

Reporte práctica 8

Introducción

Mutación

La mutación se considera como un operador secundario en los algoritmos genéticos canónicos. Es decir, su uso es menos frecuente que el de la cruce. En la práctica, se suelen recomendar porcentajes de mutación de entre 0.001 y 0.01 para la representación binaria.

Algunos investigadores, sin embargo, han sugerido que el usar porcentajes altos de mutación al inicio de la búsqueda, y luego decrementarlos exponencialmente, favorece el desempeño de un AG.

Otros autores sugieren que $p_m = \frac{1}{L}$ donde L es la longitud de la cadena cromosómica es un límite inferior para el porcentaje óptimo de la mutación. El papel que juega la mutación en el proceso evolutivo, así como su comparación con la cruce, sigue siendo tema frecuente de investigación y debate en la comunidad de computación evolutiva.

Mutación para permutaciones

A continuación se muestran los tipos de mutaciones para permutaciones y que se programaron para esta práctica:

- Inserción
- Desplazamiento
- Intercambio recíproco
- Heurística

Mutación por Inserción

Se selecciona un valor en forma aleatoria y se le inserta en una posición arbitraria.

Ejemplo de mutación por inserción.

Dada:

P = 9 4 2 1 5 7 6 10 3 8

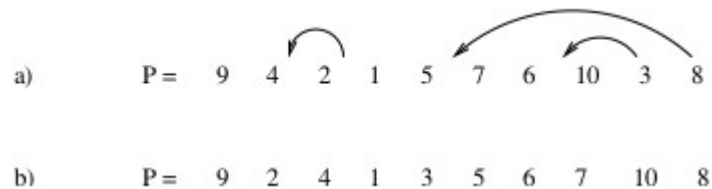
Si suponemos que elegimos la posición 7 y decidimos mover ese valor a la posición 2, tendríamos:

P' = 9 6 4 2 1 5 7 10 3 8

Mutación por desplazamiento

Es una generalización de la mutación por inserción en la que en vez de mover un solo valor, se cambian de lugar varios a la vez.

Ejemplo de mutación por desplazamiento.



Mutación por intercambio recíproco

En este caso, se seleccionan dos puntos al azar y se intercambian estos valores de posición.

Por ejemplo, dada:

P = 9 4 2 1 5 7 6 10 3 8

Tendríamos:

P' = 9 10 2 1 5 7 6 4 3 8

Mutación Heurística

Fue propuesta por Gen y Cheng. El procedimiento es el siguiente:

1. Seleccionar λ genes al azar.
2. Generar vecinos de acuerdo a todas las permutaciones posibles de los genes seleccionados.
3. Evaluar todos los vecinos y seleccionar el mejor.

Por ejemplo, consideremos el siguiente individuo.

P = 9 4 2 1 5 7 6 10 3 8

Generar todas las permutaciones de: 4 5 10

- | | |
|-----------|-----------|
| 1) 4 10 5 | 4) 10 5 4 |
| 2) 5 4 10 | 5) 10 4 5 |
| 3) 5 10 4 | |

Ruiz Beltrán Jonatan Zuriel

Individuos generados:

P1= 9 4 2 1 10 7 6 5 3 8
P2= 9 5 2 1 4 7 6 10 3 8
P3= 9 5 2 1 10 7 6 4 3 8
P4= 9 10 2 1 5 7 6 4 3 8
P5= 9 10 2 1 4 7 6 5 3 8

De entre ellas, se selecciona a la mejor. En este caso, supondremos que es P4:

P' = 9 10 2 1 5 7 6 4 3 8

Capturas de pantalla

```
[zuriel@fatboy09-pc practica8]$ ./main
Ingrese el numero de individuos que desea: 8
Ingrese el numero de alelos que desea: 6
```

Mutación por Inserción

```
Tecnicas de Mutacion:
1. Insercion
2. Desplazamiento
3. Intercambio reciproco
4. Heuristico
5. Salir
Ingrese la opcion deseada: 1
```

No.	Individuo	Descendencia
1	146235	(4->1) => 214635
2	325416	(4->3) => 324516
3	625134	(6->5) => 625143
4	352146	(4->3) => 351246
5	436152	(4->3) => 431652
6	621435	(6->4) => 621543
7	251364	(3->1) => 125364
8	354261	(5->1) => 635421

Presione la tecla enter para continuar...

