

Práctica #6 Algoritmos de cruce para permutaciones

Escuela Superior de Cómputo

Algoritmos Genéticos

Sandra Luz Morales Guitrón

Jorge Gómez Reus

Índice

1. Introducción	1
2. Contenido	3
2.1. Order Crossover	3
2.2. Partially Mapped Crossover	4
2.3. Position-based Crossover	5
2.4. Order-based Crossover	6
2.5. Cycle Crossover	6
3. Conclusión	6

1. Introducción

En esta práctica se implementaron los algoritmos:

- **Order Crossover:** Esta técnica fue propuesta por Davis. El algoritmo es el siguiente (los padres son P1 y P2):
 - Seleccionar (aleatoriamente) una sub-cadena P1.
 - Producir un hijo copiando la sub-cadena en las posiciones correspondientes a P1. Las posiciones restantes se dejan en blanco.
 - Borrar los valores que ya se encuentren en la sub-cadena de P2. La secuencia resultante contiene los valores faltantes.
 - Colocar los valores en posiciones no conocidas del hijo de izquierda a derecha.
 - Para obtener el segundo hijo, se repiten los pasos del 1 al 4, pero tomando ahora la sub-cadena de P2.
- **Partially Mapped Crossover:** Esta técnica fue propuesta por Goldberg y Lingle y tiene ciertas similitudes con Order Crossover. El algoritmo es el siguiente:
 1. Elegir aleatoriamente dos puntos de cruce.
 2. Intercambiar estos 2 segmentos en los hijos que se generan (como la cruce de 2 puntos convencional).

3. El resto de las cadenas que conforman los hijos se obtienen haciendo mapeos entre los 2 padres:
 4. Si un valor no está contenido en el segmento intercambiado, permanece igual.
 5. Si está contenido en el segmento intercambiado, entonces se sustituye por el valor que tenga dicho segmento en el otro padre.
- **Position-based Crossover:** Esta técnica fue propuesta por Syswerda como una adaptación de la cruce uniforme para permutaciones. El algoritmo es el siguiente:
 1. Seleccionar (al azar) un conjunto de posiciones de P1 (no necesariamente consecutivas).
 2. Producir un hijo borrando de P1 todos los valores, excepto aquéllos que hayan sido seleccionados en el paso anterior.
 3. Borrar los valores seleccionados de P2. La secuencia resultante de valores se usará para completar el hijo.
 4. Colocar en el hijo los valores faltantes de izquierda a derecha, de acuerdo a la secuencia de P2.
 5. Repetir los pasos del 1 al 4, pero tomando ahora la secuencia de P2.
 - **Order-based Crossover:** Esta técnica fue propuesta por Syswerda como una ligera variante de Positionbased Crossover en la que se cambia el orden de los pasos del algoritmos. En este caso, primero seleccionamos una serie de valores de P1. Luego, removemos de P2 esos valores. A continuación generamos un hijo a partir de P2'. Finalmente, completamos el hijo con los valores de la secuencia obtenida de P1 (insertada de izquierda a derecha en el orden impuesto por P1).
 - **Cycle Crossover:** Esta técnica fue propuesta por Oliver, Smith y Holland. Es similar a la Position-based Crossover, porque toma algunos valores de un padre y selecciona los restantes del otro. La principal diferencia es que los valores tomados del primer padre no se seleccionan al azar, sino de acuerdo a ciertas reglas específicas.
 1. Encontrar un ciclo que se define mediante las posiciones correspondientes de los valores entre los padres.
 2. Copiar los valores de P1 que sean parte del ciclo.
 3. Borrar de P2 los valores que estén en el ciclo.
 4. Rellenar el hijo con los valores restantes de P2 (sustituyendo de izquierda a derecha).
 5. Repetir los pasos del 1 al 4, usando ahora P2

2. Contenido

2.1. Order Crossover

```
ORDER CROSS-OVER
P1: 6 1 7 8 10 5 2 3 4 9
P2: 8 4 3 9 8 4 3 4 8
P2': 5 10 1 8 9 6 7 2
O1: 5 10 1 8 9 6 7 3 4 2
P2: 5 10 1 8 9 6 4 3 7 2
P2: 8 4 1 8 9 8 4 8 8
P1': 6 7 10 5 2 3 4
O2: 6 7 1 8 9 10 5 2 3 4
P1: 1 3 7 3 9 10 4 6 2 8
P1: 8 4 3 9 10 4 7 8 8
P2': 6 1 2 3 8 7 5
O1: 6 1 2 3 9 10 4 6 7 5
P2: 6 4 1 2 9 3 8 7 10 5
P2: 8 4 3 9 8 4 8 10 5
P1': 1 7 3 9 4 6 2 8
O2: 1 7 3 9 4 6 2 8 10 5
P1: 4 5 2 1 7 9 8 10 3 6
P1: 8 4 3 9 8 4 3 6
P2': 8 5 2 7 1 4 9 10
O1: 8 5 2 7 1 4 9 10 3 6
P2: 8 6 5 2 3 1 4 9 10
P2: 8 4 3 7 3 1 4 9 10
P1': 5 2 8 6
O2: 5 2 8 6 7 3 1 4 9 10
P1: 5 1 4 8 7 10 3 6 2 9
P1: 8 4 3 7 10 3 6 8 8
P2': 4 2 8 9 5 1
O1: 4 2 8 9 7 10 3 6 5 1
P2: 4 3 2 7 8 9 10 5 6 1
P2: 8 4 3 9 8 4 8 1
P1': 5 4 8 7 10 3 6 2 9
O2: 5 4 8 7 10 3 6 2 9 1
P1: 5 3 4 9 6 7 2 10 8 1
P1: 8 4 4 9 6 7 1 8 8 8
P2': 10 8 2 5 3 1
O1: 10 8 4 9 6 7 2 5 3 1
P2: 10 6 8 9 2 7 5 3 4 1
P2: 10 6 8 9 2 7 5 3 4 1
P1':
O2: 10 6 8 9 2 7 5 3 4 1
```

