

Investigación de Operaciones Algoritmos Genéticos

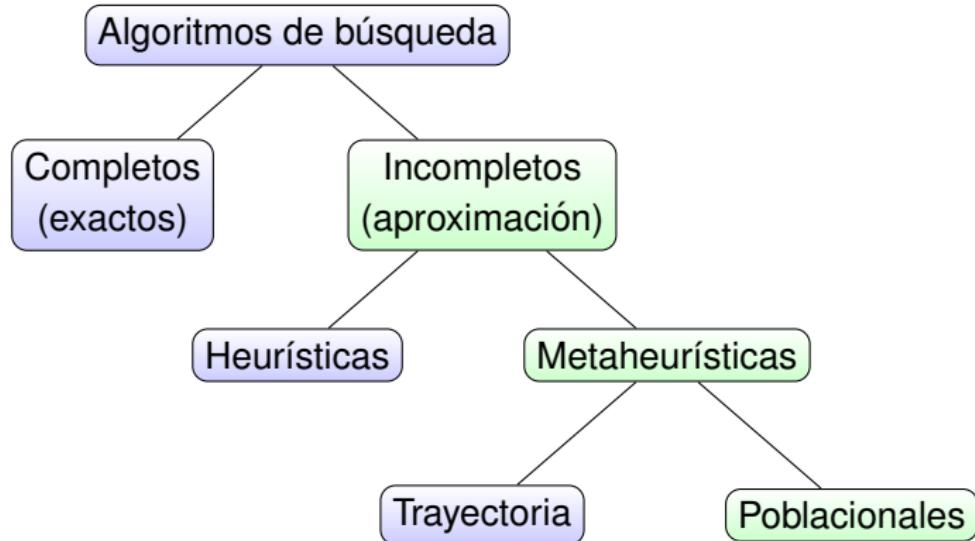
Leslie Pérez Cáceres
leslie.perez@pucv.cl

Escuela de Informática
Pontificia Universidad Católica de Valparaíso

2021



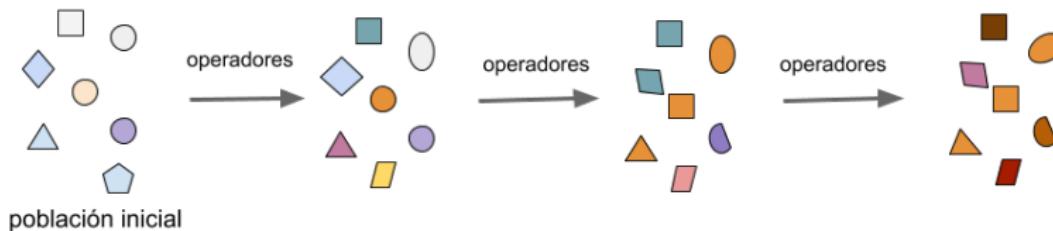
PONTIFICIA
UNIVERSIDAD
CATÓLICA DE
VALPARAÍSO



Idea: no limitar la búsqueda a una trayectoria

- + exploración → revisar varias áreas del espacio a la vez
- + explotación → más información sobre el espacio de búsqueda

Mantener un grupo de soluciones (población) en cada iteración





- Algoritmos inspirados en el proceso de evolución Darwiniana
- Definen un set de soluciones (**población**)
- En cada iteración (**generación**) las soluciones (**individuos**) son:
 - seleccionados para reproducirse
 - recombinados (combinación de material genético)
 - mutados
 - seleccionados para integrar la población
- La nueva población es utilizada para la siguiente iteración

Algoritmos Genéticos: pseudocódigo

```
G ← poblacion_inicial();
s+ ← mejor_solucion (G);
while !criterio_de_termino() do
    G* ← ∅;
    while |G*| < n_individuos do
        p1, p2 ← seleccionar_individuos (G, 2);
        s ← cruzar_individuos (p1, p2);
        s ← mutar(s);
        G* ← G* + s;
    end
    G ← seleccionar_poblacion(G, G*);
    s+ ← mejor_solucion (s+, G);
end
return s+ ;
```

Algoritmos Genéticos: pseudocódigo

```
G ← poblacion_inicial();
s+ ← mejor_solucion (G);
while !criterio_de_termino() do
    G* ← ∅;
    while |G*| < n_individuos do
        p1, p2 ← seleccionar_individuos (G, 2);
        s ← cruzar_individuos (p1, p2);
        s ← mutar (s);
        G* ← G* + s;
    end
    G ← seleccionar_poblacion (G, G*);
    s+ ← mejor_solucion (s+, G);
end
return s+;
```

Operador que selecciona dos o mas soluciones para que sean “padres” de una nueva solución:

- La selección se puede hacer de diversas maneras
 - Selección **elitista**: selección de los mejores individuos
 - Selección de **la ruleta**: asignar probabilidades de selección a cada individuo de acuerdo a su calidad
- Se seleccionan preferentemente los mejores individuos para que permanezcan sus buenos componentes en la población

Algoritmos Genéticos: selección de individuos

Selección **elitista**, asuma la siguiente población:

s_1	1 0 0 0 0 0 0 0 0	79
s_2	0 0 0 0 0 1 0 0 0	33
s_3	0 0 1 0 0 0 0 0 0	47
s_4	0 0 0 0 0 1 0 0 0	85
s_5	0 0 0 0 0 0 0 0 1	59

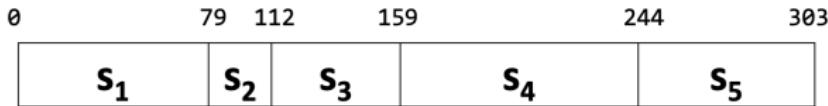
Se seleccionan las mejores soluciones:

s_1	1 0 0 0 0 0 0 0 0	79
s_4	0 0 0 0 0 1 0 0 0	85

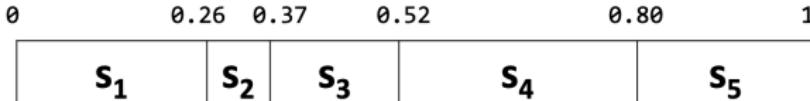
Algoritmos Genéticos: selección de individuos

Selección de la ruleta, asuma la siguiente población:

s_1	1 0 0 0 0 0 0 0 0	79
s_2	0 0 0 0 0 1 0 0 0	33
s_3	0 0 1 0 0 0 0 0 0	47
s_4	0 0 0 0 0 1 0 0 0	85
s_5	0 0 0 0 0 0 0 0 1	59



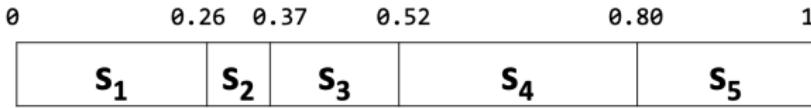
normalizamos...



