



# INSTITUTO POLITÉCNICO NACIONAL

## ESCUELA SUPERIOR DE CÓMPUTO - IPN

### TÓPICOS SELECTOS DE ALGORITMOS BIOINSPIRADOS

#### CRUZAMIENTOS AVANZADOS

*Presenta*

**Dr. DANIEL MOLINA PÉREZ**

**danielmolinaperez90@gmail.com**

INSTITUTO POLITÉCNICO NACIONAL

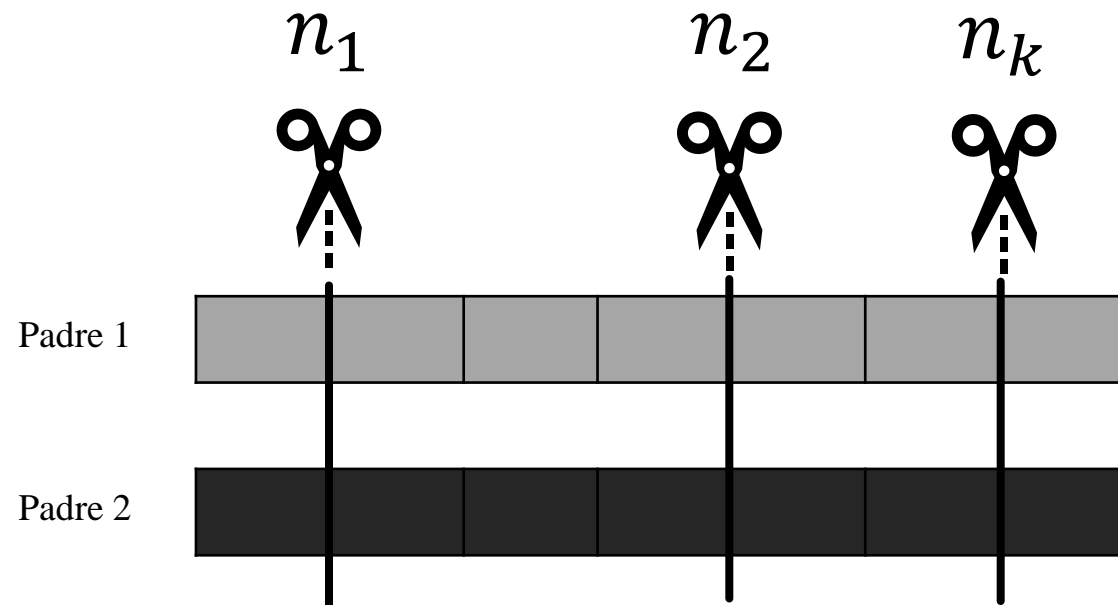


ESCOM

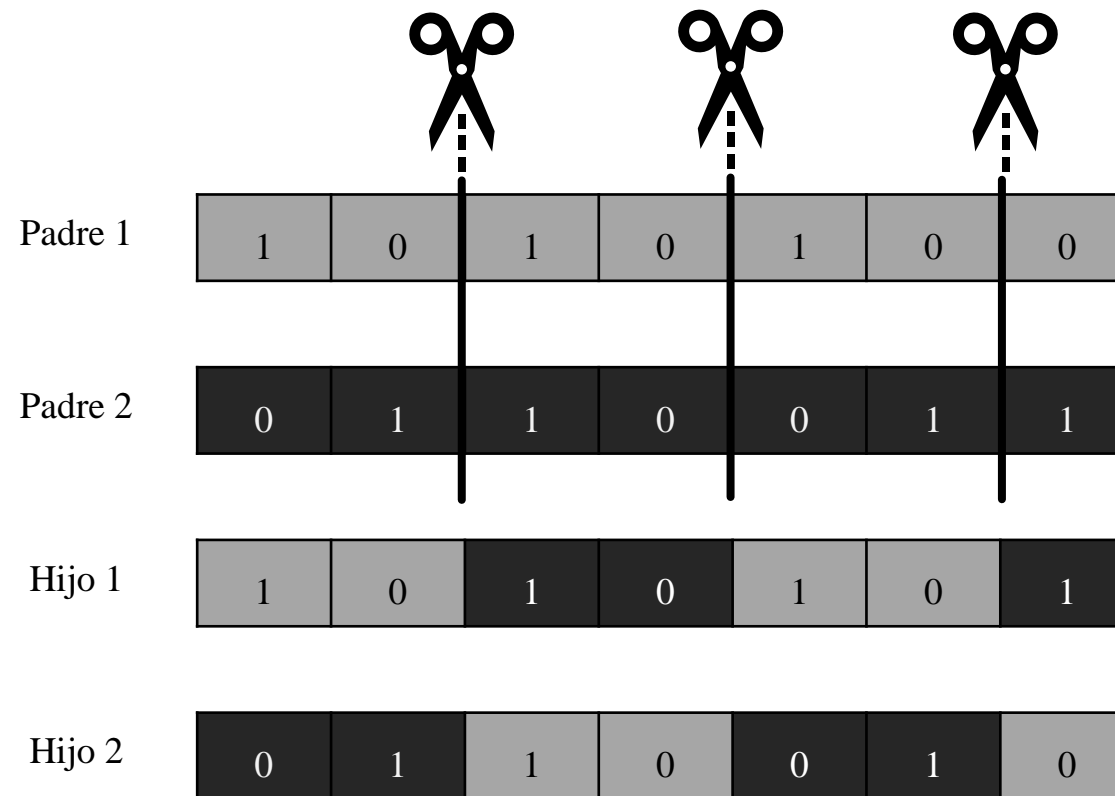
CIUDAD DE MÉXICO

## CRUZAMIENTO EN K PUNTOS (K-POINT Crossover)

En este proceso, se eligen  $k$  puntos aleatorios a lo largo de los cromosomas de ambos padres. Luego, se intercambian los fragmentos de cromosomas entre los puntos de cruce seleccionados, creando así dos descendientes. Este método permite una mayor diversidad en la descendencia, ya que no se limita a un único punto de intercambio.



## CRUZAMIENTO EN K PUNTOS (K-POINT Crossover)



## CRUZAMIENTO EN K PUNTOS (K-POINT CROSSOVER)

### **Ventajas de la Cruza en k Puntos sobre cruza en uno y dos puntos:**

**1.Mayor Diversidad Genética:** La cruza en k puntos tiende a generar descendencia con una mayor diversidad genética en comparación con la cruza en uno o dos puntos. Esto puede ayudar a explorar un espacio de búsqueda más amplio.

### **Desventajas de la Cruza en k Puntos sobre cruza en uno y dos puntos:**

**1.Mayor Riesgo de Pérdida de Información Importante:** Si se eligen demasiados puntos de cruza, existe un riesgo de que se pierda información genética crucial, ya que se fragmenta el material genético de los padres en exceso.

**2.Posible Aumento en el Tiempo de Convergencia:** La mayor diversidad genética puede llevar a una convergencia más lenta del algoritmo, ya que la población puede tardar más en enfocarse en soluciones óptimas.

## EJERCICIO

Realice la siguiente crucea en 3 puntos: Considere los puntos de crucea en 1, 5 y 6

Padre 1	1	0	1	0	1	0	0
---------	---	---	---	---	---	---	---

Padre 2	0	1	1	0	0	1	1
---------	---	---	---	---	---	---	---

Hijo 1							
--------	--	--	--	--	--	--	--

Hijo 2							
--------	--	--	--	--	--	--	--

## CRUZAMIENTO UNIFORME (UNIFORM CROSSOVER)

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 cruces anteriores. A fin de evitar un efecto excesivamente disruptivo, suele usarse con  $P_c = 0.5$ .

Se construye un vector máscara de valores 0 y 1 aleatorios.

