

# OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS (PSO)

RESOLUCIÓN: PROBLEMA DEL VIAJANTE DE COMERCIO (ATSP)

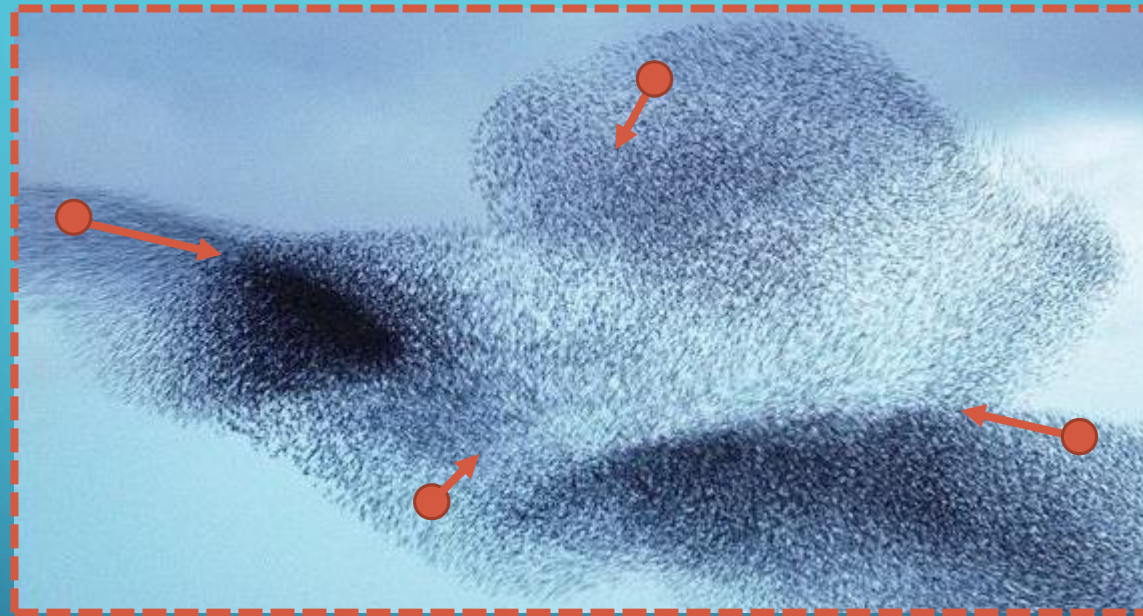
LABAYEN, FRANCO  
WALS OCHOA, LUCAS

# ÍNDICE DE TEMAS A TRATAR

1. Principios básicos del Algoritmo PSO
2. Representación del problema ATSP
3. Adaptación del Algoritmo PSO al problema ATSP: Optimización por enjambre de partículas para permutación de enteros (PSOP)
4. Resultados obtenidos
5. Conclusiones

# 1. PRINCIPIOS BÁSICOS DEL ALGORITMO PSO

## 1.1 ANALOGÍA CON LA NATURALEZA



- ☐ Enjambre / Conjunto de **Partículas**
- Posición de una **partícula** / abeja
- ↑ **Velocidad** / Rapidez y dirección

# 1. PRINCIPIOS BÁSICOS DEL ALGORITMO PSO

## 1.2 OBJETIVO Y LIMITACIONES

- Objetivo
  - Optimizar (minimizar) una función objetivo de tipo  $f: R^n \rightarrow R$ .
  - ¿Cómo? Haciendo que el enjambre converja a una solución rápidamente.
- Limitaciones
  - No garantiza encontrar la mejor solución
  - No fue diseñado originalmente para dominios discretos

# 1. PRINCIPIOS BÁSICOS DEL ALGORITMO PSO

## 1.3 ESTRUCTURAS

### Partícula

- Una **posición**  $x \in R^n$ .
- Una mejor **posición local**  $pBest \in R^n$ .
  - Una **velocidad**  $v \in R^n$ .
  - Un valor  $fitness_x \in R$ .
  - Un valor  $fitness\_pBest \in R$ .

### Enjambre

- Un conjunto de partículas.
- Una mejor **posición global**  $gBest \in R^n$ .
  - Un valor  $fitness\_gBest \in R$ .

