

Computación Bioinspirada

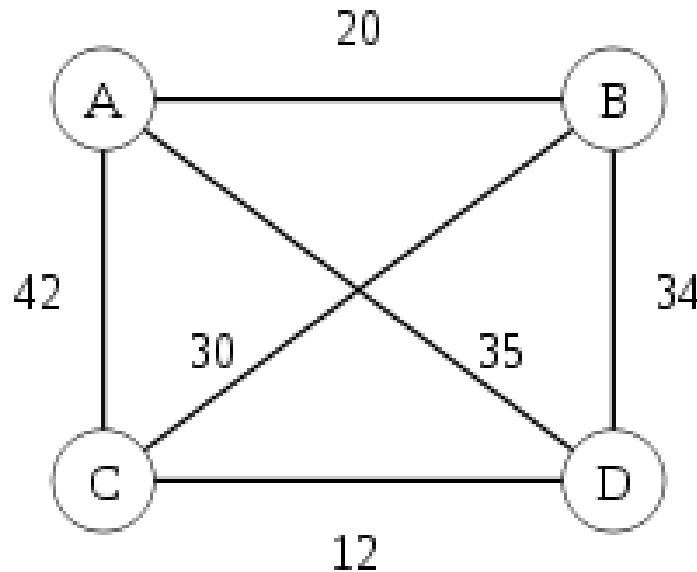
Dr. Edward Hinojosa Cárdenas
ehinojosa@unsa.edu.pe

PSO Discreto para TSP

- K.P.Wang, L. Huang, C.G. Zhou, W. Pang.
- Particle swarm optimization for traveling salesman problem, International Conference on Machine Learning and Cybernetics (2003), 1583–1585.

Problema del Vendedor Viajero

- El viajero debe visitar todas las ciudades una vez en un territorio recorriendo la menor distancia.
- Representación por permutación de n ciudades, por ejemplo 5 ciudades (A, D, C, B).



PSO para TSP

- La solución de una partícula es la permutación de todas las ciudades. Por ejemplo, (A, D, G, B, E).
- La velocidad de una partícula es una secuencia de intercambio de posiciones (swap).
- Ejemplo de velocidad:
 - Operador de intercambio (SO) = $SO(1,2)$ //intercambio de la primera ciudad visitada con la segunda ciudad visitada
 - Secuencia de intercambio (SS) = $(SO(1,2), SO(5,4), SO(5,1))$

Actualización de la Partícula

$$x_i^{k+1} = x_i^k + v_i^{k+1}$$

- Aplicar SO a una permutación:
 - $(A, D, G, B, E) + SO(1,2) = (D, A, G, B, E)$
- Aplicar una secuencia de SO (SS) a una permutación:
 - $SS = (SO(1,2), SO(5,4), SO(5,1))$
 - $(A,D,G,B,E)+SS \rightarrow (D,A,G,B,E) \rightarrow (D,A,G,E,B) \rightarrow (B,A,G,E,D)$

Actualización de la Velocidad de una Partícula

$$v_i^{k+1} = \omega \cdot v_i^k + \varphi_1 \cdot (pBest_i - x_i^k) + \varphi_2 \cdot (g_i - x_i^k)$$

- Juntar dos SS:
 - SS1 = (SO(1,2), SO(5,4), SO(5,1))
 - SS2 = (SO(1,3), SO(5,1), SO(2,1))
 - SS1 + SS2 = (SO(1,2), SO(5,4), SO(5,1), SO(1,3), SO(5,1), SO(2,1))

Restar dos Permutaciones

- P_1 : (A, C, D, E, B) y P_2 : (C, A, B, E, D)
- Existe una SS que transforma P_1 en P_2 ;
 - A está en posición 1 en P_1 y 2 en P_2 : $SO_1(1,2)$
 - $P_1 + SO_1 = P_1' = (C, A, D, E, B)$
 - B está en la posición 5 en P_1' y 3 en P_2 : $SO_2(5,3)$
 - $P_1' + SO_2(5,3) = (C, A, B, E, D)$
- $SS = (SO_1(1,2), SO_2(5,3))$
- $P_1 - P_2 = SS$

Actualización de la Velocidad de la Partícula

- Cada velocidad es una secuencia de intercambio (swap).

$$v_i^{k+1} = \omega \cdot v_i^k + \varphi_1 \cdot (pBest_i - x_i^k) + \varphi_2 \cdot (g_i - x_i^k)$$

- φ_1, φ_2 Son números aleatorios entre 0 y 1.
- La probabilidad que todos los operadores de intercambio en la secuencia de intercambio con el mejor local es definido por φ_1 .
- La probabilidad que todos los operadores de intercambio en la secuencia de intercambio con el mejor global es definido por φ_2 .
- Nosotros vamos a considerar $\varphi_1 = \varphi_2 = 1$ (100 %).

