

Introducción a la Inteligencia Artificial

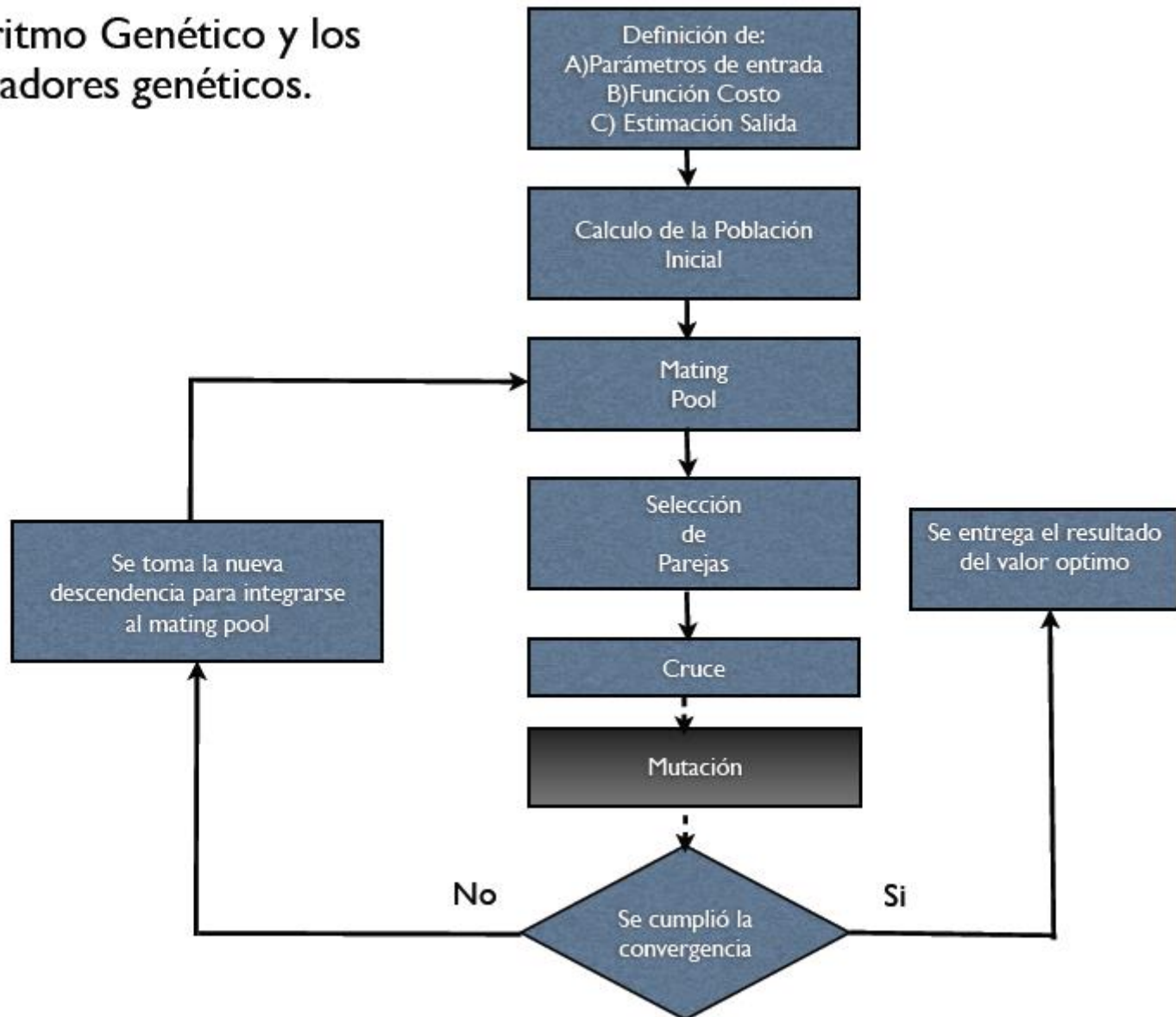
Unidad IV

Algoritmos Genéticos.

(cont.)

El Algoritmo Genético y los operadores genéticos.

El Algoritmo Genético y los
operadores genéticos.



Análisis, Diseño e Implementación de un Algoritmo Genético.

