

COMPUTACIÓN BIOLÓGICA

Pedro Isasi¹

¹Departamento de Informática
Universidad Carlos III de Madrid
Avda. de la Universidad, 30. 28911 Leganés (Madrid). Spain
email: isasi@ia.uc3m.es

Presentación

1 INTRODUCCIÓN

2 ALGORITMOS GENÉTICOS

- **Introducción a los Algoritmos Genéticos**
- Algoritmos Généticos Canónicos
- Ejemplo de Algoritmo Genético
- Propiedades de los Algoritmos Genéticos
- Métodos de compartición y soluciones múltiples
- Manejo de restricciones en Algoritmos Genéticos
- Fundamentos de los Algoritmos Genéticos

3 COMPUTACIÓN EVOLUTIVA

4 COMPUTACIÓN CON INSPIRACIÓN BIOLÓGICA

5 BIOLOGÍA Y COMPUTACIÓN

1 INTRODUCCIÓN

2 ALGORITMOS GENÉTICOS

- Introducción a los Algoritmos Genéticos
- **Algoritmos Généticos Canónicos**
- Ejemplo de Algoritmo Genético
- Propiedades de los Algoritmos Genéticos
- Métodos de compartición y soluciones múltiples
- Manejo de restricciones en Algoritmos Genéticos
- Fundamentos de los Algoritmos Genéticos

3 COMPUTACIÓN EVOLUTIVA

4 COMPUTACIÓN CON INSPIRACIÓN BIOLÓGICA

5 BIOLOGÍA Y COMPUTACIÓN

- Datos
 - Problema a resolver, \mathcal{P}
 - Espacio de soluciones, \mathcal{S}
 - Conjunto de operadores, \mathcal{O}
 - Criterio de evaluación de las soluciones, \mathcal{C}
- Obtener
 - El elemento del espacio de estados ($I \in \mathcal{S}$) que mejor resuelva el problema (\mathcal{P}), siguiendo el criterio de evaluación (\mathcal{C})

- Datos
 - Problema a resolver, \mathcal{P}
 - Espacio de soluciones, \mathcal{S}
 - Conjunto de operadores, \mathcal{O}
 - Criterio de evaluación de las soluciones, \mathcal{C}
- Obtener
 - El elemento del espacio de estados ($I \in \mathcal{S}$) que mejor resuelva el problema (\mathcal{P}), siguiendo el criterio de evaluación (\mathcal{C})

- Datos
 - Problema a resolver, \mathcal{P}
 - Espacio de soluciones, \mathcal{S}
 - Conjunto de operadores, \mathcal{O}
 - Criterio de evaluación de las soluciones, \mathcal{C}
- Obtener
 - El elemento del espacio de estados ($I \in \mathcal{S}$) que mejor resuelva el problema (\mathcal{P}), siguiendo el criterio de evaluación (\mathcal{C})

- Datos
 - Problema a resolver, \mathcal{P}
 - Espacio de soluciones, \mathcal{S}
 - Conjunto de operadores, \mathcal{O}
 - Criterio de evaluación de las soluciones, \mathcal{C}
- Obtener
 - El elemento del espacio de estados ($I \in \mathcal{S}$) que mejor resuelva el problema (\mathcal{P}), siguiendo el criterio de evaluación (\mathcal{C})

- Datos
 - Problema a resolver, \mathcal{P}
 - Espacio de soluciones, \mathcal{S}
 - Conjunto de operadores, \mathcal{O}
 - Criterio de evaluación de las soluciones, \mathcal{C}
- Obtener
 - El elemento del espacio de estados ($I \in \mathcal{S}$) que mejor resuelva el problema (\mathcal{P}), siguiendo el criterio de evaluación (\mathcal{C})

- Es un proceso genérico de optimización. Puede reducirse a:
 - **Búsqueda en espacio de estados** (Obtener el elemento que mejor resuelva un problema)
 - **Clasificación** (Busqueda del procedimiento que mejor clasifique)
 - **Predicción** (Encontrar el procedimiento que tenga menores fallos de predicción)
 - **Control** (Obtener el procedimiento que maximice una tarea de control)

- Es un proceso genérico de optimización. Puede reducirse a:
 - **Búsqueda en espacio de estados** (Obtener el elemento que mejor resuelva un problema)
 - **Clasificación** (Busqueda del procedimiento que mejor clasifique)
 - **Predicción** (Encontrar el procedimiento que tenga menores fallos de predicción)
 - **Control** (Obtener el procedimiento que maximice una tarea de control)

- Es un proceso genérico de optimización. Puede reducirse a:
 - **Búsqueda en espacio de estados** (Obtener el elemento que mejor resuelva un problema)
 - **Clasificación** (Busqueda del procedimiento que mejor clasifique)
 - **Predicción** (Encontrar el procedimiento que tenga menores fallos de predicción)
 - **Control** (Obtener el procedimiento que maximice una tarea de control)

- Es un proceso genérico de optimización. Puede reducirse a:
 - **Búsqueda en espacio de estados** (Obtener el elemento que mejor resuelva un problema)
 - **Clasificación** (Busqueda del procedimiento que mejor clasifique)
 - **Predicción** (Encontrar el procedimiento que tenga menores fallos de predicción)
 - **Control** (Obtener el procedimiento que maximice una tarea de control)