Para  $i=1$  hasta Nbits

Si máscara( $i$ )=0

Hijo1( $i$ )=Padre1( $i$ )

Hijo2( $i$ )=Padre2( $i$ )

Sino

Hijo1( $i$ )=Padre2( $i$ )

Hijo2( $i$ )=Padre1( $i$ )

Padre 1

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

Padre 2

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

Máscara

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

Hijo 1

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

Hijo 2

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

# EJERCICIO

Realice cruce uniforme considerando los siguientes padres y máscaras

Padre 1

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

Padre 2

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

Máscara

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

Hijo 1

--	--	--	--	--	--	--

Hijo 2

--	--	--	--	--	--	--

## CRUZAMIENTO ELITISTA (ELITIST CROSSOVER)

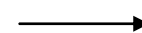
En el algoritmo genético estándar, el proceso de selección siempre es precedido por el proceso de cruzamiento. En el método EX, ambos procesos están integrados. Durante el primer paso, toda la población se baraja aleatoriamente (**Permutación**). Luego, a partir de cada par sucesivo de vectores parentales, se crean dos nuevos vectores mediante el cruzamiento. De una "familia" creada de esta manera, se seleccionan dos vectores mejores y se implementan como descendencia en la siguiente población .

La aplicación de la selección elitista de la manera tradicional, es decir, a nivel de toda la población, puede ser a menudo la razón de la convergencia prematura del algoritmo. Una selección elitista EX aplicada a nivel de la "familia" elimina este peligro según los autores.

Seleccionamos los padres por permutación, sin presión de selección (no torneo... nada)



Generamos dos hijos a partir de cada par de padres (por el cruzamiento que se defina)



De cada familia (padres con hijos) seleccionamos a los dos mejores. El remplazo por tanto ya no debe considerar elitismo.



## CRUZAMIENTO ELITISTA (ELITIST CROSSOVER)

### Ventajas del cruzamiento elitista:

- 1.Preservación de mejores individuos:** El cruzamiento elitista asegura que las mejores soluciones actuales, también conocidas como élites, se conserven en la población, incluso después del cruzamiento. Esto puede ayudar a mantener soluciones de alta calidad a lo largo de las generaciones.
- 2.Oportunidad de reproducción a todos los individuos:** No existe sesgo en la selección, por tanto todos los individuos pueden reproducirse, teniendo oportunidad de generar individuos superiores, lo cual contribuye a la exploración.
- 3.Cambio de instancia de sesgo:** El sesgo hacia los mejores individuos ocurre después del cruce, esto constituye un cambio de mecanismo en algoritmos genéticos. Todos los individuos se reproducen y después es que compiten.

### Desventajas del cruzamiento elitista:

- 1.Mayor Estancamiento en óptimos locales:** En algunos casos, el cruzamiento elitista puede llevar a que el algoritmo se estanque en óptimos locales, ya que las soluciones élites pueden dominar la población y limitar la exploración del espacio de búsqueda
- 2.Cruzamiento indefinido:** No especifica un operador de cruzamiento por lo que su efectividad estará sujeta al tipo de cruzamiento seleccionado.

## EJERCICIO

Considerando la siguiente Familia, diga quien pasa a la población de la próxima generación mediante cruzamiento por elitismo

Familia		Aptitudes (FO)
Padre 1	<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>	<div>7.2</div>
Padre 2	<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>	<div>15.5</div>
Hijo 1	<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>	<div>8.3</div>
Hijo 2	<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>	<div>9.1</div>

# CRUZAMIENTO DEL BIT ADAPTATIVO (I BIT ADAPTATION CROSSOVER)

NAVY CENTER FOR APPLIED RESEARCH AND ON ARTIFICIAL INTELLIGENCE (1992)

**Primero!!!!**¿Qué es un parámetro adaptativo? Un parámetro adaptativo se refiere a un **valor** o **configuración** que puede cambiar dinámicamente durante la ejecución del algoritmo en respuesta a ciertas condiciones específicos. Estos parámetros adaptables permiten que el sistema se ajuste automáticamente para mejorar el rendimiento en diferentes situaciones o a lo largo del tiempo.

# CRUZAMIENTO DEL BIT ADAPTATIVO (1 BIT ADAPTATION CROSSOVER)

NAVY CENTER FOR APPLIED RESEARCH AND ON ARTIFICIAL INTELLIGENCE (1992)

El último bit del vector de solución se reserva para seleccionar el tipo de cruzamiento aplicado. El valor "0" corresponde al operador de Cruzamiento Uniforme y "1" corresponde al operador de Cruzamiento de 2 Puntos.