- Los algoritmos genéticos(AG) empiezan definiendo un cromosoma (representación de un miembro de la población) como un arreglo de valores de los parámetros a ser optimizados.
- Si el cromosoma tiene N_{par} parámetros (un problema de optimización N_{par} -dimensional) dado por:

$p_1, p_2, p_3, \dots, p_{N_{par}}$

Análisis, Diseño e Implementación de un Algoritmo Genético.

- El cromosoma es escrito como un arreglo de elementos de tamaño N_{par}

$$\text{cromosoma}=[p1,p2,p3,\dots, p_{N_{par}}]$$

- Por ejemplo: Caso 1: La empresa productora de la cerveza vudweiser les encarga el diseño de una lata con capacidad de 1 litro, ¿ Cuales deben ser las dimensiones para minimizar la cantidad de metal ?
- $N_{par}=2$ altura=h, radio=r cromosoma=[h,r]

Análisis, Diseño e Implementación de un Algoritmo Genético.

- Cada cromosoma tiene un costo, que es calculado mediante la evaluación de la función costo

$$\text{Costo} = f(\text{cromosoma})$$

- En caso de la lata: $f(h,r)$

Población Inicial

- El algoritmo genético empieza con una gran comuna de cromosomas conocidos como población inicial
- La población inicial tiene N_{IPOP} cromosomas y es una matriz de $N_{IPOP} \times N_{BITS}$ llenada aleatoriamente con ceros y unos (en el caso de AG's binarios) generados de:

$$\text{Matriz IPOP} = \text{round}\{\text{random}(N_{IPOP} \times N_{BITS})\}$$

Población Inicial

- Ejemplo: Sea la función $f(x)=x^2-1$ en el rango de $0 \leq x \leq 6$ calcular la población inicial, obtener su valor máximo.

Primero debemos estimar N_{BITS} (el tamaño del cromosoma)

¿Cuántos bits necesitamos para poder representar el valor máximo de x (6)?

respuesta = 3 bits

Población Inicial

- ¿ Cuanto vale N_{IPOP} ?
- ¿ Cuantos elementos se deben considerar de la población inicial ?
- ¿ Cuantos Cromosomas ?
- Se debe hacer una estimación para probar con pocos cromosomas.
- La experiencia dirá cuantos, pero jamás

$$N_{IPOP} \leq N_{BITS}$$

Población Inicial

Paso 1

#	cromosoma		
1	1	0	1
2	0	0	1
3	1	0	0
4 N _{IPOP}	0	0	0

Población Inicial

Paso 2

#	cromosoma			x
1	1	0	1	5
2	0	0	1	1
3	1	0	0	4
4 N_{IPOP}	0	0	0	0

Población Inicial

Paso 2

#	cromosoma			x	costo
1	1	0	1	5	24
2	0	0	1	1	0
3	1	0	0	4	15
4 N_{IPOP}	0	0	0	0	-1

Mating Pool

- Debido a que la población inicial es muy grande para los procesos iterativos de los algoritmos genéticos, una gran parte de los cromosomas con los costos menos aptos son descartados por el proceso de ***Mating Pool***.
- El resultado serán los cromosomas mas aptos.
- Primero se *rankean* los cromosomas en base a los costos de los N_{IPOP} , desde el mas bajo al mas alto en caso de problemas de minimización y en maximización del costo mas alto al mas bajo.

Población Inicial

#	cromosoma			x	costo
1	1	0	1	5	24
2	1	0	0	4	15
3	0	0	1	1	0
4 N_{IPOP}	0	0	0	0	-1

Mating Pool

- A continuación se hace el primer muestreo, seleccionando a los primeros N_{POP} mejores para cada iteración, los demás son descartados.

$$N_{POP} = \text{random}(1, N_{IPOP})$$

se debe cumplir $N_{POP} \leq N_{IPOP}$

Mating Pool

- Supongamos que para nuestro ejemplo:

$$N_{\text{POP}} = \text{random}(1, 4) = 3$$

$N_{\text{POP}} =$

#	cromosoma			x	costo
1	1	0	1	5	24
2	1	0	0	4	15
3	0	0	1	1	0

Mating Pool

- De los N_{POP} cromosomas en una generación, solamente los mas aptos (N_{GOOD} mejores) sobreviven para su cruce y los peores (N_{BAD}) son descartados para hacer espacio para la descendencia:

$$N_{GOOD} = \text{random}(1, N_{POP})$$

$$N_{BAD} = N_{POP} - N_{GOOD}$$

Mating Pool

Suponiendo que $N_{\text{GOOD}} = 2$

N_{POP}	#	cromosoma			x	costo	N_{GOOD}
	1	1	0	1	5	24	
	2	1	0	0	4	15	N_{BAD}
	3	0	0	1	1	0	

Mating Pool

Permitir que solo unos cuantos cromosomas sobrevivan a la siguiente generación limita la disponibilidad de genes en los descendientes.