- **Individuos.**- Solución particular al problema ($I \in \mathcal{S}$)
- **Población.**- Conjunto finito de soluciones ($P \subset \mathcal{S}$)
- **Operadores Genéticos.**- Funciones de transformación de poblaciones

$$\mathcal{O} : \mathcal{P} \rightarrow \mathcal{P}$$

OPERADORES GENÉTICOS

- Reproducción
- Inversión
- Cruce
- Mutación
- **Función de fitness.**- Criterio de evaluación de las soluciones

$$\mathcal{F} : \mathcal{I} \rightarrow \mathbb{R}$$

ELEMENTOS

- **Individuos.**- Solución particular al problema ($I \in \mathcal{S}$)
- **Población.**- Conjunto finito de soluciones ($P \subset \mathcal{S}$)
- **Operadores Genéticos.**- Funciones de transformación de poblaciones

$$\mathcal{O} : \mathcal{P} \rightarrow \mathcal{P}$$

OPERADORES GENÉTICOS

- Reproducción
- Inversión
- Cruce
- Mutación
- **Función de fitness.**- Criterio de evaluación de las soluciones

$$\mathcal{F} : \mathcal{I} \rightarrow \mathbb{R}$$

ELEMENTOS

- **Individuos**.- Solución particular al problema ($I \in \mathcal{S}$)
- **Población**.- Conjunto finito de soluciones ($P \subset \mathcal{S}$)
- **Operadores Genéticos**.- Funciones de transformación de poblaciones

$$\mathcal{O} : \mathcal{P} \rightarrow \mathcal{P}$$

OPERADORES GENÉTICOS

- Reproducción
 - Inversión
 - Cruce
 - Mutación
- **Función de fitness**.- Criterio de evaluación de las soluciones

$$\mathcal{F} : \mathcal{I} \rightarrow \mathbb{R}$$

ELEMENTOS

- **Individuos**.- Solución particular al problema ($I \in \mathcal{S}$)
- **Población**.- Conjunto finito de soluciones ($P \subset \mathcal{S}$)
- **Operadores Genéticos**.- Funciones de transformación de poblaciones

$$\mathcal{O} : \mathcal{P} \rightarrow \mathcal{P}$$

OPERADORES GENÉTICOS

- Reproducción
- Inversión
- Cruce
- Mutación
- **Función de fitness**.- Criterio de evaluación de las soluciones

$$\mathcal{F} : \mathcal{I} \rightarrow \mathbb{R}$$

- Modelar el problema
 - Elegir una buena representación de las soluciones posibles \implies
Muy Importante
 - Diseñar la función de evaluación de dichas soluciones
- Elegir los parámetros del método
 - Elegir qué tipo de operadores de Selección y Cruce
 - Asignar las tasas de probabilidad a los operadores

- Modelar el problema
 - Elegir una buena representación de las soluciones posibles \implies
Muy Importante
 - Diseñar la función de evaluación de dichas soluciones
- Elegir los parámetros del método
 - Elegir qué tipo de operadores de Selección y Cruce
 - Asignar las tasas de probabilidad a los operadores

- Mediante cadenas de caracteres, normalmente binarios
- Si es posible es conveniente que el alfabeto sea pequeño
- Se pueden codificar valores numéricos enteros o reales (hay que determinar la precisión \Rightarrow Tamaño cromosoma)
- Se pueden codificar dominios diversos. p.ej. reglas:

Clima			Viento	
LLuvioso	Nublado	Despejado	Fuerte	Moderado
Sí/No	Sí/No	Sí/No	Sí/No	Sí/No

El clima es despejado o lluvioso, y el viento es fuerte:

Clima	Viento
101	10

- Mediante cadenas de caracteres, normalmente binarios
- Si es posible es conveniente que el alfabeto sea pequeño
- Se pueden codificar valores numéricos enteros o reales (hay que determinar la precisión \Rightarrow Tamaño cromosoma)
- Se pueden codificar dominios diversos. p.ej. reglas:

Clima			Viento	
LLuvioso	Nublado	Despejado	Fuerte	Moderado
Sí/No	Sí/No	Sí/No	Sí/No	Sí/No

El clima es despejado o lluvioso, y el viento es fuerte:

Clima	Viento
101	10

- Mediante cadenas de caracteres, normalmente binarios
- Si es posible es conveniente que el alfabeto sea pequeño
- Se pueden codificar valores numéricos enteros o reales (hay que determinar la precisión \Rightarrow Tamaño cromosoma)
- Se pueden codificar dominios diversos. p.ej. reglas:

Clima			Viento	
LLuvioso	Nublado	Despejado	Fuerte	Moderado
Sí/No	Sí/No	Sí/No	Sí/No	Sí/No

El clima es despejado o lluvioso, y el viento es fuerte:

Clima Viento
101 10

- Mediante cadenas de caracteres, normalmente binarios
- Si es posible es conveniente que el alfabeto sea pequeño
- Se pueden codificar valores numéricos enteros o reales (hay que determinar la precisión \Rightarrow Tamaño cromosoma)
- Se pueden codificar dominios diversos. p.ej. reglas:

Clima			Viento	
LLuvioso	Nublado	Despejado	Fuerte	Moderado
Sí/No	Sí/No	Sí/No	Sí/No	Sí/No

El clima es despejado o lluvioso, y el viento es fuerte:

Clima	Viento
101	10

- Se pueden codificar reglas: Si el viento es fuerte entonces se puede jugar al tenis

Clima	Viento	Jugar tenis
000	10	1

- Evitar problemas de epístasis \Rightarrow Codificación gray

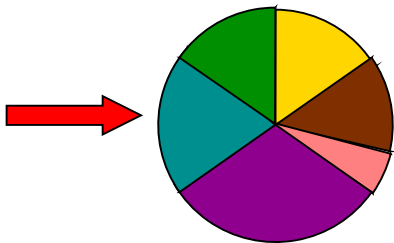
000	1
001	2
011	3
010	4
110	5
100	6
101	7
111	8

