

# Representación TSP

# Permutación de n elementos

Ejemplo Solución 1:

0 1 2 3 4 5 6 7 8 9

El recorrido sería:

[ 0 -> 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7 -> 8 -> 9 ] -> 0

Ejemplo Solución 2:

3 1 5 2 4 9 7 8 0 6

[ 3 -> 1 -> 5 -> 2 -> 4 -> 9 -> 7 -> 8 -> 0 -> 6 ] -> 3

Ejemplo Solución 3:

5 6 7 8 9 0 1 2 3 4

[ 5 -> 6 -> 7 -> 8 -> 9 -> 0 -> 1 -> 2 -> 3 -> 4 ] -> 5

La solución 1 y 3 son equivalentes.

En esta codificación, las soluciones tienen más de una representación!

# Evaluación

Ejemplo Solución 1:

0 1 2 3 4 5 6 7 8 9

El recorrido sería:

[ 0 -> 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7 -> 8 -> 9 ] -> 0

Evaluación = distancia(0, 1) + distancia(1, 2) + distancia(2, 3) + .... + distancia( 8, 9) + distancia (9, 0)

# Evaluación

Ejemplo Solución 2:

3 1 5 2 4 9 7 8 0 6

[ 3 -> 1 -> 5 -> 2 -> 4 -> 9 -> 7 -> 8 -> 0 -> 6 ] -> 3

Evaluación = distancia(3, 1) + distancia(1, 5) + distancia(5, 2) + .... + distancia( 0, 6) + distancia (6, 3)

Codificación de la evaluación:

Evaluación[0] = distancia(perm[0], perm[1] ) = distancia(3, 1)

Evaluación[1] = distancia(perm[1], perm[2] ) = distancia(1, 5)

....

Evaluación[i] = distancia(perm[i], perm[i+1] )

.....

Evaluación[9] = distancia(perm[9], perm[0] ) = distancia(6, 3)

# Fijar un nodo

Para evitar duplicidad, y reducir el espacio de búsqueda, fijamos el primer nodo. Por tanto, nuestras soluciones ahora solo serán una permutación de n-1 elementos.

Ejemplo Solución 1:

1 2 3 4 5 6 7 8 9

El recorrido sería:

0 -> [ 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7 -> 8 -> 9 ] -> 0

Ejemplo Solución 2:

3 1 5 2 4 9 7 8 6

0 -> [ 3 -> 1 -> 5 -> 2 -> 4 -> 9 -> 7 -> 8 -> 6 ] -> 0

Ejemplo Solución 3:

5 6 7 8 9 1 2 3 4

0 -> [ 5 -> 6 -> 7 -> 8 -> 9 -> 1 -> 2 -> 3 -> 4 ] -> 0

Se tiene un mapeo 1 a 1 entre la codificación y la solución que se representa.

# Evaluación

Ejemplo Solución 2:

3 1 5 2 4 9 7 8 6

0 -> [ 3 -> 1 -> 5 -> 2 -> 4 -> 9 -> 7 -> 8 -> 6 ] -> 0

Evaluación = distancia(3, 1) + distancia(1, 5) + distancia(5, 2) + .... + distancia( 0, 6) + distancia (6, 3)

Codificación de la evaluación:

Evaluación[0] = distancia(0, perm[0] ) = distancia(0, 3)

Evaluación[1] = distancia(perm[0], perm[1] ) = distancia(3, 1)

....

Evaluación[i] = distancia(perm[i-1], perm[i] )

.....

Evaluación[8] = distancia(perm[7], perm[8] )

Evaluación[9] = distancia(perm[8], 0) = distancia(6, 0)

# Intercambio de Elementos en la Permutación

Ejemplo Solución 1:

1 2 3 4 5 6 7 8 9

El recorrido sería:

0 -> [1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7 -> 8 -> 9] -> 0

Evaluación = distancia(0, 1) + distancia(1, 2) + distancia(2, 3) + **distancia(3, 4) + distancia(4, 5) +**  
distancia(5, 6) + distancia(6, 7) + **distancia(7, 8) + distancia(8, 9)** + distancia(9, 0)

Intercambiar el 3 y 8

Ejemplo Solución Resultado:

1 2 8 4 5 6 7 3 9

El recorrido sería:

0 -> [1 -> 2 -> 8 -> 4 -> 5 -> 6 -> 7 -> 3 -> 9] -> 0

Evaluación = distancia(0, 1) + distancia(1, 2) + distancia(2, 8) + **distancia(8, 4) + distancia(4, 5) +**  
distancia(5, 6) + distancia(6, 7) + **distancia(7, 3) + distancia(3, 9)** + distancia(9, 0)

# Evaluación intercambio (x, y) en la permutación

$\text{Nueva\_Evaluación} = \text{Evaluación\_Anterior} + \text{Diferencia}$

Para todo elemento  $x > 0$  de la permutación, se cumple que:

$\text{Evaluación}[x] = \text{distancia}(\text{perm}[x - 1], \text{perm}[x])$

Para  $x=0$ , se tiene que

$\text{Evaluación}[0] = \text{distancia}(0, \text{perm}[x])$



0 -> [1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7 -> 8 -> 9] -> 0

x = 2, x es la posición en perm

Evaluación[2] = distancia( perm[1], perm[2]) = distancia( 2 , 3)

Evaluación[3] = distancia( perm[2], perm[3]) = distancia( 3 , 4)

y = 7

Evaluación[7] = distancia( perm[6], perm[7]) = distancia( 7 , 8)

Evaluación[8] = distancia( perm[7], perm[8]) = distancia( 8 , 9)