Almacenar demasiados cromosomas permitirá que algunos muy malos pasen a la siguiente generación.

Selección de Pareja

- Del total de los N_{GOOD} cromosomas, se seleccionan parejas para producir nuevos descendientes con la operación genética de cruce.
- MÉTODOS DE SELECCIÓN DE PAREJA:
 - Pairing from top to bottom
 - Random pairing
 - Weigthed random pairing
 - Rank Weighting
 - Cost Weighting
 - Tournament

Pairing from top to bottom

- Empieze en lo mas alto de la lista y seleccione una pareja de cromosomas a la vez hasta que el tope de los N_{good} sean seleccionados para reproducirse.
- El algoritmo seleccionara parejas cromosoma $_{2i-1}$ con el cromosoma $_{2i}$ para $i=1,2,3....$
- Asi pues las parejas seran el cromosoma 1 y 2, 3 y 4, etc.
- Este enfoque no es el mejor modelo natural, pero es muy sencillo de programar

Random pairing

- Este enfoque utiliza un generador de numeros aleatorios para seleccionar las parejas.
- Los cromosomas son rankeados en terminos de costo desde 1 hasta N_{good} y se generan 2 numeros aleatorios para encontrar a las primeras 2 parejas.
- Un padre es seleccionado por:
 - $\text{padre} = \text{roundup}\{N_{\text{good}} \times \text{random}\}$

Random pairing

- Donde $\text{roundup}\{\}$ redondea su argumento el entero siguiente mas alto.
- Ejemplo: sea $N_{\text{good}}=6$ y los 6 numeros aleatorios en la siguiente tabla

random	$\text{roundup}\{N_{\text{good}} \times \text{random}\}$
0.1535	1
0.6781	5
0.0872	1
0.1936	2
0.7021	5
0.3933	3

Random pairing

Las parejas serían:

- cromosoma 1 con cromosoma 5
- cromosoma 1 con cromosoma 3
- cromosoma 5 con cromosoma 3

Tournament

- Otro enfoque es la competición de la reproducción más cercana a la naturaleza, se toma aleatoriamente un subconjunto de cromosomas con los costos más bajos en el subconjunto, y se forma una pareja.
- El torneo se repite para cada padre que sea necesario.
- Considere 6 torneos entre 2 cromosomas seleccionados aleatoriamente del mating pool y seleccione los mejores cromosomas de los torneos como padres.

Tournament

Torneo	6 x random (2 , 1)	Padre
1	2,1	1
2	5,5	5
3	6,3	3
4	4,5	4
5	1,1	1
6	4,5	4

Crom 1 - crom 5

Crom 3 - crom 4

Crom 1 - crom 4

Cruce

- El cruzamiento es la creacion de uno o mas descendientes de los padres seleccionados en el proceso de selección.
- Este es el primer camino que utiliza un algoritmo genetico para explorar una superficie de costo.
- Este proceso es llamado exploracion porque el algoritmo genetico hace uso de las combinaciones de bits que estan presentes en los cromosomas.

