

Inteligencia Artificial

Técnicas de Búsqueda Incompleta:

Algoritmos Genéticos

Elizabeth Montero Ureta

Departamento de Informática
Universidad Técnica Federico Santa María

2do Semestre 2018

Introducción

Teoría de evolución

Algoritmos genéticos

- Componentes

- Representación

- Función de aptitud

- Proceso de selección

- Proceso de transformación

- Consideraciones

- Intensificación y diversificación

Variaciones de algoritmos genéticos

- Representación

- Proceso de selección

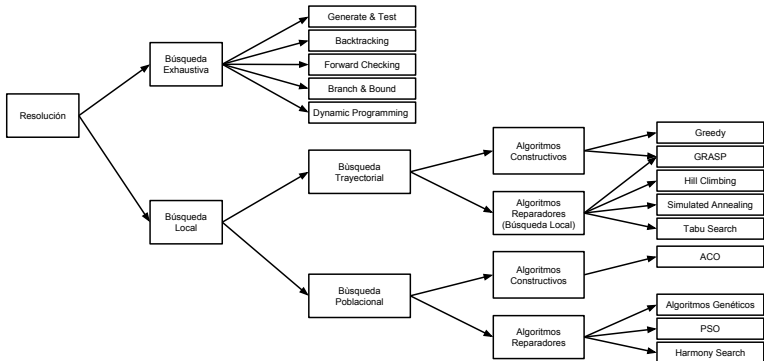
- Proceso de transformación

Seteo de parámetros

- Sintonización de parámetros

- Control de parámetros

Clasificación de Técnicas de Resolución



Algoritmos Evolutivos

- ▶ Los algoritmos genéticos son un tipo de algoritmos evolutivos.
 - ▶ Algoritmos Genéticos (Holland, 1962)
 - ▶ Programación Evolutiva (Fogel, 1962)
 - ▶ Estrategias Evolutivas (Rechemberg, 1965)
 - ▶ Programación Genética (Koza, 1989)

Algoritmos Evolutivos



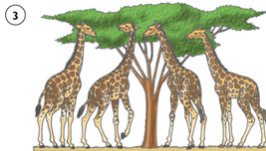
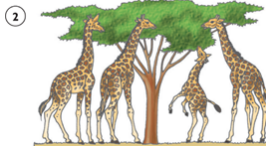
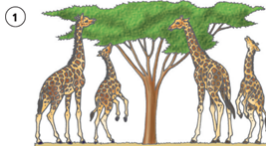
Section 2

Teoría de evolución

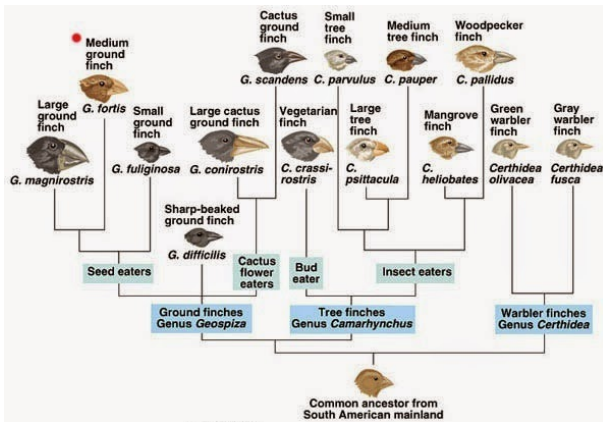
Introducción

- ▶ La teoría de la evolución muestra que los seres vivos evolucionan bajo el efecto del medio ambiente
 - ▶ Los que están mejor adaptados tienen más posibilidades de sobrevivir y de reproducirse.
- ▶ En cada generación las características de los individuos mejor adaptados tienen más posibilidades de estar presente en la población.
- ▶ La genética cuyo objetivo es estudiar los mecanismos de herencia propone un modelo que permite explicar la transmisión de estas características de una generación a otra.

Teoría de evolución



Teoría de evolución



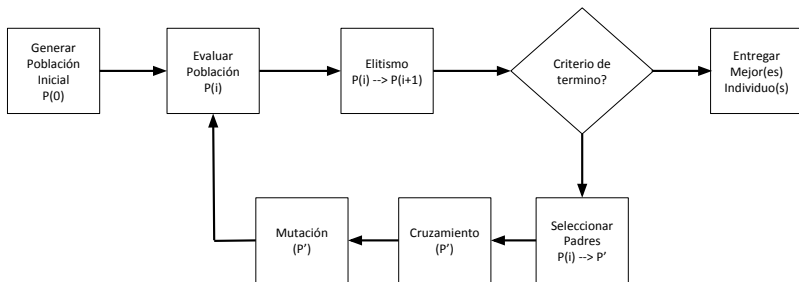
Genotipo vs Fenotipo

- ▶ Existen entidades responsables de la producción de caracteres hereditarios, que se llaman genes y el conjunto de genes de un individuo define su genotipo.

Genotipo vs Fenotipo

- ▶ Existen entidades responsables de la producción de caracteres hereditarios, que se llaman genes y el conjunto de genes de un individuo define su genotipo.
- ▶ El fenotipo de un individuo corresponde a un conjunto de características que es posible observar, medir o calificar en él (apariencia física); el fenotipo es posible que cambie en el tiempo por efecto del medio en el cual evoluciona, pero sus variaciones no pueden transmitirse.