- Selección.- Simulación computacional de la Selección Natural Darwiniana:
 - Ruleta
 - Jerárquica
 - Torneos
- Cruce.- Simulación de la reproducción sexual en los organismos:
 - Simple
 - De dos puntos
 - Uniforme
- Inversión.- Operador a medida adecuado para ciertos dominios
- Mutación.- Simulación de las mutaciones en reproducción de los organismos

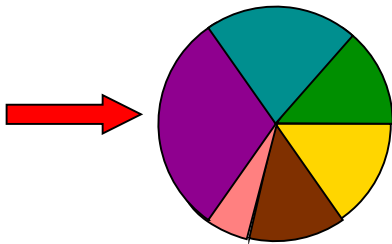
- Selección.- Simulación computacional de la Selección Natural Darwiniana:
 - Ruleta
 - Jerárquica
 - Torneos
- Cruce.- Simulación de la reproducción sexual en los organismos:
 - Simple
 - De dos puntos
 - Uniforme
- Inversión.- Operador a medida adecuado para ciertos dominios
- Mutación.- Simulación de las mutaciones en reproducción de los organismos

- Selección.- Simulación computacional de la Selección Natural Darwiniana:
 - Ruleta
 - Jerárquica
 - Torneos
- Cruce.- Simulación de la reproducción sexual en los organismos:
 - Simple
 - De dos puntos
 - Uniforme
- **Inversión.- Operador a medida adecuado para ciertos dominios**
- Mutación.- Simulación de las mutaciones en reproducción de los organismos

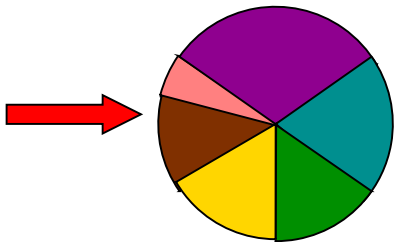
- Selección.- Simulación computacional de la Selección Natural Darwiniana:
 - Ruleta
 - Jerárquica
 - Torneos
- Cruce.- Simulación de la reproducción sexual en los organismos:
 - Simple
 - De dos puntos
 - Uniforme
- Inversión.- Operador a medida adecuado para ciertos dominios
- Mutación.- Simulación de las mutaciones en reproducción de los organismos



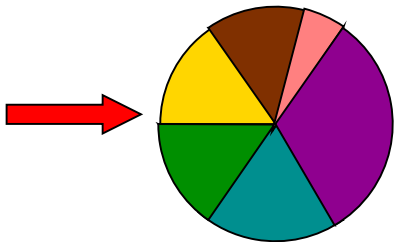
- Operador para seleccionar los individuos más aptos para reproducción
- Individuos con mayor evaluación producirán mayor número de descendientes
- Ruleta.- Probabilidad de selección proporcional al fitness. Se repite la selección tantas veces como tamaño de la población



- Operador para seleccionar los individuos más aptos para reproducción
- Individuos con mayor evaluación producirán mayor número de descendientes
- Ruleta.- Probabilidad de selección proporcional al fitness. Se repite la selección tantas veces como tamaño de la población



- Operador para seleccionar los individuos más aptos para reproducción
- Individuos con mayor evaluación producirán mayor número de descendientes
- Ruleta.- Probabilidad de selección proporcional al fitness. Se repite la selección tantas veces como tamaño de la población



- Operador para seleccionar los individuos más aptos para reproducción
- Individuos con mayor evaluación producirán mayor número de descendientes
- Ruleta.- Probabilidad de selección proporcional al fitness. Se repite la selección tantas veces como tamaño de la población

- Procedimiento método directo

- Se calcula el fitness de cada individuo f_i y la suma de todos ellos
 $f_T = \sum_{i=1}^n f_i$
- Se calcula el fitness relativo para cada individuo: $fr_i = \frac{f_i}{f_T}$
- Se asigna como número de descendientes de cada individuo D_i el resultado de: $D_i = n \times fr_i$
- La parte fraccionaria se asigna como fitness residual del individuo
- Se repite el proceso con los fitness residuales hasta que la suma de todos los D_i sea n
- Cada individuo i es copiado en la población intermedia D_i veces

- Propiedades

- No es estocástico
- Los individuos “débiles” no tienen ninguna posibilidad de reproducirse \Rightarrow Pérdida de diversidad genética
- Convergencia prematura

EJEMPLO MÉTODO DIRECTO

Ind.	Fitness	Fit. Rel.	Desc.
1	22	0.11	1.1 (1)
2	8	0.04	0.4 (0)
3	12	0.06	0.6 (1)
4	36	0.18	1.8 (2)
5	4	0.02	0.2 (0)
6	50	0.25	2.5 (3)
7	18	0.09	0.9 (1)
8	16	0.08	0.8 (1)
9	14	0.07	0.7 (1)
10	20	0.10	0.1 (0)
<hr/>			
Total		1.0	10

- Se calcula el fitness
- Se divide por el fitness total y el valor resultante es la probabilidad de cada individuo de ser seleccionado para reproducción
- La última columna es el nº de veces que se espera sea seleccionado

EJEMPLO MÉTODO DIRECTO

Ind.	Fitness	Fit. Rel.	Desc.
1	22	0.11	1.1 (1)
2	8	0.04	0.4 (0)
3	12	0.06	0.6 (1)
4	36	0.18	1.8 (2)
5	4	0.02	0.2 (0)
6	50	0.25	2.5 (3)
7	18	0.09	0.9 (1)
8	16	0.08	0.8 (1)
9	14	0.07	0.7 (1)
10	20	0.10	0.1 (0)
<hr/>			
Total		1.0	10

- Se calcula el fitness
- Se divide por el fitness total y el valor resultante es la probabilidad de cada individuo de ser seleccionado para reproducción
- La última columna es el nº de veces que se espera sea seleccionado