La elección entre ellos se realiza según las reglas:

1. Si el último bit de ambos padres tiene valor de **cero**, entonces elige el **cruzamiento uniforme**.
2. Si el último bit de ambos padres tiene valor de **uno**, entonces elige el **cruzamiento en dos puntos**.
3. Si el ultimo bit tiene valor de cero para un padre y uno para el otro padres, entonces el tipo de cruzamiento se selecciona aleatoriamente.

La aplicación de cruzamiento descrito combina la elección del operador con el vector de solución. Esta elección se realiza de manera independiente para cada par de padres, por lo que corresponde a una adaptación local. Si un par de padres generaron un buen hijo mediante un operador, es muy probable que mediante la misma operación el hijo genere buenos descendientes. **La información de la operación de cruzamiento exitosa ahora se lleva en los genes**

Padre 1



Padre 2



# CRUZAMIENTO DEL BIT ADAPTATIVO (1 BIT ADAPTATION CROSSOVER)

NAVY CENTER FOR APPLIED RESEARCH AND ON ARTIFICIAL INTELLIGENCE (1992)

## Ventajas

- 1.Flexibilidad en la elección del operador:** El esquema permite adaptarse dinámicamente eligiendo entre dos operadores de cruzamiento: Uniforme y en Dos Puntos. Esto proporciona flexibilidad para ajustar el comportamiento del algoritmo genético según las características específicas del problema.
- 2.Local adaptación independiente para cada par de padres:** La elección del operador se realiza de manera independiente para cada par de padres, lo que significa que diferentes parejas pueden utilizar diferentes tipos de cruzamiento. Esto facilita la adaptación a las variaciones locales en el espacio de búsqueda.
- 3.Simplicidad y eficiencia computacional:** La implementación del esquema adaptativo es relativamente simple, ya que se basa en la manipulación de un solo bit en el vector de solución. Además, la selección aleatoria en ciertos casos introduce variabilidad sin aumentar significativamente la complejidad computacional.

## Desventajas

- 1.Mayor complejidad conceptual:** La incorporación de un esquema adaptativo basado en el último bit del vector de solución puede aumentar la complejidad conceptual del algoritmo genético.
- 2.Sensibilidad al último bit:** La elección del operador depende exclusivamente del último bit del vector de solución. Esto significa que si este bit es perturbado por la mutación u otras operaciones, la selección del operador también cambiará. Esto podría resultar en una sensibilidad excesiva a pequeñas variaciones genéticas.

# EJERCICIO

Considerando los siguientes Padres, el cruzamiento del bit adaptativo se realiza por:

Padre 1 

						1
--	--	--	--	--	--	---

Padre 2 

						1
--	--	--	--	--	--	---

Padre 1 

						0
--	--	--	--	--	--	---

Padre 2 

						0
--	--	--	--	--	--	---

Padre 1 

						1
--	--	--	--	--	--	---

Padre 2 

						0
--	--	--	--	--	--	---

## CRUZAMIENTO ACENTUADO (PUNCTUATED CROSSOVER)

Esta técnica fue propuesta 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 crua. En vez de establecer de antemano los puntos de crua, la idea es usar una cadena binaria de “marcas” para indicar la localización de los puntos de crua. La información extra que genera la crua acentuada se agrega al cromosoma de manera que el número y localizaciones de los puntos de crua 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 crua y con ‘0’ las posiciones donde no la hay. Asimismo, se suelen usar signos de admiración para facilitar la escritura de las cadenas. La figura siguiente muestra un ejemplo de la crua acentuada. El algoritmo de la crua acentuada es el siguiente:

- Copiar los bits de cada padre hacia sus hijos, de uno en uno.
- En el momento en que se encuentra un signo de admiración en cualquiera de los padres, se efectúa la crua (es decir, se invierte la procedencia de los bits en los hijos).