Tipos de Cruce

- Tipos básicos de Operación cruce
 - Sencillo
 - Doble
 - Uniforme

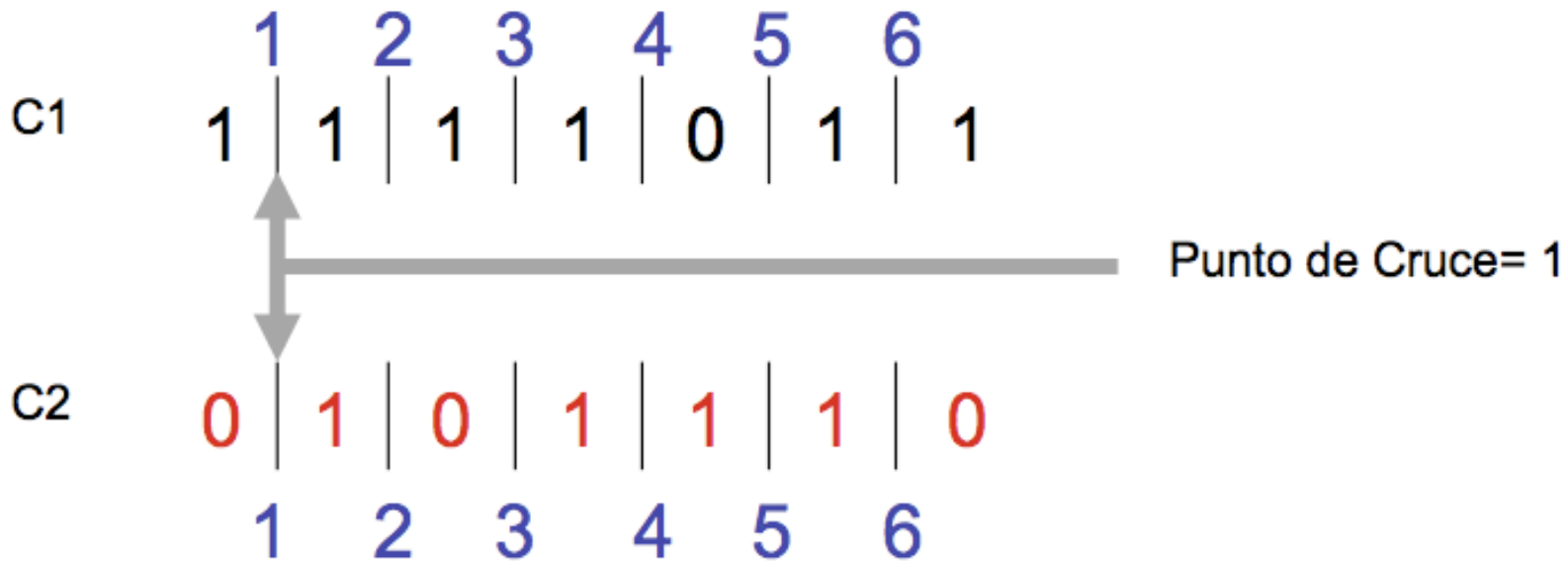
Cruce Sencillo

- Dados una pareja de cromosomas C_1 y C_2 donde $|C_1| = |C_2|$
- Se genera un random con la siguiente formula:
- Punto de cruce = $\text{roundup} \{ \text{random} \times (|C_1| - 1) \}$

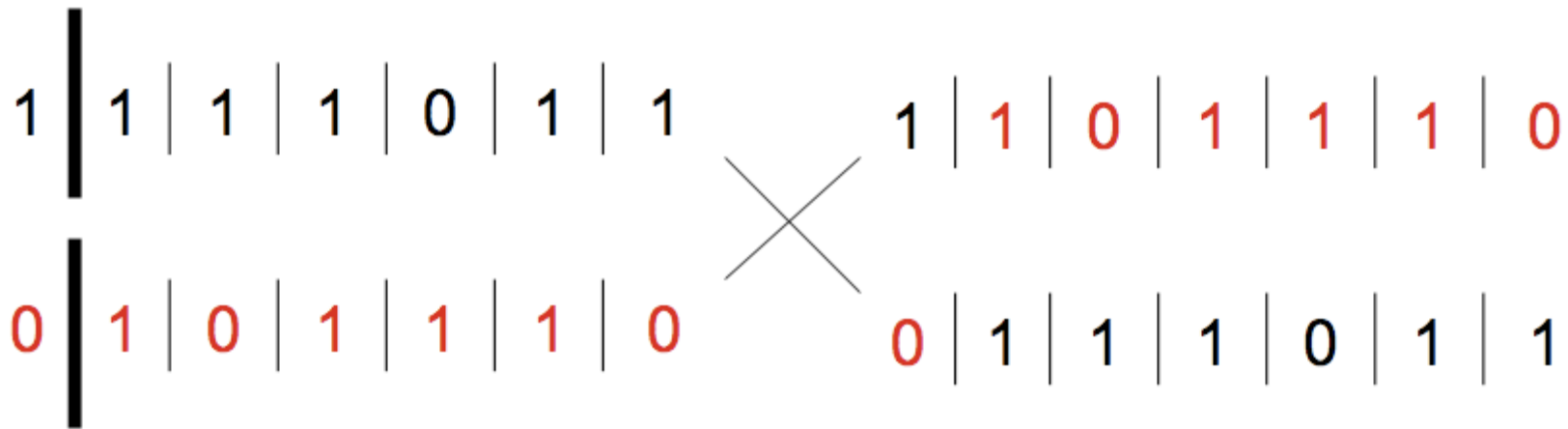
Cruce Sencillo

- Sean $C1 = 1111011$ y $C2 = 0101110$
- $|C1| = 7$ $\text{random} = 0.1535$
- Punto de cruce = $\text{roundup} \{ 0.1535 \times (7-1) \}$
- Punto de cruce = $\text{roundup} \{ 0.921 \}$
- Punto de cruce = 1