Algoritmos Genéticos: selección de individuos

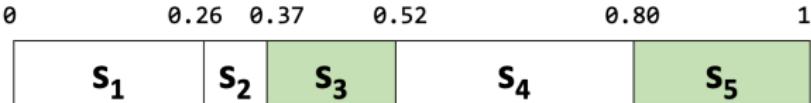
Selección de la ruleta, asuma la siguiente población:

s_1	1 0 0 0 0 0 0 0 0	79	(+)	303
s_2	0 0 0 0 0 1 0 0 0	33		
s_3	0 0 1 0 0 0 0 0 0	47		
s_4	0 0 0 0 0 1 0 0 0	85		
s_5	0 0 0 0 0 0 0 0 1	59		



`random() = 0.88 → s_5`

`random() = 0.51 → s_3`



Selección de la ruleta, individuos seleccionados:

s_1	1 0 0 0 0 0 0 0 0	79
s_2	0 0 0 0 0 1 0 0 0	33
s_3	0 0 1 0 0 0 0 0 0	47
s_4	0 0 0 0 0 1 0 0 0	85
s_5	0 0 0 0 0 0 0 0 1	59

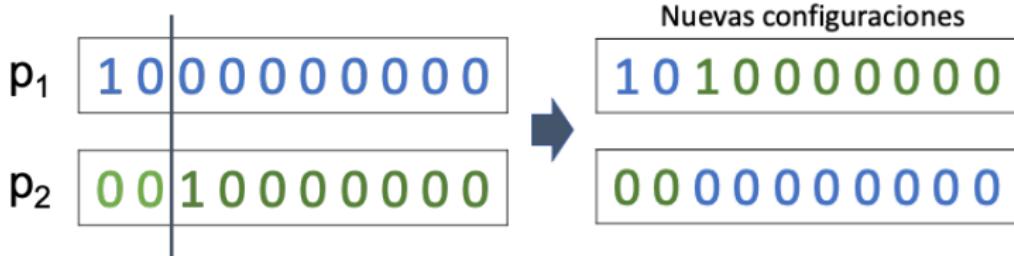
Algoritmos Genéticos: pseudocódigo

```
G ← poblacion_inicial();
s+ ← mejor_solucion (G);
while !criterio_de_termino() do
    G* ← ∅;
    while |G*| < n_individuos do
        p1, p2 ← seleccionar_individuos (G, 2);
        s ← cruzar_individuos (p1, p2);
        s ← mutar (s);
        G* ← G* + s;
    end
    G ← seleccionar_poblacion (G, G*);
    s+ ← mejor_solucion (s+, G);
end
return s+;
```

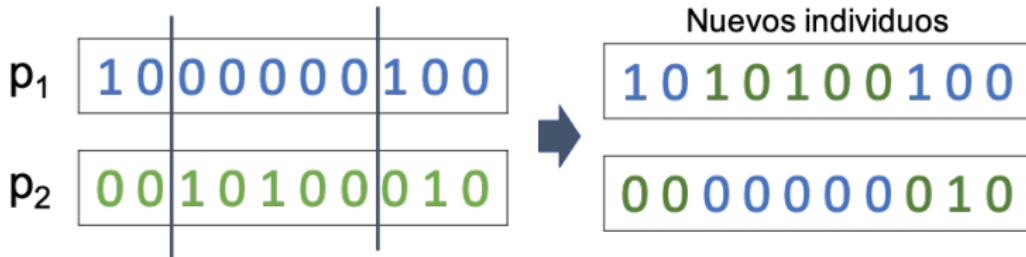
Operador que genera nuevos individuos a partir de otros

- El término general es **recombinación**
- Combinar los componentes de las soluciones (cromosomas)
- El objetivo es preservar lo bueno de las soluciones
→ **explotación!**
- El operador elegido depende del problema y representación
- Ejemplos son:
 - cruzamiento en un punto
 - cruzamiento en dos puntos
 - cruzamiento uniforme

Ejemplo: Cruzamiento de un punto

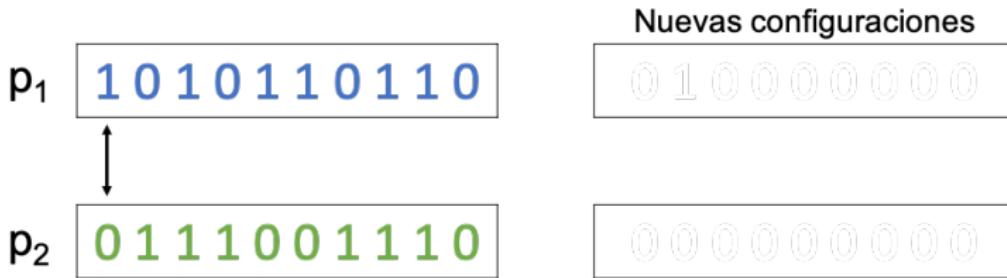


Ejemplo: Cruzamiento en dos puntos



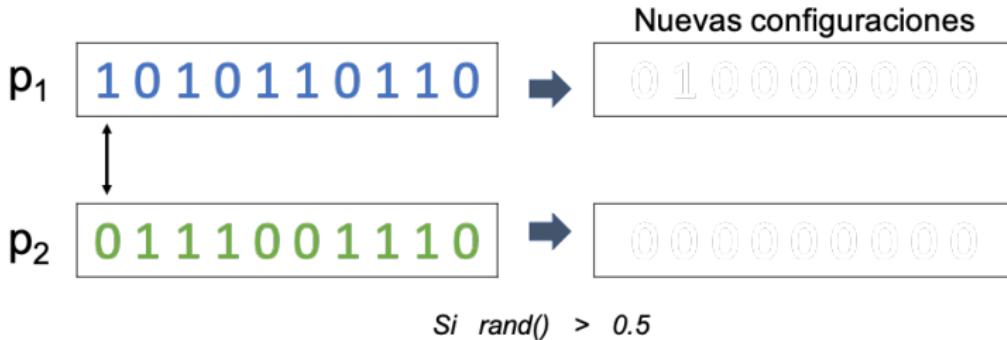
Algoritmos Genéticos: cruzamiento uniforme

