

Operadores de cruce y reemplazo de individuos

Computación evolutiva

7 de octubre de 2015

David Atienza Guillermo Delgado Javier López
Rubén Martín Álvaro Muñoz Gustavo Puig

Departamento de Inteligencia Artificial



CAMPUS
DE EXCELENCIA
INTERNACIONAL

Algoritmos genéticos

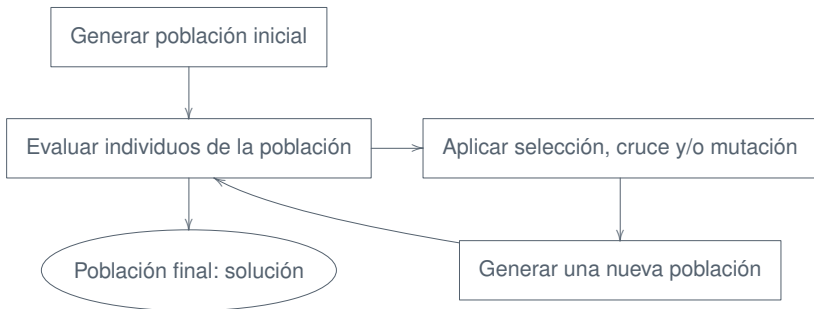
Operadores de cruce con alfabeto finito

Operadores de cruce con codificación real

Reemplazo de individuos

Algoritmos genéticos

Funcionamiento general





- ▶ Operador cruce:
 - ▶ Obtener descendientes a partir de los padres
 - ▶ Los nuevos individuos tienen parte del genotipo de los padres
 - ▶ Diferentes métodos entre alfabeto finito y codificación real
- ▶ Operador reemplazo:
 - ▶ Reemplazar individuos de la antigua población por los nuevos individuos



OBJETIVO: Todo algoritmo genético necesita establecer un equilibrio entre dos factores aparentemente opuestos:

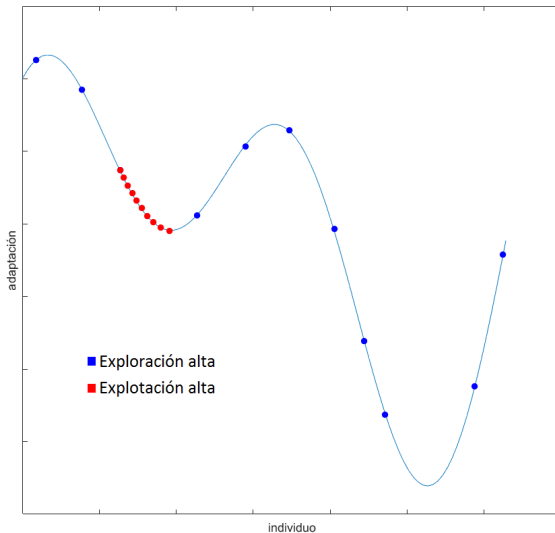
- ▶ Exploración:
 - ▶ Diversidad de soluciones
 - ▶ Explorar todo el espacio de soluciones para localizar zonas prometedoras
 - ▶ Evitar óptimos locales
- ▶ Explotación:
 - ▶ Convergencia de soluciones
 - ▶ Centrar la búsqueda en regiones exploradas para mejorar las soluciones encontradas
 - ▶ Evitar la búsqueda aleatoria

Exploración y explotación



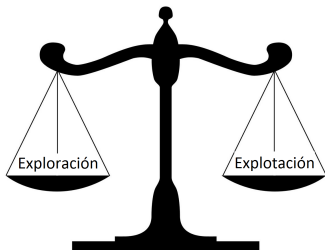
POLITÉCNICA

5



Existen diversas formas de variar la exploración y la explotación:

- ▶ Operador mutación: Aumenta la exploración.
- ▶ Operador selección: Elegir a los mas aptos aumenta la explotación
- ▶ Operador cruce: Descendientes con genotipo similar a los padres aumenta la explotación
- ▶ Reemplazo: Eliminar a los individuos menos adaptados aumenta la explotación.





