

# Maestría y Doctorado en Ciencia de la Computación

Inteligencia Artificial

Algoritmos Genéticos

Dr. Edward Hinojosa Cárdenas  
ehinojosa@unsa.edu.pe  
22 de Agosto del 2020



UNSA

UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA

# Índice



Objetivos del Curso

Evolución Natural

Computación Evolutiva

Optimización Combinatoria

Algoritmos Genéticos

Algoritmos Genéticos - Representación Real

Algoritmos Genéticos - Representación por Permutación

# Objetivos del Curso



- ▶ **Conocer, comprender e implementar algoritmos evolutivos para resolver problemas complejos.**
- ▶ Conocer, comprender e implementar algoritmos de inteligencia de enjambre para resolver problemas complejos.
- ▶ Conocer, comprender e implementar algoritmos inmunes artificiales para resolver problemas complejos.
- ▶ Conocer, comprender e implementar sistemas basados en lógica difusa para resolver problemas complejos.



# Origen de las Especies

- Charles Robert Darwin, naturalista inglés, presentó en 1859 una teoría para explicar porque de la existencia de una variedad (diversidad) tan grande de seres vivos (organismos) en la naturaleza [3].

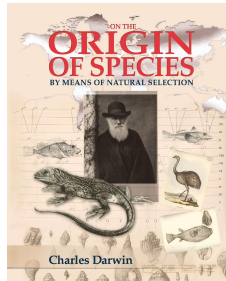


Figure: El Origen de las Especies

# Origen de las Especies



- ▶ Su teoría presenta la “Selección Natural” como el principal mecanismo para la manutención de las variaciones favorables a la supervivencia y reproducción de un organismo en su ambiente.
- ▶ En resumen, la teoría Darwiniana propone que la evolución y el resultado de una (o más) poblaciones de individuos está sujeta a los siguientes procesos:
  - ▶ Reproducción con herencia.
  - ▶ Variación.
  - ▶ Selección natural.

# Origen de las Especies



- ▶ Además, resume en los siguiente argumentos la defensa a la teoría de la selección natural (entre otros):
  - ▶ Perfección gradual de cualquier órgano o instinto existen o podrían haber existido;
  - ▶ Todos los órganos e instintos son variables;
  - ▶ Existe una lucha por la supervivencia que lleva la preservación de las variaciones favorables a la vida y reproducción de un organismo.

# Origen de las Especies

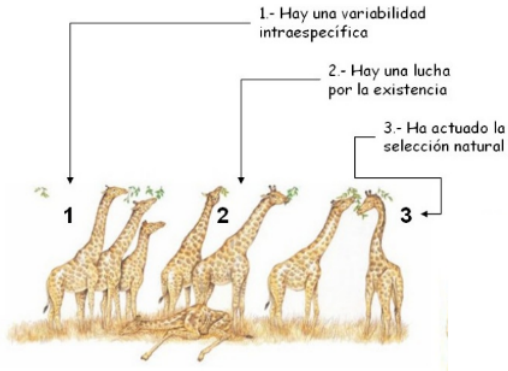


Figure: Selección Natural

# Origen de las Especies



- ▶ En la evolución biológica, las especies se seleccionan positiva o negativamente en función de su éxito relativo en la supervivencia y reproducción en el medio ambiente.
- ▶ La supervivencia diferencial y la generación de variedades durante la reproducción son el motor de la evolución.



# Origen de las Especies

- Este proceso puede ser considerado como un proceso de "búsqueda", donde el objetivo es mejorar continuamente la calidad de los individuos en la población.

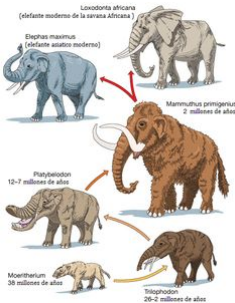


Figure: Búsqueda Evolutiva

# Computación Evolutiva



- ▶ Los procesos evolutivos, que sirven de inspiración a la Computación Evolutiva, se resumen en los siguientes:
  - ▶ Una población de entidades,
  - ▶ Mecanismos de selección,
  - ▶ Adaptación, y
  - ▶ La generación de la variedad.
- ▶ Los procesos mencionados se realizan cíclicamente como muestra la siguiente figura:

# Computación Evolutiva

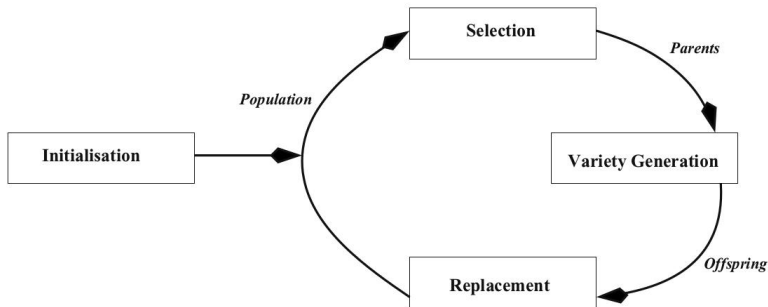


Figure: Ciclo Evolutivo



# Algoritmos Evolutivos

- El Ciclo Evolutivo sirvió de inspiración para definir los Algoritmos Evolutivos:

