

Instituto Politécnico Nacional

Escuela Superior de Computo

Algoritmos Genéticos

Práctica 6: Cruza

Hernández López Ángel Zait

2014080682

3CM5

Periodo: 2019/01

Planteamiento del problema

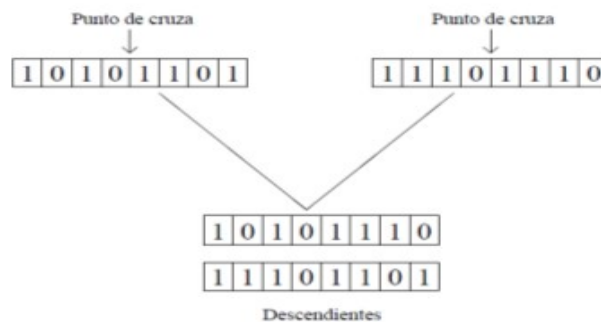
Realizar un programa el cual se tenga que desarrollar los tipos de cruza, los cuales son cruza por un punto, cruza por dos puntos, cruza uniforme y cruza acentuada.

Introducción

Los tipos de cruza que se solicitan son para individuos de bits. A continuación se explicará brevemente cada una de las cruza.

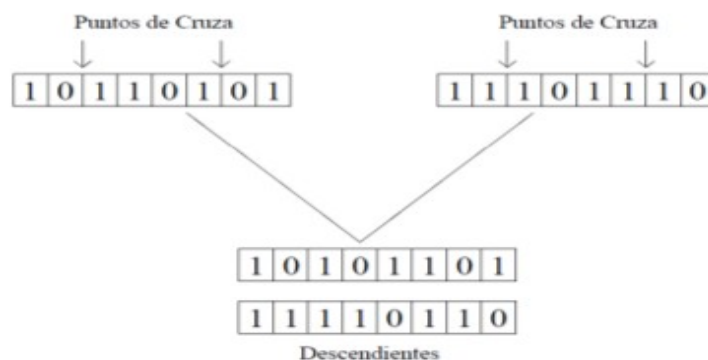
Cruza por un punto:

En este tipo de cruza, tenemos dos individuos los cuales son los padres. Para realizar la cruza, se elige un alelo al azar, y después, se intercambian los alelos después de esa posición. Para crear a los hijos.



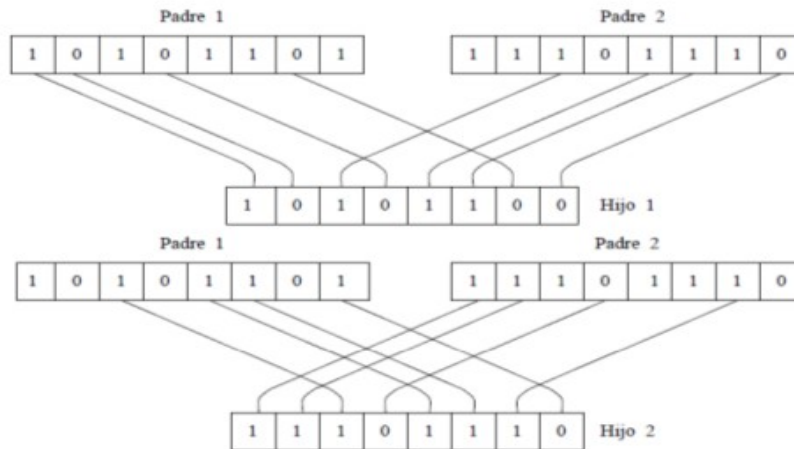
Cruza por dos puntos:

Es el mismo principio que la cruza por un punto, en lugar de elegir un alelo, se escogen dos, los cuales son los puntos de cruza, entre esos alelos, se intercambiarán los alelos de los padres, para poder crear un nuevo hijo.



Cruza uniforme:

En este tipo de cruza Se van a intercambiar el mismo numero de alelos de ambos padres, es decir, si se seleccionan cuatro alelos al azar del padre uno, esos mismo alelos serán del hijo uno, y los alelos restantes que falten serán del padre dos. De igual forma para el otro hijo, los alelos seleccionados del padre dos, serán los alelos del hijo dos y los restantes del padre dos.



Cruza acentuada:

Este tipo de cruce puede parecerse a un poco de todo, ya que en los alelos, en cualquiera de los padres, se tiene un identificador, el cual indica que después de ese identificador se hace el intercambio de bits, hasta encontrar otro identificador en cualquiera de los padres.