- ▶ Los puntos de la función a optimizar utilizan un **alfabeto finito**:

$$A = \{0, 1, \dots, m\}$$

- ▶ Los **individuos** de la población son los puntos del espacio de búsqueda :

$$c_1, c_2, \dots, c_l \quad \forall c_k \in A, k \in \{1, 2, \dots, l\}$$

- ▶ Individuos seleccionadas de la **población** (con probabilidad p_c):

Individuos progenitores o **padres**

- ▶ Individuos generados por el operador:

Individuos **descendientes**

Operadores de cruce con alfabeto finito II

Operador de cruce ordinario U operador basado en un punto



- Padres:

01010 y 10101

- Lugar de cruce $k \in [1, 5]$:

010 10
 \updownarrow
101 01 $k = 3$

- Descendientes:

01001 y 10110

Operadores de cruce con alfabeto finito III

Operador de cruce basado en dos puntos



9

- Padres:

01010 y 10101

- Lugares de cruce $k, k' \in [1, 5]$:

01	01	0	
	↑		
	↓		
10	10	1	

$k = 2$ y $k' = 4$

- Descendientes:

01100 y 10011

Operadores de cruce con alfabeto finito IV

Operador de cruce uniforme



- ▶ Padres:

01010 y 10101

- ▶ Máscara de cruce aleatoria:

$m = 01001$

- ▶ ¿Cómo aplicar la máscara?

- ▶ Descendiente 1:

a_i si m_i es 0, b_i si m_i es 1

- ▶ Descendiente 2:

a_i si m_i es 1, b_i si m_i es 0

- ▶ Descendientes:

00011 y 11100

Operadores de cruce con alfabeto finito V

Operador de cruce generalizado



- ▶ Se trabaja con cadenas binarias $S = \{0, 1\}^l$.
- ▶ Existe una función biyectiva para transformar las cadenas:

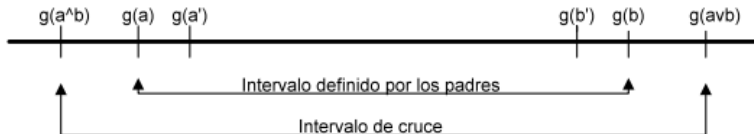
$$g : S \rightarrow \{0, 1, \dots, 2^l - 1\}$$

- ▶ Descendiente 1:

$$a' \in g^{-l}([g(a \wedge b), g(a \vee b)])$$

- ▶ Descendiente 2:

$$b' \in g^{-l}(g(a) + g(b) - g(a'))$$





Reflexiones

- ▶ ¿Qué ocurre con el ordenamiento de los genes?
- ▶ ¿Y si los padres tienen el mismo gen?
- ▶ ¿Puede llegar a ocurrir que se convierta en una búsqueda aleatoria?
- ▶ ¿Equilibrio entre exploración y explotación?
- ▶ ¿Son capaces todos los operadores de encontrar cualquier solución?
- ▶ ¿Todos los operadores funcionan de forma similar?



¿Que ocurre si usamos un método de cruce de los explicados anteriormente?

- Supongamos que tenemos lo siguiente:

- $a = a_1, a_2, \dots, a_l$ y $b = b_1, b_2, \dots, b_l$
 $a_i, b_i \in \mathbb{R}, \forall i \in \{1, \dots, l\}$

- lo cruzamos y obtenemos:

- $c = c_1, c_2, \dots, c_l$
 $c_i \in \{a_1, \dots, a_l, b_1, \dots, b_l\}, \forall i \in \{1, \dots, l\}$

3	8	5
---	---	---

6	2	3
---	---	---

3	2	5
---	---	---



6	8	3
---	---	---

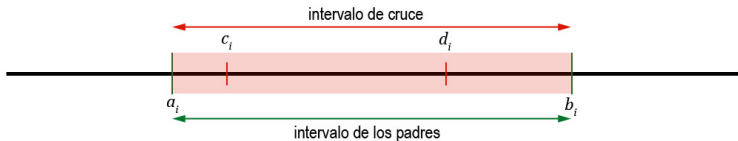
- ▶ A partir de 2 progenitores generamos 2 descendientes
- ▶ Utiliza un intervalo de números reales (intervalo de cruce)
- ▶ Se escoge un valor aleatorio dentro del intervalo