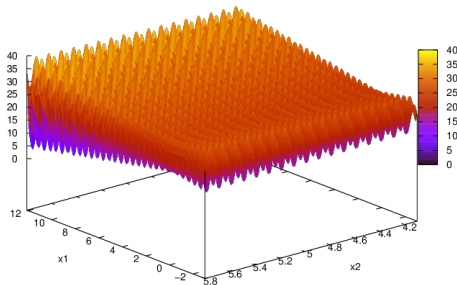
Diagrama de Flujo Programa Evolutivo



Section 3

Algoritmos genéticos

Algoritmos Genéticos



$$f(x_1, x_2) = 21.5 + x_1 * \sin(4\pi x_1) + x_2 * \sin(20\pi x_2) \quad (1)$$

Algoritmo Genético Estándar

Los AG fueron pensados para resolver problemas de optimización continua que los métodos matemáticos no podían resolver

Algoritmo Genético

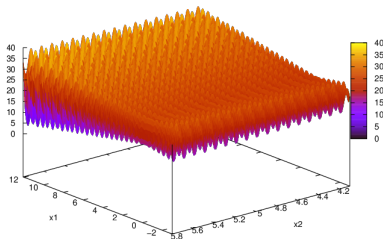
Componentes de los AG

1. Representación
2. Función de Evaluación/Aptitud
3. Proceso de Selección
4. Proceso de Transformación

Representación

- ▶ Codifica las soluciones candidatas para ser manipuladas por el algoritmo
- ▶ Determina el espacio de búsqueda donde se encuentran las soluciones
- ▶ La representación de los Algoritmos Genéticos Estándar (AGE) es **Binaria**
- ▶ **Existe un costo de codificación - decodificación asociado**
- ▶ El tamaño del dominio y la precisión escogida determina la cantidad de bits a utilizar

Representación

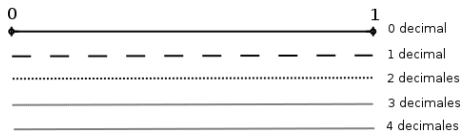


$$f(x_1, x_2) = 21.5 + x_1 * \sin(4\pi x_1) + x_2 * \sin(20\pi x_2)$$

Ejemplo

- ▶ Los dominios son: $-3.0 \leq x_1 \leq 12.1$ y $4.1 \leq x_2 \leq 5.8$
- ▶ Para x_1 , el tamaño de su dominio es 15.1
- ▶ Para x_2 , el tamaño de su dominio es 1.7

Representación



$$f(x_1, x_2) = 21.5 + x_1 * \sin(4\pi x_1) + x_2 * \sin(20\pi x_2)$$

Ejemplo

- ▶ Los dominios son: $-3.0 \leq x_1 \leq 12.1$ y $4.1 \leq x_2 \leq 5.8$
- ▶ Para x_1 , el tamaño de su dominio es 15.1
- ▶ Para x_2 , el tamaño de su dominio es 1.7

Representación

$$f(x_1, x_2) = 21.5 + x_1 * \sin(4\pi x_1) + x_2 * \sin(20\pi x_2)$$

Cantidad de Bits

- ▶ Para x_1 , el tamaño de su dominio es 15.1 y cada entero se dividirá en 10000 partes
- ▶ Es decir, para todo el dominio necesitamos 151.000 secciones
- ▶ Luego observamos cómo podemos representar esa cantidad en binario:
 - ▶ $2^{17} = 131.072$

Representación

$$f(x_1, x_2) = 21.5 + x_1 * \sin(4\pi x_1) + x_2 * \sin(20\pi x_2)$$

Cantidad de Bits

- ▶ Para x_1 , el tamaño de su dominio es 15.1 y cada entero se dividirá en 10000 partes
- ▶ Es decir, para todo el dominio necesitamos 151.000 secciones
- ▶ Luego observamos cómo podemos representar esa cantidad en binario:
 - ▶ $2^{17} = 131.072$ - Faltarían 19.928
 - ▶ $2^{18} = 262.144$

Representación

$$f(x_1, x_2) = 21.5 + x_1 * \sin(4\pi x_1) + x_2 * \sin(20\pi x_2)$$

Cantidad de Bits

- ▶ Para x_1 , el tamaño de su dominio es 15.1 y cada entero se dividirá en 10000 partes
- ▶ Es decir, para todo el dominio necesitamos 151.000 secciones
- ▶ Luego observamos cómo podemos representar esa cantidad en binario:
 - ▶ $2^{17} = 131.072$ - Faltarían 19.928
 - ▶ $2^{18} = 262.144$
- ▶ Necesitamos 18 bits para representar x_1

Representación

$$f(x_1, x_2) = 21.5 + x_1 * \sin(4\pi x_1) + x_2 * \sin(20\pi x_2)$$

Cantidad de Bits

- ▶ Para x_2 , el tamaño de su dominio es 1.7 y cada entero se dividirá en 10000 partes
- ▶ Es decir, para todo el dominio necesitamos 17000 secciones
- ▶ Luego observamos como podemos representar esa cantidad en binario:
 - ▶ $2^{14} = 16.384$
 - ▶ $2^{15} = 32.768$

Representación

$$f(x_1, x_2) = 21.5 + x_1 * \sin(4\pi x_1) + x_2 * \sin(20\pi x_2)$$

Cantidad de Bits

- ▶ Para x_2 , el tamaño de su dominio es 1.7 y cada entero se dividirá en 10000 partes
- ▶ Es decir, para todo el dominio necesitamos 17000 secciones
- ▶ Luego observamos como podemos representar esa cantidad en binario:
 - ▶ $2^{14} = 16.384$
 - ▶ $2^{15} = 32.768$
- ▶ Necesitamos 15 bits para representar x_2