Algoritmo de PSO para TSP

Random initialization of permutation and swap sequences.

For each time step

 Update gbest if needed,

 Update pbest if needed.

 For each particle in the swarm

$$v_i^{k+1} = \omega \cdot v_i^k + \varphi_1 \cdot (pBest_i - x_i^k) + \varphi_2 \cdot (g_i - x_i^k)$$

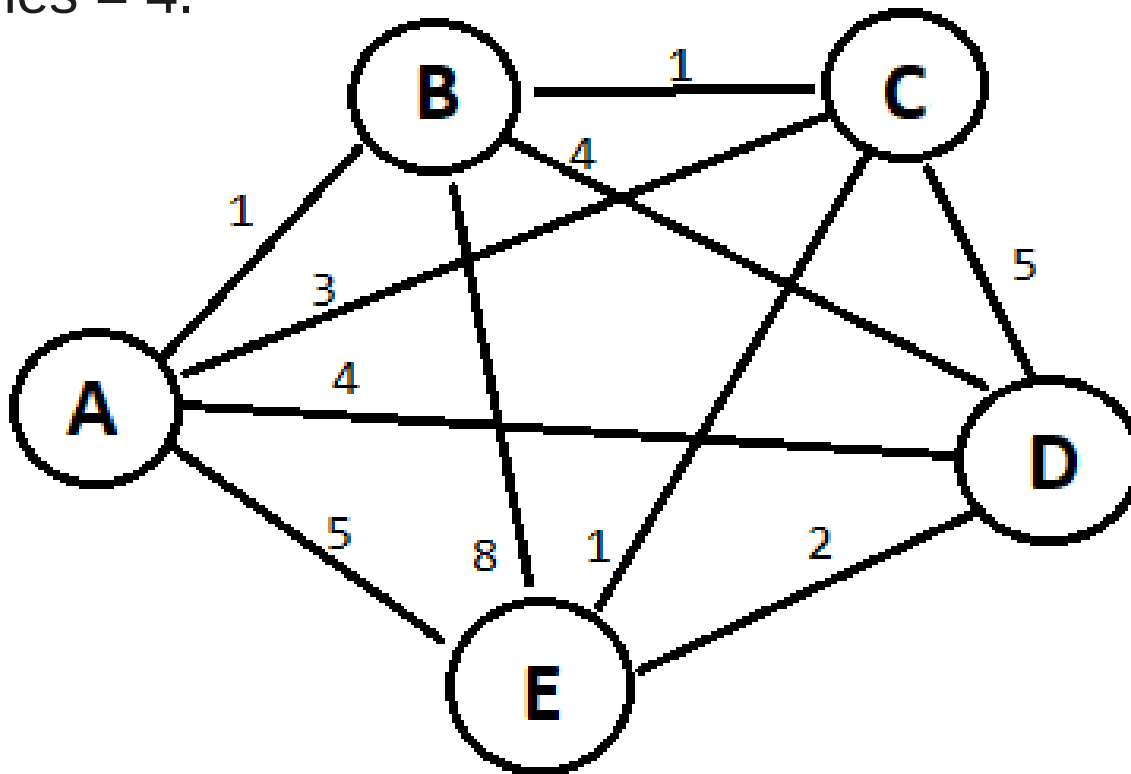
$$x_i^{k+1} = x_i^k + v_i^{k+1}$$

 End

End

Ejemplo

- Aplicar PSO discreto al siguiente problema TSP.
 - Cantidad de partículas 4.
 - $\varphi_1 = \varphi_2 = 1$.
 - Iteraciones = 4.