2.2. Partially Mapped Crossover

```
PARTIALLY MAPPED CROSS-OVER
P1: 8 10 4 2 7 5 1 3 9 6
P2: 3 4 2 7 1 5 10 9 8 6
H1: X X X 7 1 5 X X X X
H2: X X X 2 7 5 X X X X
H1: 8 10 4 7 1 5 2 3 9 6
H2: 3 4 1 2 7 5 10 9 8 6

P1: 2 10 4 6 8 3 9 5 7 1
P2: 4 3 1 8 10 7 2 9 5 6
H1: X X X 8 10 7 X X X X
H2: X X X 6 8 3 X X X X
H1: 2 3 4 8 10 7 9 5 6 1
H2: 4 7 1 6 8 3 2 9 5 10

P1: 9 10 7 3 8 2 6 4 5 1
P2: 1 2 10 3 4 9 8 5 6 7
H1: X X X 3 4 9 X X X X
H2: X X X 3 8 2 X X X X
H1: 2 10 7 3 4 9 6 8 5 1
H2: 1 9 10 3 8 2 4 5 6 7

P1: 1 3 2 8 4 10 5 6 9 7
P2: 4 9 10 5 2 8 3 6 7 1
H1: X X X 5 2 8 X X X X
H2: X X X 8 4 10 X X X X
H1: 1 3 10 5 2 8 4 6 9 7
H2: 5 9 2 8 4 10 3 6 7 1

P1: 2 8 5 10 6 7 9 4 1 3
P2: 10 4 3 8 2 5 6 7 9 1
H1: X X X 8 2 5 X X X X
H2: X X X 10 6 7 X X X X
H1: 7 10 6 8 2 5 9 4 1 3
H2: 5 4 3 10 6 7 8 2 9 1
```

2.3. Position-based Crossover

```
POSITION BASED CROSS-OVER
P1: 1 10 6 2 4 3 9 7 5 8
H1: 8 2 X 2 4 8 9 7 X 8
P2: 5 3 10 1 6
H1: 5 3 10 2 4 1 9 7 6 8
P2: 5 3 9 10 2 4 7 1 8 6
H2: 8 2 9 10 2 4 7 1 X X
P1: 6 5 8
H2: 6 3 9 10 2 4 7 1 5 8
P1: 10 2 9 3 7 4 6 8 1 5
H1: 8 2 X 4 8 X 8 9 X
P2: 7 1 4 3 6 9 2 10 5
H1: 7 1 4 3 6 9 2 8 10 5
P2: 7 1 8 4 3 6 9 2 10 5
H2: 8 X X 3 6 X X 10 X
P1: 2 9 7 4 8 1 5
H2: 2 9 7 4 3 6 8 1 10 5
P1: 6 5 9 4 7 1 2 8 1 10
H1: 8 2 X X X 1 2 X X 10
P2: 4 7 3 9 6 8
H1: 4 5 7 3 9 1 2 6 6 10
P2: 4 10 5 2 1 7 3 9 6 8
H2: 8 10 5 2 1 X X X 6 X
P1: 9 4 7 8 3
H2: 9 10 5 2 1 4 7 8 6 3
P1: 7 9 2 4 10 5 6 1 3 8
H1: 8 X X 4 X X 6 X X 8
P2: 7 4 1 3 5 10 2 8
H1: 7 4 1 3 5 10 6 2 9 8
P2: 8 7 4 1 3 5 10 6 2 9
H2: 8 7 X 4 3 X X X 9
P1: 2 4 10 5 6 1 8
H2: 2 7 4 10 3 5 6 1 8 9
P1: 10 9 8 7 4 6 3 5 1 2
H1: 8 9 X 7 X 6 3 5 X X
P2: 2 1 8 10 4
H1: 2 9 1 7 8 6 3 5 10 4
P2: 2 1 6 9 8 7 10 4 5 3
H2: 8 X X X X 10 4 X X
P1: 9 8 7 6 3 5 1 2
H2: 9 8 7 6 3 5 10 4 1 2
```

