



# ***Algoritmos Bioinspirados***

## ***Unidad II : Cómputo Evolutivo - Algoritmos Genéticos***

***Programa: Ingeniería en Inteligencia Artificial***

## Algoritmos Genéticos – Operadores de reproducción

**Cruza:** es un operador esencial para la evolución natural. Toma dos genotipos y produce un nuevo genotipo mezclando los genes que se encuentra en los originales.

La descendencia, simplemente combinando todas las buenas características de sus padres, puede superar a sus antepasados.

## Probabilidad de Cruza

La probabilidad de cruza ( $P_c$ ) es un parámetro básico de los AG que indica si todos los individuos seleccionados como padres se reproducen.

- ❖  $P_c = 1$  , todos los individuos seleccionados se cruzan, no hay supervivientes.
- ❖  $P_c = 0$ , ningún individuo se cruza. La nueva generación está hecha de copias exactas de los cromosomas de la población anterior
- ❖ La cruza se realiza con la esperanza de que los nuevos cromosomas sean mejores. Sin embargo, es bueno dejar que una parte de la población antigua sobreviva a la próxima generación

- ❖ Estudios empíricos demuestran que una probabilidad de cruce entre 0,65 y 0,85 logra buenos resultados.
- ❖ Esto significa que un cromosoma tiene una probabilidad entre 0.15 y 0.35 de sobrevivir a la siguiente generación.

$$P_c = 0.85$$

Padre 1 

1	0	1	0	0	1	1	1	0	0
---	---	---	---	---	---	---	---	---	---

Padre 2 

1	1	0	0	1	0	0	1	0	1
---	---	---	---	---	---	---	---	---	---

- ❖ Generar un número aleatorio entre 0 y 1.  $r_c = \text{rand.uniform}(0,1)$
- ❖ Si el número generado es menor que  $P_c$  se realiza la cruce.
- ❖ En caso contrario, los individuos seleccionados como padres pasan sin modificación a la siguiente generación (supervivientes).

$P_c = 0.85$  Generar número aleatorio: 0.75, es menor que  $P_c$

Se reproducen.

Padre 1 

1	0	1	0	0	1	1	1	0	0
---	---	---	---	---	---	---	---	---	---

Hijo 1 

1	1	0	0	0	1	1	1	0	1
---	---	---	---	---	---	---	---	---	---

Padre 2 

1	1	0	0	1	0	0	1	0	1
---	---	---	---	---	---	---	---	---	---

Hijo 2 

1	0	1	0	0	0	0	1	0	0
---	---	---	---	---	---	---	---	---	---

$P_c = 0.85$  Generar número aleatorio: 0.92, es mayor que  $P_c$

No se reproducen.

Padre 1 

1	0	1	0	0	1	1	1	0	0
---	---	---	---	---	---	---	---	---	---

Padre 1 

1	0	1	0	0	1	1	1	0	0
---	---	---	---	---	---	---	---	---	---

Padre 2 

1	1	0	0	1	0	0	1	0	1
---	---	---	---	---	---	---	---	---	---

Padre 2 

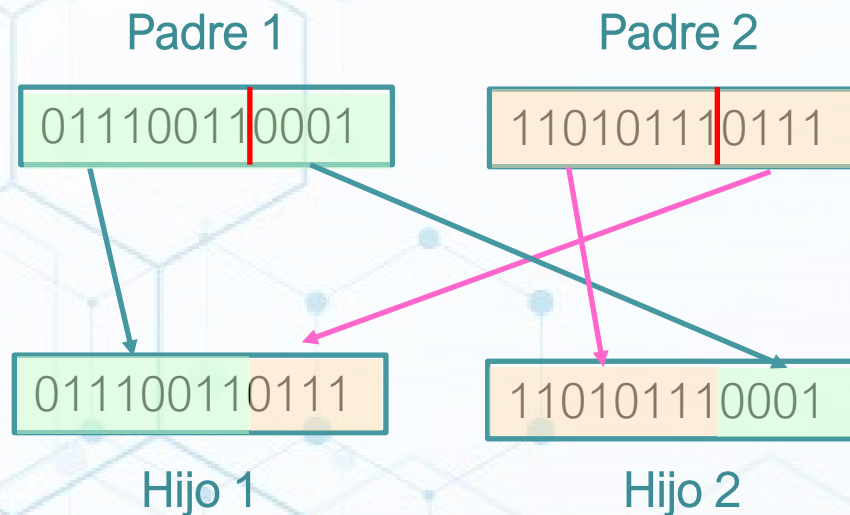
1	1	0	0	1	0	0	1	0	1
---	---	---	---	---	---	---	---	---	---