EJEMPLO MÉTODO DIRECTO

Ind.	Fitness	Fit. Rel.	Desc.
1	22	0.11	1.1 (1)
2	8	0.04	0.4 (0)
3	12	0.06	0.6 (1)
4	36	0.18	1.8 (2)
5	4	0.02	0.2 (0)
6	50	0.25	2.5 (3)
7	18	0.09	0.9 (1)
8	16	0.08	0.8 (1)
9	14	0.07	0.7 (1)
10	20	0.10	0.1 (0)
Total		1.0	10

- Se calcula el fitness
- Se divide por el fitness total y el valor resultante es la probabilidad de cada individuo de ser seleccionado para reproducción
- La última columna es el nº de veces que se espera sea seleccionado

EJEMPLO MÉTODO DIRECTO

Ind.	Fitness	Fit. Rel.	Desc.
1	22	0.11	1.1 (1)
2	8	0.04	0.4 (0)
3	12	0.06	0.6 (1)
4	36	0.18	1.8 (2)
5	4	0.02	0.2 (0)
6	50	0.25	2.5 (3)
7	18	0.09	0.9 (1)
8	16	0.08	0.8 (1)
9	14	0.07	0.7 (1)
10	20	0.10	0.1 (0)
<hr/>			
Total		1.0	10

- Se calcula el fitness
- Se divide por el fitness total y el valor resultante es la probabilidad de cada individuo de ser seleccionado para reproducción
- La última columna es el n° de veces que se espera sea seleccionado

- Procedimiento método acumulado

- Se calcula el fitness de cada individuo f_i y la suma de todos ellos

$$f_T = \sum_{i=1}^n f_i$$

- Se calcula el fitness relativo para cada individuo: $fr_i = \frac{f_i}{f_T}$

- Se genera un vector de n elementos como sigue:

$$\vec{S} = \{fr_1, fr_1 + fr_2, \dots, \sum_{i=1}^j fr_i, \dots, \sum_{i=1}^n fr_i = 1\}$$

- Repetir n veces:

- Se genera un número aleatorio (a) entre $[0 - 1]$
- Pasar a la población intermedia el individuo k siempre que:

$$S_{i-1} < a < S_i$$

- Propiedades

- Es estocástico
- Los individuos “débiles” tienen posibilidad de reproducirse

EJEMPLO MÉTODO ACUMULADO

Ind.	Fitness	Fit. Rel.	\vec{S}
1	22	0.11	0.11 (0)
2	8	0.04	0.15 (0)
3	12	0.06	0.21 (1)
4	36	0.18	0.39 (2)
5	4	0.02	0.41 (0)
6	50	0.25	0.66 (4)
7	18	0.09	0.75 (1)
8	16	0.08	0.83 (1)
9	14	0.07	0.9 (1)
10	20	0.10	1 (0)

Se generan 10 n° aleatorios, p.ej.:

EJEMPLO MÉTODO ACUMULADO

Ind.	Fitness	Fit. Rel.	\vec{S}
1	22	0.11	0.11 (0)
2	8	0.04	0.15 (0)
3	12	0.06	0.21 (1)
4	36	0.18	0.39 (2)
5	4	0.02	0.41 (0)
6	50	0.25	0.66 (4)
7	18	0.09	0.75 (1)
8	16	0.08	0.83 (1)
9	14	0.07	0.9 (1)
10	20	0.10	1 (0)

Se generan 10 n^0 aleatorios, p.ej.: 0.24

EJEMPLO MÉTODO ACUMULADO

Ind.	Fitness	Fit. Rel.	\vec{S}
1	22	0.11	0.11 (0)
2	8	0.04	0.15 (0)
3	12	0.06	0.21 (1)
4	36	0.18	0.39 (2)
5	4	0.02	0.41 (0)
6	50	0.25	0.66 (4)
7	18	0.09	0.75 (1)
8	16	0.08	0.83 (1)
9	14	0.07	0.9 (1)
10	20	0.10	1 (0)

Se generan 10 n^0 aleatorios, p.ej.: 0.58

EJEMPLO MÉTODO ACUMULADO

Ind.	Fitness	Fit. Rel.	\vec{S}
1	22	0.11	0.11 (0)
2	8	0.04	0.15 (0)
3	12	0.06	0.21 (1)
4	36	0.18	0.39 (2)
5	4	0.02	0.41 (0)
6	50	0.25	0.66 (4)
7	18	0.09	0.75 (1)
8	16	0.08	0.83 (1)
9	14	0.07	0.9 (1)
10	20	0.10	1 (0)

Se generan 10 n^0 aleatorios, p.ej.: 0.76

EJEMPLO MÉTODO ACUMULADO

Ind.	Fitness	Fit. Rel.	\vec{S}
1	22	0.11	0.11 (0)
2	8	0.04	0.15 (0)
3	12	0.06	0.21 (1)
4	36	0.18	0.39 (2)
5	4	0.02	0.41 (0)
6	50	0.25	0.66 (4)
7	18	0.09	0.75 (1)
8	16	0.08	0.83 (1)
9	14	0.07	0.9 (1)
10	20	0.10	1 (0)

Se generan 10 n^0 aleatorios, p.ej.: 0.31

EJEMPLO MÉTODO ACUMULADO

Ind.	Fitness	Fit. Rel.	\vec{S}
1	22	0.11	0.11 (0)
2	8	0.04	0.15 (0)
3	12	0.06	0.21 (1)
4	36	0.18	0.39 (2)
5	4	0.02	0.41 (0)
6	50	0.25	0.66 (4)
7	18	0.09	0.75 (1)
8	16	0.08	0.83 (1)
9	14	0.07	0.9 (1)
10	20	0.10	1 (0)