Cruce Sencillo



Cruce Sencillo



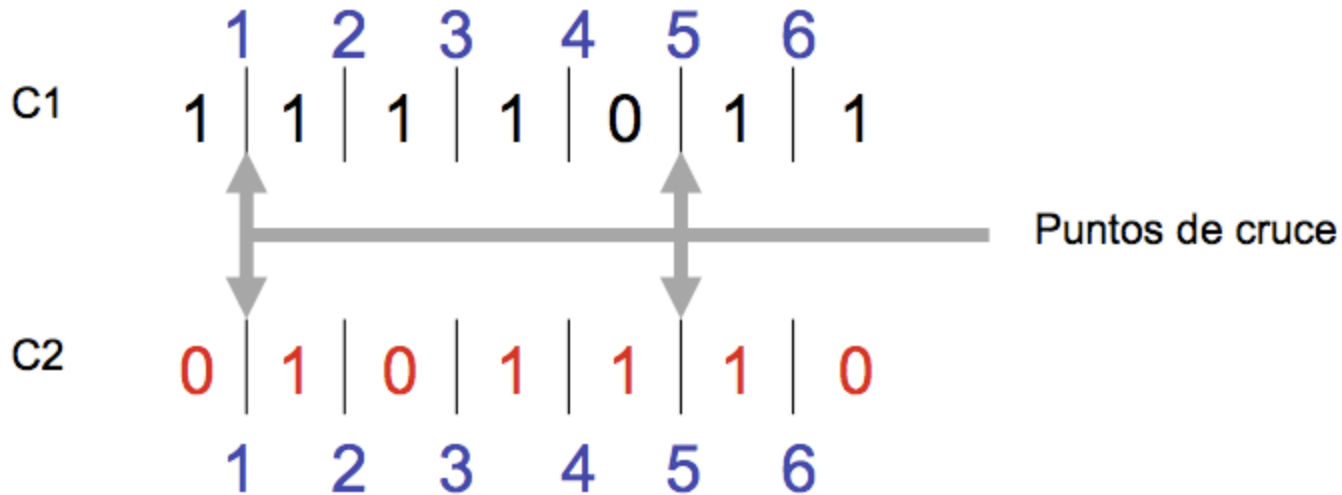
Padres

Descendientes

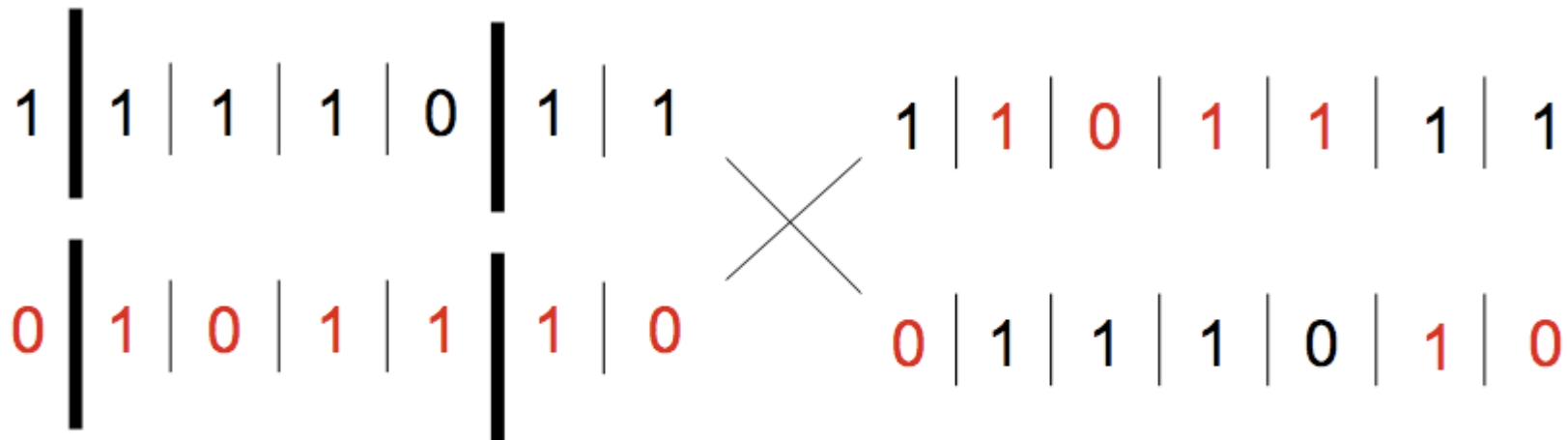
Cruce Doble

- Los dos cromosomas se cortan por 2 puntos, y el material genético situado entre ellos se intercambia.
- Sean $C1 = 1111011$ y $C2 = 0101110$
- $|C1| = 7$ $\text{random}_1 = 0.1535$ $\text{random}_2 = 0.8112$
- Nota $\text{random}_1 \neq \text{random}_2$
- Punto de cruce₁ = roundup $\{ 0.1535 \times (7-1) \}$
- Punto de cruce₁ = roundup $\{ 0.921 \} = 1$
- Punto de cruce₂ = roundup $\{ 0.8112 \times (7-1) \}$
- Punto de cruce₂ = roundup $\{ 4.86 \} = 5$

Cruce Doble



Cruce Doble



Padres

Descendientes

Cruce uniforme

- Se genera un patrón aleatorio de 1s y 0s del mismo tamaño de los cromosomas.
- Se intercambian los bits de los dos cromosomas que coincidan donde hay un 1 en el patrón.

Cruce Uniforme

A

1 | 1 | 1 | 1 | 0 | 1 | 1

0 | 1 | 0 | 1 | 1 | 1 | 0

C

0 | 1 | 0 | 1 | 1 | 1 | 0

B

Padres A y B

Patron C

1 | 1 | 1 | 1 | 1 | 1 | 1

0 | 1 | 0 | 1 | 0 | 1 | 0

Descendientes

Mutación

- En la Evolución, una mutación es un suceso bastante poco común (sucede aproximadamente una de cada mil replicaciones), como ya se ha visto anteriormente.
- En la mayoría de los casos las mutaciones son letales, pero en promedio, contribuyen a la diversidad genética de la especie.
- En un algoritmo genético tendrán el mismo papel, y la misma frecuencia (es decir, muy baja).

Mutación