2.4. Order-based Crossover

```
ORDER-BASED CROSS-OVER
P1: 2 4 10 7 9 6 1 3 8 5
P2: 10 5 1 3 4 10 9 8 3 X 5
Cycle: 10
P1: 7 6 1 8 2
P2: 7 4 10 6 9 1 8 3 2 5
P1: 7 3 6 10 9 1 5 8 2 4
P2: 10 5 1 3 4 10 9 8 3 X 5
P1: 2 4 10 7 8 3 6 5
P2: 2 4 10 7 9 1 8 3 6 5
P1: 5 3 9 10 7 8 2 1 4 6
P2: 10 5 1 3 4 10 9 8 3 1 5 6
P1: 5 4 10 7 9 2 1 3 6
P2: 5 4 10 7 9 2 8 1 6 3
P1: 5 4 10 7 9 2 8 1 6 3
P2: 10 5 1 3 4 10 9 8 3 1 5 6
P1: 5 10 7 2 1
P2: 5 4 10 7 9 2 8 1 6 3
P1: 8 6 5 7 1 4 9 2 3 10
P2: 10 5 1 3 4 10 9 8 3 1 5 6
P1: 8 9 10 1 7 4 3
P2: 8 6 5 9 10 1 7 2 4 3
P1: 5 8 9 10 1 2 7 4 6 3
P2: 10 5 1 3 4 10 9 8 3 1 5 6
P1: 8 6 5 7 1 4 9 3 10
P2: 8 6 5 7 1 2 4 9 3 10
P1: 5 6 2 3 8 10 9 4 7 1
P2: 10 5 1 3 4 10 9 8 3 1 5 6
P1: 7 3 2 1 5
P2: 7 6 2 2 8 10 9 4 1 5
P1: 7 9 2 2 4 1 8 1 10 6
P2: 10 5 1 3 4 10 9 8 3 1 5 6
P1: 5 6 3 10 7
P2: 5 9 6 2 4 1 8 3 10 7
P1: 4 5 8 7 1 2 9 10 3 6
P2: 10 5 1 3 4 10 9 8 3 1 5 6
P1: 2 1 5 6 4 10 9 8 3
P2: 2 1 5 7 6 4 10 9 8 3
P1: 2 1 5 6 7 4 10 9 8 3
P2: 10 5 1 3 4 10 9 8 3 1 5 6
P1: 7 1 2 9 10 6
P2: 7 1 2 2 9 4 10 6 8 3
```

2.5. Cycle Crossover

```
CYCLE CROSS-OVER
P1: 10 5 6 1 2 4 3 7 9 8
P2: 10 9 5 3 4 1 6 8 2 7
Cycle: 10
P1: 10 4 X X X X X X X
P2: 9 5 3 4 1 6 8 2 7
P1: 10 9 5 3 4 1 6 8 2 7
P2: 10 4 X X X X X X X
P1: 5 6 3 2 4 3 7 9 8
P2: 10 5 6 1 2 4 3 7 9 8
P1: 9 6 2 1 3 5 8 4 7 10
P2: 4 8 10 3 9 6 1 2 7 5
Cycle: 10
P1: 8 4 X X X X X X X
P2: 4 8 3 9 6 1 2 7 5
P1: 4 8 3 9 6 1 2 7 5
P2: 8 4 10 3 9 6 1 2 7 5
P1: 9 6 2 1 3 5 8 4 7
P2: 9 6 10 2 1 3 5 8 4 7
P1: 2 6 4 8 1 10 3 5 9 7
P2: 8 10 6 9 5 3 2 1 7 4
Cycle: 10
P1: 8 4 X X X X X X X
P2: 8 6 9 5 3 2 1 7 4
P1: 8 6 9 5 3 10 2 1 7 4
P2: 8 4 10 3 9 6 1 2 7 5
P1: 2 6 4 8 1 3 5 9 7
P2: 2 10 6 4 8 1 3 5 9 7
P1: 6 4 8 2 10 1 9 7 3 5
P2: 6 5 10 2 8 7 1 9 3 4
Cycle: 10
P1: 8 4 X X X X X X X
P2: 6 5 2 8 7 1 9 3 4
P1: 6 5 2 8 10 7 1 9 3 4
P2: 8 4 10 3 9 6 1 2 7 5
P1: 6 4 8 2 1 9 7 3 5
P2: 6 4 10 8 2 1 9 7 3 5
P1: 8 4 6 3 10 2 5 7 1 9
P2: 1 10 2 8 5 7 3 6 4 9
Cycle: 10
P1: 8 4 X X X X X X X
P2: 1 2 8 5 7 3 6 4 9
P1: 1 2 8 5 10 7 3 6 4 9
P2: 8 4 10 3 9 6 1 2 7 5
P1: 8 4 6 3 2 5 7 1 9
P2: 8 10 6 6 2 5 7 1 9
```

3. Conclusión

En esta práctica se pudieron ver las diferencias de algunos de los algoritmos más populares de cruce para permutaciones, se implementaron y se analizaron.