# 1. PRINCIPIOS BÁSICOS DEL ALGORITMO PSO

## 1.4 FUNCIONAMIENTO

Generación aleatoria  
de **posiciones** y  
**velocidades**



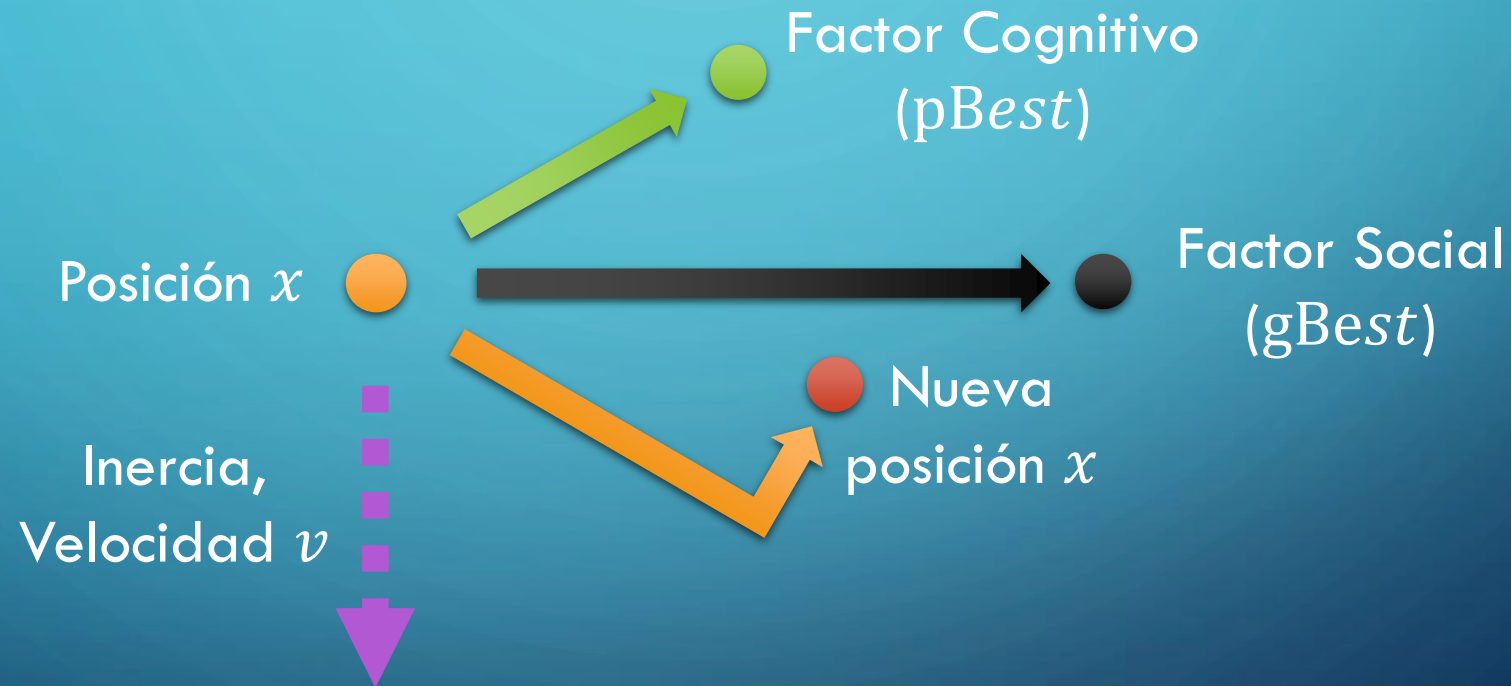
Criterio de  
Parada





# 1. PRINCIPIOS BÁSICOS DEL ALGORITMO PSO

## 1.4 FUNCIONAMIENTO



# 1. PRINCIPIOS BÁSICOS DEL ALGORITMO PSO

## 1.5 SISTEMA MATEMÁTICO: ECUACIONES

Operador Actualización de velocidad

$$\mathbf{v}_i^t = \underbrace{\omega \cdot v_i^{t-1}}_{\text{Factor de inercia}} + \underbrace{\varphi_1 \cdot r_1 \cdot (pBest_i - x_i^{t-1})}_{\text{Factor cognitivo}} + \underbrace{\varphi_2 \cdot r_2 \cdot (gBest - x_i^{t-1})}_{\text{Factor social}}$$

Factor de  
inercia

Factor  
cognitivo

Factor  
social

Operador Movimiento

$$\mathbf{x}_i^t = x_i^{t-1} + \mathbf{v}_i^t$$



# 1. PRINCIPIOS BÁSICOS DEL ALGORITMO PSO

## 1.5 CLASIFICACIÓN

- PSO Global

- La búsqueda es guiada por un único punto global.
- Es en el cual basamos la implementación.