Pasa a la siguiente generación

### Cruza de 1 punto:

- Selección de 2 individuos para ser padres.
- Se elige un punto de cruce al azar (posición de un gen dentro del individuo).
- Finalmente, los genes de los 2 individuos se intercambian a partir del punto de cruce

Sirve para codificación con valores binarios, enteros (no ordinales) y reales



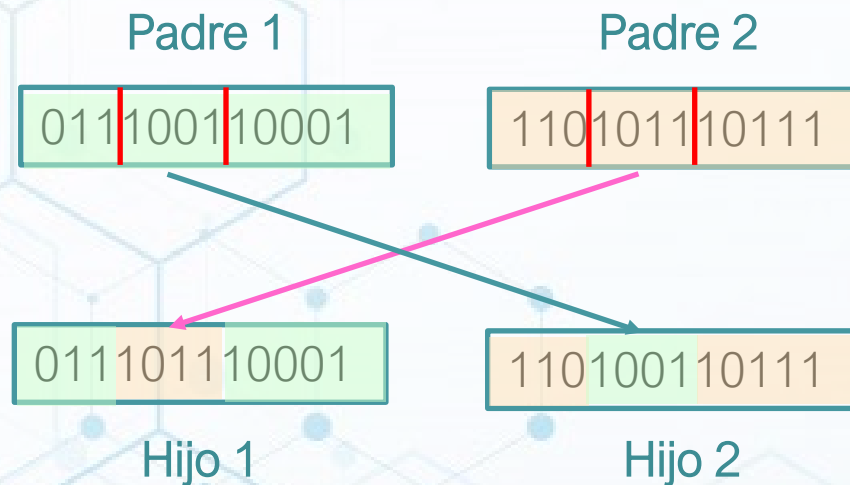
Punto de  
cruce = 8

Cruzas de 1  
punto

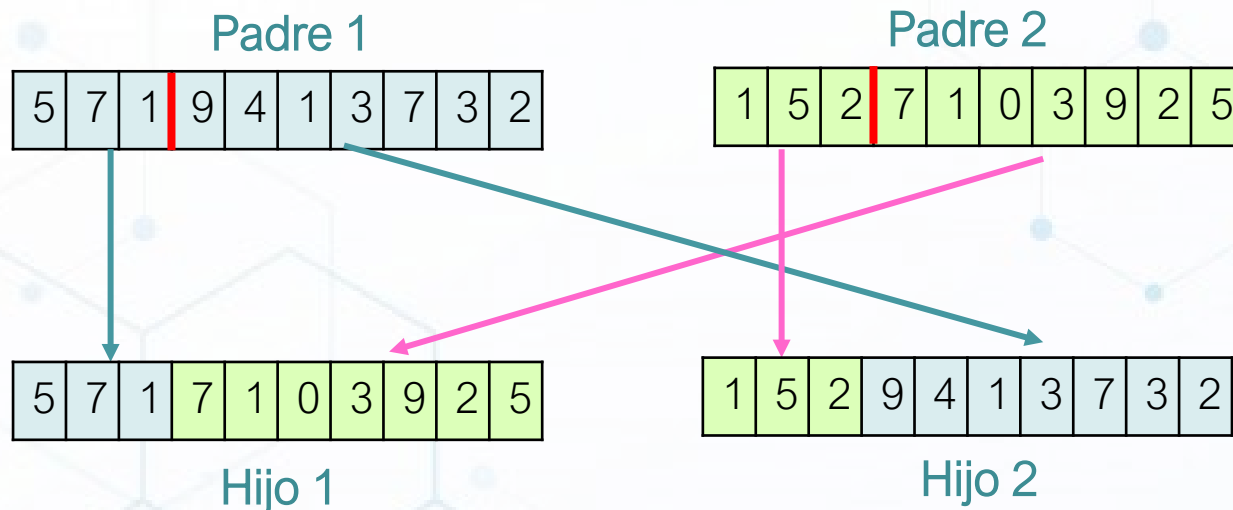
### Cruza de 2 puntos

- Selección de 2 individuos para ser padres.
- Se elige dos puntos de cruce al azar (posición de un gen dentro del individuo).
- Finalmente, los genes de los 2 individuos se intercambian a partir de los puntos de cruce.

