

Primer Proyecto Programado

Tema: Algoritmos Genéticos

Curso: Análisis de algoritmos

Profesor:

José Castro Mora

Estudiantes:

Ricardo Guerrero Matus

Gabriela Garro Abdykerimov



# EXPLICACION DE LOS ALGORITMOS

## Selección de los padres:

- **Azar:** Generamos un número random entre el padre número 0 y el ultimo padre de la y ese es el padre seleccionado.
- **Torneo:** Seleccionamos una cantidad N de padres de la generación actual y en esa cantidad N comparamos cual padre tiene mejor fitness y ese padre es seleccionado.
- **Ruleta:** Sumamos el fitness individual de todos los individuos de la población luego generamos un número random N el cual su máximo tamaño será la suma de todos los fitness de cada individuo de la generación de esta manera se irán sumando cada uno de los individuos de la generación de 1 en 1 y cuando la suma de los individuos sea mayor o igual al random ese individuo será seleccionado como padre para que así los que tengan un mejor fitness (mas alto) tengan mejores posibilidades de ser escogidos como padres.

## Explicación **del Cruce:**

Para el cruce lo que utilizamos fue un algoritmo de un punto el cual divide los genes de los 2 padres a la mitad y de esta forma cada padre le reparte una mitad de sus genes distinta a cada hijo para así generar 2 nuevos individuos y luego estos 2 nuevos individuos va a proceder a remplazar a los 2 individuos menos aptos para sobrevivir de la generación próxima así se irá mejorando la generación cada vez mas.

# CUADROS COMPARATIVOS DE DISTINTAS GENERACIONES

## Viajero:

datos entrada txt:

```
8
3 1 5 1 7 2 1
1 4 3 5 2 2
6 2 1 3 3
1 7 2 4
4 1 5
1 6
7
3 1 2 4
4 1 2 3
2 1 3 4
1 4 3 2|
```

Comando Entrada:

```
-----EJEMPLO-----
Orden de los datos: Mochila-Viajero ArchivoEntrada Generaciones TamañoPoblacion PorcentajeMutacion (0:Azar 1:Ruleta 2:Torneo) elitismo ArchivoSalida
Cada dato debe ser separado por un espacio en el orden dicho
»Ejemplo: Mochila entradaMochila.txt 100 1000 3 2 0 output.txt
»Ejemplo: Viajero capibara.txt 100 100 5 2 0 output.txt
-----
Inserte el comando para iniciar la Aplicacion:
Viajero capibara.txt 100 100 5 2 0 output.txt|
```

## Resultados de las distintas Primeras Generaciones:

Matriz de distancias entre ciudades:

```
0 3 1 5 1 7 2 1
3 0 1 4 3 5 2 2
1 1 0 6 2 1 3 3
5 4 6 0 1 7 2 4
1 3 2 1 0 4 1 5
7 5 1 7 4 0 1 6
2 2 3 2 1 1 0 7
1 2 3 4 5 6 7 0
```

Generacion numero 1

[[2, 7, 3, 8, 4, 1, 5, 6] fitness: 27, [3, 1, 2, 7, 6, 8, 5, 4] fitness: 25, [2, 5, 1, 8, 4, 7, 3, 6] fitness: 20,  
Mejor fitness: 13 Peor fitness: 37

Generacion numero 2

[[3, 1, 5, 8, 2, 7, 4, 6] fitness: 21, [2, 5, 1, 8, 4, 3, 7, 6] fitness: 24, [5, 6, 3, 1, 7, 4, 8, 2] fitness: 19,  
Mejor fitness: 14 Peor fitness: 37

Generacion numero 3

[[7, 6, 5, 4, 2, 3, 1, 8] fitness: 20, [1, 7, 6, 3, 5, 4, 8, 2] fitness: 16, [7, 3, 6, 5, 2, 4, 8, 1] fitness: 22,  
Mejor fitness: 16 Peor fitness: 38

Generacion numero 4

[[1, 7, 6, 3, 5, 8, 4, 2] fitness: 22, [7, 4, 8, 1, 5, 3, 6, 2] fitness: 18, [2, 5, 4, 1, 7, 6, 8, 3] fitness: 22,  
Mejor fitness: 17 Peor fitness: 38