Ejemplo: Cruzamiento uniforme (probabilidad = 0.5)



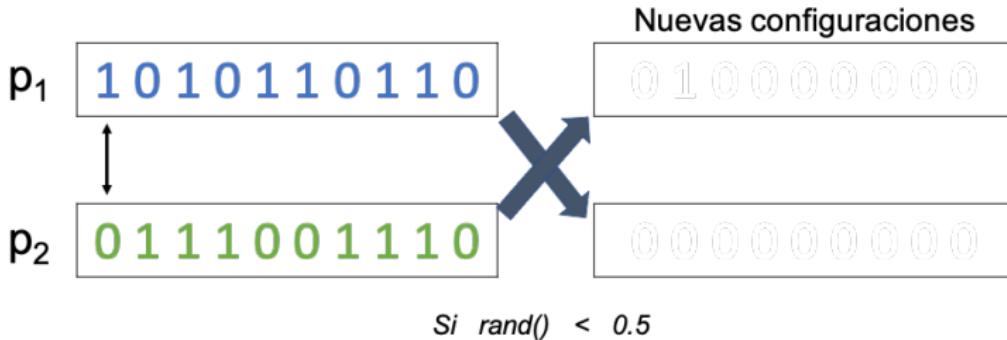
Algoritmos Genéticos: cruzamiento uniforme

Ejemplo: Cruzamiento uniforme (probabilidad = 0.5)



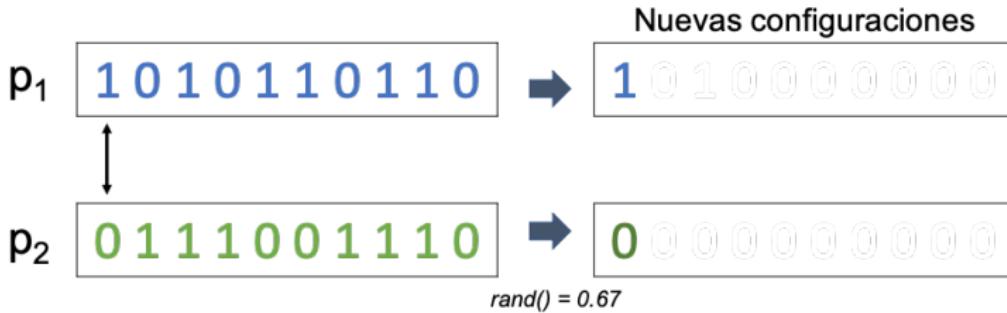
Algoritmos Genéticos: cruzamiento uniforme

Ejemplo: Cruzamiento uniforme (probabilidad = 0.5)



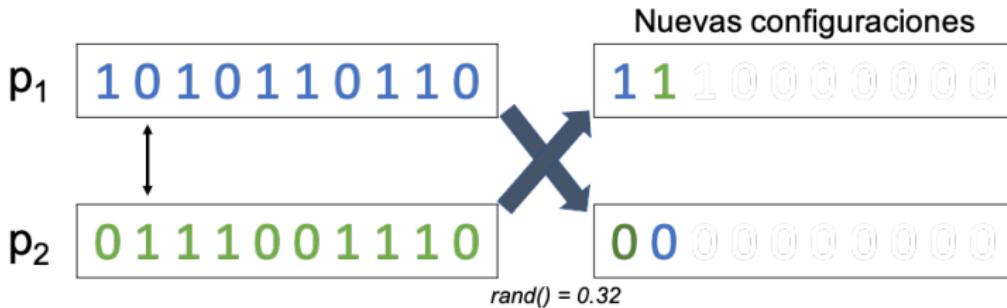
Algoritmos Genéticos: cruzamiento uniforme

Ejemplo: Cruzamiento uniforme (probabilidad = 0.5)



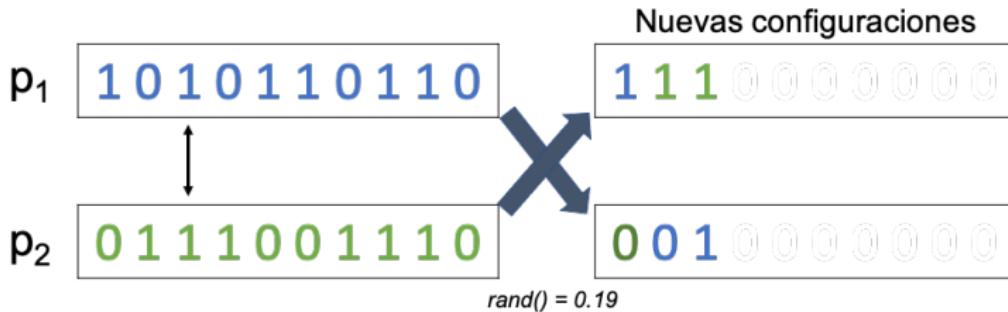
Algoritmos Genéticos: cruzamiento uniforme

Ejemplo: Cruzamiento uniforme (probabilidad = 0.5)



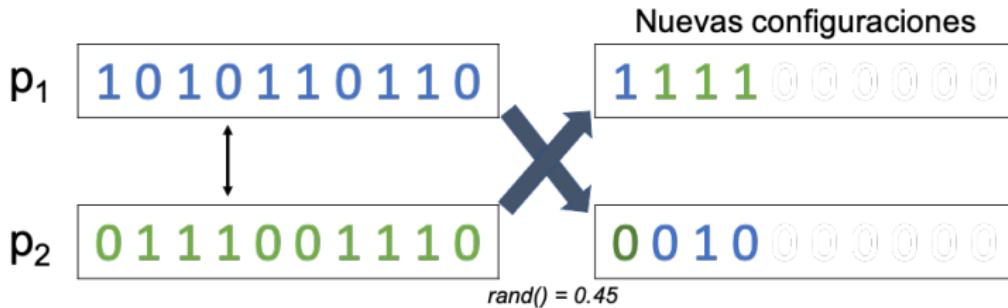
Algoritmos Genéticos: cruzamiento uniforme