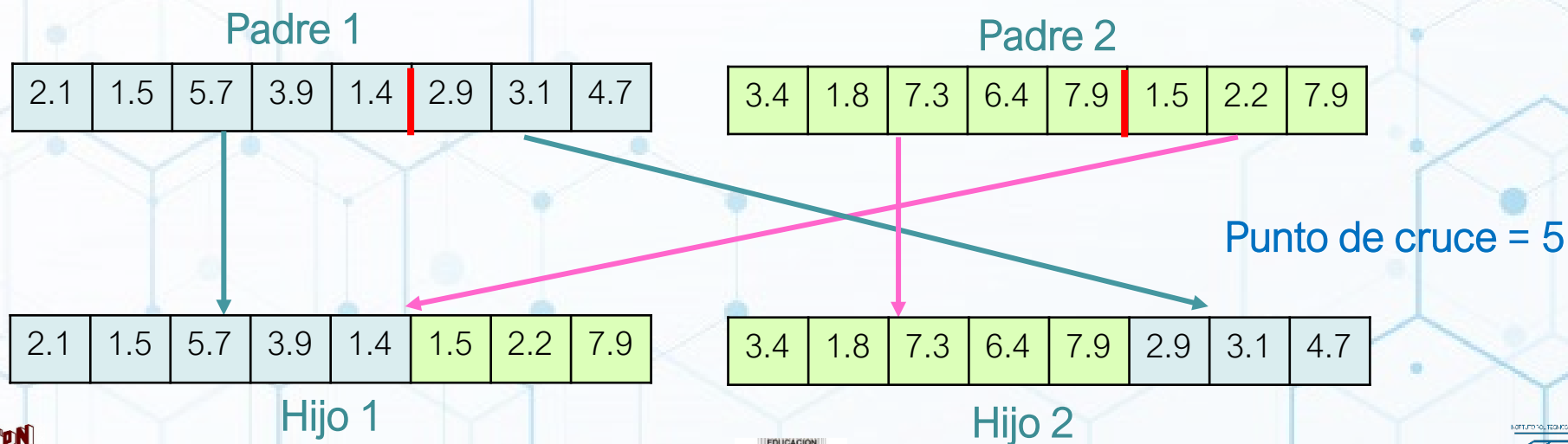
Sirve para codificación con valores binarios, enteros (no ordinales) y reales



## Ejemplo de cruce de 1 punto con valores enteros



## Ejemplo de cruce de 1 punto con valores reales





### Cruza uniforme:

- ❖ Tratar cada gen de forma independiente y elegir aleatoriamente de qué padre se debe heredar cada gen.
- ❖ Esto se implementa generando una variable aleatoria del tamaño de cromosoma a partir de una distribución uniforme  $[0,1]$ . Si el valor es menor que un umbral ( $u=0.5$ ), se heredan los genes del 1er padre, de otra forma se heredan del segundo. Los genes se invierten en el 2do hijo.

Sirve para codificación con valores binarios, enteros (no ordinales) y reales

Padre 1

1	0	1	0	0	1	1	1	0	0
---	---	---	---	---	---	---	---	---	---

Padre 2

1	1	0	0	1	0	0	1	0	1
---	---	---	---	---	---	---	---	---	---

### Variable aleatoria

1	2	3	4	5	6	7	8	9	10
0.3	0.6	0.1	0.4	0.8	0.7	0.3	0.5	0.2	0.6

Hijo 1

1	1	1	0	1	0	1	1	0	1
---	---	---	---	---	---	---	---	---	---

Hijo 2

1	0	0	0	0	1	0	1	0	0
---	---	---	---	---	---	---	---	---	---

Recombinación uniforme:

- ❖ También se puede implementar generando una variable aleatoria del tamaño de cromosoma con valores  $[0,1]$

Padre 1

1	0	1	0	0	1	1	1	0	0
---	---	---	---	---	---	---	---	---	---

Padre 2

1	1	0	0	1	0	0	1	0	1
---	---	---	---	---	---	---	---	---	---

Variable aleatoria

1	2	3	4	5	6	7	8	9	10
1	0	1	1	0	0	1	0	1	0

Hijo 1

1	1	1	0	1	0	1	1	0	1
---	---	---	---	---	---	---	---	---	---

Hijo 2

1	0	0	0	0	1	0	1	0	0
---	---	---	---	---	---	---	---	---	---

## Cruza parcialmente mapeada:

Usar cuando la codificación del cromosoma es entera, pero además representa un orden (ordinales).

1. Elija 2 puntos de cruce al azar. Puntos de cruce = 3 y 7

Padre 1

1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---

Padre 2

9	3	7	8	2	6	5	1	4
---	---	---	---	---	---	---	---	---

2. Copiar el segmento entre ellos desde el primer padre (P1) al primer hijo.

Hijo 1

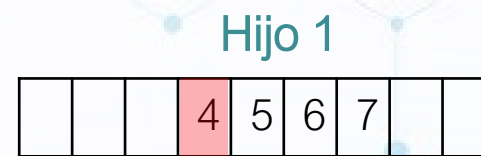
			4	5	6	7		
--	--	--	---	---	---	---	--	--

## Cruza parcialmente mapeada (cont.):

3. Por cada valor en el segmento del padre 2 que no se encuentra en el hijo 1. Iniciando en el primer punto de cruce, por cada valor  $A$  en el padre 2 que no está en el hijo 1, identificar el valor  $B$  que se encuentra en la misma posición en el hijo 1.