Generacion numero 5

[[5, 6, 3, 4, 2, 7, 8, 1] fitness: 26, [7, 6, 5, 8, 1, 2, 3, 4] fitness: 23, [1, 3, 4, 5, 7, 8, 6, 2] fitness: 30,  
Mejor fitness: 14 Peor fitness: 39

Generacion numero 6

[[2, 3, 6, 1, 8, 4, 5, 7] fitness: 18, [7, 6, 3, 8, 4, 5, 1, 2] fitness: 16, [4, 7, 2, 3, 8, 1, 5, 6] fitness: 21,  
Mejor fitness: 15 Peor fitness: 39

Generacion numero 7

[[5, 4, 3, 1, 8, 2, 6, 7] fitness: 18, [4, 8, 2, 1, 3, 6, 7, 5] fitness: 14, [8, 3, 6, 7, 4, 5, 1, 2] fitness: 14,  
Mejor fitness: 14 Peor fitness: 38

# Viajero:

## datos entrada txt:

```
13.5
3
1.4 2.8
6.5 3.2
1.1 3.1
6
0 1 0
0 1 0
0 1 0
0 1 1
0 1 1
1 1 1
```

## Comando Entrada:

run:

### -----EJEMPLO-----

Orden de los datos: Mochila-Viajero ArchivoEntrada Generaciones TamañoPoblacion PorcentajeMutacion (0:Azar 1:Ruleta 2:Torneo) elitismo ArchivoSalida

Cada dato debe ser separado por un espacio en el orden dicho

»Ejemplo: Mochila entradaMochila.txt 100 1000 3 2 0 output.txt

»Ejemplo: Viajero capibara.txt 100 100 5 2 0 output.txt

Inserte el comando para iniciar la Aplicacion:

Mochila entradaMochila.txt 100 1000 3 2 0 output.txt|

## Resultados de las distintas Primeras Generaciones:

\*\*\*\*\*

ELEMENTOS DE LA MOCHILA:

[[1.4,2.8] , [6.5,3.2] , [1.1,3.1]]

Peor Individuo: Peso Total: 7.9 Costo Total: 6.0 |

[1 , 1 , 0]

Mejor Individuo: Peso Total: 2.5 Costo Total: 5.8999996

[1 , 0 , 1]

\*\*\*\*\*

ELEMENTOS DE LA MOCHILA:

[[1.4,2.8] , [6.5,3.2] , [1.1,3.1]]

Peor Individuo: Peso Total: 7.9 Costo Total: 6.0

[1 , 1 , 0]

Mejor Individuo: Peso Total: 2.5 Costo Total: 5.8999996

[1 , 0 , 1]

\*\*\*\*\*

ELEMENTOS DE LA MOCHILA:

[[1.4,2.8] , [6.5,3.2] , [1.1,3.1]]

Peor Individuo: Peso Total: 6.5 Costo Total: 3.2

[0 , 1 , 0]

Mejor Individuo: Peso Total: 2.5 Costo Total: 5.8999996

[1 , 0 , 1]

\*\*\*\*\*

ELEMENTOS DE LA MOCHILA:

[[1.4,2.8] , [6.5,3.2] , [1.1,3.1]]

Peor Individuo: Peso Total: 7.9 Costo Total: 6.0

[1 , 1 , 0]

Mejor Individuo: Peso Total: 9.0 Costo Total: 9.1

[1 , 1 , 1]

\*\*\*\*\*

ELEMENTOS DE LA MOCHILA:

[[1.4,2.8] , [6.5,3.2] , [1.1,3.1]]

Peor Individuo: Peso Total: 7.9 Costo Total: 6.0

[1 , 1 , 0]

Mejor Individuo: Peso Total: 9.0 Costo Total: 9.1

[1 , 1 , 1]

\*\*\*\*\*

ELEMENTOS DE LA MOCHILA:

[[1.4,2.8] , [6.5,3.2] , [1.1,3.1]]

Peor Individuo: Peso Total: 6.5 Costo Total: 3.2

[0 , 1 , 0]

Mejor Individuo: Peso Total: 9.0 Costo Total: 9.1

[1 , 1 , 1]