## Evolutionary Algorithm

Initialise the population of candidate solutions;

**repeat**

    Select individuals (parents) for breeding from the current population;

    Generate new individuals (offspring) from these parents;

    Replace some or all of the current population with the newly generated individuals;

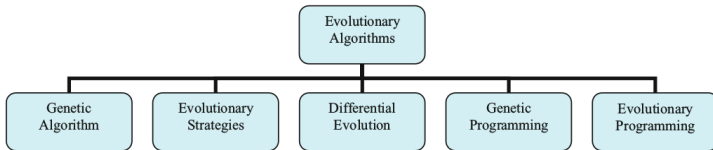
**until** *terminating condition*;

Figure: Algoritmo Evolutivo

# Algoritmos Evolutivos



- Existen varias formas de operativizar cada uno de los pasos de este algoritmo; en consecuencia, existen varios algoritmos evolutivos diferentes, pero relacionados.



# Algoritmos Evolutivos



- ▶ Aunque las ramas anteriores son distintas, y cada una de ellas tenía sus propios autores, las líneas entre todos estos enfoques inspirados en la evolución se están desdibujando con representaciones y estrategias que se utilizan indistintamente entre los distintos algoritmos.
- ▶ Como tal, hoy en día es común utilizar el término algoritmo evolutivo para abarcar todos los enfoques anteriores.

# Optimización Combinatoria



- ▶ Antes de comenzar con nuestro primer Algoritmo Evolutivo, veremos resumidamente el concepto de Optimización Combinatoria.
- ▶ Un problema de Optimización Combinatoria  $P = (S, f)$  puede ser definido como [2]:
  - ▶ Un conjunto de variables  $X = \{x_1, x_2, \dots, x_n\}$ ;
  - ▶ Variables de Dominio  $D_1, D_2, \dots, D_n$ ;
  - ▶ Restricciones entre las variables;
  - ▶ Una función objetivo  $f$  a ser minimizada (o maximizada) donde  $f : D_1 \times D_2 \times \dots \times D_n \rightarrow \mathbb{R}^+$ ;
  - ▶  $S$  es el conjunto de todas las combinaciones que satisfagan todas las restricciones  $S = \{s = \{(x_1, v_1), \dots, (x_n, v_n)\} | v_i \in D_i\}$



# Optimización Combinatoria

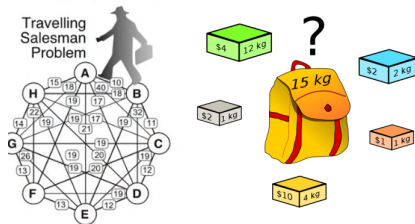
- La siguiente tabla muestra las tasas de crecimiento de funciones, según su complejidad, medidas en nanosegundos [1]:

$n / f(n)$	$\log(n)$	$n$	$n \log(n)$	$n^2$	$2^n$	$n!$
10	0.003 $\mu$ s	0.01 $\mu$ s	0.033 $\mu$ s	0.1 $\mu$ s	1 $\mu$ s	3.63 ms
20	0.004 $\mu$ s	0.02 $\mu$ s	0.086 $\mu$ s	0.4 $\mu$ s	1ms	77.1 years
30	0.005 $\mu$ s	0.03 $\mu$ s	0.147 $\mu$ s	0.9 $\mu$ s	1s	$8.4 \times 10^{15}$ years
40	0.005 $\mu$ s	0.04 $\mu$ s	0.213 $\mu$ s	1.6 $\mu$ s	18.3 min	
50	0.006 $\mu$ s	0.05 $\mu$ s	0.282 $\mu$ s	2.5 $\mu$ s	13 days.	
100	0.007 $\mu$ s	0.1 $\mu$ s	0.644 $\mu$ s	10 $\mu$ s	$4 \times 10^{13}$ years	
1,000	0.010 $\mu$ s	1 $\mu$ s	9.966 $\mu$ s	1ms		
10,000	0.013 $\mu$ s	10 $\mu$ s	130 $\mu$ s	100ms		
100,000	0.017 $\mu$ s	0.10 ms	1.67 ms	10s		
1,000,000	0.020 $\mu$ s	1ms	19.93 ms	16.7 min		
10,000,000	0.023 $\mu$ s	0.01 s	0.23 s	1.16 days		
100,000,000	0.027 $\mu$ s	0.1 s	2.66 s	115.7 days		
1,000,000,000	0.030 $\mu$ s	1s	29.90 s	31.7 years		



# Optimización Combinatoria

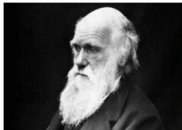
- ▶ Algunos problemas de optimización combinatoria NP-Completo son:
  - ▶ El problema del viajante.
  - ▶ El problema de la mochila.
  - ▶ El problema del parqueo de vehículos.
  - ▶ ...