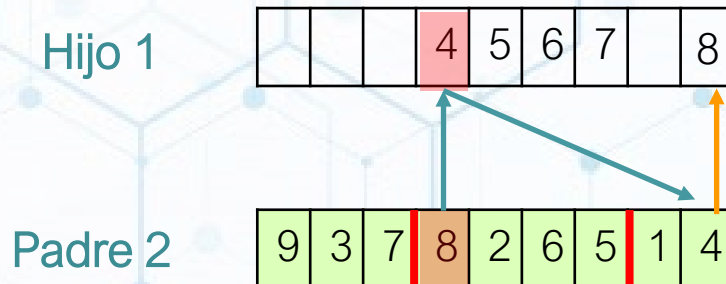


*valor A en la pos. i*



*valor B en la pos. i*

4. Colocar  $A$ , en el hijo 1, en la posición que ocupa  $B$  en el padre 2.



El valor 8 está en la posición 4 del padre 2.

El valor 4 ocupa la misma posición en el hijo 1.

El valor 4 se encuentra en la posición 9 del padre 2.

El valor 8 se coloca en la posición 9 del hijo 1.

## Cruza parcialmente mapeada (cont.):

5. Si la posición del valor  $B$  en el padre 2, ya está ocupado en el hijo 1, por un tercer valor  $C$ , colocar  $A$  en la posición que ocupa  $C$  en el padre 2.

El valor 2 ( $A$ ) está en la posición 5 del padre 2.

El valor 5 ( $B$ ) ocupa la misma posición en el hijo 1.

El valor 5 ocupa la posición 7 del padre 2.

Esta posición ya está ocupada en el hijo 1 por el valor 7 ( $C$ ), seguir el proceso usando ahora la posición del valor 7 en el padre 2.

El valor 7 se encuentra en la posición 3 en el padre 2.

Usar la posición del 7 en el padre 2, para colocar el 2 en el hijo 1, es decir poner el 2 en la posición 3.

Hijo 1

			4	5	6	7		8
--	--	--	---	---	---	---	--	---

Padre 2

9	3	7	8	2	6	5	1	4
---	---	---	---	---	---	---	---	---

Hijo 1

		2	4	5	6	7		8
--	--	---	---	---	---	---	--	---

## Cruza parcialmente mapeada (cont.):

6. Si el valor en el padre 2 ya se encuentra en el hijo no hacer nada.

Hijo 1

		2	4	5	6	7		8
--	--	---	---	---	---	---	--	---

Los valores 5 y 6 ya están en el hijo, por lo que ya no se hace nada.

Padre 2

9	3	7	8	2	6	5	1	4
---	---	---	---	---	---	---	---	---

7. Las posiciones faltantes del hijo se rellenan con los valores del padre 2, en la posición actual.

Hijo 1

9	3	2	4	5	6	7	1	8
---	---	---	---	---	---	---	---	---

Padre 2

9	3	7	8	2	6	5	1	4
---	---	---	---	---	---	---	---	---



Cruza aritmética simple:

- Se elige un punto de cruce al azar  $k$ .
- El hijo 1 toma los primeros  $k$  posiciones del padre 1.
- El resto de posiciones es el promedio aritmético de los 2 padres.
- Para el otro hijo el proceso es análogo invirtiendo el orden de los padres.

Sirve para codificación con valores enteros (no ordinales) y reales

Punto de cruce = 5

Padre 1

2.1	1.5	5.7	3.9	1.4	2.9	3.1	4.7
-----	-----	-----	-----	-----	-----	-----	-----

Padre 2

3.4	1.8	7.3	6.4	7.9	1.5	2.2	7.9
-----	-----	-----	-----	-----	-----	-----	-----

2.1	1.5	5.7	3.9	1.4	2.2	2.6	6.3
-----	-----	-----	-----	-----	-----	-----	-----

3.4	1.8	7.3	6.4	7.9	2.2	2.6	6.3
-----	-----	-----	-----	-----	-----	-----	-----

Hijo 1

Hijo 2



**Recombinación aritmética completa:** se calcula la suma ponderada de los alelos en cada gen de los 2 padres. Donde  $x_i$  representa los alelos del padre 1,  $y_i$  los alelos del padre 2 y  $z_i$  los alelos en el hijo