$$a = a_1, a_2, \dots, a_l$$
$$b = b_1, b_2, \dots, b_l$$



$$c = c_1, c_2, \dots, c_l$$
$$d = d_1, d_2, \dots, d_l$$

$$C_i = [\min(a_i, b_i), \max(a_i, b_i)]$$



Operador de cruce lineal

Wright



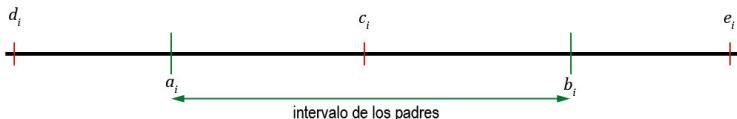
- ▶ A partir de 2 progenitores generamos 3 descendientes
- ▶ Se escoge en el primer caso el valor intermedio del intervalo
- ▶ Los otros individuos se escogen fuera del intervalo de los progenitores, concretamente se alejan la misma distancia que la que hay al centro

$$a = a_1, a_2, \dots, a_l$$
$$b = b_1, b_2, \dots, b_l$$



$$c = c_1, c_2, \dots, c_l$$
$$d = d_1, d_2, \dots, d_l$$
$$e = e_1, e_2, \dots, e_l$$

$$c_i = \frac{a_i + b_i}{2}$$
$$d_i = 1,5 \cdot a_i - 0,5 \cdot b_i$$
$$e_i = -0,5 \cdot a_i + 1,5 \cdot b_i$$



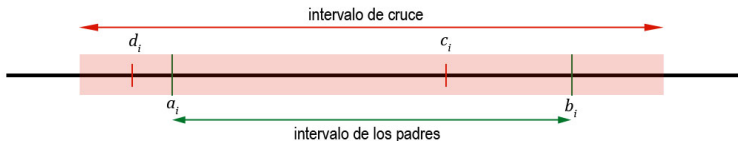
Operador de cruce combinado (BLX-a)



- ▶ A partir de 2 progenitores generamos 2 descendientes
- ▶ Se escoge un valor aleatorio dentro del intervalo de cruce
- ▶ Generamos un intervalo que puede ser mas grande que el intervalo de los progenitores dado por α

$$\begin{array}{l} a = a_1, a_2, \dots, a_l \\ b = b_1, b_2, \dots, b_l \end{array} \quad \begin{array}{c} \curvearrowright \\ \curvearrowleft \end{array} \quad \begin{array}{l} c = c_1, c_2, \dots, c_l \\ d = d_1, d_2, \dots, d_l \end{array}$$

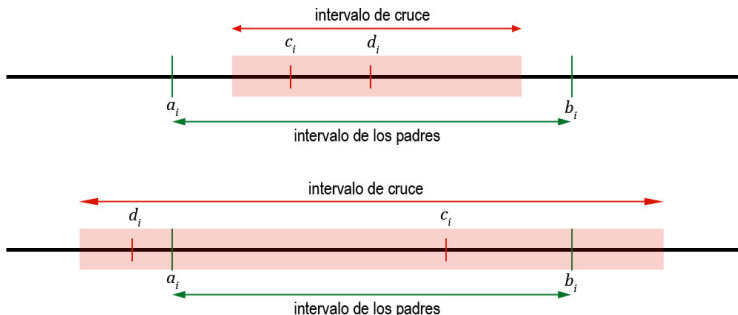
$$C_i = [a_i - \alpha l, b_i - \alpha l]$$



Operador de cruce morfológico (MMX)



- ▶ A partir de un número n impar de progenitores obtenemos dos descendientes
- ▶ Se genera un intervalo de cruce que **depende de la diversidad genética**
- ▶ Los descendientes se generan eligiendo números aleatorios dentro del intervalo de cruce
- ▶ Se normaliza el dominio



- ▶ Se ejecutan los siguientes pasos:
 1. Cálculo de la medida de la diversidad genética
 2. Cálculo de los intervalos de cruce
 3. Obtención de la descendencia

Definición: análisis de la tendencia de la evolución genética para deducir dinámicamente que técnicas evolutivas aplicar.

► Técnicas:

1. Explotación.
2. Exploración.

OBJETIVO: dado un subconjunto de una población de λ individuos, estudiar cuán grande es la variedad genética con respecto al individuo con características medias.

¿CÓMO SE ANALIZA? Utilizando Morfología Matemática aplicada a imágenes binarias → Gradiente Morfológico.

David Atienza, Guillermo Delgado, Javier López, Rubén Martín, Álvaro Muñoz, Gustavo Puig | Operadores de cruce y reemplazo de individuos

MORFOLOGÍA MATEMÁTICA SOBRE IMÁGENES:

- ▶ **MORFOLOGÍA** → *MORFOS* → *forma*.
- ▶ **VISION ARTIFICIAL** → Gradiente Morfológico
→ segmentación de imágenes para obtención de formas.