Ejemplo

Iteracion 1 *****

Actual: ['A', 'D', 'E', 'C', 'B']
Costo Actual: 9
Mejor: ['A', 'D', 'E', 'C', 'B']
Costo del Mejor: 9
Velocidad: []

Actual: ['A', 'C', 'E', 'B', 'D']
Costo Actual: 20
Mejor: ['A', 'C', 'E', 'B', 'D']
Costo del Mejor: 20
Velocidad: []

Actual: ['A', 'E', 'D', 'C', 'B']
Costo Actual: 14
Mejor: ['A', 'E', 'D', 'C', 'B']
Costo del Mejor: 14
Velocidad: []

Actual: ['A', 'E', 'C', 'B', 'D']
Costo Actual: 15
Mejor: ['A', 'E', 'C', 'B', 'D']
Costo del Mejor: 15
Velocidad: []

Ejemplo

Iteracion 1 *****

Actual: ['A', 'D', 'E', 'C', 'B']
Costo Actual: 9
Mejor: ['A', 'D', 'E', 'C', 'B']
Costo del Mejor: 9
Velocidad: []

Actual: ['A', 'C', 'E', 'B', 'D']
Costo Actual: 20
Mejor: ['A', 'C', 'E', 'B', 'D']
Costo del Mejor: 20
Velocidad: []

Actual: ['A', 'E', 'D', 'C', 'B']
Costo Actual: 14
Mejor: ['A', 'E', 'D', 'C', 'B']
Costo del Mejor: 14
Velocidad: []

Actual: ['A', 'E', 'C', 'B', 'D']
Costo Actual: 15
Mejor: ['A', 'E', 'C', 'B', 'D']
Costo del Mejor: 15
Velocidad: []

$pbest - x(t-1) = []$
 $(pbest - x(t-1)) + (gbest - x(t-1)) = []$
[]
 $pbest - x(t-1) = []$
 $(pbest - x(t-1)) + (gbest - x(t-1)) = [(1, 3, 1), (3, 4, 1)]$
[(1, 3, 1), (3, 4, 1)]
 $pbest - x(t-1) = []$
 $(pbest - x(t-1)) + (gbest - x(t-1)) = [(1, 2, 1)]$
[(1, 2, 1)]
 $pbest - x(t-1) = []$
 $(pbest - x(t-1)) + (gbest - x(t-1)) = [(1, 2, 1), (2, 3, 1), (3, 4, 1)]$
[(1, 2, 1), (2, 3, 1), (3, 4, 1)]

Ejemplo

Iteracion 2 *****

Actual: ['A', 'D', 'E', 'C', 'B']
Costo Actual: 9
Mejor: ['A', 'D', 'E', 'C', 'B']
Costo del Mejor: 9
Velocidad: []

Actual: ['A', 'B', 'E', 'D', 'C']
Costo Actual: 19
Mejor: ['A', 'B', 'E', 'D', 'C']
Costo del Mejor: 19
Velocidad: [(1, 3, 1), (3, 4, 1)]

Actual: ['A', 'D', 'E', 'C', 'B']
Costo Actual: 9
Mejor: ['A', 'D', 'E', 'C', 'B']
Costo del Mejor: 9
Velocidad: [(1, 2, 1)]

Actual: ['A', 'C', 'B', 'D', 'E']
Costo Actual: 15
Mejor: ['A', 'E', 'C', 'B', 'D']
Costo del Mejor: 15
Velocidad: [(1, 2, 1), (2, 3, 1), (3, 4, 1)]

```
pbest - x(t-1) = []  
(pbest - x(t-1)) + (gbest - x(t-1)) = []  
[]  
pbest - x(t-1) = []  
(pbest - x(t-1)) + (gbest - x(t-1)) = [(1, 4, 1), (3, 4, 1)]  
[(1, 4, 1), (3, 4, 1)]  
pbest - x(t-1) = []  
(pbest - x(t-1)) + (gbest - x(t-1)) = []  
[]  
pbest - x(t-1) = [(1, 2, 1), (2, 3, 1), (3, 4, 1)]  
(pbest - x(t-1)) + (gbest - x(t-1)) = [(1, 2, 1), (2, 3, 1), (3, 4, 1), (1, 3, 1), (2, 4, 1)]  
[(1, 2, 1), (2, 3, 1), (3, 4, 1), (1, 3, 1), (2, 4, 1)]
```