Ejemplo: Cruzamiento uniforme (probabilidad = 0.5)



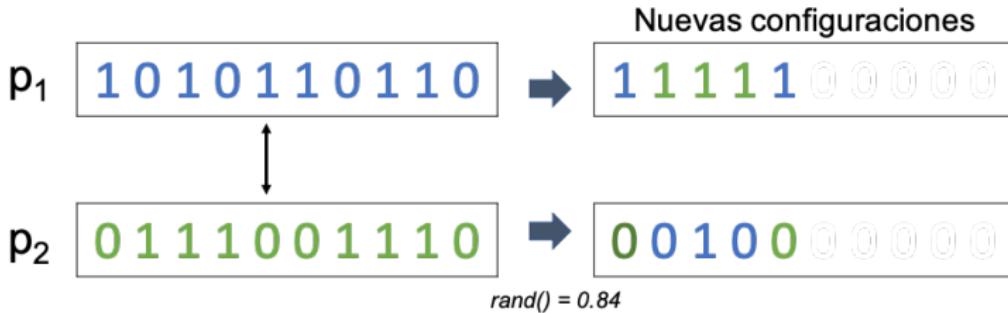
Algoritmos Genéticos: cruzamiento uniforme

Ejemplo: Cruzamiento uniforme (probabilidad = 0.5)



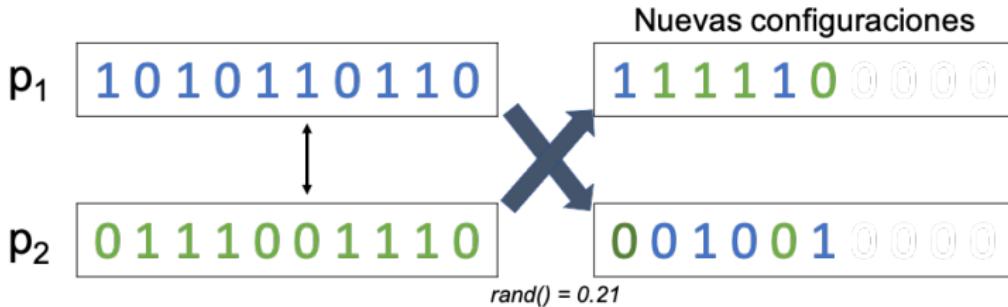
Algoritmos Genéticos: cruzamiento uniforme

Ejemplo: Cruzamiento uniforme (probabilidad = 0.5)



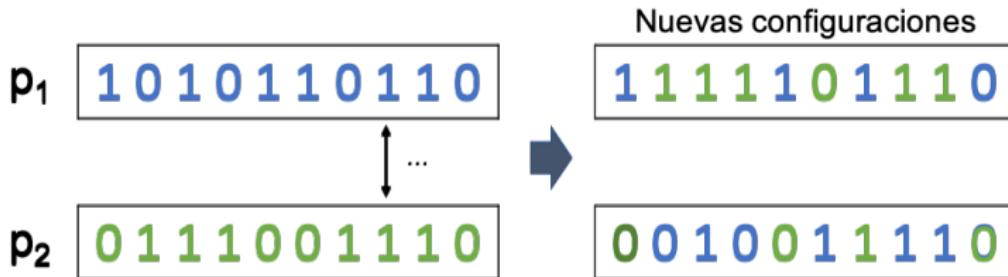
Algoritmos Genéticos: cruzamiento uniforme

Ejemplo: Cruzamiento uniforme (probabilidad = 0.5)



Algoritmos Genéticos: cruzamiento uniforme

Ejemplo: Cruzamiento uniforme (probabilidad = 0.5)



El cruzamiento es operador complejo cuando existen restricciones
→ Por ejemplo, ¿cómo cruzamos dos soluciones del TSP?

$s_1:$	A	B	C	D	E	F	G
$s_2:$	F	C	A	D	E	B	G

Cruzamiento ordenado: un operador especial para TSP

Algoritmos Genéticos: cruzamiento ordenado

Definimos un cruzamiento en el orden de las ciudades:

$p_1:$	A	B	C	D	E	F	G
$p_2:$	F	C	A	D	E	B	G

Paso 1. Seleccionar aleatoriamente un segmento de los padres:

$p_1:$	A	B	C	D	E	F	G
$p_2:$	F	C	A	D	E	B	G

Paso 2. Copiar segmento de padres a hijos ($p_1 \rightarrow h_1$ y $p_2 \rightarrow h_2$)

$h_1:$							
--------	--	--	--	--	--	--	--

$h_2:$							
--------	--	--	--	--	--	--	--

Algoritmos Genéticos: cruzamiento ordenado

Definimos un cruzamiento en el orden de las ciudades:

$p_1:$	A	B	C	D	E	F	G
$p_2:$	F	C	A	D	E	B	G

Paso 1. Seleccionar aleatoriamente un segmento de los padres:

$p_1:$	A	B	C	D	E	F	G
$p_2:$	F	C	A	D	E	B	G

Paso 2. Copiar segmento de padres a hijos ($p_1 \rightarrow h_1$ y $p_2 \rightarrow h_2$)

$h_1:$			C	D	E		
--------	--	--	---	---	---	--	--

$h_2:$			A	D	E		
--------	--	--	---	---	---	--	--

Definimos un cruzamiento en el orden de las ciudades:

$p_1:$	A	B	C	D	E	F	G
$p_2:$	F	C	A	D	E	B	G

Paso 1. Seleccionar aleatoriamente un segmento de los padres:

$p_1:$	A	B	C	D	E	F	G
$p_2:$	F	C	A	D	E	B	G

Paso 3. Copiar el resto de las ciudades en orden ($p_2 \rightarrow h_1$ y $p_1 \rightarrow h_2$)

$h_1:$			C	D	E		
--------	--	--	---	---	---	--	--

$h_2:$			A	D	E		
--------	--	--	---	---	---	--	--

Algoritmos Genéticos: cruzamiento ordenado

Definimos un cruzamiento en el orden de las ciudades:

$p_1:$	A	B	C	D	E	F	G
$p_2:$	F	C	A	D	E	B	G

Paso 1. Seleccionar aleatoriamente un segmento de los padres:

$p_1:$	A	B	C	D	E	F	G
$p_2:$	F	C	A	D	E	B	G

Paso 3. Copiar el resto de las ciudades en orden ($p_2 \rightarrow h_1$ y $p_1 \rightarrow h_2$)

$h_1:$	F		C	D	E		
--------	---	--	---	---	---	--	--

$h_2:$			A	D	E		
--------	--	--	---	---	---	--	--

Definimos un cruzamiento en el orden de las ciudades:

$p_1:$	A	B	C	D	E	F	G
$p_2:$	F	C	A	D	E	B	G

Paso 1. Seleccionar aleatoriamente un segmento de los padres:

$p_1:$	A	B	C	D	E	F	G
$p_2:$	F	C	A	D	E	B	G

Paso 3. Copiar el resto de las ciudades en orden ($p_2 \rightarrow h_1$ y $p_1 \rightarrow h_2$)

$h_1:$	F	A	C	D	E		
--------	---	---	---	---	---	--	--

$h_2:$			A	D	E		
--------	--	--	---	---	---	--	--

Definimos un cruzamiento en el orden de las ciudades:

$s_1:$	A	B	C	D	E	F	G
$s_2:$	F	C	A	D	E	B	G

Paso 1. Seleccionar aleatoriamente un segmento de los padres:

$s_1:$	A	B	C	D	E	F	G
$s_2:$	F	C	A	D	E	B	G

Paso 3. Copiar el resto de las ciudades en orden ($p_2 \rightarrow h_1$ y $p_1 \rightarrow h_2$)

$h_1:$	F	A	C	D	E	B	
--------	---	---	---	---	---	---	--

$h_2:$			A	D	E		
--------	--	--	---	---	---	--	--

Definimos un cruzamiento en el orden de las ciudades:

$p_1:$	A	B	C	D	E	F	G
$p_2:$	F	C	A	D	E	B	G

Paso 1. Seleccionar aleatoriamente un segmento de los padres:

$p_1:$	A	B	C	D	E	F	G
$p_2:$	F	C	A	D	E	B	G

Paso 3. Copiar el resto de las ciudades en orden ($p_2 \rightarrow h_1$ y $p_1 \rightarrow h_2$)

$h_1:$	F	A	C	D	E	B	G
--------	---	---	---	---	---	---	---

$h_2:$			A	D	E		
--------	--	--	---	---	---	--	--

Definimos un cruzamiento en el orden de las ciudades:

$p_1:$	A	B	C	D	E	F	G
$p_2:$	F	C	A	D	E	B	G

Paso 1. Seleccionar aleatoriamente un segmento de los padres:

$p_1:$	A	B	C	D	E	F	G
$p_2:$	F	C	A	D	E	B	G

Paso 3. Copiar el resto de las ciudades en orden ($p_2 \rightarrow h_1$ y $p_1 \rightarrow h_2$)

$h_1:$	F	A	C	D	E	B	G
--------	---	---	---	---	---	---	---

$h_2:$	B	C	A	D	E	F	G
--------	---	---	---	---	---	---	---

```
G ← poblacion_inicial();  
s+ ← mejor_solucion (G);  
while !criterio_de_termino() do  
    G* ← ∅;  
    while |G*| < n_individuos do  
        p1, p2 ← seleccionar_individuos (G, 2);  
        s ← cruzar_individuos (p1, p2);  
        s ← mutar (s);  
        G* ← G* + s;  
    end  
    G ← seleccionar_poblacion (G, G*);  
    s+ ← mejor_solucion (s+, G);  
end  
return s+;
```

Operador que modifica una solución para generar una nueva solución

- El objetivo es insertar nuevo material genético
→ **exploración!**
- Se utiliza normalmente un parámetro (prob_mutacion) que representa la probabilidad de aplicar la mutación
- Normalmente el valor de prob_mutacion es pequeño
- Ejemplos:
 - bit-flip: con probabilidad prob_mutacion cambiar el valor de cada variable
 - swap: con probabilidad prob_mutacion aplicar swap aleatorio

Operador bit-flip: con probabilidad $\text{prob_mutacion} = 0.1$ cambiar el valor de cada variable

1 0 1 0 1 0 0 1 0 0

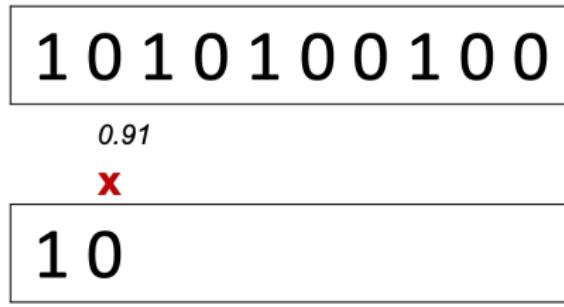
0.45

X

1

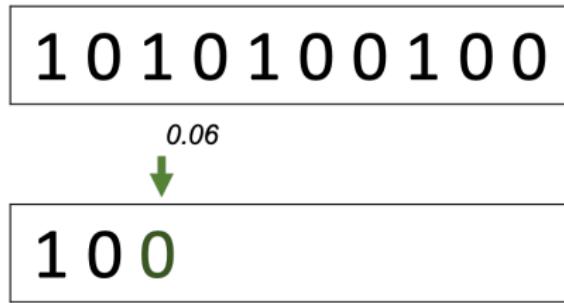
$\text{prob_mutacion} = 0.1$

Operador bit-flip: con probabilidad $\text{prob_mutacion} = 0.1$ cambiar el valor de cada variable



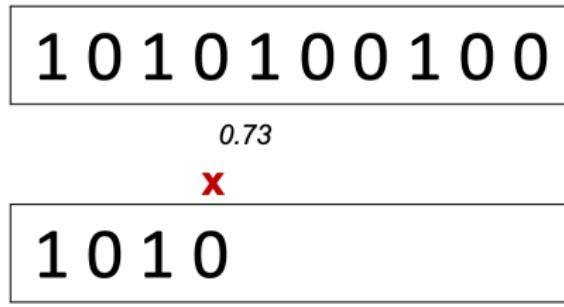
$$\text{prob_mutacion} = 0.1$$

Operador bit-flip: con probabilidad $\text{prob_mutacion} = 0.1$ cambiar el valor de cada variable