Original



Normalizado/Binarizado



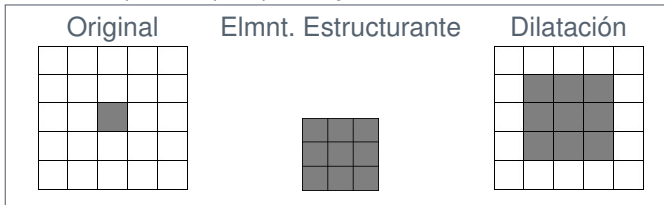
Grad. Morfológico



MATEMÁTICAMENTE SOBRE UNA IMAGEN BIDIMENSIONAL:

Ingeniería inversa: $g_m(f) = \text{dilatation}(f) - \text{erotion}(f)$

- DILATACIÓN: $(f \oplus b)(s, t) = \max\{f(s - x, t - y) + b(x, y) : (s - x, t - y) \in D_f; (x, y) \in D_b\}$

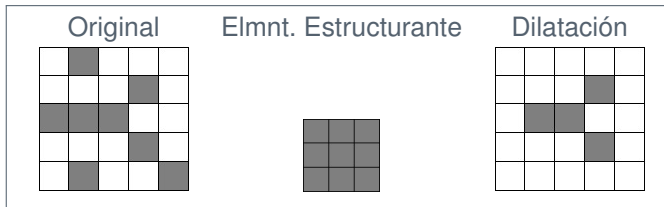


- Si nuestro objetivo es obtener la máxima variación, el estructurante debe ser nulo para no limitar la dilatación:

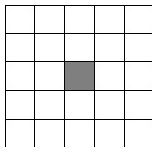


PARA EL CASO DE LA EROSIÓN:

- $(f \ominus b)(s, t) = \min\{f(s + x, t + y) - b(x, y) : (s + x, t + y) \in D_f; (x, y) \in D_b\}$



- Si nuestro objetivo es obtener la máxima variación, el estructurante debe ser nulo para no limitar la erosión:



EXTRAPOLANDO A NUESTRO CASO:

$$G[\text{matrizprogenitora}] = \begin{pmatrix} a_{0,0} & a_{0,1} & \dots & a_{0,l-1} \\ a_{1,0} & a_{1,1} & \dots & a_{1,l-1} \\ a_{2,0} & a_{2,1} & \dots & a_{2,l-1} \\ a_{3,0} & a_{3,1} & \dots & a_{3,l-1} \\ \vdots & \vdots & \dots & \vdots \\ a_{n,0} & a_{n,1} & \dots & a_{n,l-1} \end{pmatrix}$$

donde n ($n \leq \lambda$) es el N. de progenitores y l la longitud de cada cromosoma.

- Maximizar información \rightarrow analizar la evolución de los genes en particular (y no los cromosomas/individuos en general).

$$f_0 = (a_{0,0}, a_{1,0} a_{2,0} a_{3,0} \dots a_{n-1,0})$$

CASO PARTICULAR:

- ▶ Si tomamos $n = 5$ con cardinal $[1, 6]$; $f_0 = (6, 2, 4, 1, 3, 5)$.

Nuestro interés es averiguar la desviación con respecto el gen que ocupa la posición media.

Teniendo en cuenta que nuestro elemento estructurante es nulo no ofrece restricciones a la dilatación.

- ▶ El valor máximo dilatado que podemos obtener será 6.
- ▶ Del mismo modo, la menor de las erosiones de nuestra muestra será el menor de todos los valores: 1.

CONCLUSIÓN: $g_m(f_i) = \max(f_i) - \min(f_i)$.

ANÁLISIS DE RESULTADOS:

Si $g_m(f_i)$ es muy elevado:

- ▶ Los genes de en la posición i de nuestro progenitores toman valores dispersos \rightarrow nuestra población sobreexplora el espacio de búsqueda.
- ▶ Debemos aplicar cruces que fomenten la explotación \rightarrow los genes (i) de sus hijos deben tomar valores más centrados en la media de la imagen.