# Algoritmos Genéticos

- Los Algoritmos Genéticos fueron introducidos por John Holland en los años 1960, definiendolos como "planes reproductivos", pero se hicieron populares bajo el nombre de Algoritmos Genéticos tras la publicación de su libro en 1975 [4].



**Charles Robert Darwin** (1809–1882), father of Evolution, who developed the theory of evolution whereby changes in species are driven, over time, by natural and sexual selection.



**Prof. John Holland**, father of genetic algorithms and pioneer in complex systems invented genetic algorithms in the 1960's, proving a number of theorems about them before they had been so named

Figure: Charles R. Darwin y Jhon Holland

# Algoritmos Genéticos



- ▶ Se basan en la teoría de la evolución natural de las especies.
- ▶ Los individuos son mejorados a través de un proceso evolutivo y los mejor adaptados al medio sobreviven.
- ▶ Sufren influencia del medio en que viven.
- ▶ Se reproducen y sufren alteraciones genéticas.

# Algoritmos Genéticos



- Podemos realizar una analogía entre los conceptos y entidades dentro de la Evolución Natural y los Algoritmos Genéticos:

<b>Evolución Natural</b>	<b>Algoritmos Genéticos</b>
Individuo	Solución
Genotipo (Cromosoma)	Representación
Reproducción	Cruzamiento
Mutación	Mutación
Población	Conjunto de Soluciones
Generaciones	Ciclos del Algoritmo

# Algoritmos Genéticos



20

- A continuación se muestra el Pseudocódigo de un Algoritmo Genético:

## Canonical Genetic Algorithm

Determine how the solution is to be encoded as a genotype and define the fitness function;  
Create an initial population of genotypes;  
Decode each genotype into a solution and calculate the fitness of each of the  $n$  solution candidates in the population;

### **repeat**

Select  $n$  members from the current population of encodings (the *parents*) in order to create a mating pool;

### **repeat**

Select two parents randomly from the mating pool;

With probability  $p_{\text{cross}}$ , perform a crossover process on the encodings of the selected parent solutions, to produce two new (*child*) solutions;  
Otherwise, crossover is not performed and the two children are simply copies of their parents;

With probability  $p_{\text{mut}}$ , apply a mutation process to each element of the encodings of the two child solutions;

**until**  $n$  new child solutions have been created;

Replace the old population with the newly created one (this constitutes a generation);

**until** terminating condition;



# Estructura del Cromosoma

- Existen diferentes formas de representar un cromosoma. Entre las más conocidas tenemos la representación binaria y la representación numérica:

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

Representación Binaria

3	1	2	5	0	1	1	2	3	0	2	2	3	4	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Representación Decimal

# Función de Actitud



- ▶ La función de aptitud (o fitness) atribuye un valor a cada individuo, determinando su potencial.
- ▶ Es constituida por la información del problema que lleva a una función objetivo a ser minimizada o maximizada.

# Métodos de Selección



- ▶ Consiste en definir la forma de como se seleccionarán los individuos que deben sufrir operaciones genéticas.
- ▶ Los métodos más utilizados son:
  - ▶ Selección por Ruleta.
  - ▶ Selección por Torneo.





# Métodos de Selección por Ruleta

- ▶ En el método de selección por ruleta el intervalo de selección es inversamente proporcional al valor de fitness (en el caso de minimización), por tanto los mejores individuos tienen mayor oportunidad de ser seleccionados.
- ▶ En el caso de maximización el intervalo de selección es proporcional al valor de fitness, como muestra la siguiente figura:

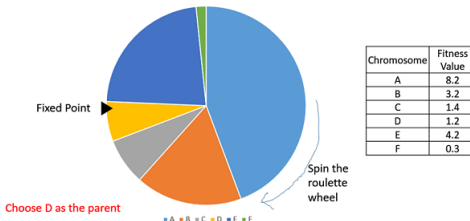
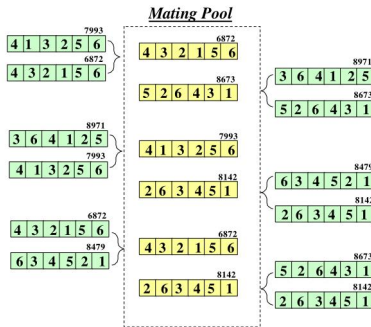


Figure: Método de Selección por Ruleta



# Métodos de Selección por Torneo

- En este método,  $N$  individuos son escogidos aleatoriamente con una misma probabilidad, siendo el mejor de los  $N$  individuos seleccionado para sufrir las operaciones genéticas.



# Operadores Genéticos - Cruzamiento



- ▶ El operador de cruce es análogo a la reproducción o al cruce biológico. En esto, se selecciona más de un padre y se producen uno o más descendientes utilizando el material genético de los padres.
- ▶ El cruzamiento generalmente se aplica generalmente en un GA con una alta probabilidad  $p_c$ .
- ▶ Formas de cruzamiento:
  - ▶ Cruzamiento de un punto de corte.
  - ▶ Cruzamiento de  $n$  puntos de corte.
  - ▶ Cruzamiento uniforme.