Representación

Cantidad de Bits

- ▶ Necesitamos 33 bits para representar ambas variables
- ▶ Los primeros 18 serán para x_1 y los restantes 15 para x_2
- ▶ Por ejemplo: **010001001011010000**111110010100010

Representación

Cantidad de Bits

- ▶ Necesitamos 33 bits para representar ambas variables
- ▶ Los primeros 18 serán para x_1 y los restantes 15 para x_2
- ▶ Por ejemplo: **010001001011010000**111110010100010

Conversión

- ▶ Si $x_i \in [a, b]$
- ▶ El string binario de largo k que representa la variable x_i es $(i_1 i_2 i_3 \dots i_k)$
- ▶ El valor de x_i corresponde a:

$$x_i = a + \text{decimal}(i_1 i_2 i_3 \dots i_k) \cdot \frac{(b - a)}{(2^k - 1)}$$

Representación

Conversión

- ▶ Para convertir $x_1 \in [-3, 0; 12, 1]$
- ▶ El string de largo 18 que representa la variable x_i es (010001001011010000)
- ▶ El valor de x_1 corresponde a:

$$x_1 = -3, 0 + \text{decimal}(010001001011010000) \cdot \frac{(12, 1 + 3, 0)}{(2^{18} - 1)}$$

Representación

Conversión

- ▶ Para convertir $x_1 \in [-3, 0; 12, 1]$
- ▶ El string de largo 18 que representa la variable x_i es (010001001011010000)
- ▶ El valor de x_1 corresponde a:

$$x_1 = -3, 0 + \text{decimal}(010001001011010000) \cdot \frac{(12, 1 + 3, 0)}{(2^{18} - 1)}$$

$$x_1 = -3.0 + 70352 \cdot \frac{(15, 1)}{262143} = 1, 0524$$

Representación

Conversión

- ▶ Para convertir $x_2 \in [4, 1; 5, 8]$
- ▶ El string de largo 15 que representa la variable x_i es (111110010100010)
- ▶ El valor de x_1 corresponde a:

$$x_2 = 4,1 + \text{decimal}(111110010100010) \cdot \frac{(5,8 - 4,1)}{(2^{15} - 1)}$$

Representación

Conversión

- ▶ Para convertir $x_2 \in [4, 1; 5, 8]$
- ▶ El string de largo 15 que representa la variable x_i es (111110010100010)
- ▶ El valor de x_1 corresponde a:

$$x_2 = 4,1 + \text{decimal}(111110010100010) \cdot \frac{(5,8 - 4,1)}{(2^{15} - 1)}$$

$$x_2 = 4,1 + 31906 \cdot \frac{1,7}{32767} = 5,7553$$

Representación

Conversión

- ▶ Para convertir $x_2 \in [4, 1; 5, 8]$
- ▶ El string de largo 15 que representa la variable x_i es (111110010100010)
- ▶ El valor de x_1 corresponde a:

$$x_2 = 4,1 + \text{decimal}(111110010100010) \cdot \frac{(5,8 - 4,1)}{(2^{15} - 1)}$$

$$x_2 = 4,1 + 31906 \cdot \frac{1,7}{32767} = 5,7553$$

- ▶ Por lo que el **cromosoma 010001001011010000111110010100010** representa $(x_1; x_2) = (1,0524; 5,7553)$

Representación

Conversión

- Código Gray

Función de Evaluación/Aptitud

Componentes de los AG

1. Representación
2. **Función de Evaluación/Aptitud**
3. Proceso de Selección
4. Proceso de Transformación

Función de Evaluación/Aptitud

- ▶ Indica la calidad de una solución candidata
- ▶ **En AG es conocida como Función de Aptitud**
- ▶ **En AGE requiere decodificación**

$$f(x_1; x_2) = 21.5 + x_1 * \sin(4\pi x_1) + x_2 * \sin(20\pi x_2)$$

- ▶ Para el ejemplo anterior, bastaría evaluar x_1 y x_2 en la función
- ▶ $f(1,0524; 5,7553) = 20,2526$

Proceso de Selección

Componentes de los AG

1. Representación
2. Función de Evaluación/Aptitud
3. **Proceso de Selección**
4. Proceso de Transformación

Proceso de Selección

- ▶ Es el filtro con el cual el algoritmo determina qué individuos serán padres para crear la próxima generación
- ▶ El grado de consideración de la calidad de los individuos, determina la presión de selección del proceso
- ▶ **En AGE se aplica la selección por Ruleta y consiste en asignar a cada individuo una probabilidad de ser seleccionado proporcional su calidad**
- ▶ **Se seleccionan N padres de la población actual y cada uno tiene una probabilidad de ser seleccionado.**

$$P_i = \frac{F_i}{\sum_{i=1}^N F_i} \quad (2)$$

(caso maximización).