Hijo 1

$$z_i = \alpha \cdot x_i + (1 - \alpha) \cdot y_i$$

Hijo 2

$$z_i = \alpha \cdot y_i + (1 - \alpha) \cdot x_i$$

Sirve para codificación con valores enteros (no ordinales) y reales

Ejemplo con  
 $\alpha = 0.7$

Padre 1

2.1	1.5	5.7	3.9	1.4	2.9
-----	-----	-----	-----	-----	-----

$$x_1 = 2.1 \quad y_1 = 3.4$$

$$z_1 = (0.7 \times 2.1) + (0.3 \times 3.4)$$

Hijo 1

2.5					
-----	--	--	--	--	--

Primer gen

Todos los genes

2.5	1.6	6.2	4.7	3.3	2.5
-----	-----	-----	-----	-----	-----

Padre 2

3.4	1.8	7.3	6.4	7.9	1.5
-----	-----	-----	-----	-----	-----

$$z_1 = (0.7 \times 3.4) + (0.3 \times 2.1)$$

Hijo 2

3.0					
-----	--	--	--	--	--

3.0	1.7	6.8	5.6	6.0	1.9
-----	-----	-----	-----	-----	-----



## Algoritmos Genéticos – Operadores de variación

**Mutación:** consiste en cambiar el valor de los genes. En la evolución natural generalmente produce individuos no aptos. Sin embargo, es la optimización, algunos cambios aleatorios pueden ser una buena manera de explorar el espacio de búsqueda rápidamente.

- ❖ Evita quedar atrapados en mínimos o máximos locales.
- ❖ Si la cruza permite explotar la solución actual para encontrar otras mejores, la mutación ayudará a explorar todo el espacio de búsqueda.
- ❖ La mutación puede verse como un operador de fondo para mantener la diversidad genética en la población al introducir nuevo material genético.



## Probabilidad de Mutación

La probabilidad de mutación ( $P_m$ ) es otro parámetro básico de los AG que indica que parte del cromosoma debe ser mutado

- ❖  $P_m = 1$ , todos los genes del individuo deben ser mutados.
- ❖  $P_m = 0$ , ningún gen del individuo debe ser mutado.
- ❖ La mutación no debería ocurrir muy a menudo, porque entonces el AG se transforma en una búsqueda aleatoria, por lo que se eligen valores bajos de  $P_m$ .



## Mutación simple

Usar probabilidad de mutación ( $P_m$ ) es para decidir si el cromosoma será mutado o no.

- ❖ Generar un número aleatorio entre 0 y 1 por cada gen del individuo.
- ❖ Si el número correspondiente a un gen es menor que  $P_m$  se muta el gen.
- ❖ En caso contrario, el gen es dejado sin modificación.

$$P_m = 0.1 \quad r_m = \text{rand.uniform}(0,1)$$

Si  $r_m \leq P_m$  realizar algún cambio pequeño en el individuo, sino dejar sin mutar

Ejemplo: escoger de forma aleatoria un gen del individuo y cambiarlo.

$$\text{pos} = \text{rand.int}(1, \text{tamaño del individuo}) \quad \text{pos} = 5$$

1	1	0	0	1	0	0	1	0	1
1	1	0	0	0	0	0	1	0	1

## Mutación uniforme

- ❖ Generar un número aleatorio entre 0 y 1 por cada gen del individuo.
- ❖ Si el número correspondiente a un gen es menor que  $P_m$  se muta el gen.
- ❖ En caso contrario, el gen es dejado sin modificación.

$$P_m = 0.1$$

Hijo1

1	0	1	0	0	1	1	1	0	0
---	---	---	---	---	---	---	---	---	---

0.25	0.32	0.17	0.02	0.65	0.92	0.08	0.16	0.29	0.78
------	------	------	------	------	------	------	------	------	------

1	0	1	1	0	1	0	1	0	0
---	---	---	---	---	---	---	---	---	---

Hijo2

1	1	0	0	1	0	0	1	0	1
---	---	---	---	---	---	---	---	---	---

0.65	0.24	0.06	0.18	0.68	0.71	0.63	0.27	0.17	0.19
------	------	------	------	------	------	------	------	------	------

