_k(t)$
    - **si**  $f(\mathbf{y}_k) < f(\hat{\mathbf{y}}) \rightarrow \hat{\mathbf{y}} = \mathbf{y}_k$
  - 2.2 **para** cada partícula  $k = 1, 2, \dots, N$ 
    - $v_{ki}(t+1) = v_{ki}(t) + c_1 r_{1i} [y_{ki} - x_{ki}(t)]$

## Algoritmo 4: Enjambre del mejor local ( $\ell$ EP)

1. Inicializar  $\mathbf{x}_{ki}(0) \leftarrow U(\mathbf{x}_i^{\min}, \mathbf{x}_i^{\max})$
2. **repetir**
  - 2.1 **para** cada partícula  $k = 1, 2, \dots, N$ 
    - **si**  $f(\mathbf{x}_k(t)) < f(\mathbf{y}_k) \rightarrow \mathbf{y}_k = \mathbf{x}_k(t)$
    - **si**  $f(\mathbf{y}_k) < f(\hat{\mathbf{y}}) \rightarrow \hat{\mathbf{y}} = \mathbf{y}_k$
  - 2.2 **para** cada partícula  $k = 1, 2, \dots, N$ 
    - $$v_{ki}(t+1) = v_{ki}(t) + c_1 r_{1i} [y_{ki} - x_{ki}(t)] + c_2 r_{2i} [\hat{y}_{ki} - x_{ki}(t)]$$

## Algoritmo 4: Enjambre del mejor local ( $\ell$ EP)

1. Inicializar  $\mathbf{x}_{ki}(0) \leftarrow U(\mathbf{x}_i^{\min}, \mathbf{x}_i^{\max})$
2. **repetir**
  - 2.1 **para** cada partícula  $k = 1, 2, \dots, N$ 
    - **si**  $f(\mathbf{x}_k(t)) < f(\mathbf{y}_k) \rightarrow \mathbf{y}_k = \mathbf{x}_k(t)$
    - **si**  $f(\mathbf{y}_k) < f(\hat{\mathbf{y}}) \rightarrow \hat{\mathbf{y}} = \mathbf{y}_k$
  - 2.2 **para** cada partícula  $k = 1, 2, \dots, N$ 
    - $$v_{ki}(t+1) = v_{ki}(t) + c_1 r_{1i} [y_{ki} - x_{ki}(t)] + c_2 r_{2i} [\hat{y}_{ki} - x_{ki}(t)]$$

$$r_{1i}, r_{2i} \leftarrow U(0, 1)$$

$$\hat{\mathbf{y}}_k = \arg \min_{\mathbf{y}_u \in \mathcal{E}_k} \{f(\mathbf{y}_u)\}$$

## Algoritmo 4: Enjambre del mejor local ( $\ell$ EP)

1. Inicializar  $\mathbf{x}_{ki}(0) \leftarrow U(\mathbf{x}_i^{\min}, \mathbf{x}_i^{\max})$
2. **repetir**
  - 2.1 **para** cada partícula  $k = 1, 2, \dots, N$ 
    - **si**  $f(\mathbf{x}_k(t)) < f(\mathbf{y}_k) \rightarrow \mathbf{y}_k = \mathbf{x}_k(t)$
    - **si**  $f(\mathbf{y}_k) < f(\hat{\mathbf{y}}) \rightarrow \hat{\mathbf{y}} = \mathbf{y}_k$
  - 2.2 **para** cada partícula  $k = 1, 2, \dots, N$ 
    - $$\begin{aligned} \mathbf{v}_{ki}(t+1) = & \mathbf{v}_{ki}(t) + c_1 r_{1i} [\mathbf{y}_{ki} - \mathbf{x}_{ki}(t)] \\ & + c_2 r_{2i} [\hat{\mathbf{y}}_{ki} - \mathbf{x}_{ki}(t)] \\ & r_{1i}, r_{2i} \leftarrow U(0, 1) \\ & \hat{\mathbf{y}}_k = \arg \min_{\mathbf{y}_u \in \mathcal{E}_k} \{f(\mathbf{y}_u)\} \end{aligned}$$
    - $\mathbf{x}_k(t+1) = \mathbf{x}_k(t) + \mathbf{v}_k(t+1)$

**hasta** cumplir con la condición de finalización

3. **devolver** la mejor partícula encontrada