Proceso de Selección

P=

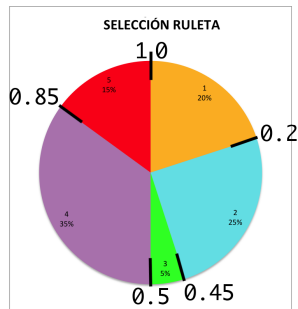
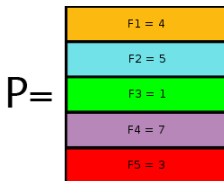
F1 = 4
F2 = 5
F3 = 1
F4 = 7
F5 = 3

Proceso de Selección

P=

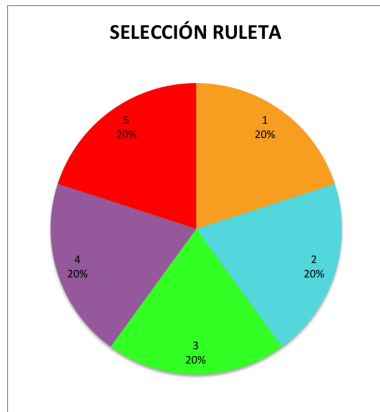
F1 = 4
F2 = 5
F3 = 1
F4 = 7
F5 = 3

Proceso de Selección



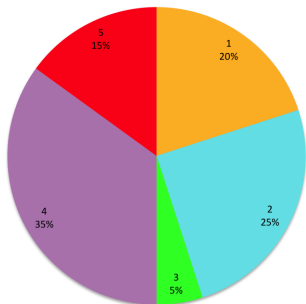
Proceso de Selección

F1=200524~P1=0.200
F2=200525~P2=0.2010
F3=200521~P3=0.1997
F4=200527~P4=0.2029
F5=200523~P5=0.1990

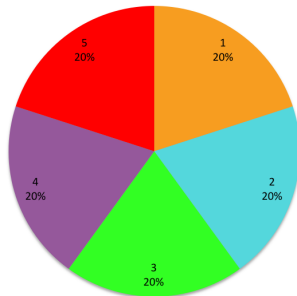


Proceso de Selección

SELECCIÓN RULETA



SELECCIÓN RULETA



Proceso de Transformación

Componentes de los AG

1. Representación
2. Función de Evaluación/Aptitud
3. Proceso de Selección
4. **Proceso de Transformación**

Proceso de Transformación

1. El proceso de transformación consiste en aplicar operadores de transformación unarios y binarios a los individuos
2. En AG se aplica un operador binario llamado **Operador de Cruzamiento**, donde se generan 2 hijos a partir de 2 padres
3. Luego, se aplica un operador unario llamado **Operador de Mutación**, alterando un individuo determinado
4. La decisión de aplicar estos operadores es probabilística y está asociada a dos parámetros del algoritmo: p_c y p_m .

Proceso de Transformación

Operador de Cruzamiento

- ▶ Genera dos hijos a partir de dos padres
- ▶ Se utiliza información conocida y presente en la población
- ▶ Se realizan con la apuesta de obtener algún individuo interesante recombining individuos existentes

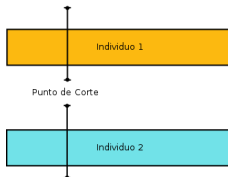
Proceso de Transformación

Cruzamiento en un punto (AGE)



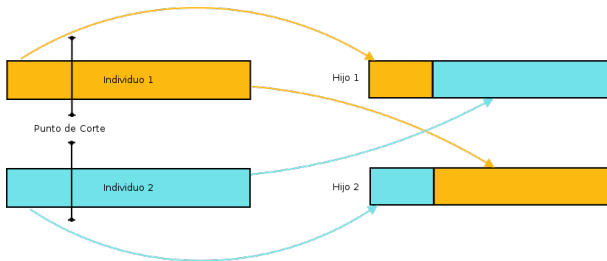
Proceso de Transformación

Cruzamiento en un punto (AGE)



Proceso de Transformación

Cruzamiento en un punto (AGE)



Proceso de Transformación

$$f(x_1, x_2) = 21.5 + x_1 * \sin(4\pi x_1) + x_2 * \sin(20\pi x_2)$$

Necesitamos un punto de corte aleatorio (Ej:pos=9)!

P1	100011000 101101001111000001110010	$x_1 = 5,2786$	$x_2 = 5,5934$	$f_1 = 17,4067$
P2	111011101 101110000100011111011110	$x_1 = 11,0890$	$x_2 = 5,0545$	$f_2 = 30,0602$

Proceso de Transformación

$$f(x_1, x_2) = 21.5 + x_1 * \sin(4\pi x_1) + x_2 * \sin(20\pi x_2)$$

Necesitamos un punto de corte aleatorio (Ej:pos=9)!

P1	100011000 101101001111000001110010	$x_1 = 5,2786$	$x_2 = 5,5934$	$f_1 = 17,4067$
P2	111011101 101110000100011111011110	$x_1 = 11,0890$	$x_2 = 5,0545$	$f_2 = 30,0602$
H1	100011000 101110000100011111011110			$f_1 = 18,2010$
H2	111011101 101101001111000001110010			$f_2 = 29,2156$

Proceso de Transformación

Operador de Mutación

Realiza modificaciones a un individuo con el objetivo de incluir información nueva

Proceso de Transformación

Operador de Mutación

Realiza modificaciones a un individuo con el objetivo de incluir información nueva

Ejemplos

1. Bit-Flip (AGE)

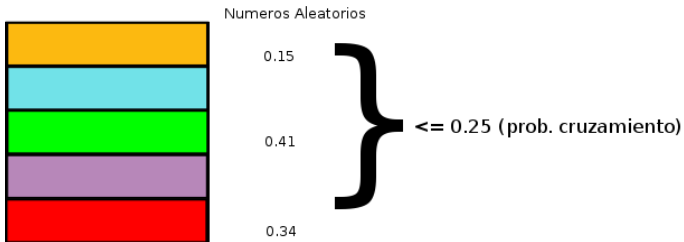
► [1 - 1 - 1 - 1 - 1] → [0 - 1 - 1 - 1 - 1]