1	1	1	0	1	0	0	1	0	1
---	---	---	---	---	---	---	---	---	---

## Mutación uniforme

### Ejemplo con valores reales

Individuo sin  
mutar

1.7	2.3	1.2	8.3	3	1.8
-----	-----	-----	-----	---	-----

Cromosoma  
de mutación

0.7	0.02	0.4	0.9	0.08	0.3
-----	------	-----	-----	------	-----

Individuo  
mutado

1.7	3.7	1.2	8.3	2.5	1.2
-----	-----	-----	-----	-----	-----

Valores permitidos:

$$x_1, x'_1 \in [1.1, 2.5]$$

$$x_2, x'_2 \in [0, 5]$$

$$x_3, x'_3 \in [1, 4.7]$$

$$x_4, x'_4 \in [2.5, 10]$$

$$x_5, x'_5 \in [1.1, 4]$$

$$x_6, x'_6 \in [1.1, 2.5]$$

### Mutación por inversión (*flip*) de bits:

- Generar un cromosoma de mutación con valores aleatorios de 0 y 1.
- Las posiciones que correspondan con valores menores a la probabilidad de mutación son invertidas.

- Sirve para codificación con binaria.
- Puede ser usada con mutación simple y con mutación uniforme

### Ejemplo con mutación uniforme

Individuo sin  
mutar

1	0	1	0	0	1	1	1	0	0
---	---	---	---	---	---	---	---	---	---

Cromosoma  
de mutación

0.7	0.02	0.4	0.9	0.08	0.3	0.7	0.04	0.8	0.06
-----	------	-----	-----	------	-----	-----	------	-----	------

Individuo  
mutado

1	1	1	0	1	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---

### Mutación por intercambio:

- Seleccionar 2 posiciones al azar
- Intercambiar los bits correspondientes a esas posiciones.

- Sirve para codificación con binaria, entera y real.
- Puede ser usada con mutación simple

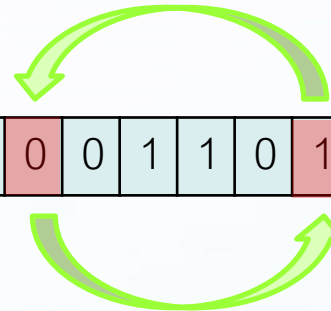
### Ejemplo con valores binarios

Individuo sin  
mutar

1	0	1	0	0	1	1	0	1	0
---	---	---	---	---	---	---	---	---	---

Individuo  
mutado

1	0	1	1	0	1	1	0	0	0
---	---	---	---	---	---	---	---	---	---





### Mutación aleatoria:

- i. Una vez seleccionados el gen(es) a mutar usando el método de mutación simple o uniforme, cambiar de forma aleatoria el valor del gen usando el rango de valores permitidos para ese gen (variable).
- Sirve para codificación con valores enteros o reales.
  - Puede ser usada con mutación simple y con mutación uniforme

Valores permitidos: {0 – 20}

Individuo sin  
mutar

3	7	12	8	3	9	1	4	18	0
---	---	----	---	---	---	---	---	----	---

Cromosoma  
de mutación

0.7	0.02	0.4	0.9	0.08	0.3	0.7	0.04	0.8	0.06
-----	------	-----	-----	------	-----	-----	------	-----	------

Individuo  
mutado

3	5	12	8	15	9	1	3	18	2
---	---	----	---	----	---	---	---	----	---



### Mutación por cambio de orden:

- Seleccionar una sección al azar del cromosoma.
- Cambiar el orden de la secuencia en la sección seleccionada.

- Sirve para codificación con valores binarios, enteros o reales.
- Puede ser usada con mutación simple

### Ejemplo invirtiendo el orden completamente

Individuo sin  
mutar

3	7	10	8	1	5	2	4	9	6
---	---	----	---	---	---	---	---	---	---

Individuo  
mutado

3	7	10	2	5	1	8	4	9	6
---	---	----	---	---	---	---	---	---	---

### Mutación por mezclado (scramble):

- Cambiar el orden de todos los genes del cromosoma completo
- No es lo mejor, en su lugar mejor usar el método en la diapositiva anterior (parcial).

Individuo sin  
mutar

3	7	10	8	1	5	2	4	9	6
---	---	----	---	---	---	---	---	---	---

Completo

Individuo  
mutado

2	6	9	3	4	10	7	1	8	5
---	---	---	---	---	----	---	---	---	---

Parcial

Individuo  
mutado

3	10	1	8	7	10	7	1	8	5
---	----	---	---	---	----	---	---	---	---

### Mutación aritmética no uniforme:

- i. Una vez seleccionados el gen(es) a mutar usando el método de mutación simple o uniforme, cambiar el valor del gen sumándole una cantidad aleatoria extraída de una distribución gaussiana con media cero y desviación estándar especificada por el usuario (acotar el valor si hay restricciones).
- Sirve para codificación con reales.
  - Puede ser usada con mutación simple y con mutación uniforme