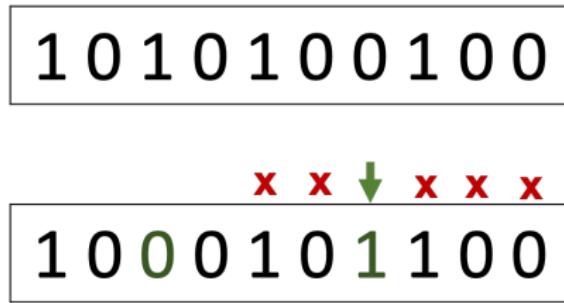
$$\text{prob_mutacion} = 0.1$$

Operador bit-flip: con probabilidad $\text{prob_mutacion} = 0.1$ cambiar el valor de cada variable



$$\text{prob_mutacion} = 0.1$$

Operador bit-flip: con probabilidad $\text{prob_mutacion} = 0.1$ cambiar el valor de cada variable



$$\text{prob_mutacion} = 0.1$$

Operador swap: con probabilidad $\text{prob_mutacion} = 0.1$ realizar un swap aleatorio

h

B	D	C	A	E
---	---	---	---	---

$\text{rand}() = 0.07 < \text{prob_mutacion} \rightarrow \text{si mutar!}$

realizar un swap aleatorio:

→ $\text{rand}() = 0.45$

→ $\text{rand}() = 0.86$

h :

B	D	C	A	E
---	---	---	---	---

 →

B	D	E	A	C
---	---	---	---	---

Algoritmos Genéticos: pseudocódigo

```
G ← poblacion_inicial();
s+ ← mejor_solucion (G);
while !criterio_de_termino() do
    G* ← ∅;
    while |G*| < n_individuos do
        p1, p2 ← seleccionar_individuos (G, 2);
        s ← cruzar_individuos (p1, p2);
        s ← mutar (s);
        G* ← G* + s;
    end
    G ← seleccionar_poblacion (G, G*);
    s+ ← mejor_solucion (s+, G);
end
return s+;
```

Operador que selecciona los individuos sobrevivientes que pasan a la nueva población

- Se pueden implementar diversas estrategias de selección
→ balance entre exploración y explotación
- Elitismo: seleccionar los mejores individuos, normalmente definido como parámetro. Ejemplo: $e = 0.2$ (20%)
- Proporcional al fitness: seleccionar los individuos de acuerdo a su calidad (por ejemplo selección de la ruleta)

Se utiliza generalmente una nomenclatura para clasificar los algoritmos evolutivos basados en su estrategia de evolución.

- μ : número de individuos en una generación
- λ : número de hijos creados en una generación

Se pueden definir las siguientes estrategias:

- (μ, λ)
- $(\mu + \lambda)$

- (μ, λ) :
 - población tamaño μ
 - población de hijos tamaño λ
 - los μ individuos para la próxima generación se seleccionan únicamente de los λ hijos
 - siempre $\lambda \geq \mu$

- $(\mu + \lambda)$:
 - población tamaño μ
 - población de hijos tamaño λ
 - los μ individuos para la próxima generación se seleccionan únicamente se seleccionan del set de los λ hijos y los μ padres

Supongamos el siguiente Algoritmo Genético (4 + 2):

Componentes:

- Población inicial: (dada por el ejercicio)
- Selección de padres: Selección Ruleta
- Cruzamiento: Operador ordenado
- Mutación: Operador swap
- Estrategia evolutiva: $(\mu + \lambda)$
- Selección de población: Elitista

Parámetros:

- Tamaño de la población: $\mu = 4$
- Tamaño de la población de hijos: $\lambda = 2$
- Probabilidad de mutación: $\text{prob_mutacion} = 0.2$

Ejecute paso a paso la primera iteración del algoritmo genético definido sobre esta instancia del TSP:

	A	B	C	D	E
A	-	8	4	9	9
B	8	-	6	7	10
C	4	6	-	5	6
D	9	7	5	-	4
E	9	10	6	4	-

Utilice los siguientes números aleatorios en la ejecución:

0.11, 0.77, 0.02, 0.46, 0.16, 0.43, 0.64, 0.85

Ejecute paso a paso el algoritmo genético definido:

	A	B	C	D	E
A	-	8	4	9	9
B	8	-	6	7	10
C	4	6	-	5	6
D	9	7	5	-	4
E	9	10	6	4	-

- Selección de padres: Selección Ruleta
- Cruzamiento: Operador ordenado
- Mutación: Operador swap
- $\text{prob_mutacion} = 0.2$
- Estrategia evolutiva: $(4 + 2)$
- Selección de población: Elitista

Numeros aleatorios:

0.11, 0.77, 0.02, 0.46, 0.16, 0.43, 0.64, 0.85

	A	B	C	D	E
A	-	8	4	9	9
B	8	-	6	7	10
C	4	6	-	5	6
D	9	7	5	-	4
E	9	10	6	4	-

*Selección de padres ruleta

*Cruzamiento ordenado

*Mutación swap

*prob_mutacion = 0.2

*Estrategia (4 + 2)

*Selección de población elitista

Numeros aleatorios:

0.11, 0.77, 0.02, 0.46, 0.16, 0.43,

0.64, 0.85

Población inicial

s_1	C	A	E	B	D	$f(s_1) = 35$
s_2	E	A	D	C	B	$f(s_2) = 39$
s_3	C	D	B	A	E	$f(s_3) = 35$
s_4	B	A	D	E	C	$f(s_4) = 33$

Paso 1: seleccionar padres

Calcular las probabilidades de la ruleta:

→ **minimización?**

$$\hat{f}(s) = \min + \max - f(s)$$

Ejercicio TSP

	A	B	C	D	E
A	-	8	4	9	9
B	8	-	6	7	10
C	4	6	-	5	6
D	9	7	5	-	4
E	9	10	6	4	-

*Selección de padres ruleta

*Cruzamiento ordenado

*Mutación swap

*prob_mutacion = 0.2

*Estrategia (4 + 2)