2. Al mutar 111011101101101001111000001110010 - cambia de $f_2 = 29.21563$ a $f'_2 = 33.3518$

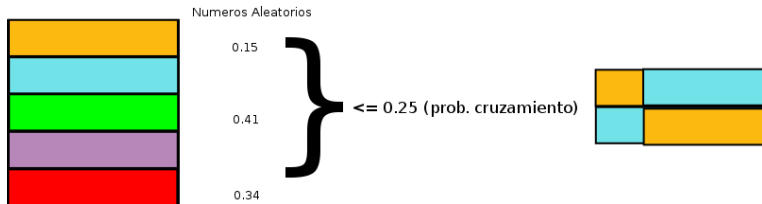
Síntesis



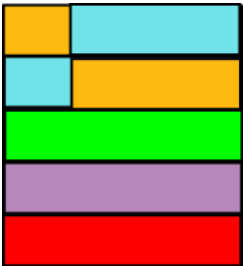
Síntesis



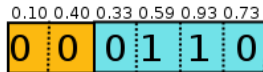
Síntesis



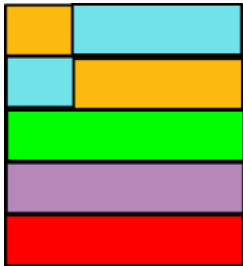
Síntesis



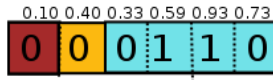
Probabilidad de Mutación= 0.11



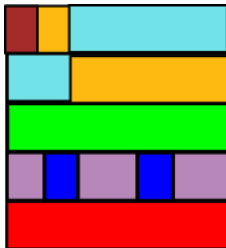
Síntesis



Probabilidad de Mutación= 0.11



Síntesis



Consideraciones

Consideraciones

- ▶ Este algoritmo necesita de algunos parámetros a definir previos a su ejecución
 - ▶ Cómo inicializar la Población
 - ▶ Tamaño Población
 - ▶ Cantidad de Generaciones
 - ▶ Probabilidades p_c y p_m
- ▶ Es importante observar cómo estos parámetros influyen en el comportamiento del algoritmo

Intensificación y Diversificación

Componente	Diversificación	Intensificación
Selección	Presión de selección baja	Presión de selección alta
Transformaciones	Mutación	Cruzamiento
Tamaño Población	Grande	Pequeño

Section 4

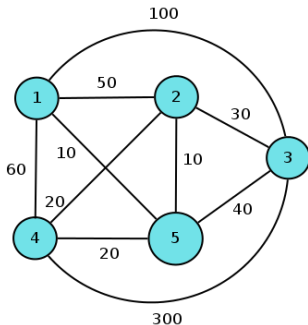
Variaciones de algoritmos genéticos

Variaciones de Algoritmos Genéticos

Variaciones de Algoritmos Genéticos

1. **Representación**
2. Proceso de Selección
3. Proceso de Transformación

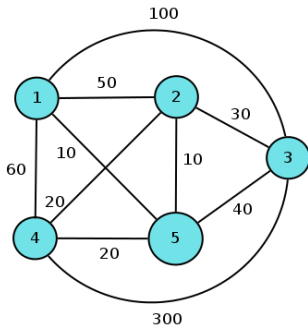
Representación



Tour: [1 - 5 - 2 - 3 - 4]

1. Camino - [1 - 5 - 2 - 3 - 4]

Representación

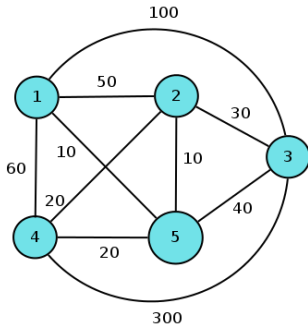


Tour: [1 - 5 - 2 - 3 - 4]

1. Camino - [1 - 5 - 2 - 3 - 4]
2. Matriz Binaria

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

Representación



Tour: [1 - 5 - 2 - 3 - 4]

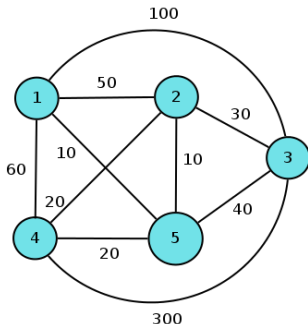
1. Camino - [1 - 5 - 2 - 3 - 4]
2. Matriz Binaria

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

3. Lista de Adyacencia

Casilla	1	2	3	4	5
Tour	5	3	4	1	2

Representación



Tour: [1 - 5 - 2 - 3 - 4]

1. Camino - [1 - 5 - 2 - 3 - 4]
2. Matriz Binaria

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

3. Lista de Adyacencia

Casilla	1	2	3	4	5
Tour	5	3	4	1	2

4. Lista de Referencia

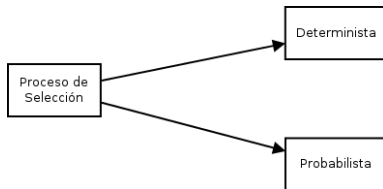
Ciudades	1	2	3	4	5
Tour	1	4	1	1	1

Variaciones de Algoritmos Genéticos

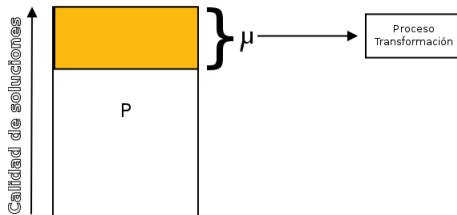
Variaciones de Algoritmos Genéticos

1. Representación
2. **Proceso de Selección**
3. Proceso de Transformación

Proceso de Selección



Proceso de Selección



Selección Determinista

- ▶ Considera los μ mejores individuos de la población actual para ser padres
- ▶ Estos procesos convergen más rápidamente \rightarrow se estancan en óptimos locales