Ejemplo

Iteracion 3 *****

Actual: ['A', 'D', 'E', 'C', 'B']

Costo Actual: 9

Mejor: ['A', 'D', 'E', 'C', 'B']

Costo del Mejor: 9

Velocidad: []

Actual: ['A', 'C', 'E', 'B', 'D']

Costo Actual: 20

Mejor: ['A', 'B', 'E', 'D', 'C']

Costo del Mejor: 19

Velocidad: [(1, 4, 1), (3, 4, 1)]

Actual: ['A', 'D', 'E', 'C', 'B']

Costo Actual: 9

Mejor: ['A', 'D', 'E', 'C', 'B']

Costo del Mejor: 9

Velocidad: []

Actual: ['A', 'E', 'C', 'B', 'D']

Costo Actual: 15

Mejor: ['A', 'E', 'C', 'B', 'D']

Costo del Mejor: 15

Velocidad: [(1, 2, 1), (2, 3, 1), (3, 4, 1), (1, 3, 1), (2, 4, 1)]

Ejemplo

```
pbest - x(t-1) = []  
(pbest - x(t-1)) + (gbest - x(t-1)) = []  
[]  
pbest - x(t-1) = [(1, 4, 1), (3, 4, 1)]  
(pbest - x(t-1)) + (gbest - x(t-1)) = [(1, 4, 1), (3, 4, 1), (1, 3, 1), (3, 4, 1)]  
[(1, 4, 1), (3, 4, 1), (1, 3, 1), (3, 4, 1)]  
pbest - x(t-1) = []  
(pbest - x(t-1)) + (gbest - x(t-1)) = []  
[]  
pbest - x(t-1) = []  
(pbest - x(t-1)) + (gbest - x(t-1)) = [(1, 2, 1), (2, 3, 1), (3, 4, 1)]  
[(1, 2, 1), (2, 3, 1), (3, 4, 1)]
```

Ejemplo

Iteracion 4 *****

Actual: ['A', 'D', 'E', 'C', 'B']

Costo Actual: 9

Mejor: ['A', 'D', 'E', 'C', 'B']

Costo del Mejor: 9

Velocidad: []

Actual: ['A', 'C', 'E', 'B', 'D']

Costo Actual: 20

Mejor: ['A', 'B', 'E', 'D', 'C']

Costo del Mejor: 19

Velocidad: [(1, 4, 1), (3, 4, 1), (1, 3, 1), (3, 4, 1)]

Actual: ['A', 'D', 'E', 'C', 'B']

Costo Actual: 9

Mejor: ['A', 'D', 'E', 'C', 'B']

Costo del Mejor: 9

Velocidad: []

Actual: ['A', 'C', 'B', 'D', 'E']

Costo Actual: 15

Mejor: ['A', 'E', 'C', 'B', 'D']

Costo del Mejor: 15

Velocidad: [(1, 2, 1), (2, 3, 1), (3, 4, 1)]

Ejemplo

```
pbest - x(t-1) = []  
(pbest - x(t-1)) + (gbest - x(t-1)) = []  
[]  
pbest - x(t-1) = [(1, 4, 1), (3, 4, 1)]  
(pbest - x(t-1)) + (gbest - x(t-1)) = [(1, 4, 1), (3, 4, 1), (1, 3, 1), (3, 4, 1)]  
[(1, 4, 1), (3, 4, 1), (1, 3, 1), (3, 4, 1)]  
pbest - x(t-1) = []  
(pbest - x(t-1)) + (gbest - x(t-1)) = []  
[]  
pbest - x(t-1) = [(1, 2, 1), (2, 3, 1), (3, 4, 1)]  
(pbest - x(t-1)) + (gbest - x(t-1)) = [(1, 2, 1), (2, 3, 1), (3, 4, 1), (1, 3, 1), (2, 4, 1)]  
[(1, 2, 1), (2, 3, 1), (3, 4, 1), (1, 3, 1), (2, 4, 1)]
```

Ejemplo

Mejor Global: ['A', 'D', 'E', 'C', 'B'] | Costo: 9

Práctica

- Replicar el ejemplo mostrado.

GRACIAS

Dr. Edward Hinojosa Cárdenas
ehinojosa@unsa.edu.pe