



**INSTITUTO POLITÉCNICO NACIONAL**  
**ESCUELA SUPERIOR DE CÓMPUTO**



**GENETIC ALGORITHMS**

***REPORTE PRACTICA VI***

***“CRUZA”***

ALUMNO: SOLIS SANCHEZ JHOVANY

GRUPO: 3CM5

BOLETA: 2015630489

## INTRODUCCIÓN

Cruza de un punto.

Esta técnica fue propuesta por Holland, y fue muy popular durante muchos años. Hoy en día, sin embargo, no suele usarse mucho en la práctica debido a sus inconvenientes.

Cruza de dos puntos.

DeJong fue el primero en implementar una cruce de  $n$  puntos, como una generalización de la cruce de un punto. El valor  $n = 2$  es el que minimiza los efectos disruptivos (o destructivos) de la cruce y de ahí que sea usado con gran frecuencia. No existe consenso en torno al uso de valores para  $n$  que sean mayores o iguales a 3. Los estudios empíricos al respecto proporcionan resultados que no resultan concluyentes respecto a las ventajas o desventajas de usar dichos valores. En general, sin embargo, es aceptado que la cruce de dos puntos es mejor que la cruce de un punto. Asimismo, el incrementar el valor de  $n$  se asocia con un mayor efecto disruptivo de la cruce

Cruza uniforme.

Esta técnica fue propuesta originalmente por Ackley, aunque se le suele atribuir a Syswerda. En este caso, se trata de una cruce de  $n$  puntos, pero en la cual el número de puntos de cruce no se fija previamente. La cruce uniforme tiene un mayor efecto disruptivo que cualquiera de las 2 cruces anteriores. A fin de evitar un efecto excesivamente disruptivo, suele usarse con  $P_c = 0.5$ . Algunos investigadores, sin embargo, sugieren usar valores más pequeños de  $P_c$ . Cuando se usa  $P_c = 0.5$ , hay una alta probabilidad de que todo tipo de cadena binaria de longitud  $L$  sea generada como máscara de copiado de bits.

Cruza Acentuada.

Esta técnica fue propuesta por Schaffer y Morishima, en un intento por implementar un mecanismo de auto-adaptación para la generación de los patrones favorables (o sea, los buenos bloques constructores) de la cruce. En vez de calcular directamente la máscara (o patrón) de cruce, la idea es usar una cadena binaria de “marcas” para indicar la localización de los puntos de cruce. La idea fue sugerida por Holland [127], aunque en un sentido distinto. La información extra que genera la cruce acentuada se agrega al cromosoma de manera que el número y localizaciones de los puntos de cruce pueda ser objeto de manipulación por el AG. Por tanto, las cadenas tendrán una longitud del doble de su tamaño original. La convención que suele adoptarse es la de marcar con ‘1’ las posiciones donde hay cruce y con ‘0’ las posiciones donde no la hay. Asimismo, se suelen usar signos de admiración para facilitar la escritura de las cadenas.

## DESARROLLO

En esta practica se implementaran cada una de las técnicas de cruce anteriormente descritas. Para ello se realizaran como pruebas del algoritmo cruza entre dos individuos de 10 bits dentro de un población.

### Cruza de un punto.

En esta implementación el algoritmo mostrara los individuos originales, el punto donde se realizo la cruce así como sus correspondientes resultados.

```
-- Cruza de un punto --
323      |0|1|0|1|0|0|0|0|1|1|
307      |0|1|0|0|1|1|0|0|1|1|

          Cruza en: 8

323      |0|1|0|1|0|0|0|0|1|1|
307      |0|1|0|0|1|1|0|0|1|1|
```

*Illustration 1: Un Punto*

Para su implementación se invierten los bits de los padres dependiendo de la posición donde se realice la cruce.

### Cruza de dos puntos.

Al igual que la anterior se mostraran los los padres, sus descendientes y los puntos que se consideraron para hacer la cruce.

```
-- Cruza de dos puntos --
145      |0|0|1|0|0|1|0|0|0|1|
256      |0|1|0|0|0|0|0|0|0|0|

          Cruza en: 0 5

273      |0|1|0|0|0|1|0|0|0|1|
128      |0|0|1|0|0|0|0|0|0|0|
```

*Illustration 2: Dos Puntos*

Los puntos donde se realiza la cruce son generados aleatoriamente para ambos casos con la única diferencia que el caso del la cruce de dos puntos estos se tienen que ordenar de manera ascendente.

### Cruza Uniforme.

Para esta técnica se generaron aleatoriamente los puntos que se tomarían de cada padre para formar un nuevo descendiente.

```
-- Cruza Uniforme --

836      |1|1|0|1|0|0|0|1|0|0|
124      |0|0|0|1|1|1|1|1|0|0|

          Cruza en: 5 7 1 3 2
          Cruza en: 2 4 1 8 5

596      |1|0|0|1|0|1|0|1|0|0|
332      |0|1|0|1|0|0|1|1|0|0|
```

*Illustration 3: Uniforme*

Para cada descendiente se generan puntos que en total es la mitad de bits de la longitud de los padres.

### Cruza acentuada.

Para esta técnica de cruce se genera un patrón para cada descendiente el cual al igual que las demás técnicas será generado aleatoriamente. El algoritmo comenzará guardando los bits del padre A al mismo tiempo que recorre el patrón bit en bit con el fin de que cuando encuentre un cambio en el valor comenzará a guardar los bits del padre B y seguirá hasta encontrar otro cambio el cual lo regrese a leer los bits del padre A.

```
-- Cruza Acentuada --

958      |1|1|1|0|1|1|1|1|1|0|
305      |0|1|0|0|1|1|0|0|0|1|

          Patron A: 0 0 0 1 0 0 0 0 0 0
          Patron B: 1 0 1 0 0 0 0 0 0 1

945      |1|1|1|0|1|1|0|0|0|1|
816      |1|1|0|0|1|1|0|0|0|0|
```

*Illustration 4: Acentuada*

La marca indica cuando el algoritmo tiene que cambiar el padre que está leyendo. Con el fin de optimizar el patrón es almacenado en una cadena de longitud igual al doble de la longitud de los padres.

## **CONCLUSIÓN**

La cruce de individuos es muy importante ya que con ella es posible poder tener mejores individuos conforme pasen las generaciones. Eso algoritmos implementados pueden hasta el momentos manejar cadenas de individuos de un tamaño grande con las limitantes de memoria y procesamiento de la computadora donde se este ejecutando. Aun estos algoritmos es posible opalizarlos para mejorar el tiempo de ejecución y memoria para cuando se ejecuten en conjunto con las técnicas de selección y mutación así como para grande poblaciones o gran cantidad de bits por individuo.