Ruiz Beltrán Jonatan Zuriel

Mutación por Desplazamiento

```
Tecnicas de Mutacion:
1. Insercion
2. Desplazamiento
3. Intercambio reciproco
4. Heuristico
5. Salir
Ingrese la opcion deseada: 2

-----+-----+-----+
      No.  | Individuo  | Descendencia
-----+-----+-----+
      1    | 214635    | (2->1) => (6->4) => 124563
      2    | 324516    | (6->5) => (5->4) => (4->1) => (3->1) => (6->5) => 263415
      3    | 625143    | (6->2) => (6->5) => (6->5) => (3->2) => (6->5) => 623541
      4    | 351246    | (6->5) => (5->1) => (6->5) => (6->1) => (5->1) => 126354
      5    | 431652    | (6->5) => (4->1) => 643125
      6    | 621543    | (4->3) => (6->5) => 625134
      7    | 125364    | (5->2) => (2->1) => (6->5) => 612543
      8    | 635421    | (6->5) => (4->2) => 643512

Presione la tecla enter para continuar...
```

Mutación por intercambio recíproco

```
Tecnicas de Mutacion:
1. Insercion
2. Desplazamiento
3. Intercambio reciproco
4. Heuristico
5. Salir
Ingrese la opcion deseada: 3

-----+-----+-----+
      No.  | Individuo  | Descendencia
-----+-----+-----+
      1    | 124563    | (3->1) => 421563
      2    | 263415    | (5->3) => 261435
      3    | 623541    | (5->4) => 623451
      4    | 126354    | (6->4) => 126453
      5    | 643125    | (6->5) => 643152
      6    | 625134    | (5->4) => 625314
      7    | 612543    | (2->1) => 162543
      8    | 643512    | (6->4) => 643215

Presione la tecla enter para continuar...
```

Mutación Heurística

```
Tecnicas de Mutacion:
1. Insercion
2. Desplazamiento
3. Intercambio reciproco
4. Heuristico
5. Salir
Ingrese la opcion deseada: 4

-----+-----+
No.      | Individuo
-----+-----+
1        | 421563
    Generando permutaciones de: 4 2 1
    1) 4 1 2 => 4 1 2 5 6 3
    2) 2 4 1 => 2 4 1 5 6 3
    3) 2 1 4 => 2 1 4 5 6 3
    4) 1 2 4 => 1 2 4 5 6 3
    5) 1 4 2 => 1 4 2 5 6 3
    Seleccionamos el mejor: 4 1 2 5 6 3

-----+-----+
2        | 261435
    Generando permutaciones de: 6 4 5
    1) 6 5 4 => 2 6 1 5 3 4
    2) 4 6 5 => 2 4 1 6 3 5
    3) 4 5 6 => 2 4 1 5 3 6
    4) 5 4 6 => 2 5 1 4 3 6
    5) 5 6 4 => 2 5 1 6 3 4
    Seleccionamos el mejor: 2 6 1 5 3 4

-----+-----+
3        | 623451
    Generando permutaciones de: 6 2 3
    1) 6 3 2 => 6 3 2 4 5 1
    2) 2 6 3 => 2 6 3 4 5 1
    3) 2 3 6 => 2 3 6 4 5 1
    4) 3 2 6 => 3 2 6 4 5 1
    5) 3 6 2 => 3 6 2 4 5 1
    Seleccionamos el mejor: 6 3 2 4 5 1

-----+-----+
4        | 126453
    Generando permutaciones de: 2 6 4
    1) 2 4 6 => 1 2 4 6 5 3
    2) 6 2 4 => 1 6 2 4 5 3
    3) 6 4 2 => 1 6 4 2 5 3
    4) 4 6 2 => 1 4 6 2 5 3
    5) 4 2 6 => 1 4 2 6 5 3
    Seleccionamos el mejor: 1 6 4 2 5 3

-----+-----+
5        | 643152
    Generando permutaciones de: 4 1 5
    1) 4 5 1 => 6 4 3 5 1 2
    2) 1 4 5 => 6 1 3 4 5 2
    3) 1 5 4 => 6 1 3 5 4 2
    4) 5 1 4 => 6 5 3 1 4 2
    5) 5 4 1 => 6 5 3 4 1 2
    Seleccionamos el mejor: 6 5 3 4 1 2

-----+-----+
```

Conclusiones

En conclusión, el objetivo de esta práctica fue el implementar algunas técnicas de mutación para permutaciones en los individuos, los algoritmos implementados son sencillos y fácil de programar. Para implementar estos algoritmos en la práctica nuestras cadenas ahora son representadas por enteros para representar a una permutación. La permutación es una operación que se usa menos frecuente en un algoritmo genético, ya que esta operación puede o no mejorar a nuestro individuo, suele recomendarse utilizar porcentajes de mutación entre 0.001 y 0.01 para la representación binaria.

Referencias bibliograficas

- Coello, C. (2008). Introducción a la Computación Evolutiva (Notas de Curso). México, pp.165-167.