Se generan 10 n° aleatorios, p.ej.: 0.67

EJEMPLO MÉTODO ACUMULADO

Ind.	Fitness	Fit. Rel.	\vec{S}
1	22	0.11	0.11 (0)
2	8	0.04	0.15 (0)
3	12	0.06	0.21 (1)
4	36	0.18	0.39 (2)
5	4	0.02	0.41 (0)
6	50	0.25	0.66 (4)
7	18	0.09	0.75 (1)
8	16	0.08	0.83 (1)
9	14	0.07	0.9 (1)
10	20	0.10	1 (0)

Se generan 10 n° aleatorios, p.ej.: 0.48

EJEMPLO MÉTODO ACUMULADO

Ind.	Fitness	Fit. Rel.	\vec{S}
1	22	0.11	0.11 (0)
2	8	0.04	0.15 (0)
3	12	0.06	0.21 (1)
4	36	0.18	0.39 (2)
5	4	0.02	0.41 (0)
6	50	0.25	0.66 (4)
7	18	0.09	0.75 (1)
8	16	0.08	0.83 (1)
9	14	0.07	0.9 (1)
10	20	0.10	1 (0)

Se generan 10 n° aleatorios, p.ej.: 0.62

EJEMPLO MÉTODO ACUMULADO

Ind.	Fitness	Fit. Rel.	\vec{S}
1	22	0.11	0.11 (0)
2	8	0.04	0.15 (0)
3	12	0.06	0.21 (1)
4	36	0.18	0.39 (2)
5	4	0.02	0.41 (0)
6	50	0.25	0.66 (4)
7	18	0.09	0.75 (1)
8	16	0.08	0.83 (1)
9	14	0.07	0.9 (1)
10	20	0.10	1 (0)

Se generan 10 n° aleatorios, p.ej.: 0.54

EJEMPLO MÉTODO ACUMULADO

Ind.	Fitness	Fit. Rel.	\vec{S}
1	22	0.11	0.11 (0)
2	8	0.04	0.15 (0)
3	12	0.06	0.21 (1)
4	36	0.18	0.39 (2)
5	4	0.02	0.41 (0)
6	50	0.25	0.66 (4)
7	18	0.09	0.75 (1)
8	16	0.08	0.83 (1)
9	14	0.07	0.9 (1)
10	20	0.10	1 (0)

Se generan 10 n° aleatorios, p.ej.: 0.16

EJEMPLO MÉTODO ACUMULADO

Ind.	Fitness	Fit. Rel.	\vec{S}
1	22	0.11	0.11 (0)
2	8	0.04	0.15 (0)
3	12	0.06	0.21 (1)
4	36	0.18	0.39 (2)
5	4	0.02	0.41 (0)
6	50	0.25	0.66 (4)
7	18	0.09	0.75 (1)
8	16	0.08	0.83 (1)
9	14	0.07	0.9 (1)
10	20	0.10	1 (0)

Se generan 10 n° aleatorios, p.ej.: 0.86

EJEMPLO MÉTODO ACUMULADO

Ind.	Fitness	Fit. Rel.	\vec{S}
1	22	0.11	0.11 (0)
2	8	0.04	0.15 (0)
3	12	0.06	0.21 (1)
4	36	0.18	0.39 (2)
5	4	0.02	0.41 (0)
6	50	0.25	0.66 (4)
7	18	0.09	0.75 (1)
8	16	0.08	0.83 (1)
9	14	0.07	0.9 (1)
10	20	0.10	1 (0)

- Convergencia prematura:
 - Si un individuo tiene un valor de adecuación muy alto, la población intermedia tendrá muchos clones de dicho individuo
 - Los descendientes se parecerán mucho al progenitor y no se generarán nuevas soluciones
 - El método se estanca en soluciones sub-óptimas
- Estancamiento:
 - Después de unas cuantas generaciones, todos los individuos tienen un valor de adecuación muy parecido entre ellos
 - Entonces $f_i = f_j \forall i, j$ y no hay presión selectiva
 - Todos los individuos tienen la misma probabilidad de ser seleccionados, y se eligen casi aleatoriamente

- Convergencia prematura:
 - Si un individuo tiene un valor de adecuación muy alto, la población intermedia tendrá muchos clones de dicho individuo
 - Los descendientes se parecerán mucho al progenitor y no se generarán nuevas soluciones
 - El método se estanca en soluciones sub-óptimas
- Estancamiento:
 - Después de unas cuantas generaciones, todos los individuos tienen un valor de adecuación muy parecido entre ellos
 - Entonces $f_i = f_j \forall i, j$ y no hay presión selectiva
 - Todos los individuos tienen la misma probabilidad de ser seleccionados, y se eligen casi aleatoriamente

Dependiendo del tipo de reemplazamiento de los descendientes en la población, entre generaciones consecutivas, se distinguen dos tipos de Algoritmos Genéticos:

- **Generacional**. Se reemplaza toda la población
- **Estado Estacionario**. El reemplazamiento es parcial (n individuos). En este caso hay diferentes criterios de reemplazo:
 - Los n individuos con peor valor de adecuación son sustituidos por los nuevos
 - Los nuevos reemplazan a sus n progenitores
 - Los individuos más parecidos. Cada individuo nuevo reemplaza a aquel cuya distancia Hamming sea menor
- Estado estacionario puro. Se reemplaza un único individuo

- Se ordena la población de mayor a menor según fitness:
 $f_i > f_j \forall i > j$
- Se selecciona el individuo proporcionalmete al rango que ocupa en la población
- Se calcula una probabilidad de ser seleccionado según:
 - Esquema lineal: $p_i = A \cdot i + B$
 - Esquema exponencial: $p_i = A \cdot e^{B \cdot i + C}$
- Se eliminan efectos de fitness muy grandes o muy pequeños
- Se uniformiza la diferencia de probabilidades entre los individuos
- Ayuda a mantener la variabilidad genética