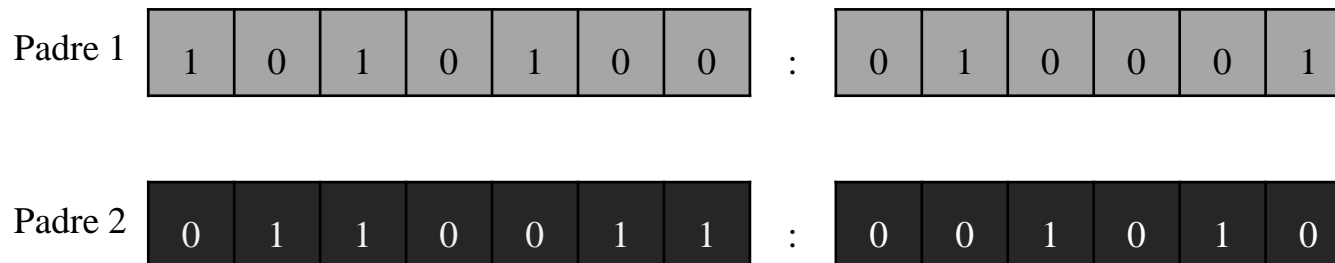
## CRUZAMIENTO ACENTUADO (PUNCTUATED CROSSOVER)

Esta técnica fue propuesta 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 establecer de antemano los puntos de cruce, la idea es usar una cadena binaria de “marcas” para indicar la localización de los puntos de cruce. Cuando la cadena tenga valor 1 significa que hay un punto de cruce.

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 también de **evolución**. Por tanto, las cadenas tendrán una longitud casi doble de su tamaño original.

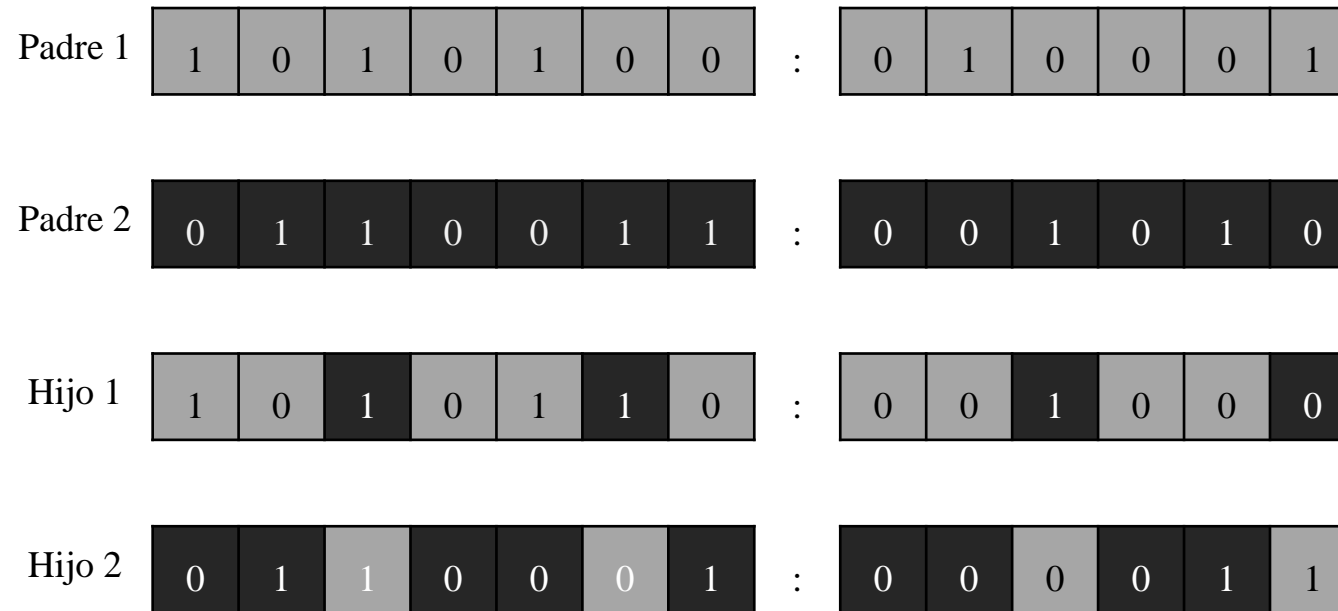
El algoritmo de la cruce acentuada es el siguiente:

- Copiar los bits de cada padre hacia sus hijos, de uno en uno.
- En el momento en que se encuentra un 1 se efectúa la cruce (es decir, se invierte la procedencia de los bits en los hijos).