- Una vez establecida la frecuencia de mutación, por ejemplo, uno por mil, se examina cada bit de cada cadena cuando se vaya a crear la nueva criatura a partir de sus padres.
- Se genera un número aleatorio para cada bit del cromosoma.
- Si el numero aleatorio es > 0.5 entonces de 0 a 1 o de 1 a 0.
- Si no, se dejará como está. Dependiendo del número de individuos que haya y del número de bits por individuo, puede resultar que las mutaciones sean extremadamente raras en una sola generación.

Generaciones siguientes

- Después que las mutaciones son llevadas a cabo, se calculan los costos asociados con los descendientes y con los cromosomas mutados.

Descendientes sin mutar y mutados	costo
xxxxxxxxxxxxx	aaaa
xxxxxxxxxyxxx	aaab
yyyyxxxxxxxxx	baba

Convergencia

- El numero de generaciones que evoluciona, depende del numero de iteraciones que sean alcanzadas o excedidas.
- Despues de que todos los cromosomas en y sus costos asociados en la nueva generacion se calcularon y no hubo mutaciones entonces el algoritmo debe detenerse

Convergencia

- La mayoría de los algoritmos genéticos almacenan información estadística de la población.
- Esta información normalmente es la media, desviación estándar y el costo mínimo.
- Cualquier combinación de estos parámetros puede servir como medida de convergencia.

Caso Práctico

$$f(x) = -x * \text{sen}(\sqrt{|x|})$$

Nota: raíz cuadrada del valor absoluto de x

- Calcular el valor mínimo.
- $-500 \leq x \leq 500$
- $N_{\text{IPOP}} = 20$.
- Selección de Pareja = ~~Ruleta (random pairing)~~
- Cruce sencillo. Top to Bottom
- Sin Mutación.
- $\neq 2$ iteración.