Individuo sin  
mutar

1.7	2.3	1.2	8.3	3	1.8
-----	-----	-----	-----	---	-----

Cromosoma  
de mutación

0.7	0.02	0.4	0.9	0.08	0.3
-----	------	-----	-----	------	-----

Individuo  
mutado

1.7	2.8	1.2	8.3	3.1	2.3
-----	-----	-----	-----	-----	-----

```
import random
```

```
mu = 0          # media  
sigma = 0.8     # desv. estandar
```

```
print(random.gauss(mu, sigma))
```

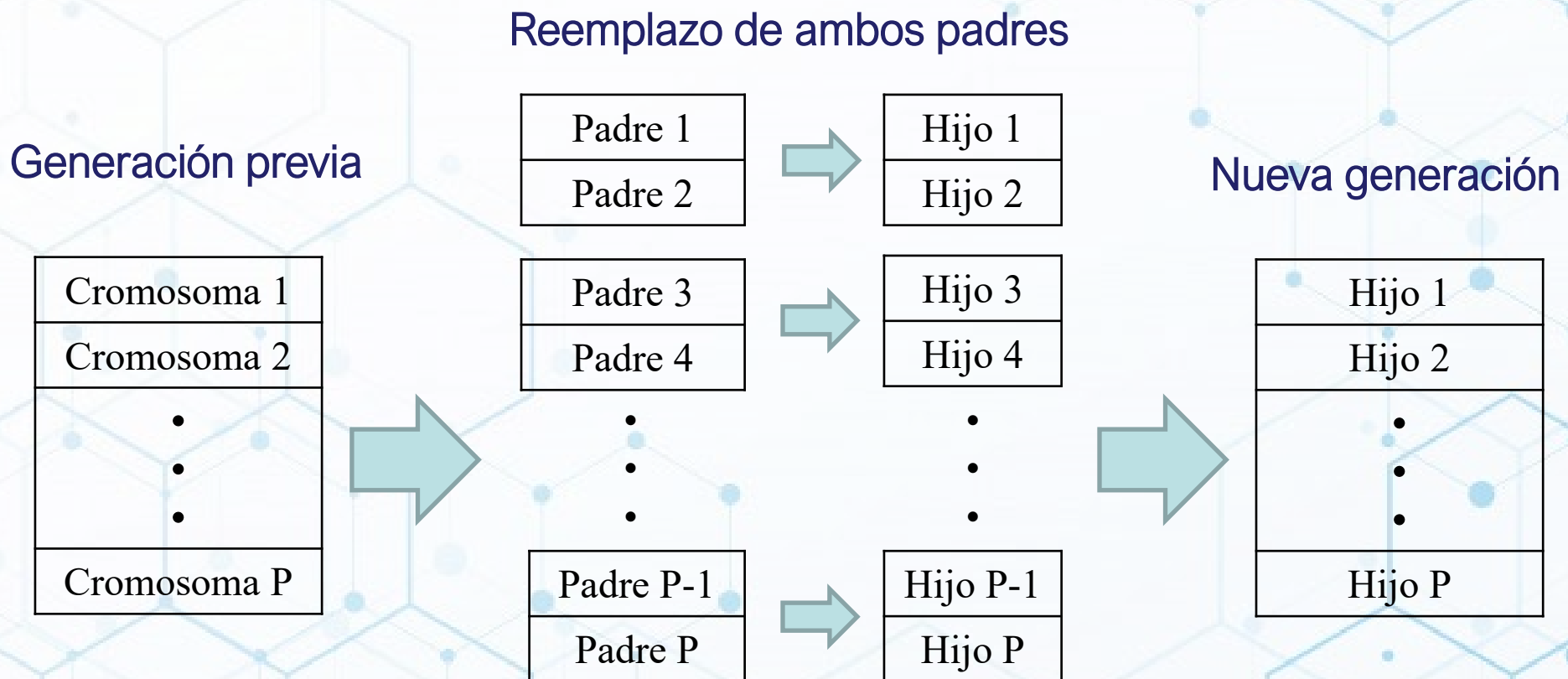
## Reemplazo (selección de supervivientes)

El proceso de reducir un conjunto de  $P$  padres y  $L$  hijos a un conjunto de  $P$  individuos que conforman la siguiente generación.

- ❖ Es el último paso del ciclo de reproducción.
- ❖ Una vez que se producen los descendientes, un método debe determinar cuál de los miembros actuales de la población, si alguno, debe ser reemplazado por las nuevas soluciones.

## Actualización Generacional

Este esquema consiste en elegir  $P$  individuos, que formaran la nueva población, a partir de  $P$  padres y  $L$  hijos.