# Cruzamiento de un punto de corte

- Un punto de cruzamiento es escogido.
- Para el primer descendiente, los genes del cromosoma desde el comienzo hasta ese punto son copiados del primer padre y el resto copiado del otro padre. De forma contraria para el segundo descendiente.

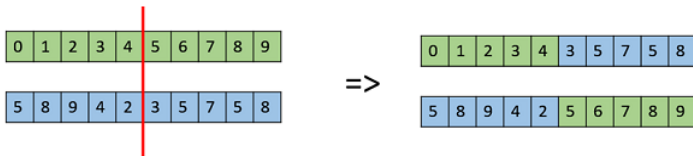


Figure: Cruzamiento de un punto de corte



# Cruzamiento de 2 puntos de corte

- Consiste en definir dos puntos e intercambiar genes entre ellos, tanto para el primer y segundo descendiente.

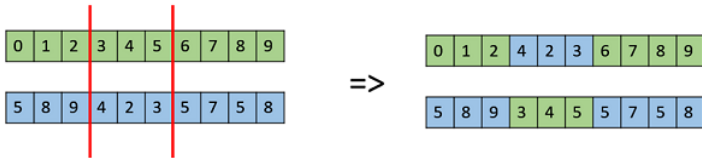


Figure: Cruzamiento de 2 puntos de corte



# Cruzamiento Uniforme

- Mediante una máscara se determina que genes son heredados de los padres para cada descendiente. Un valor de 1 en la máscara indica que el gen correspondiente del primer padre es heredado por el primer descendiente y el gen correspondiente al segundo padre es heredado por el segundo descendiente. Para el valor 0 ocurre lo inverso.

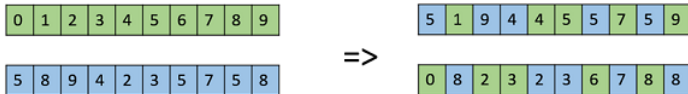


Figure: Cruzamiento Uniforme - Máscara: 0100110101

# Operadores Genéticos - Mutación



- ▶ La mutación se puede definir como un pequeño ajuste aleatorio en el cromosoma, para obtener una nueva solución. Se utiliza para mantener e introducir diversidad en la población genética y generalmente se aplica con baja probabilidad  $p_m$ .
- ▶ Formas de mutación:
  - ▶ Bit Flip Mutation
  - ▶ Swap Mutation.
  - ▶ Scramble Mutation.
  - ▶ Inversion Mutation.



# Operadores Genéticos - Mutación

Bit Flip Mutation

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

=>

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

Swap Mutation

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

=>

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

Scramble Mutation

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

=>

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

Inversion Mutation

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

=>

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

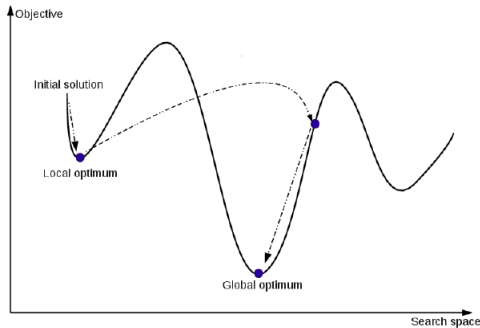
Figure: Operadores de Mutación





# Rol de los Operadores Genéticos

- Todos los operadores genéticos permiten realizar modificaciones guiadas dentro del espacio de búsqueda que podrían llevarnos a obtener el óptimo global (**no se garantiza**).



# Parámetros



- ▶ El desempeño de un Algoritmo Genético es influenciado por la definición de los parámetros que son utilizados. Es objetivo de varias investigaciones.
- ▶ Los principales parámetros son:
  - ▶ Tamaño de la población.
  - ▶ Tasa de cruzamiento (Recomendado: 70% al 90%).
  - ▶ Tasa de mutación (Recomendado: 0,5% al 5%).
  - ▶ Criterios de parada.

# Parámetros



- ▶ No existe un criterio exacto para detener la ejecución de un Algoritmo Genético.
- ▶ Los criterios más utilizados son:
  - ▶ Número de generaciones.
  - ▶ Tiempo de ejecución.
  - ▶ Número de validaciones.
  - ▶ Llegar a un valor en la función objetivo.
  - ▶ Convergencia de la población: Cuando no ocurre una mejora significativa en el cromosoma de mayor aptitud para una determinado número de generaciones. Cuando más del 90% de los individuos o cromosomas tienen los mismos genes.