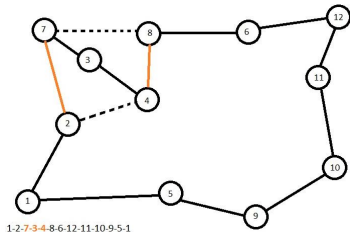
Proceso de Selección

Selección Probabilista

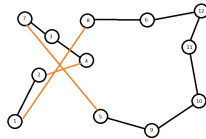
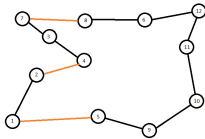
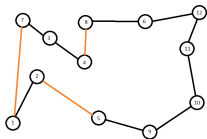
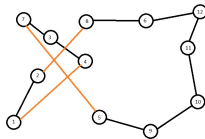
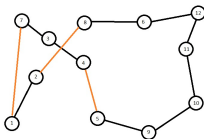
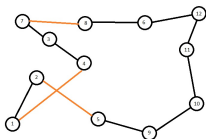
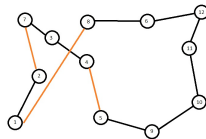
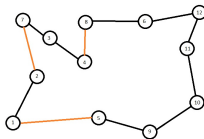
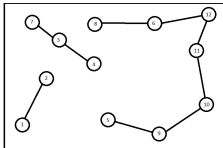
- ▶ Selección por Torneo:
 - ▶ Se considera una cantidad k de individuos, tomados de manera aleatoria
 - ▶ Se selecciona el mejor entre los k individuos
 - ▶ La presión de selección es determinada por el tamaño de k
 - ▶ Si $k \rightarrow \mu$, presión de selección alta
 - ▶ Si $k \rightarrow 2$, presión de selección baja

Proceso de Transformación

1. Intercambio (o Swap)
 - ▶ 1-2-7-3-4-8-6-12-11-10-9-5-1
 - ▶ 1-8-7-3-4-2-6-12-11-10-9-5-1
2. K-opt (K-exchange): Es un procedimiento que reemplaza K arcos de un tour de TSP por K nuevos arcos, tal que el tour resultante es un tour factible del TSP.
 - ▶ 2-opt:
1-2-4-3-7-8-6-12-11-10-9-5-1 //reemplazar 2-4 y 7-8 por 2-7 y 4-8
1-2-7-3-4-8-6-12-11-10-9-5-1



3-opt



Proceso de Transformación

Operador de Cruzamiento

1. Cruzamiento Uniforme (Ejemplo - Problema de la Mochila)

P1 [0 - **1** - 0 - 1 - **1**]

P2 [**1** - 0 - **1** - **0** - 0]

Proceso de Transformación

Operador de Cruzamiento

1. Cruzamiento Uniforme (Ejemplo - Problema de la Mochila)

P1	[0 - 1 - 0 - 1 - 1]
P2	[1 - 0 - 1 - 0 - 0]
<hr/>	
Hijo	[1 - 1 - 1 - 0 - 1]

Proceso de Transformación

Operador de Cruzamiento

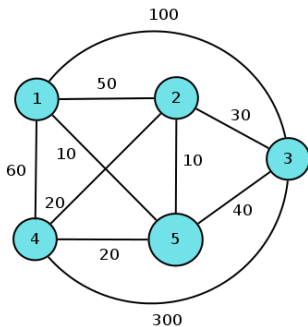
1. Cruzamiento Uniforme (Ejemplo - Problema de la Mochila)

$$\begin{array}{r}
 \text{P1} \quad [0 - \mathbf{1} - 0 - 1 - \mathbf{1}] \\
 \text{P2} \quad [\mathbf{1} - 0 - \mathbf{1} - \mathbf{0} - 0] \\
 \hline
 \text{Hijo} \quad [1 - 1 - 1 - 0 - 1]
 \end{array}$$

2. Cruzamiento en un punto (Ejemplo - Problema de la Mochila)

$$\begin{array}{r}
 \text{P1} \quad [1 - 1 \mid 1 - 1 - 1] \\
 \text{P2} \quad [0 - 0 \mid 0 - 0 - 0] \\
 \hline
 \text{Hijo 1} \quad [1 - 1 \mid 0 - 0 - 0] \\
 \text{Hijo 2} \quad [0 - 0 \mid 1 - 1 - 1]
 \end{array}$$

Proceso de Transformación



Cruzamiento en un punto

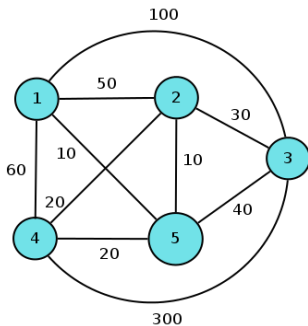
Tour 1 [1 - 4 | 3 - 2 - 5]

Tour 2 [5 - 1 | 4 - 3 - 2]

Hijo 1 [1 - **4** | **4** - 3 - 2]

Hijo 2 [**5** - 1 | 3 - 2 - **5**]

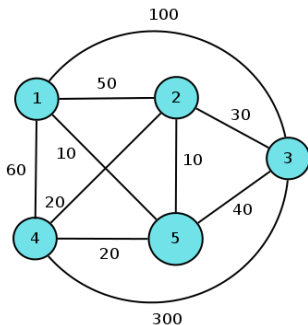
Proceso de Transformación



Cruzamiento en un punto - Lista de Referencia

Lista	[1 - 2 - 3 - 4 - 5]
Padre 1	[1 - 4 - 3 - 2 - 5]