# CRUZAMIENTO ACENTUADO (PUNCTUATED CROSSOVER)



# OPERADORES DE MUTACIÓN

## **Restablecimiento aleatorio**

El reinicio aleatorio es una extensión del cambio de un bit. En este caso, se define un parámetro  $n$  que indique cuantos bits se pretenden restablecer en cada mutación. Se determinan  $n$  posiciones aleatorias y se asignan valores aleatorios.

## **Mutación de inversión**

En la mutación por inversión, seleccionamos un subconjunto de alelos, y simplemente invertimos la cadena del subconjunto.



# EJERCICIOS



# ORDENE LOS OPERADORES DE ALGORITMOS GENÉTICOS

- \_\_\_ **Mutación**
- \_\_\_ **Cruzamiento**
- \_\_\_ **Sustitución de la nueva generación**
- \_\_\_ **Selección de padres**
- \_\_\_ **Generación de la población inicial**

# ENLACE LOS OPERADORES CON SU FUNDAMENTO

## **Mutación**

**Selección de los individuos que conformarán la población de la próxima generación**

## **Cruzamiento**

**Generación de descendencia con características combinadas de los padres.**

## **Sustitución de la nueva generación**

**Generación de un vecindario de individuos**

## **Selección de padres**

**Generación aleatoria de los individuos de la primera población**

**Generación de nueva información genética basada en aprendizaje**

## **Generación de la población inicial**

**Alteración aleatoria de la información genética de un descendiente**

**Competencia entre individuos que serán los progenitores de la nueva generación**

## COMPLETE EL ESPACIO EN BLANCO

La población inicial se genera de manera \_\_\_\_\_, luego esa población se \_\_\_\_\_ a reales y se \_\_\_\_\_ en la función objetivo. La selección de \_\_\_\_\_ es un mecanismo que determina los progenitores de la nueva población de individuos. El operador de \_\_\_\_\_ combina características de los padres en los descendientes. La \_\_\_\_\_ genera un cambio aleatorio en la información genética de los descendientes. La sustitución selecciona los individuos de la población de la próxima \_\_\_\_\_





# EJEMPLO TORNEO

## Ejemplo:

Considerando las aptitudes que se muestran diga el ganador de los siguientes torneos:

Individuo 1 vs. Individuo 4 \_\_\_\_\_

Individuo 2 vs Individuo 3 \_\_\_\_\_

		Aptitud
Individuo 1		10.2
Individuo 2		15.5
Individuo 3		8.3
Individuo 4		9.1

# EJEMPLO CRUZAMIENTO EN DOS PUNTOS

**Ejemplo:**

**Haga cruzamiento en dos puntos entre:**

**a) padre 1 y padre 2 con puntos de cruce en 4 y 6**

**b) padre 3 y padre 4 con puntos de cruce en 2 y 5**

Padre 1 

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

Padre 2 

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

Hijo 1 

--	--	--	--	--	--	--

Hijo 2 

--	--	--	--	--	--	--

Padre 3 

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

Padre 4 

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

Hijo 3 

--	--	--	--	--	--	--

Hijo 4 

--	--	--	--	--	--	--



# EJEMPLO CRUZAMIENTO EN UN PUNTO

## Ejemplo:

**Padre 1 cruzó con padre 2 mediante cruzamiento en un punto con punto de cruce en 4, detecta el error en cada hijo:**

Padre 1 

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

Padre 2 

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

Hijo 1 

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

Hijo 2 

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

# EJEMPLO CRUZAMIENTO EN UN PUNTO

## Ejemplo:

Ya se encuentra generada la población de la siguiente generación, sin embargo, usaremos elitismo. Qué individuo de la población anterior pasa a la siguiente generación considerando las aptitudes:

Población anterior								Aptitudes (FO)	
Individuo 1								7.1	
Individuo 2								7	
Individuo 3								6.9	
Individuo 4								9.1	

## TAREA 2. FUNCIÓN DE RASTRIGIN

**Hacer una comparación de desempeño entre dos variantes de Algoritmos genéticos en la solución de la función de Rastrigin.**

**Variante 1: Algoritmos genético con (i) selección por torneo (ii) cruzamiento en dos puntos (iii) sustitución extintiva con elitismo (la variante básica)**

**Variante 2: Algoritmo genético con (i) Cruzamiento elitista (recuerde que en este cruzamiento todos los miembros de la población son seleccionados como padres, cruzan mediante algún operador de cruzamiento (utilicen dos puntos), luego pasan a la próxima generación los mejores entre padres e hijos). Noten que queda definido el tipo de selección, de cruzamiento y de sustitución)**

## TAREA 2. FUNCIÓN DE RASTRIGIN

Ejecución 1 (200 evaluaciones) -----mejor solución encontrada  
Ejecución 2 (200 evaluaciones) -----mejor solución encontrada  
Ejecución 3 (200 evaluaciones) -----mejor solución encontrada  
.  
.  
.  
Ejecución 10 (200 evaluaciones) -----mejor solución encontrada



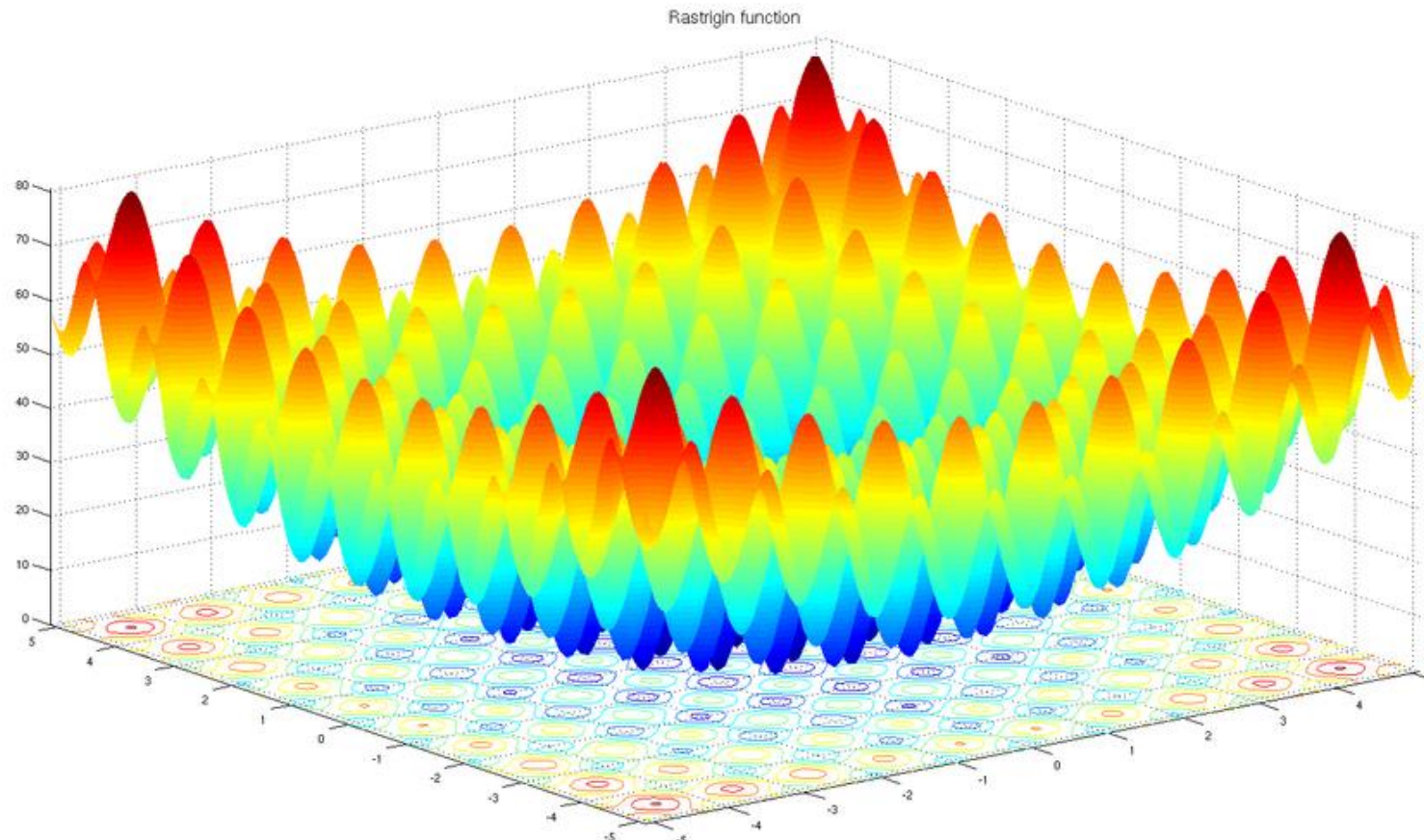
Indicador	Variante 1	Variante 2
Mejor		
Media		
Peor		
Desv. estandar		