- PSO Local

- Añade el concepto de vecindarios.
- La búsqueda es guiada por varios puntos globales, uno por cada vecindario.



# 1. PRINCIPIOS BÁSICOS DEL ALGORITMO PSO

## 1.6 PSEUDOCÓDIGO

Para cada partícula  $i = 1, \dots$ , hasta el total de partículas:

- Inicializar la posición  $x_i \in R^n$  aleatoriamente.
- Inicializar la velocidad  $v_i \in R^n$  aleatoriamente.
- Inicializar la posición  $pBest_i \in R^n$  con el valor de  $x_i$ :  $pBest_i \leftarrow x_i$
- Inicializar  $fitness\_x_i \in R$  de la siguiente forma:  $fitness\_x_i \leftarrow f(x_i)$
- Inicializar  $fitness\_pBest_i \in R$  de la siguiente forma:  $fitness\_pBest_i \leftarrow fitness\_x_i$
- Si ( $fitness\_pBest_i < fitness\_gBest$ ) entonces:
  - Actualizar el mejor resultado global:  $fitness\_gBest \leftarrow fitness\_pBest_i$
  - Actualizar la mejor posición global:  $gBest \leftarrow pBest_i$

Mientras no se alcance el límite máximo de iteraciones, o no se cumpla el criterio de parada:

- Para cada partícula  $i = 1, \dots$ , hasta el total de partículas:
  - Actualizar la velocidad  $v_i$ : 
$$v_i \leftarrow \omega \cdot v_i + \varphi_1 \cdot rand_1 \cdot (pBest_i - x_i) + \varphi_2 \cdot rand_2 \cdot (gBest - x_i)$$
  - Trasladar la posición  $x_i$ :  $x_i \leftarrow x_i + v_i$
- Para cada partícula  $i = 1, \dots$ , hasta el total de partículas:
  - Reevaluar  $fitness_{x_i}$ :  $fitness_{x_i} \leftarrow f(x_i)$
  - Si ( $fitness_{x_i} < fitness_{pBest_i}$ ) entonces:
    - Actualizar el mejor resultado local:  $fitness_{pBest_i} \leftarrow fitness_{x_i}$
    - Actualizar la mejor posición local:  $pBest_i$ :  $pBest_i \leftarrow x_i$
    - Si ( $fitness_{pBest_i} < fitness_{gBest}$ ) entonces:
      - Actualizar el mejor resultado global:  $fitness_{gBest} \leftarrow fitness_{pBest_i}$
      - Actualizar la mejor posición global:  $gBest \leftarrow pBest_i$

Devolver  $gBest$  como la mejor solución.

## 2. REPRESENTACIÓN DEL PROBLEMA ATSP

### 2.1 INSTANCIA

- Un problema en particular será representado por una matriz cuadrada asimétrica que llamaremos  $C$ .

- Consideraciones

- Dimensión = Cantidad de Ciudades.
- El costo de ir de una ciudad  $i$  a otra  $j$ , está dado por el coeficiente  $C_{ij}$ .

| C | 0        | 1        | 2        | 3        | 4        |
|---|----------|----------|----------|----------|----------|
| 0 | $\infty$ | 42       | 11       | 22       | 34       |
| 1 | 29       | $\infty$ | 95       | 19       | 15       |
| 2 | 23       | 43       | $\infty$ | 11       | 51       |
| 3 | 80       | 28       | 10       | $\infty$ | 45       |
| 4 | 10       | 17       | 74       | 89       | $\infty$ |

## 2. REPRESENTACIÓN DEL PROBLEMA ATSP

### 2.2 TOUR

- Camino:  $0 \rightarrow 3 \rightarrow 1 \rightarrow 4 \rightarrow 2 \rightarrow 0$



- Representación:  $[0, 3, 1, 4, 2]$
- Condiciones
  - Longitud de lista ( $n$ ) = Dimensión de la matriz  $C$ .
  - Números enteros comprendidos  $[0, n - 1]$ .
  - Números no repetidos.