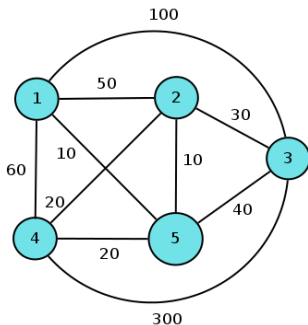
Proceso de Transformación



Cruzamiento en un punto - Lista de Referencia

Lista	[1 - 2 - 3 - 4 - 5]
Padre 1	[1 - 4 - 3 - 2 - 5]
Padre 1 (c)	[1 - 3 2 - 1 - 1]

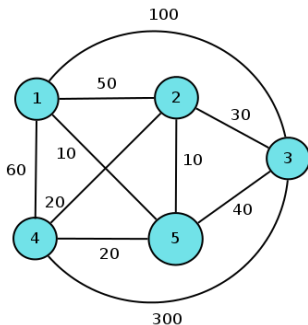
Proceso de Transformación



Cruzamiento en un punto - Lista de Referencia

Lista	[1 - 2 - 3 - 4 - 5]
Padre 1	[1 - 4 - 3 - 2 - 5]
Padre 1 (c)	[1 - 3 2 - 1 - 1]
Padre 2	[5 - 1 - 4 - 3 - 2]

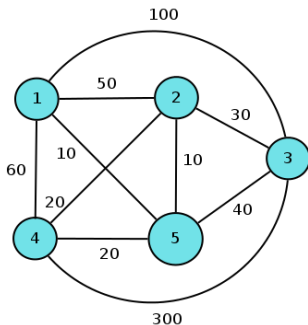
Proceso de Transformación



Cruzamiento en un punto - Lista de Referencia

Lista	[1 - 2 - 3 - 4 - 5]
Padre 1	[1 - 4 - 3 - 2 - 5]
Padre 1 (c)	[1 - 3 2 - 1 - 1]
Padre 2	[5 - 1 - 4 - 3 - 2]
Padre 1 (c)	[5 - 1 3 - 2 - 1]

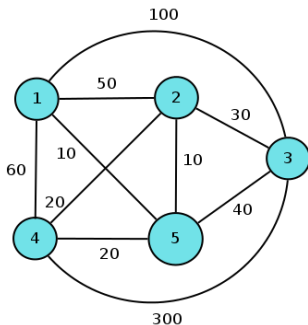
Proceso de Transformación



Cruzamiento en un punto - Lista de Referencia

Lista	[1 - 2 - 3 - 4 - 5]
Padre 1	[1 - 4 - 3 - 2 - 5]
Padre 1 (c)	[1 - 3 2 - 1 - 1]
Padre 2	[5 - 1 - 4 - 3 - 2]
Padre 1 (c)	[5 - 1 3 - 2 - 1]
Hijo 1 (c)	[1 - 3 3 - 2 - 1]
Hijo 2 (c)	[5 - 1 2 - 1 - 1]

Proceso de Transformación



Cruzamiento en un punto - Lista de Referencia

Lista	[1 - 2 - 3 - 4 - 5]
Padre 1	[1 - 4 - 3 - 2 - 5]
Padre 1 (c)	[1 - 3 2 - 1 - 1]
Padre 2	[5 - 1 - 4 - 3 - 2]
Padre 2 (c)	[5 - 1 3 - 2 - 1]
Hijo 1 (c)	[1 - 3 3 - 2 - 1]
Hijo 2 (c)	[5 - 1 2 - 1 - 1]
Hijo 1	[1 - 4 5 - 3 - 2]
Hijo 2	[5 - 1 3 - 2 - 4]

Proceso de Transformación

- Cruzamiento en dos puntos para Permutaciones (Ejemplo - TSP)

P1	[1 - 2 5 - 3 4]
P2	[4 - 3 - 5 - 1 - 2]
<hr/>	
Hijo	[5 - 3 4 - 1 - 2]

Epistasis

- El fenómeno de la epistasis consiste en la existencia de fuertes vinculaciones/interacciones entre los genes, de forma que el efecto de unos genes inhibe o potencia otros. En algunos casos, el resultado es tan complejo que no tiene sentido considerar los efectos de los genes de forma individual sino conjuntamente.

Section 5

Seteo de parámetros

Parámetros de los algoritmos

- ▶ Hill-Climbing
 - ▶ Representación
 - ▶ Función de evaluación
 - ▶ Vecindario - Movimiento

Parámetros de los algoritmos

- ▶ Hill-Climbing
 - ▶ Representación
 - ▶ Función de evaluación
 - ▶ Vecindario - Movimiento
- ▶ Hill-Climbing+Restart
 - ▶ + Número de reinicios

Parámetros de los algoritmos

- ▶ Hill-Climbing
 - ▶ Representación
 - ▶ Función de evaluación
 - ▶ Vecindario - Movimiento
- ▶ Hill-Climbing+Restart
 - ▶ + Número de reinicios
- ▶ Tabu Search
 - ▶ + Información a almacenar en la lista
 - ▶ + Largo de la lista
 - ▶ + Número de iteraciones/evaluaciones (Criterio de parada)

Parámetros de los algoritmos

- ▶ Hill-Climbing
 - ▶ Representación
 - ▶ Función de evaluación
 - ▶ Vecindario - Movimiento
- ▶ Hill-Climbing+Restart
 - ▶ + Número de reinicios
- ▶ Tabu Search
 - ▶ + Información a almacenar en la lista
 - ▶ + Largo de la lista
 - ▶ + Número de iteraciones/evaluaciones (Criterio de parada)
- ▶ Simulated Annealing
 - ▶ + Temperatura inicial
 - ▶ + α : parámetro de enfriamiento del sistema
 - ▶ + Cantidad de iteraciones entre enfriamientos
 - ▶ + Cantidad de iteraciones entre recalentamientos