Si $g_m(f_i)$ es muy pequeño:

- ▶ Los genes de en la posición i de nuestro progenitores toman valores cercanos \rightarrow nuestra población sobreexplota el espacio de búsqueda.
- ▶ Debemos aplicar cruces que fomenten la exploración \rightarrow los genes (i) de sus hijos deben tomar valores más separados de la media de la imagen.

- ▶ Buscamos los intervalos C_i
- ▶ Estos intervalos dependerán de una función $\varphi : \mathbb{R} \rightarrow \mathbb{R}$ denominada **función de exploración/explotación**.
- ▶ Sean g_{imax} gen máximo y g_{imin} gen mínimo:
 - ▶ $g_{imax} = \underbrace{(f_i \oplus b)(E(n/2) + 1)}_{\text{Dilataci}nf_i\text{enelpuntomedio}} - \varphi(g_i) = mx(f_i) - \varphi(g_i)$
 - ▶ $g_{imin} = \underbrace{(f_i \ominus b)(E(n/2) + 1)}_{\text{Erosinf}_i\text{enelpuntomedio}} + \varphi(g_i) = mn(f_i) + \varphi(g_i)$
- ▶ Definiremos los intervalos de cruce: $C_i = [g_{imin}, g_{imax}]$ con $i=0, \dots, l-1$.

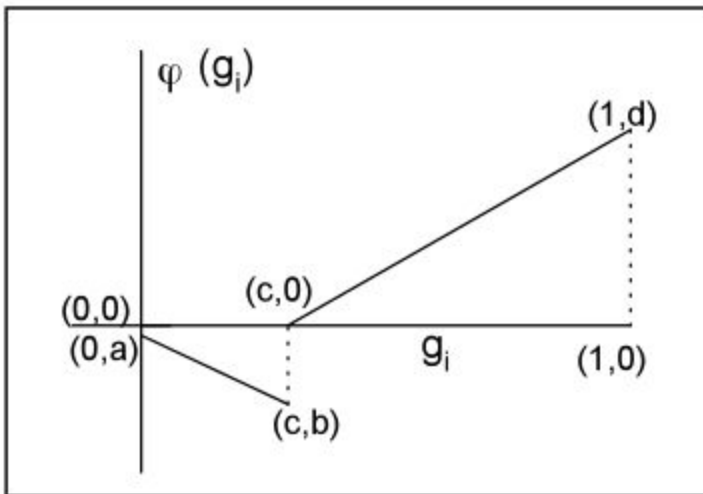
La función de exploración/explotación, FEE, es crucial en la determinación de los intervalos.

- ▶ Si $\varphi(g_i) = 0, \forall g_i$ entonces:
 - ▶ Los intervalos de cruce serán de la forma
 $C_i = [mn(f_i), mx(f_i)]$ (*intervalo definido por los padres o intervalo de referencia*).
 - ▶ Genera únicamente individuos pertenecientes al intervalo de cruce definido por los padres.
 - ▶ Aplica únicamente técnicas de explotación \rightarrow convergencia prematura.
- ▶ La FEE al depender de g_i nos permite establecer una regla que distinga si debemos optar por la exploración o la explotación.

Estudio casos:

- ▶ Si los valores que toma un determinado gen son muy similares(convergencia hacia un punto):
 - ▶ g_i estará próximo a 0.
 - ▶ La FEE deberá permitir la expansión del intervalo para evitar la convergencia a un punto que no sea el óptimo(exploración).
 - ▶ $\varphi(g_i)$ deberá tomar valores negativos para expandir el intervalo.
- ▶ Si los valores son muy distintos(dispersión de la población en el espacio de búsqueda):
 - ▶ g_i estará próximo a 1.
 - ▶ La FEE deberá permitir la expansión del intervalo para evitar la convergencia a un punto que no sea el óptimo(exploración).
 - ▶ $\varphi(g_i)$ deberá tomar valores positivos para estrechar el intervalo.