*Selección de población elitista

Numeros aleatorios:

0.11, 0.77, 0.02, 0.46, 0.16, 0.43,
0.64, 0.85

Paso 1: seleccionar padres

s_1	C	A	E	B	D	$f(s_1) = 35$
s_2	E	A	D	C	B	$f(s_2) = 39$
s_3	C	D	B	A	E	$f(s_3) = 35$
s_4	B	A	D	E	C	$f(s_4) = 33$

$$\hat{f}(s_1) = \min + \max - f(s_1)$$

$$33 + 39 - 35$$

$$37$$

$$\hat{f}(s_2) = 33 + 39 - 39 = 33$$

$$\hat{f}(s_3) = 33 + 39 - 35 = 37$$

$$\hat{f}(s_4) = 33 + 39 - 33 = 39$$

Ejercicio TSP

	A	B	C	D	E
A	-	8	4	9	9
B	8	-	6	7	10
C	4	6	-	5	6
D	9	7	5	-	4
E	9	10	6	4	-

*Selección de padres ruleta

*Cruzamiento ordenado

*Mutación swap

*prob_mutacion = 0.2

*Estrategia (4 + 2)

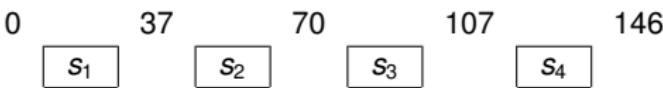
*Selección de población elitista

Numeros aleatorios:

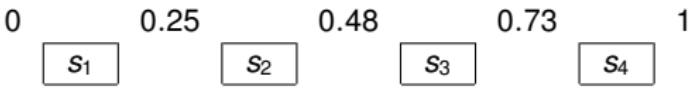
0.11, 0.77, 0.02, 0.46, 0.16, 0.43,
0.64, 0.85

Paso 1: seleccionar padres

s_1	C	A	E	B	D	$\hat{f}(s_1) = 37$
s_2	E	A	D	C	B	$\hat{f}(s_2) = 33$
s_3	C	D	B	A	E	$\hat{f}(s_3) = 37$
s_4	B	A	D	E	C	$\hat{f}(s_4) = 39$



Normalizamos dividiendo por 146:



Seleccionamos: $\text{rand}() = 0.11 \rightarrow s_1$

Ejercicio TSP

	A	B	C	D	E
A	-	8	4	9	9
B	8	-	6	7	10
C	4	6	-	5	6
D	9	7	5	-	4
E	9	10	6	4	-

*Selección de padres ruleta

*Cruzamiento ordenado

*Mutación swap

*prob_mutacion = 0.2

*Estrategia (4 + 2)

*Selección de población elitista

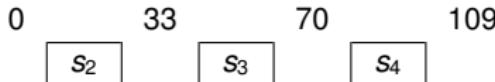
Numeros aleatorios:

0.111, 0.77, 0.02, 0.46, 0.16, 0.43,
0.64, 0.85

Paso 1: seleccionar padres

s_2	E	A	D	C	B	$\hat{f}(s_2) = 33$
s_3	C	D	B	A	E	$\hat{f}(s_3) = 37$
s_4	B	A	D	E	C	$\hat{f}(s_4) = 39$

Selección segundo parente:



Normalizamos dividiendo por 109:



Seleccionamos: rand() = 0.77 → s₄

Ejercicio TSP

	A	B	C	D	E
A	-	8	4	9	9
B	8	-	6	7	10
C	4	6	-	5	6
D	9	7	5	-	4
E	9	10	6	4	-

*Selección de padres ruleta

*Cruzamiento ordenado

*Mutación swap

*prob_mutacion = 0.2

*Estrategia (4 + 2)

*Selección de población elitista

Numeros aleatorios:

0.111, 0.777, 0.02, 0.46, 0.16, 0.43,

0.64, 0.85

Padres seleccionados:

$p_1 = s_1$	C	A	E	B	D
$p_2 = s_4$	B	A	D	E	C

Ejercicio TSP

	A	B	C	D	E
A	-	8	4	9	9
B	8	-	6	7	10
C	4	6	-	5	6
D	9	7	5	-	4
E	9	10	6	4	-

*Selección de padres ruleta

*Cruzamiento ordenado

*Mutación swap

*prob_mutacion = 0.2

*Estrategia (4 + 2)

*Selección de población elitista

Numeros aleatorios:

0.111, 0.777, 0.02, 0.46, 0.16, 0.43,

0.64, 0.85

Paso 2: aplicar cruzamiento

p_1	C	A	E	B	D
p_2	B	A	D	E	C

Seleccionar una sección a copiar:

- $\text{rand}() = 0.02$
- $\text{rand}() = 0.46$



p_1	C	A	E	B	D
p_2	B	A	D	E	C

Ejercicio TSP

	A	B	C	D	E
A	-	8	4	9	9
B	8	-	6	7	10
C	4	6	-	5	6
D	9	7	5	-	4
E	9	10	6	4	-

*Selección de padres ruleta

*Cruzamiento ordenado

*Mutación swap

*prob_mutacion = 0.2

*Estrategia (4 + 2)

*Selección de población elitista

Numeros aleatorios:

0.111, 0.777, 0.02, 0.46, 0.16, 0.43,

0.64, 0.85

Paso 2: aplicar cruzamiento

Copiar segmentos a hijos:

p_1	C	A	E	B	D
p_2	B	A	D	E	C

→ p_1 se copia a h_1 :

h_1					
-------	--	--	--	--	--

→ p_2 se copia a h_2 :

h_2					
-------	--	--	--	--	--

Ejercicio TSP

	A	B	C	D	E
A	-	8	4	9	9
B	8	-	6	7	10
C	4	6	-	5	6
D	9	7	5	-	4
E	9	10	6	4	-

*Selección de padres ruleta

*Cruzamiento ordenado

*Mutación swap

*prob_mutacion = 0.2

*Estrategia (4 + 2)

*Selección de población elitista

Numeros aleatorios:

0.111, 0.777, 0.02, 0.46, 0.16, 0.43,

0.64, 0.85

Paso 2: aplicar cruzamiento

Copiar segmentos a hijos:

p_1	C	A	E	B	D
p_2	B	A	D	E	C

→ p_1 se copia a h_1 :

h_1	C	A	E		
-------	---	---	---	--	--

→ p_2 se copia a h_2 :

h_2	B	A	D		
-------	---	---	---	--	--

Ejercicio TSP

	A	B	C	D	E
A	-	8	4	9	9
B	8	-	6	7	10
C	4	6	-	5	6
D	9	7	5	-	4
E	9	10	6	4	-

*Selección de padres ruleta

*Cruzamiento ordenado

*Mutación swap

*prob_mutacion = 0.2

*Estrategia (4 + 2)

*Selección de población elitista

Numeros aleatorios:

0.111, 0.777, 0.02, 0.46, 0.16, 0.43,

0.64, 0.85

Paso 2: aplicar cruzamiento

Completar ciudades en orden:

p_1	C	A	E	B	D
p_2	B	A	D	E	C

→ orden en p_2 en h_1 :

h_1	C	A	E		
-------	---	---	---	--	--

→ orden en p_1 en h_2 :

h_2	B	A	D		
-------	---	---	---	--	--

Ejercicio TSP

	A	B	C	D	E
A	-	8	4	9	9
B	8	-	6	7	10
C	4	6	-	5	6
D	9	7	5	-	4
E	9	10	6	4	-

*Selección de padres ruleta

*Cruzamiento ordenado

*Mutación swap

*prob_mutacion = 0.2

*Estrategia (4 + 2)

*Selección de población elitista

Numeros aleatorios:

0.111, 0.777, 0.02, 0.46, 0.16, 0.43,

0.64, 0.85

Paso 2: aplicar cruzamiento

Completar ciudades en orden:

p_1	C	A	E	B	D
p_2	B	A	D	E	C

→ orden en p_2 en h_1 :

h_1	C	A	E	B	D
-------	---	---	---	---	---

→ orden en p_1 en h_2 :

h_2	B	A	D	C	E
-------	---	---	---	---	---

Ejercicio TSP

	A	B	C	D	E
A	-	8	4	9	9
B	8	-	6	7	10
C	4	6	-	5	6
D	9	7	5	-	4
E	9	10	6	4	-

*Selección de padres ruleta

*Cruzamiento ordenado

*Mutación swap

*prob_mutacion = 0.2

*Estrategia (4 + 2)

*Selección de población elitista

Numeros aleatorios:

0.111, 0.777, 0.02, 0.46, 0.16, 0.43,

0.64, 0.85

Hijos obtenidos del cruzamiento:

h_1	C	A	E	B	D
h_2	B	A	D	C	E

Ejercicio TSP

	A	B	C	D	E
A	-	8	4	9	9
B	8	-	6	7	10
C	4	6	-	5	6
D	9	7	5	-	4
E	9	10	6	4	-

*Selección de padres ruleta

*Cruzamiento ordenado

*Mutación swap

*prob_mutacion = 0.2

*Estrategia (4 + 2)

*Selección de población elitista

Numeros aleatorios:

0.111, 0.777, 0.02, 0.46, 0.16, 0.43,
0.64, 0.85

Paso 3: aplicar mutación

h_1	C	A	E	B	D
h_2	B	A	D	C	E

¿Aplicar mutación en hijo h_1 ?

→ rand() = 0.16 < prob_mutacion

h_1 **se debe mutar!**

Seleccionar posiciones para swap:

rand() = 0.43

rand() = 0.64

Aplicar swap:

h_1	C	A	B	E	D
-------	---	---	---	---	---

	A	B	C	D	E
A	-	8	4	9	9
B	8	-	6	7	10
C	4	6	-	5	6
D	9	7	5	-	4
E	9	10	6	4	-

*Selección de padres ruleta

*Cruzamiento ordenado

*Mutación swap

*prob_mutacion = 0.2

*Estrategia (4 + 2)

*Selección de población elitista

Numeros aleatorios:

0.111, 0.777, 0.02, 0.46, 0.16, 0.43,

0.64, 0.85,

Paso 3: aplicar mutación

h_1	C	A	E	B	D
h_2	B	A	D	C	E

¿Aplicar mutación en hijo h_2 ?

→ $\text{rand}() = 0.85 > \text{prob_mutacion}$

h_2 **no se debe mutar!**

Hijos después de la mutación:

h_1	C	A	B	E	D	$f(h_1) = 31$
h_2	B	A	D	C	E	$f(h_2) = 38$

	A	B	C	D	E
A	-	8	4	9	9
B	8	-	6	7	10
C	4	6	-	5	6
D	9	7	5	-	4
E	9	10	6	4	-

*Selección de padres ruleta

*Cruzamiento ordenado

*Mutación swap

*prob_mutacion = 0.2

*Estrategia (4 + 2)

*Selección de población elitista

Numeros aleatorios:

0.111, 0.777, 0.02, 0.46, 0.16, 0.43,

0.64, 0.85

Paso 4: seleccionar población

Selección elitista (4 + 2)

h_1	C	A	B	E	D	$f(h_1) = 31$
s_4	B	A	D	E	C	$f(s_4) = 33$
s_1	C	A	E	B	D	$f(s_1) = 35$
s_3	C	D	B	A	E	$f(s_3) = 35$
h_2	B	A	D	C	E	$f(h_2) = 38$
s_2	E	A	D	C	B	$f(s_2) = 39$

¿Cuáles seleccionamos?

	A	B	C	D	E
A	-	8	4	9	9
B	8	-	6	7	10
C	4	6	-	5	6
D	9	7	5	-	4
E	9	10	6	4	-

*Selección de padres ruleta

*Cruzamiento ordenado

*Mutación swap

*prob_mutacion = 0.2

*Estrategia (4 + 2)

*Selección de población elitista

Numeros aleatorios:

0.111, 0.777, 0.02, 0.46, 0.16, 0.43,

0.64, 0.85

Paso 4: seleccionar población

Población seleccionada:

h_1	C	A	B	E	D	$f(h_1) = 31$
s_4	B	A	D	E	C	$f(s_4) = 33$
s_1	C	A	E	B	D	$f(s_1) = 35$
s_3	C	D	B	A	E	$f(s_3) = 35$

Primera iteración terminada!