### Procedimiento:

Se ejecuta 10 veces la Variante 1 (en cada ejecución se corren 200 generaciones). En cada ejecución debe obtenerse la mejor solución. De las 10 soluciones obtener la mejor, la mediana, y la peor.

Se ejecuta 10 veces la Variante 2 (en cada ejecución se corren 200 generaciones). En cada ejecución debe obtenerse la mejor solución. De las 10 soluciones obtener la mejor, la mediana, y la peor.

**Nota:** Usar precisión de 5 en la codificación



**FUNCIÓN RASTRIGIN:**  $f(x, y) = 20 + (x^2 - 10 \cos(2\pi x)) + (y^2 - 10 \cos(2\pi y))$

s. a  $x, y \in [-5.12, 5.12]$

Solución:  $[0, 0; 0]$

**Procedimiento AG\_binario\_cruzamiento\_elitista()**

1. % Configuración de parámetros
2. % Definir límites según el problema
3. % Generar población inicial
4. % Evaluar población inicial (VFOp)  
% Ciclo principal  
Mientras iter <= NG
5. % Selección sin sesgo
6. seleccionados = aleatorio\_sin\_reemplazo(Np);
7. Padres = p(seleccionados, :);
8. VFOPadres = VFOp(seleccionados);
9. % Cruzamiento (cruce en dos puntos)
10. % Mutación (por bit flip)
11. % Evaluación de hijos
12. hcont = decodificacion(Hijos, Nvar, Np, Nbit, pr, lb, ub);
13. VFOHijos = FO(hcont);

14. % Sustitución (pasan los dos mejores individuos de la familia)
15. p=[];
16. VFOp=[];
17. for i=1:2:Np
  - %Familia
  - Familia=[Padres(i:i+1,:);Hijos(i:i+1,:)];
  - %ordenamiento de la familia de mejor a peor
  - [aptitud,idx]=sort([VFOPadres(i:i+1);VFOHijos(i:i+1)]);
  - %ganadores de la familia conforman nueva población
  - p(i:i+1,:)=Familia(idx(1:2),:); %individuos
  - VFOp(i:i+1,1)= aptitud(1:2); %aptitudes
18. end
19. % Actualizar el mejor individuo de todos
20. Fin Mientras

## TAREA 2. FUNCIÓN DE RASTRIGIN

**Entrega:** un reporte desde se declare el problema, se describan brevemente las variantes de **AG**, y se haga una breve discusión sobre los resultados (media pagina) (a que variante le fue mejor, influyó la existencia de mínimos locales en el desempeño, la selección de mejores individuos por familias ayudó, por qué cree una variante tuvo mejor desempeño que la otra...)

**Formato:** un comprimido con nombres y apellidos, dentro una carpeta con nombres y apellidos, y dentro Documento PDF y códigos

**Algunos parámetros:** Precisión 5  
Np=100