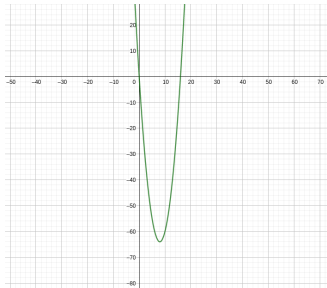


# Ejemplo de un Algoritmo Genético

- Utilice un Algoritmo Genético con codificación binaria para minimizar la siguiente función:

$$f(x) = x^2 - 16x$$

$$0 \leq x \leq 31$$



# Ejemplo de un Algoritmo Genético



- ▶ Considere los siguientes parámetros:
  - ▶ Representación Binaria
  - ▶ Cantidad de Individuos: 8
  - ▶ Cantidad de Genes por Individuo: 5
  - ▶ Selección por torneo (2)
  - ▶ Probabilidad de Cruzamiento: 0.7
  - ▶ Cruzamiento de un Punto (Punto 3)
  - ▶ Probabilidad de Mutación: 0.05
  - ▶ Mutación Bit Flip
  - ▶ Cantidad de Iteraciones: 500

# Representación Real



- Para entender mejor esta sección, consideraremos el siguiente problema para maximizar la siguiente función con dos parámetros  $x$  y  $y$  reales.

$$f(x, y) = (x + 2y - 7)^2 + (2x + y - 5)^2$$

$$-10.0 \leq x \leq 10.0$$

$$-10.0 \leq y \leq 10.0$$

# Representación Real



38

## Binario

$$s_1 = 01101001001001101000001000111000100001110010$$

Paso 1: Dividir

$$0110100100100110100000 \text{ e } 1000111000100001110010$$

Paso 2: Convertir

$$1722784 \text{ e } 2328690.$$

Paso 3: Mapear

$$x_1 = \frac{1722784}{2^{22} - 1} [100 - (-100)] + 100 = -17,851$$

$$y_1 = \frac{2328690}{2^{22} - 1} [100 - (-100)] + 100 = 11,041$$

# Representación Real



- ▶ Recordemos, para cada punto decimal adicionado en la presentación, es necesario adicionar 3,3 bit en la cadena. En caso, 8 números decimales sean necesarios,  $8 * 3.3 \approx 27$  bits utilizados para cada parámetro.
- ▶ Cuando existen muchos parámetros, se obtiene largas cadenas de bits que pueden hacer que el algoritmos converja lentamente.
- ▶ Además, no existe uniformidad en los operadores, por ejemplo, mutación en los primeros bits tiene un efecto diferente a los últimos.



# Representación Real



- ▶ La representación real (con punto flotante) genera cromosomas de menor tamaño y es comprendida mejor por las personas que las cadenas de bits.
- ▶ En el problema anterior, el cromosoma sería representado por un vector de dos números con punto flotante. Por ejemplo,  $[-17.5963, 11.2449]$ .
- ▶ Otra ventaja de la representación real es la facilidad de utilizar nuevos operadores genéticos (lo veremos a continuación).



# Representación Real Cruzamiento

- ▶ Veremos un conjunto de operadores aritméticos de cruzamiento para la representación real.
- ▶ Considere la siguiente notación, los cromosomas padres ( $p_1$  y  $p_2$ ) y el cromosoma hijo ( $c$ ) serán representados por:

$$p_1 = (p_{11}, p_{12}, \dots, p_{1g})$$

$$p_2 = (p_{21}, p_{22}, \dots, p_{2g})$$

$$c = (c_1, c_2, \dots, c_g)$$

$$p_{ij} \in \mathbb{R}, c_i \in \mathbb{R}$$

# Representación Real Cruzamiento



- ▶ Los operadores convencionales son resultado de las adaptaciones de los operadores utilizados para representación binaria.
- ▶ Los operadores convencionales (por ejemplo: cruzamiento de  $n$  puntos y uniforme) funcionan en la representación real, ellos básicamente cambian la posición de los genes, pero sus resultados generalmente son repetitivos, por ello se recomienda utilizar operadores aritméticos.
- ▶ Los operadores aritmético realizan algún tipo de combinación lineal entre los cromosomas padres.

# Representación Real

## Cruzamiento Medio



- Dado dos cromosomas  $p_1$  y  $p_2$ , es producido un cromosoma  $c$  de la siguiente forma:

$$(c_1, c_2, \dots, c_g) = \left( \frac{p_{11} + p_{21}}{2}, \frac{p_{12} + p_{22}}{2}, \dots, \frac{p_{1g} + p_{2g}}{2} \right)$$

# Representación Real

## Cruzamiento Medio Geométrico



- Dado dos cromosomas  $p_1$  y  $p_2$ , es producido un cromosoma  $c$  de la siguiente forma:

$$(c_1, c_2, \dots, c_g) = (\sqrt{p_{11} \times p_{21}}, \sqrt{p_{12} \times p_{22}}, \dots, \sqrt{p_{1g} \times p_{2g}})$$



# Representación Real

## Cruzamiento BLX- $\alpha$

- ▶ El cruzamiento medio tiende a llevar los genes para el medio del intervalo permitido, lo que puede causar pérdida de diversidad. Ello puede ser mejorado con el cruzamiento BLX- $\alpha$ .
- ▶ Cruzamiento BLX- $\alpha$  o cruzamiento mixtura (del inglés BLeNd crossover).
- ▶ Dado dos cromosomas  $p_1$  y  $p_2$ , es producido un cromosoma  $c$  de la siguiente forma:

$$c_i = p_{1i} + \beta \times (p_{2i} - p_{1i})$$

$$\beta \in U(-\alpha, 1 + \alpha)$$



# Representación Real

## Cruzamiento BLX- $\alpha$

- El BLX- $\alpha$  es mostrado en la siguiente figura, en la cual es escogido un único valor de  $\beta$  para todos los pares de genes:

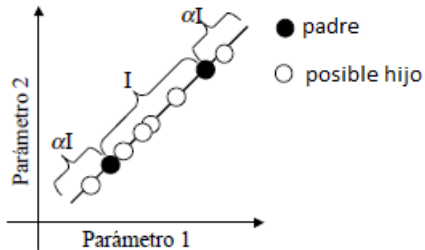


Figure: Cruzamiento BLX- $\alpha$

# Representación Real

## Cruzamiento BLX- $\alpha$



- ▶ Cuando  $\alpha = 0$  el hijo se sitúa sobre el intervalo  $I$  entre los dos puntos que representan sus padres. El parámetro  $\alpha$  extiende el intervalo  $I$ .
- ▶ Por ejemplo, si  $\alpha = 0.5$ , el intervalo  $I$  es extendido  $\alpha * I$  en ambos lados.
- ▶ El BLX-0.5 balancea la tendencia de generar hijos próximos al centro del intervalo  $I$  evitando la pérdida de diversidad.
- ▶ El BLX- $\alpha$  ha sido usado en varios trabajos dando buenos resultados y puede ser el operador más utilizado para la representación real.





# Representación Real

## Cruzamiento BLX- $\alpha$

$$p_1 = (30,173; 85,342)$$

$$p_2 = (75,989; 10,162)$$

$$\text{BLX- } 0,5 \quad \beta = 1,262 \quad \beta \in U(-0,5; 1,5))$$

$$c_1 = 30,173 + 1,262(75,989 - 30,173) = 87,993$$

$$c_2 = 85,342 + 1,262(10,162 - 85,342) = -9,535$$

Hijo Generado:

$$c = (87,993; -9,535)$$

# Representación Real

## Cruzamiento BLX- $\alpha$



- ▶ Si el hijo  $c$  no fuera factible, entonces se genera otro hijo con un nuevo  $\beta$ . El proceso es repetido hasta obtener un hijo factible.
- ▶ Se nota que este ejemplo, fue utilizado en  $\beta$  para todos los genes. Alternativamente, se puede usar un  $\beta$  diferente para cada par de genes. En ese caso, un posible hijo se sitúa en un lugar de un área limitada por un rectángulo.



# Representación Real

## Cruzamiento BLX- $\alpha$

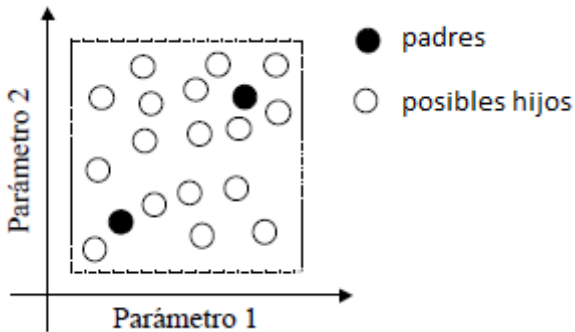


Figure: Cruzamiento BLX- $\alpha$  con  $\beta$  variable

# Representación Real

## Cruzamiento Lineal



- Dado dos cromosomas  $p_1$  y  $p_2$  se obtienen tres hijos  $c_x, c_y, c_z$  de la siguiente forma:

$$c_{xi} = 0.5 \times p_{1i} + 0.5 \times p_{2i}$$

$$c_{yi} = 1.5 \times p_{1i} - 0.5 \times p_{2i}$$

$$c_{zi} = -0.5 \times p_{1i} + 1.5 \times p_{2i}$$



# Representación Real

## Cruzamiento Aritmético

- ▶ Se diferencia porque no extrapola el intervalo  $p_1$  y  $p_2$ .
- ▶ Dado dos cromosomas  $p_1$  y  $p_2$  se obtienen tres hijos  $c_x$ ,  $c_y$  de la siguiente forma:

$$c_{xi} = \beta \times p_{1i} + (1 - \beta) \times p_{2i}$$

$$c_{yi} = (1 - \beta) \times p_{1i} + \beta \times p_{2i}$$

$$\beta \in U(0, 1)$$



# Representación Real

## Cruzamiento Heurístico

- ▶ Realiza una extrapolación lineal entre los padres usando la información de la aptitud.
- ▶ Dado dos cromosomas  $p_1$  y  $p_2$  en que  $p_1$  es mejor que  $p_2$  en términos de aptitud. Entonces el cromosoma  $c$  se obtiene de la siguiente forma:

$$c_i = p_{1i} + r \times (p_{2i} - p_{1i})$$

$$f(p_1) > f(p_2)$$

$$r \in U(0, 1)$$

- ▶ Caso el cruzamiento produce un hijo no factible, se generado otro número aleatorio  $r$ , y se obtiene un nuevo hijo. Si en  $t$  tentativas el hijo continua no factible, entonces el cruzamiento se detiene sin producir hijos.



