

“ESCUELA SUPERIOR DE CÓMPUTO”

MATERIA: GENETIC ALGORITHMS

PROFESOR: MORALES GUITRON SANDRA LUZ

GRUPO: 3CM5

ALUMNO: ALBARRAN CRUZ CARLOS
ALBERTO

“PRÁCTICA 7”

INTRODUCCIÓN

La representación de permutaciones se usa frecuentemente en problemas de optimización combinatoria, como el del viajero y consiste básicamente en usar cadenas de enteros para representar una permutación.

Al efectuar cualquiera de las cruzas antes descritas entre 2 cadenas que usan representación de permutaciones, los hijos invariablemente serán no válidos.

Este problema requiere la definición de procedimientos de “reparación” de las cadenas inválidas que se producen a consecuencia de la cruce.

Las siguientes cruzas que se presentan son:

- Order Crossover.
- Partially Mapped Crossover.
- Position-Based Crossover.
- Order-Based Crossover.
- Cycle Crossover

RESULTADOS

Order Crossover

```
Poblacion inicial
5 1 3 8 7 2 9 10 6 4
6 1 10 7 5 3 8 2 4 9
7 10 9 2 8 1 4 5 6 3
10 3 4 7 5 9 1 8 6 2
5 7 4 9 10 6 1 8 2 3
9 4 7 5 1 3 2 10 6 8

Nueva Poblacion
5 1 3 6 10 7 8 2 4 9
6 1 10 5 3 8 7 2 9 4
7 10 9 3 4 5 1 8 6 2
10 3 4 7 9 2 8 1 5 6
5 7 4 9 1 3 2 10 6 8
9 4 7 5 10 6 1 8 2 3
```

Partially Mapped Crossover

```
Poblacion inicial
5 7 1 8 2 4 9 3 6 10
7 9 1 2 10 4 8 6 5 3
2 7 9 5 4 6 1 8 3 10
3 4 10 5 8 6 1 9 7 2
4 8 2 6 5 3 10 9 1 7
3 1 7 5 4 9 2 8 10 6
```

```
Nueva Poblacion
5 7 1 8 2 4 9 3 6 10
7 9 1 2 10 4 8 6 5 3
2 7 10 5 4 6 1 8 3 9
3 4 9 5 8 6 1 10 7 2
4 8 7 6 5 3 10 9 1 2
3 1 2 5 4 9 7 8 10 6
```

Positioned-Based Crossover

```
Poblacion inicial
5 9 7 3 10 4 1 6 8 2
5 6 1 4 7 3 10 2 9 8
3 5 7 2 1 8 10 6 4 9
7 5 10 4 3 6 2 8 1 9
4 8 2 3 5 9 1 7 10 6
7 4 10 6 5 9 3 1 8 2
```

```
Nueva Poblacion
5 6 1 3 7 4 10 2 9 8
5 9 7 4 10 3 1 6 8 2
3 5 7 10 6 2 8 1 4 9
7 5 2 4 3 1 8 10 6 9
4 7 10 3 5 6 9 1 8 2
8 4 2 9 5 1 3 7 10 6
```

Order-Based Crossover

```
Poblacion inicial
7 9 3 4 2 10 8 6 1 5
6 10 4 8 7 5 2 3 9 1
9 5 1 8 3 2 7 4 10 6
5 1 7 9 10 6 2 3 4 8
3 6 4 7 2 10 5 1 9 8
8 9 4 10 7 5 3 6 1 2

Nueva Poblacion
6 7 9 3 4 10 2 8 1 5
10 4 8 7 2 5 3 6 9 1
9 5 1 8 3 6 2 7 4 10
5 1 7 9 10 2 3 4 8 6
3 4 7 10 5 1 9 6 8 2
8 6 9 4 2 10 7 5 3 1
```

Cycled Crossover

```
Poblacion inicial
2 3 8 4 5 10 1 7 9 6
9 4 2 5 7 8 1 3 6 10
6 1 8 10 2 3 4 9 7 5
5 1 4 9 10 7 6 3 2 8
2 1 4 8 7 5 6 10 3 9
2 3 6 5 1 10 4 8 7 9

Nueva Poblacion
2 4 8 5 7 10 1 3 9 6
9 3 2 4 5 8 1 7 6 10
6 1 8 9 10 7 4 3 2 5
5 1 4 10 2 3 6 9 7 8
2 3 6 5 1 10 4 8 7 9
2 1 4 8 7 5 6 10 3 9
```

CONCLUSIONES

Al estar programando cada una de estas diferentes cruzas se puede notar que hay mucha similitud con el cómo es que sus algoritmos se parecen bastantes, casi siempre se toma algún padre y se le quita una subcadena ya evaluada y después se toma el otro padre para que el resto de valores que no estaban en la otra cadena se ponga en el hijo. A diferencia de las otras cruzas hechas en la práctica 6, estas tienen un nivel de complejidad más alto.