# 3. ADAPTACIÓN DEL ALGORITMO PSO (PSOP)

## 3.1 FUNDAMENTOS

- PSO original no nos permite trabajar con entornos discretos:
  - No es viable con la representación planteada.
- Surge la necesidad de formular:
  - Nuevas estructuras.
  - Un nuevo sistema matemático.





# 3. ADAPTACIÓN DEL ALGORITMO PSO (PSOP)

## 3.2 CAMBIOS ESTRUCTURALES

### Posición

Las posiciones de una partícula estarán formadas por una lista de N posibles valores enteros sin que existan repeticiones u omisiones.

$$x = [4, 3, 2, 0, 5, 1]$$

**Una posición representará un tour**

# 3. ADAPTACIÓN DEL ALGORITMO PSO (PSOP)

## 3.2 CAMBIOS ESTRUCTURALES

### Velocidad

Las velocidades estarán representadas por una lista de listas, donde cada sublista representará un par de enteros  $[i \rightarrow j]$ .

$$v = [[9 \rightarrow 1], [4 \rightarrow 6], [1 \rightarrow 5]]$$

**Una velocidad simbolizará un conjunto de intercambios o permutaciones a realizar sobre los elementos de una posición**

# 3. ADAPTACIÓN DEL ALGORITMO PSO (PSOP)

## 3.3 CAMBIOS EN LAS ECUACIONES

Operador Actualización de Velocidad

$$v_i^t = \omega \otimes v_i^{t-1} \circ \varphi_1 \otimes (pBest_i \theta x_i^{t-1}) \circ \varphi_2 \otimes (gBest \theta x_i^{t-1})$$

### Nuevos Operadores

Operador Movimiento

$$x_i^t = x_i^{t-1} \oplus v_i^t$$

# 3. ADAPTACIÓN DEL ALGORITMO PSO (PSOP)

## 3.3 CAMBIOS EN LAS ECUACIONES: OPERADOR $\theta$

Resta de posiciones


$$[1,2,3,4,5,6] \theta [5,2,4,6,3,1]$$


=

$$[[5 \rightarrow 1], [2 \rightarrow 2], [4 \rightarrow 3], [6 \rightarrow 4], [3 \rightarrow 5], [1 \rightarrow 6]]$$

### 3. ADAPTACIÓN DEL ALGORITMO PSO (PSOP)

#### 3.3 CAMBIOS EN LAS ECUACIONES: OPERADOR

Suma de velocidades


$$\begin{aligned} & [[5 \rightarrow 1], [2 \rightarrow 2]] \circ [[1 \rightarrow 1], [5 \rightarrow 1], [2 \rightarrow 4]] \\ & \quad = \\ & \quad [[5 \rightarrow 1], [2 \rightarrow 4]] \end{aligned}$$

## 3. ADAPTACIÓN DEL ALGORITMO PSO (PSOP)

### 3.3 CAMBIOS EN LAS ECUACIONES: OPERADOR

Producto Coeficiente  $\varphi$  Velocidad  $v$

- Si  $\varphi$  se encuentra entre  $[0,1]$ , se obtiene  $\varphi' = rand(0,1)$  y se compara:

- $\varphi' < \varphi \Rightarrow [i \rightarrow j] = [i \rightarrow i]$
- $\varphi' \geq \varphi \Rightarrow [i \rightarrow j] = [i \rightarrow j]$

- Si  $\varphi > 1$ , tal que  $\varphi = k + \varphi'$ , con  $k$  entero y  $\varphi' < 1$ :

$$\underbrace{v \circ v \circ \dots \circ v}_k \circ \varphi' \otimes v$$

$k$  veces



### 3. ADAPTACIÓN DEL ALGORITMO PSO (PSOP)

#### 3.3 CAMBIOS EN LAS ECUACIONES: OPERADOR $\otimes$

Producto Coeficiente  $\varphi$  Velocidad  $v$

$$0.5 \otimes [[1 \rightarrow 3], [2 \rightarrow 5], [4 \rightarrow 4], [3 \rightarrow 2]]$$