# Representación Real

## Mutación Uniforme

- Es la simple sustitución de un gen por un número aleatorio. Es decir, dado un cromosoma  $p_1$  con el  $j$ -ésimo gen seleccionado para mutación, es producido un cromosoma  $c_1$  de la siguiente forma:

$$c_{1i} = \begin{cases} U(a_{1i}, b_{1i}) & i = j \\ p_{1i} & \text{caso contrario} \end{cases}$$



# Representación Real

## Mutación Creep

- Adiciona al gen un número aleatorio. Alternativamente, la mutación creep puede ser realizada mutliplicando el gen por un número aleatorio cercano a 1.

$$c_{1i} = \begin{cases} p_{1i} + \text{random} & i = j \\ p_{1i} & \text{caso contrario} \end{cases}$$

$$c_{1i} = \begin{cases} p_{1i} * U(0, 1) & i = j \\ p_{1i} & \text{caso contrario} \end{cases}$$



# Representación Real

## Mutación Límite



- Es la sustitución del gen por uno de los límites del intervalo permitido para ese gen.

$$c_{1i} = \begin{cases} a_{1i} & r < 0.5 \text{ y } i = j \\ b_{1i} & r \geq 0.5 \text{ y } i = j \\ p_{1i} & \text{caso contrario} \end{cases}$$



# Representación Real

## Ejemplo de un Algoritmo Genético

- Utilice un Algoritmo Genético para minimizar la siguiente función:

$$f(x, y) = (x + 2y - 7)^2 + (2x + y - 5)^2$$

$$-10.0 \leq x \leq 10.0$$

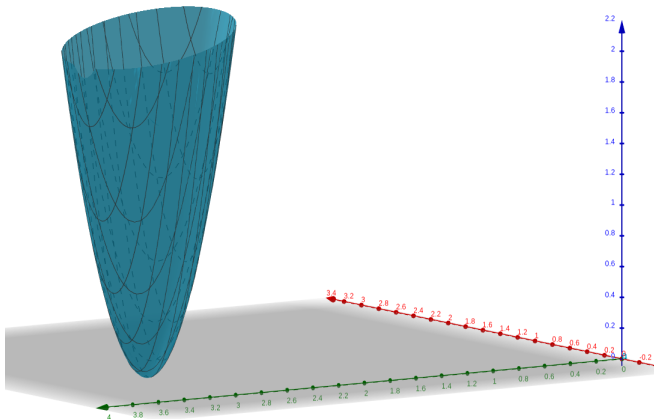
$$-10.0 \leq y \leq 10.0$$

- Otras Funciones de Optimización - [Link](#)



# Representación Real

## Ejemplo de un Algoritmo Genético





# Representación Real

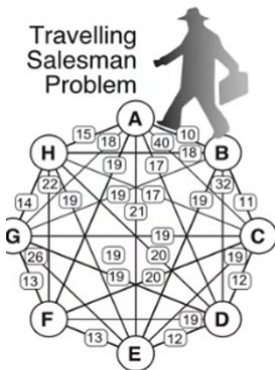
## Ejemplo de un Algoritmo Genético

- ▶ Considere los siguientes parámetros:
  - ▶ Representación Real (Considere hasta 5 decimales)
  - ▶ Cantidad de Individuos: 16
  - ▶ Cantidad de Genes por Individuo: 2
  - ▶ Selección por torneo (2)
  - ▶ Probabilidad de Cruzamiento: 0.7
  - ▶ Cruzamiento BLX- $\alpha$ ,  $\alpha = 0.5$
  - ▶ Probabilidad de Mutación: 0.05
  - ▶ Mutación Uniforme
  - ▶ Cantidad de Iteraciones: 5000

# Representación por Permutación

## El Problema del Vendedor Viajero

- Travelling Salesman Problem (TSP).





# Representación por Permutación

## El Problema del Vendedor Viajero

- ▶ El objetivo es mostrar como aplicar los AGs en problemas de naturaleza diferente de aquellos anteriormente descritos (optimización de funciones numéricas).
- ▶ Por ejemplo, problemas que dependen del orden con las acciones o tareas a ejecutar. Tales problemas han sido exhaustivamente estudiados en la literatura de AGs y se han propuesto varios operadores genéticos específicos.

# Representación por Permutación

## El Problema del Vendedor Viajero



- ▶ Este problema es NP-complejo, lo que significa que los algoritmos conocidos para encontrar su solución exacta son intratables por el computador (es decir, requieren una cantidad de tiempo computacional que aumentan exponencialmente con el tamaño del problema).
- ▶ Problemas NP-complejo han sido resueltos con algoritmos heurísticos y metaheurísticos que no garantizan encontrar la solución óptima, pero reducen el tiempo de procesamiento.

# Representación por Permutación

## El Problema del Vendedor Viajero

- En el TSP con  $n$  ciudades, el número total de caminos posible es  $n!$ .
- Imagínense si tenemos 24 ciudades, (o 24 los departamentos del Perú).