Antes de la cruce:

P1 = a a a a a a a! b b b b b b b
P2 = c c c c! d d d d d d! e e e e

Después de la cruce:

H1 = a a a a d d d b b b e e e e
H2 = c c c c! a a a! d d d! b b b b

Contenido

Para el desarrollo de esta practica se crearon funciones, el cual una de ellas se le tiene asignado un tipo de cruce, a continuación, se explicará un poco de las funciones principales para la resolución del problema:

`void CruzaPorPunto(void):` Realiza el punto de cruce para cada dos individuos, se generan individuos divididos entre dos números al azar, en el rango de cero y el numero de alelos de los individuos. Después de eso, se realiza el método de cruce por un punto.

| Cruza por un punto. | | | |
|---------------------|------------|----------------|--------------|
| No. | Cruza | Punto de cruce | Descendencia |
| 1 | 0101101001 | 4 | 0101001000 |
| 2 | 0001001000 | 4 | 0001101001 |
| 3 | 0101100001 | 3 | 0100100101 |
| 4 | 0010100101 | 3 | 0011100001 |
| 5 | 0010001101 | 8 | 0010001100 |
| 6 | 1010100100 | 8 | 1010100101 |
| 7 | 1000000010 | 9 | 1000000011 |
| 8 | 1101001101 | 9 | 1101001100 |
| 9 | 0111101101 | 3 | 0110110101 |
| 10 | 1110110101 | 3 | 1111101101 |

void CruzaPorDosPuntos(void): Realiza el punto de cruza para cada dos individuos, se generan individuos números al azar, en el rango de cero y el numero de alelos de los individuos. Después de eso, se realiza el método de cruza por dos puntos.

Cruza por dos puntos.

| No. | Cruza | Punto de cruza | Descendencia |
|-----|------------|----------------|--------------|
| 1 | 0101101001 | 4 | 0101101001 |
| 2 | 0001001000 | 3 | 0001001000 |
| 3 | 0101100001 | 8 | 0101100001 |
| 4 | 0010100101 | 9 | 0010100101 |
| 5 | 0010001101 | 3 | 0010001101 |
| 6 | 1010100100 | 2 | 1010100100 |
| 7 | 1000000010 | 7 | 1001001010 |
| 8 | 1101001101 | 2 | 1100000101 |
| 9 | 0111101101 | 2 | 0111101101 |
| 10 | 1110110101 | 1 | 1110110101 |

void CruzaUniforme(void): Genera la cruza uniforme para cada dos individuos, recorriendo alelo por alelo y generando un verdadero o falso, para saber si en ese lugar habrá o no un intercambio de bit para generar a los hijos.

Cruza uniforme.

| No. | Cruza | Descendencia |
|-----|------------|--------------|
| 1 | 0101101001 | 0101101001 |
| 2 | 0001001000 | 0001001000 |
| 3 | 0101100001 | 0011100101 |
| 4 | 0010100101 | 0100100001 |
| 5 | 0010001101 | 1010000101 |
| 6 | 1010100100 | 0010101100 |
| 7 | 1000000010 | 1101001000 |
| 8 | 1101001101 | 1000000111 |
| 9 | 0111101101 | 1110111101 |
| 10 | 1110110101 | 0111100101 |

void CruzaAcentuada(void): Genera la cruza acentuada para cada dos individuos, en cada individuo se generan acentuaciones al azar, que no pasen del numero de alelos, y dependiendo donde esté la acentuación, se generará el intercambio de bits.

Cruza acentuada.

| No. | Cruza | Descendencia |
|-----|------------|--------------|
| 1 | 0101101001 | 0101101001 |
| 2 | 0001001000 | 0001001000 |
| 3 | 0101100001 | 0101100001 |
| 4 | 0010100101 | 0010100101 |
| 5 | 0010001101 | 0010001101 |
| 6 | 1010100100 | 1010100100 |
| 7 | 1000000010 | 1001001010 |
| 8 | 1101001101 | 1100000101 |
| 9 | 0111101101 | 0111101101 |
| 10 | 1110110101 | 1110110101 |

Conclusiones

Se pudo observar los distintos tipos de cruza que se pueden usar en una cadena de bits, esto puede servir para mejorar las generaciones futuras de una población inicial. Al igual de que no solo hay una forma de hacer cruza, si no que pueden haber varias, y dependiendo el caso o la problemática, usar uno para que se tenga un mejor resultado.