Parámetros de los algoritmos

- ▶ Algoritmos Genéticos
 - ▶ + Tamaño de la población
 - ▶ + Operador de selección
 - ▶ + Operador de mutación
 - ▶ + Probabilidad de mutación
 - ▶ + Operador de cruzamiento
 - ▶ + Probabilidad de cruzamiento

El problema de Asignación de valores a parámetros

Importancia:

- ▶ Naturaleza estocástica de las metaheurísticas
- ▶ Parámetros propios de cada instancia/problema
- ▶ Parámetros interrelacionados
- ▶ Proceso costoso en tiempo
- ▶ Problema de Optimización Combinatoria en sí
- ▶ Parámetros que varían durante la ejecución

El problema de Asignación de valores a parámetros

Importancia:

- ▶ Naturaleza estocástica de las metaheurísticas
- ▶ Parámetros propios de cada instancia/problema
- ▶ Parámetros interrelacionados
- ▶ Proceso costoso en tiempo
- ▶ Problema de Optimización Combinatoria en sí
- ▶ Parámetros que varían durante la ejecución

Acercamientos:

[Eiben and Schut, 2008]

- ▶ Sintonización
- ▶ Control
 - ▶ Determinista
 - ▶ Adaptivo
 - ▶ Auto-adaptivo

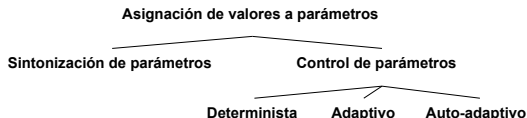
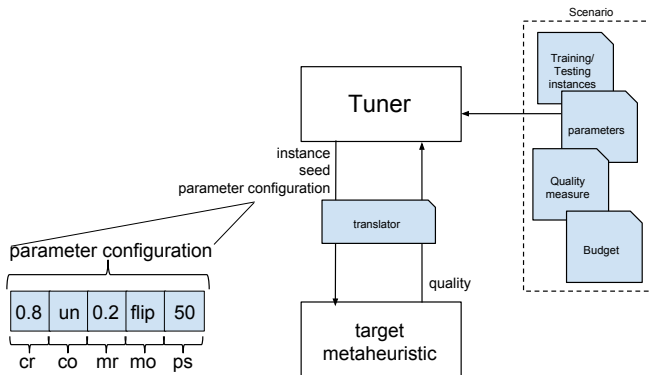


Figura: Taxonomía global del problema de asignación de valores a los parámetros en algoritmos evolutivos

Proceso de sintonización

- ▶ Costoso en tiempo
 - ▶ Requiere ejecutar el algoritmo a sintonizar muchas veces de manera de obtener información relevante de la dependencia de su desempeño respecto a los valores de sus parámetros.
 - ▶ Estas ejecuciones se realizan antes de la ejecución “real” del algoritmo.
- ▶ Mantiene los valores de parámetros fijos durante la búsqueda.
- ▶ Tipos de parámetros:
 - ▶ Numérico: Permiten definir medidas de distancia. Gran conjunto de valores posibles.
 - ▶ Categórico: NO permiten definir medidas de distancia. Conjunto reducido de valores posibles.
 - ▶ Condicional: Su importancia para el algoritmo depende del valor de otro parámetro del algoritmo.

Proceso de sintonización



Control de Parámetros

Criterios de clasificación

- ▶ ¿Qué se cambia?
 - ▶ Parámetros típicos del algoritmo
- ▶ ¿Cómo se cambia?
 - ▶ Determinista
 - ▶ Adaptivo
 - ▶ Auto-adaptivo
- ▶ ¿Cuál es la evidencia del cambio?
 - ▶ Qué se *monitorea* en el algoritmo

¿Cómo se cambia?

Control Determinista

Los cambios se realizan de forma determinista con respecto a alguna forma de medición de “tiempo” en el algoritmo. Cada “n” iteraciones disminuir el valor del parámetro “x” en 10%.

Control Adaptivo

Los cambios se realizan en base a la información de la búsqueda. La idea es monitorear ciertos indicadores del algoritmo que gatillen cambios en los valores de los parámetros. Los valores de los parámetros son globales para el algoritmo.

Control Auto-adaptivo

Los cambios se realizan en base a la información de la búsqueda. La idea es monitorear ciertos indicadores del algoritmo que gatillen cambios en los valores de los parámetros. Los valores de los parámetros son propios de cada solución/individuo (algoritmos poblacionales).

Sintonización vs Control

Sintonización

Proceso previo a la ejecución del algoritmo

- ▶ Altamente consumidor de tiempo
- ▶ Valores de parámetros útiles para el problema/instancia particular
- ▶ Valores de parámetros permanecen fijos durante toda la ejecución

Control

Proceso conjunto a la ejecución del algoritmo

- ▶ Tiempo de diseño (estrategias de monitoreo y actualización)
- ▶ Tiempo adicional durante la ejecución (monitoreo y actualización)
- ▶ Valores de parámetros pueden cambiar durante la ejecución

Bibliografía



Eiben, A. and Schut, M. (2008).

Advances in Metaheuristics for Hard Optimization, chapter New Ways to Calibrate Evolutionary Algorithms, pages 153–177.

Natural Computing. Springer.