- Se ordena la población de mayor a menor según fitness:
 $f_i > f_j \forall i > j$
- Se selecciona el individuo proporcionalmete al rango que ocupa en la población
- Se calcula una probabilidad de ser seleccionado según:
 - Esquema lineal: $p_i = A \cdot i + B$
 - Esquema exponencial: $p_i = A \cdot e^{B \cdot i + C}$
- Se eliminan efectos de fitness muy grandes o muy pequeños
- Se uniformiza la diferencia de probabilidades entre los individuos
- Ayuda a mantener la variabilidad genética

- Se ordena la población de mayor a menor según fitness:
 $f_i > f_j \forall i > j$
- Se selecciona el individuo proporcionalmete al rango que ocupa en la población
- Se calcula una probabilidad de ser seleccionado según:
 - Esquema lineal: $p_i = A \cdot i + B$
 - Esquema exponencial: $p_i = A \cdot e^{B \cdot i + C}$
- Se eliminan efectos de fitness muy grandes o muy pequeños
- Se uniformiza la diferencia de probabilidades entre los individuos
- Ayuda a mantener la variabilidad genética

- Se ordena la población de mayor a menor según fitness:
 $f_i > f_j \forall i > j$
- Se selecciona el individuo proporcionalmete al rango que ocupa en la población
- Se calcula una probabilidad de ser seleccionado según:
 - Esquema lineal: $p_i = A \cdot i + B$
 - Esquema exponencial: $p_i = A \cdot e^{B \cdot i + C}$
- Se eliminan efectos de fitness muy grandes o muy pequeños
- Se uniformiza la diferencia de probabilidades entre los individuos
- Ayuda a mantener la variabilidad genética

- Se ordena la población de mayor a menor según fitness:
 $f_i > f_j \forall i > j$
- Se selecciona el individuo proporcionalmete al rango que ocupa en la población
- Se calcula una probabilidad de ser seleccionado según:
 - Esquema lineal: $p_i = A \cdot i + B$
 - Esquema exponencial: $p_i = A \cdot e^{B \cdot i + C}$
- Se eliminan efectos de fitness muy grandes o muy pequeños
- Se uniformiza la diferencia de probabilidades entre los individuos
- Ayuda a mantener la variabilidad genética

- Se ordena la población de mayor a menor según fitness:
 $f_i > f_j \forall i > j$
- Se selecciona el individuo proporcionalmete al rango que ocupa en la población
- Se calcula una probabilidad de ser seleccionado según:
 - Esquema lineal: $p_i = A \cdot i + B$
 - Esquema exponencial: $p_i = A \cdot e^{B \cdot i + C}$
- Se eliminan efectos de fitness muy grandes o muy pequeños
- Se uniformiza la diferencia de probabilidades entre los individuos
- Ayuda a mantener la variabilidad genética

- Se ordena la población de mayor a menor según fitness:
 $f_i > f_j \forall i > j$
- Se selecciona el individuo proporcionalmete al rango que ocupa en la población
- Se calcula una probabilidad de ser seleccionado según:
 - Esquema lineal: $p_i = A \cdot i + B$
 - Esquema exponencial: $p_i = A \cdot e^{B \cdot i + C}$
- Se eliminan efectos de fitness muy grandes o muy pequeños
- **Se uniformiza la diferencia de probabilidades entre los individuos**
- Ayuda a mantener la variabilidad genética

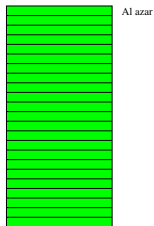
- Se ordena la población de mayor a menor según fitness:
 $f_i > f_j \forall i > j$
- Se selecciona el individuo proporcionalmente al rango que ocupa en la población
- Se calcula una probabilidad de ser seleccionado según:
 - Esquema lineal: $p_i = A \cdot i + B$
 - Esquema exponencial: $p_i = A \cdot e^{B \cdot i + C}$
- Se eliminan efectos de fitness muy grandes o muy pequeños
- Se uniformiza la diferencia de probabilidades entre los individuos
- Ayuda a mantener la variabilidad genética

EJEMPLO MÉTODO JERÁRQUICO

Ind.	Fitness	Ranking	Prob.
6	50	1	0.18
4	36	2	0.16
1	22	3	0.15
10	20	4	0.13
7	18	5	0.11
8	16	6	0.09
9	14	7	0.07
3	12	8	0.05
2	8	9	0.04
5	4	10	0.02

Se calcula la probabilidad según esquema lineal con $A = \frac{1}{55}$ y $B = 0$
La probabilidad es independiente del fitness, depende del ranking

REPRODUCCIÓN-TORNEOS



Poblacion (t)