Figure: Departamentos del Perú



# Representación por Permutación

## Permutación



- ▶ Una permutación de  $n$  elementos es una secuencia de  $n$  elementos en que ningún elemento es repetido.
- ▶ Por ejemplo,  $(A, B, C)$  y  $(C, A, B)$  son ejemplos de dos permutaciones del conjunto  $(A, B, C)$ .

# Algoritmos Genéticos

## Representación por Permutación



- ▶ Veremos algunos operadores de cruzamiento para permutaciones.
- ▶ La mutación es relativamente simple.
- ▶ Los otros pasos son similares a los visto con anterioridad.



# Representación por Permutación

## Cruzamiento OBX

- ▶ Cruzamiento OBX (Order - Based Crossover).
- ▶ Comienza seleccionando un conjunto de posiciones aleatoriamente (cada posición tiene una probabilidad igual a 0.5 de ser seleccionado).
- ▶ Después, se le impone a los elementos del padre 1, en las posiciones seleccionadas, el mismo orden que estos elementos presenta en el padre 2. El nuevo orden en la posiciones seleccionados del padre 1, es copiado al hijo 1.



# Representación por Permutación

## Cruzamiento OBX

- Los elementos en la posiciones no seleccionadas del padre 1, son copiados sin alteraciones para el hijo 1. El cromosoma hijo 2 es obtenido a través de un proceso similar, por ejemplo:

Padre 1 :	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>F</i>	<i>E</i>	<i>G</i>
Padre 2 :	<i>C</i>	<i>E</i>	<i>G</i>	<i>A</i>	<i>D</i>	<i>F</i>	<i>B</i>
		*		*	*		
Hijo 1 :	<i>A</i>	<i>D</i>	<i>C</i>	<i>F</i>	<i>B</i>	<i>E</i>	<i>G</i>
Hijo 2 :	<i>C</i>	<i>A</i>	<i>G</i>	<i>D</i>	<i>E</i>	<i>F</i>	<i>B</i>

Figure: Cruzamiento OBX



# Representación por Permutación

## Cruzamiento PBX

- Cruzamiento PBX (Position- Based Crossover).
- También comienza seleccionando un conjunto de posiciones aleatorias. Sin embargo, en vez de imponer un orden, imponen la posición.
- En las posiciones seleccionadas, el hijo 1 tendrá los mismos elementos que el padre 2. Los demás elementos del hijo 1 vienen del padre 1, manteniendo el mismo orden presente en el padre 1. El hijo 2 es obtenido de forma similar. Por ejemplo:

Padre 1 :	A	B	C	D	F	E	G
Padre 2 :	C	E	G	A	D	F	B
		*		*	*		
Hijo 1 :	B	E	C	A	D	F	G
Hijo 2 :	C	B	E	D	F	G	A

Figure: Cruzamiento PBX



# Representación por Permutación

## Mutación de Intercambio

- Consiste en seleccionar aleatoriamente dos elementos e intercambiarlos.

A	B	F	C	D	E	G
	*				*	
A	E	F	C	D	B	G

Figure: Mutación por Intercambio



# Representación por Permutación

## Ejemplo de un Algoritmo Genético

- Utilice un Algoritmo Genético para encontrar la ruta con menos costo (minimización) para el siguiente problema TSP:

	A	B	C	D	E	F	G	H	I	J
A	0	1	3	23	11	5	83	21	28	45
B	1	0	1	18	3	41	20	61	95	58
C	3	1	0	1	56	21	43	17	83	16
D	23	18	1	0	1	46	44	45	50	11
E	11	3	56	1	0	1	93	38	78	41
F	5	41	21	46	1	0	1	90	92	97
G	83	20	43	44	93	1	0	1	74	29
H	21	61	17	45	38	90	1	0	1	28
I	28	95	83	50	78	92	74	1	0	1
J	45	58	16	11	41	97	29	28	1	0



# Representación por Permutación

## Ejemplo de un Algoritmo Genético

- ▶ Considere los siguientes parámetros:
  - ▶ Representación por Permutación.
  - ▶ Cantidad de Individuos: 50
  - ▶ Selección por ruleta
  - ▶ Probabilidad de Cruzamiento: 0.9
  - ▶ Cruzamiento PBX.
  - ▶ Probabilidad de Mutación: 0.5
  - ▶ Mutación de Intercambio
  - ▶ Cantidad de Iteraciones: 2000



# ¡GRACIAS!





# Bibliografía I

- [1] E. Azar and M. E. Alebicto.  
*Swift Data Structure and Algorithms*.  
Packt Publishing, 2016.
- [2] C. Blum and A. Roli.  
Metaheuristics in combinatorial optimization: Overview and  
conceptual comparison.  
*ACM Comput. Surv.*, 35:268–308, 01 2001.
- [3] C. Darwin.  
*On the Origin of Species by Means of Natural Selection*.  
Murray, London, 1859.  
or the Preservation of Favored Races in the Struggle for Life.

## Bibliografía II



- [4] J. H. Holland.  
*Adaptation in Natural and Artificial Systems.*  
University of Michigan Press, Ann Arbor, MI, 1975.