=

$$[[1 \rightarrow 1], [2 \rightarrow 5], [4 \rightarrow 4], [3 \rightarrow 3]]$$

Para valores de  $\varphi' = 0,2 - 0,8 - 0,5 - 0,3$

$$\varphi' < \varphi \Rightarrow [i \rightarrow j] = [i \rightarrow i]$$

$$\varphi' \geq \varphi \Rightarrow [i \rightarrow j] = [i \rightarrow j]$$

### 3. ADAPTACIÓN DEL ALGORITMO PSO (PSOP)

#### 3.3 CAMBIOS EN LAS ECUACIONES: OPERADOR $\oplus$

Suma de Posición con Velocidad

$$[2,4,3,6,1,5] \oplus [[3 \rightarrow 1], [6 \rightarrow 6], [1 \rightarrow 5]]$$

=

$$[2,4,1,6,3,5] \xrightarrow{\text{orange arrow}} [2,4,1,6,3,5] \xrightarrow{\text{orange arrow}} [2,4,5,6,3,1]$$

$$\begin{array}{c} \text{orange arrow} \\ \text{red arrow} \\ [3 \rightarrow 1] \end{array}$$

$$\begin{array}{c} \text{orange arrow} \\ \text{red arrow} \\ [6 \rightarrow 6] \end{array}$$

$$\begin{array}{c} \text{orange arrow} \\ \text{red arrow} \\ [1 \rightarrow 5] \end{array}$$

### 3. ADAPTACIÓN DEL ALGORITMO PSO

#### 3.4 OPTIMIZACIÓN DE ECUACIONES: $\varphi_1 = \varphi_2$

Posición Intermedia:

$$pInt_i = pBest_i \oplus \frac{1}{2} \otimes (gBest \theta pBest_i)$$

Operador Actualización de Velocidad

$$v_i^t = c1 \otimes v_i^{t-1} \circ c2 \otimes (pInt_i \theta x_i^{t-1})$$

Operador Movimiento

$$x_i^t = x_i^{t-1} \oplus v_i^t$$

# 3. ADAPTACIÓN DEL ALGORITMO PSO

## 3.5 PSEUDOCÓDIGO

Para cada partícula  $i = 1, \dots$ , hasta el total de partículas:

- ❖ Inicializar la posición  $x_i$  con un tour generado aleatoriamente.
- ❖ Inicializar la velocidad de la partícula  $v_i$  mediante la resta de dos posiciones aleatorias.
- ❖ Inicializar la posición  $pBest_i$  con el valor de  $x_i$ :  $pBest_i \leftarrow x_i$
- ❖ Inicializar  $fitness\_x_i$  de la siguiente forma:  $fitness\_x_i \leftarrow f(x_i)$
- ❖ Inicializar  $fitness\_pBest_i$  de la siguiente forma:  $fitness\_pBest_i \leftarrow fitness\_x_i$
- ❖ Si ( $fitness\_pBest_i < fitness\_gBest$ ) entonces:
  - Actualizar el mejor resultado global:  $fitness\_gBest \leftarrow fitness\_pBest_i$
  - Actualizar la mejor posición global:  $gBest \leftarrow pBest_i$

Mientras no se alcance el límite máximo de iteraciones, o no se cumpla el criterio de parada:

❖ Para cada partícula  $i = 1, \dots$ , hasta el total de partículas:

- Actualizar la velocidad:  $v_i^t = c1 \otimes v_i^{t-1} \circ c2 \otimes (pInt_i \theta x_i^{t-1})$
- Actualizar la posición:  $x_i^t = x_i^{t-1} \oplus v_i^t$

❖ Para cada partícula  $i = 1, \dots$ , hasta el total de partículas:

- Reevaluar  $fitness\_x_i$ :  $fitness\_x_i \leftarrow f(x_i)$
- Si ( $fitness\_x_i < fitness\_pBest_i$ ) entonces:
  - Actualizar el mejor resultado local:  $fitness\_pBest_i \leftarrow fitness\_x_i$
  - Actualizar la mejor posición local:  $pBest_i$ :  $pBest_i \leftarrow x_i$
  - Si ( $fitness\_pBest_i < fitness\_gBest$ ) entonces:
    - Actualizar el mejor resultado global:  $fitness\_gBest \leftarrow fitness\_pBest_i$
    - Actualizar la mejor posición global:  $gBest \leftarrow pBest_i$

Devolver  $gBest$  como la mejor solución.