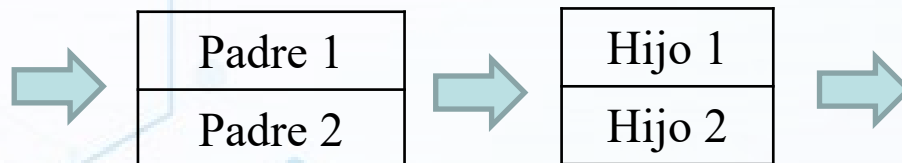
## Actualización de Estado Estacionario

Los nuevos individuos se insertan en la población tan pronto como son creados.

Generación previa

Cromosoma 1
Padre 1
Cromosoma 2
• • •
Padre 2
• • •
Cromosoma P

Reemplazo de ambos padres



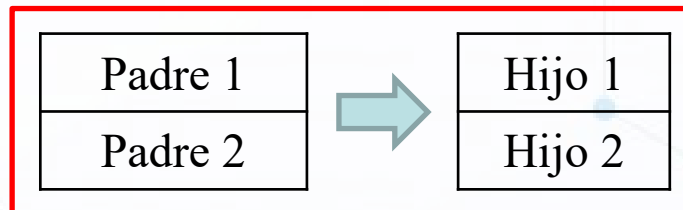
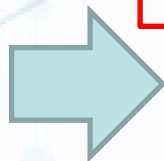
Nueva generación

Cromosoma 1
Hijo 1
Cromosoma 2
• • •
Hijo 2
• • •
Cromosoma P

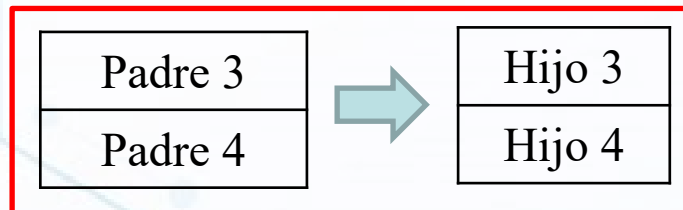
## Actualización Generacional Reemplazo padre más débil

Generación  
previa

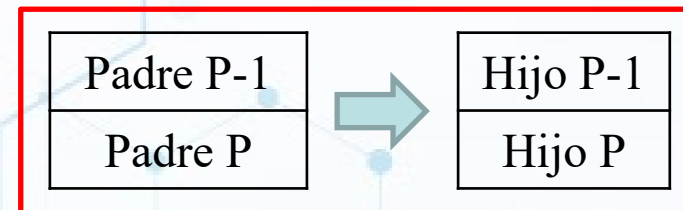
Cromosoma 1
Cromosoma 2
•
•
•
Cromosoma P



Seleccionar  
los 2 mejores



Seleccionar  
los 2 mejores



Seleccionar  
los 2 mejores

Nueva  
generación

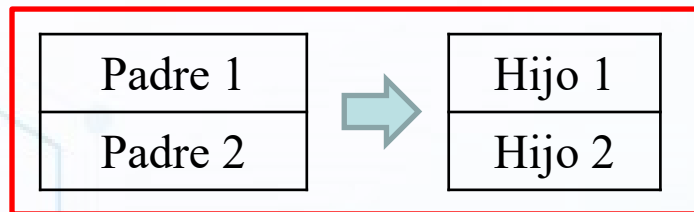


## Actualización de Estado Estacionario Reemplazo padre más débil

Generación  
previa

Cromosoma 1
Padre 1
Cromosoma 2
•
•
•
Padre 2
•
•
•
Cromosoma P

Seleccionar  
los 2 mejores



Nueva  
Generación

Cromosoma 1
Mejor individuo 1
Cromosoma 2
•
•
•
Mejor individuo 2
•
•
•
Cromosoma P



## Criterios de finalización

**Generaciones máximas:** el algoritmo genético se detiene cuando ha evolucionado el número especificado de generaciones.

**Tiempo transcurrido:** el proceso genético finalizará cuando haya transcurrido un tiempo específico.

**Sin cambios en la aptitud:** el proceso genético terminará si no hay cambios en el mejor valor de aptitud de la población para un número específico de generaciones

## Bibliografía

1. A. E. Eiben and James E. Smith. 2015. Introduction to Evolutionary Computing (2nd. ed.). Springer Publishing Company, Incorporated.
2. S. N. Sivanandam and S. N. Deepa. 2010. Introduction to Genetic Algorithms (1st. ed.). Springer Publishing Company, Incorporated.





¡ Gracias !

Thanks !

Obrigado

Xie xie ni

Domo arigatou

Спасибо

Merci

Grazie

Alfa Beta