Convergencia con codificación binaria



Resumen de las condiciones que debe cumplir $\varphi(g_i)$:

- ▶ 1- Debe proporcionar valores positivos para valores elevados de g_i y negativos para valores próximos a 0.
- ▶ 2- Coste computacional bajo.
- ▶ 3- El dominio debe ser el intervalo real $[0,1]$
- ▶ 4- Se debe cumplir que $g_{imin} \leq g_{imax}$
- ▶ 5- $\varphi(g_i) \neq 0$

- ▶ Una vez conocidos los intervalos de cruce C_i de cada gen, podemos calcular nuevos individuos.
- ▶ Crearemos dos individuos, $o = (o_0, \dots, o_{l-1})$ y $o' = (o'_0, \dots, o'_{l-1})$.
- ▶ Como cada gen tiene un intervalo de cruce diferente, se debe aplicar el siguiente procedimiento por cada gen:
 1. o_i se escoge aleatoriamente en el intervalo de cruce C_i .
 2. o'_i se calcula con la expresión:

$$o'_i = (\min(f_i) + \max(f_i)) - o_i$$

- ▶ Se demuestra que $g_{imin} \leq o'_i \leq g_{imax}$.
- ▶ o_i y o'_i se encuentran equidistantes al centro del intervalo de cruce.



Algunas consideraciones iniciales:

- ▶ Es importante asegurarse de que el coste computacional del cruce no sea demasiado elevado \Rightarrow se repite muchas veces.
- ▶ La multiplicación es una operación más costosa que una suma / resta.
- ▶ Repetiremos el proceso de creación de σ_i y σ'_i una vez por cada gen, es decir, l veces. Ese proceso debería ser lo menos costoso posible.
- ▶ Estamos realizando el mismo proceso en los l genes \Rightarrow fácilmente paralelizable en un ordenador con l procesadores.



Podemos dividir el proceso de cruce en tres partes, que se deben repetir en cada gen:

1. En primer lugar se calcula el gradiente morfológico,
 $g = \max(f_i) - \min(f_i)$.
 - ▶ Para calcular el valor máximo / mínimo, necesitamos realizar n sumas / restas.
 2. Posteriormente, se calcula el intervalo $[g_{imin}, g_{imax}]$.
 - ▶ $g_{imin} = \max(f_i) - \varphi(x)$
 - ▶ $g_{imax} = \min(f_i) + \varphi(x)$
 - ▶ Calcular $\varphi(x)$ supone una multiplicación y una suma.
 3. Por último, se calculan los descendientes o_i y o'_i .
 - ▶ $o'_i = (\min(f_i) + \max(f_i)) - o_i$
- ▶ En total se realizan $2n + 6$ sumas y 1 multiplicación por cada gen.

- ▶ Se dispone de una población con λ individuos y conocemos cómo generar descendientes utilizando el operador de cruce.
- ▶ Se deben obtener λ individuos que formaran parte de la población de la siguiente generación.
- ▶ ¿Cómo decidir que individuos formarán parte de la siguiente generación?

- ▶ La tasa de reemplazamiento generacional, t_{tg} , indica el porcentaje de hijos generados respecto al total de la población inicial.
- ▶ Con $t_{tg} = \lambda^{-1}$, se realiza la sustitución de un individuo de la población por un descendiente.
- ▶ Aquellos algoritmos genéticos en los que se sustituyen unos pocos individuos se los denomina **SSGA (Steady-State Replacement Genetic Algorithms)**.

Otros reemplazamientos:

- ▶ **Reducción simple:** se realiza un reemplazamiento en bloque, esto es, $t_{tg} = 1$.
- ▶ **Reducción elitista de grado λ :** Se seleccionan a los λ mejores individuos a partir de los λ individuos iniciales y λ descendientes.
- ▶ **Algoritmos genéticos modificados:**
 - ▶ r_1 individuos seleccionados por la reproducción para ser cruzados.
 - ▶ r_2 individuos seleccionados para morir.
 - ▶ $\lambda - (r_1 + r_2)$ individuos neutros.
 - ▶ Cuanto más adaptado está un individuo tiene más probabilidades de ser seleccionado por la reproducción y menores sus probabilidades de morir

Los operadores catastróficos evalúan la convergencia para evitar acabar en un óptimo local.

- ▶ Se sustituyen individuos de la población por otros nuevos (y generados aleatoriamente).
- ▶ Se aplican cuando el algoritmo esta convergiendo para escapar de un óptimo local mediante exploración.
- ▶ No se eliminan a los mejores individuos de la población.
- ▶ Principales operadores catastróficos:
 1. **Empaquetado**: Todos los individuos con el mismo valor de adaptación son eliminados excepto uno. Evita los individuos repetidos.
 2. **El día del juicio final**: Solo se conserva el individuo más adaptado. El resto se eliminan.

¿Alguna pregunta?



CAMPUS
DE EXCELENCIA
INTERNACIONAL