## 4. RESULTADOS

### 4.1 CONFIGURACIÓN BASADA EN BR17

- Cantidad de Partículas: 200.
- Cantidad Máxima de Iteraciones: 10000.
- $c1$ : 0,4.
- $c2$ : 0,1.

|       | PI      | PTS   | PE         | PTT    | PC       | MS    |
|-------|---------|-------|------------|--------|----------|-------|
| Br17  | 569,47  | 0,63  | 881732,5   | 9,93   | 39,03    | 39    |
| Ftv33 | 4831,73 | 15,11 | 12257107,5 | 30,58  | 1798,03  | 1574  |
| Ft53  | 7132,77 | 46,77 | 17314307,5 | 64,93  | 11209,40 | 10219 |
| Ft70  | 7935,83 | 85,08 | 14271325   | 106,54 | 48830,47 | 46952 |



## 4. RESULTADOS

### 4.1 CONFIGURACIÓN BASADA EN FTV33

- Cantidad de Partículas: 250.
- Cantidad Máxima de Iteraciones: 75000.
- $c1$ : 0,1.
- $c2$ : 0,4.

|       | PI       | PTS    | PE         | PTT     | PC       | MS    |
|-------|----------|--------|------------|---------|----------|-------|
| Br17  | 3526,93  | 4,53   | 113893,33  | 98,88   | 39       | 39    |
| Ftv33 | 49028,43 | 199,38 | 966346,67  | 298,99  | 1665,43  | 1491  |
| Ft53  | 69257,23 | 593,33 | 1426553,33 | 642,72  | 10898,37 | 9425  |
| Ft70  | 57085,30 | 781,36 | 1587166,67 | 1020,04 | 54075,47 | 47468 |

## 4. RESULTADOS

### 4.1 CONFIGURACIÓN BASADA EN FT53

- Cantidad de Partículas: 100.
- Cantidad Máxima de Iteraciones: 150000.
- $c1$ : 0,2.
- $c2$ : 1,8.

|       | PI        | PTS    | PE       | PTT    | PC       | MS    |
|-------|-----------|--------|----------|--------|----------|-------|
| Br17  | 3442,00   | 2,24   | 344200   | 94,91  | 39,17    | 39    |
| Ftv33 | 80401,53  | 153,58 | 8040153  | 281,71 | 1733,73  | 1521  |
| Ft53  | 118616,13 | 470,82 | 11861613 | 594,16 | 10857,33 | 9934  |
| Ft70  | 132425,47 | 863,76 | 13242547 | 976,93 | 50525,20 | 46968 |

## 4. RESULTADOS

### 4.1 CONFIGURACIÓN BASADA EN FT70

- Cantidad de Partículas: 300.
- Cantidad Máxima de Iteraciones: 50000.
- $c1$ : 0,25.
- $c2$ : 0,25.

|       | PI       | PTS    | PE       | PTT    | PC       | MS    |
|-------|----------|--------|----------|--------|----------|-------|
| Br17  | 578,17   | 0,90   | 173451   | 78,10  | 39,00    | 39    |
| Ftv33 | 17767,6  | 85,15  | 5330280  | 231,85 | 1692,67  | 1464  |
| Ft53  | 35092,17 | 351,80 | 10527651 | 497,08 | 10490,00 | 9274  |
| Ft70  | 42237,03 | 700,53 | 12671109 | 826,27 | 47210,47 | 45108 |

## 5. CONCLUSIONES

- Para lograr mejores aproximaciones, generalmente es mejor aumentar el número de partículas, en lugar al de iteraciones.
- Recomendamos valores de  $c1$  y  $c2$  chicos.
- La configuración basada en ft70 es relativamente buena para las cuatro instancias.
- La selección de la configuración es un problema en sí mismo.