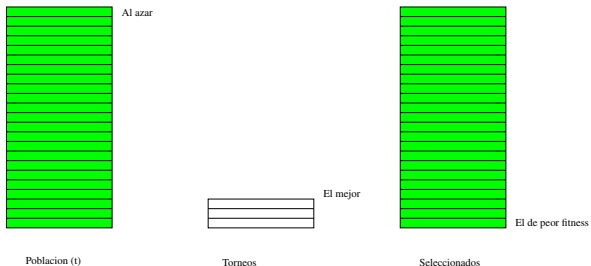
Torneos



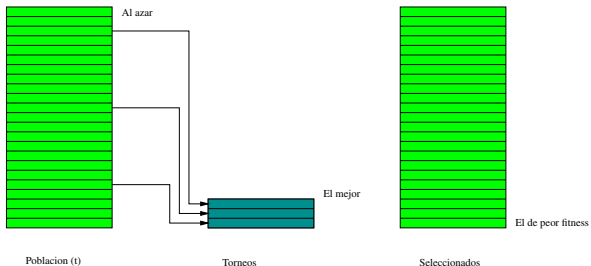
Seleccionados

REPRODUCCIÓN-TORNEOS

- Se asigna un tamaño de torneos (t). Normalmente pequeño

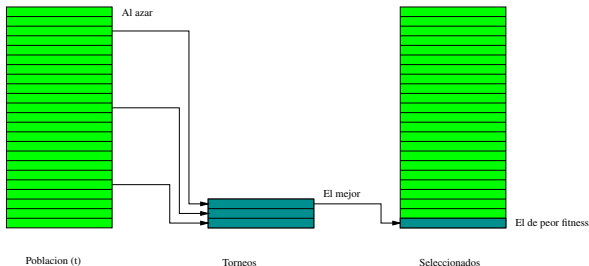


REPRODUCCIÓN-TORNEOS



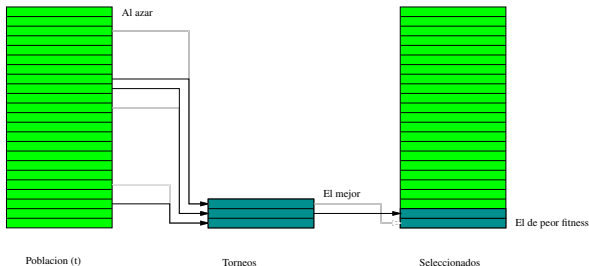
- Se asigna un tamaño de torneos (t). Normalmente pequeño
- Se seleccionan t individuos de manera uniforme

REPRODUCCIÓN-TORNEOS



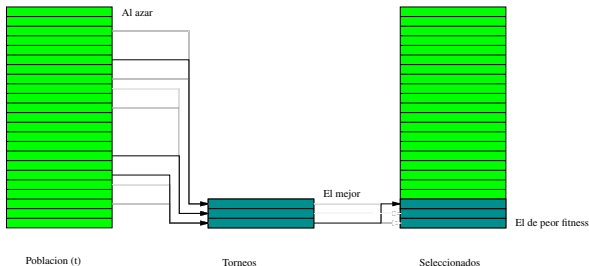
- Se asigna un tamaño de torneos (t). Normalmente pequeño
- Se seleccionan t individuos de manera uniforme
- Se incluye en la siguiente población el de mayor fitness

REPRODUCCIÓN-TORNEOS



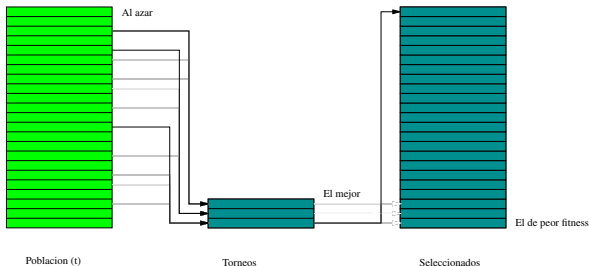
- Se asigna un tamaño de torneos (t). Normalmente pequeño
- Se seleccionan t individuos de manera uniforme
- Se incluye en la siguiente población el de mayor fitness
- Se repite el proceso tantas veces como sea el tamaño población

REPRODUCCIÓN-TORNEOS



- Se asigna un tamaño de torneos (t). Normalmente pequeño
- Se seleccionan t individuos de manera uniforme
- Se incluye en la siguiente población el de mayor fitness
- Se repite el proceso tantas veces como sea el tamaño población

REPRODUCCIÓN-TORNEOS

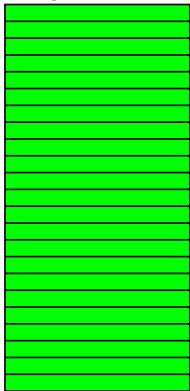


- Se asigna un tamaño de torneos (t). Normalmente pequeño
- Se seleccionan t individuos de manera uniforme
- Se incluye en la siguiente población el de mayor fitness
- Se repite el proceso tantas veces como sea el tamaño población

- El tamaño de torneos permite asignar la presión selectiva
- No es necesaria función de evaluación, solo comparar individuos
- Es fácil pasar de exploración a explotación
- Permite mantener y regular la variabilidad genética
- Se elimina el efecto de los super-individuos
- Se ajusta fácilmente a problemas multiobjetivo

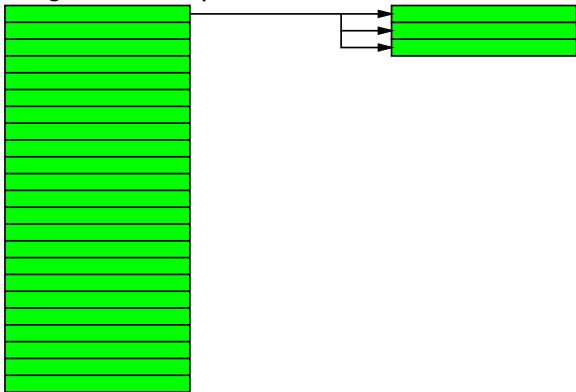
OPERADOR DE REPRODUCCIÓN

Se genera una población intermedia del mismo tamaño que la original:



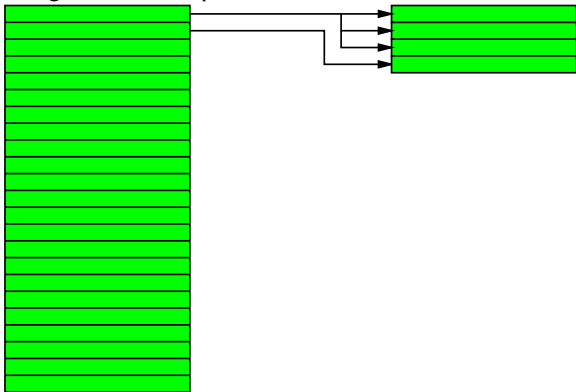
Poblacion (t)

Se genera una población intermedia del mismo tamaño que la original:



Poblacion (t)

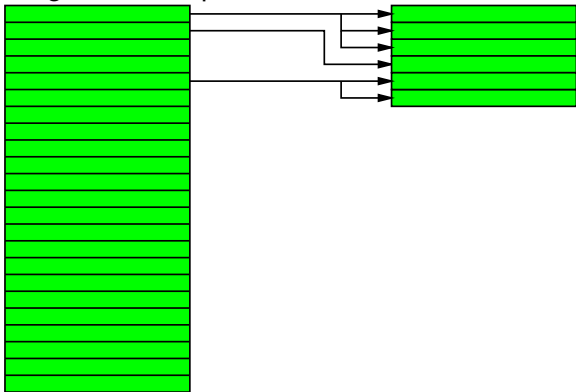
Se genera una población intermedia del mismo tamaño que la original:



Poblacion (t)

OPERADOR DE REPRODUCCIÓN

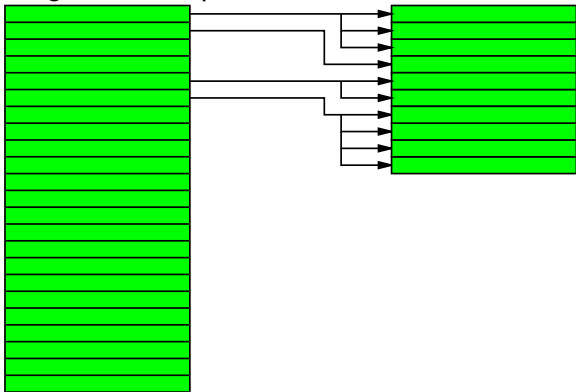
Se genera una población intermedia del mismo tamaño que la original:



Poblacion (t)

OPERADOR DE REPRODUCCIÓN

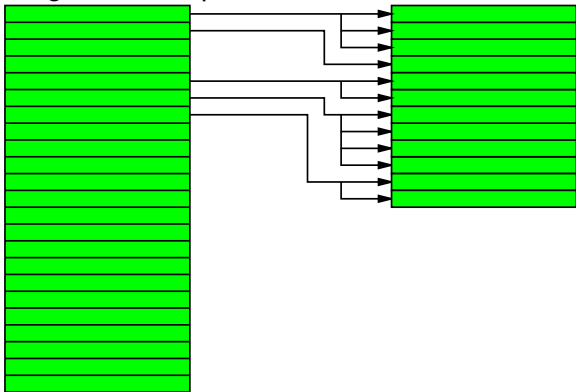
Se genera una población intermedia del mismo tamaño que la original:



Poblacion (t)

OPERADOR DE REPRODUCCIÓN

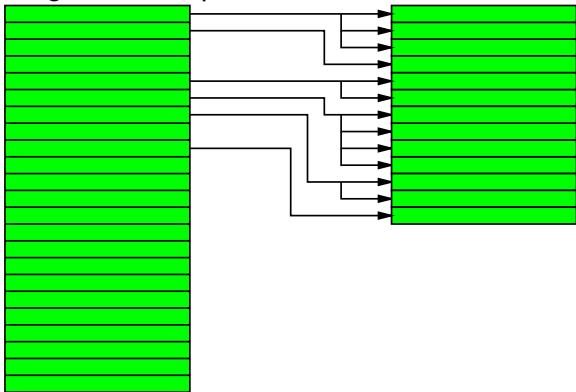
Se genera una población intermedia del mismo tamaño que la original:



Poblacion (t)

OPERADOR DE REPRODUCCIÓN

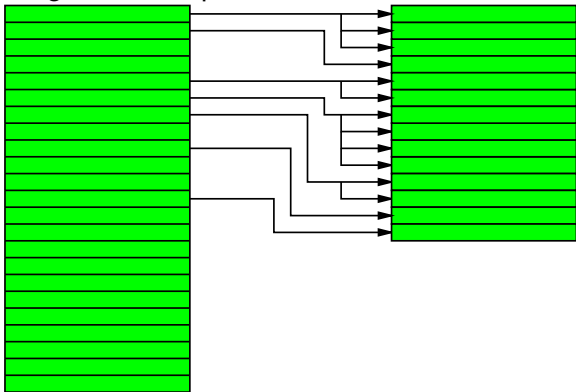
Se genera una población intermedia del mismo tamaño que la original:



Poblacion (t)

OPERADOR DE REPRODUCCIÓN

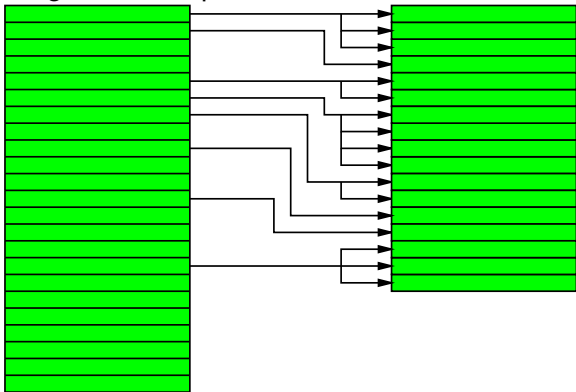
Se genera una población intermedia del mismo tamaño que la original:



Poblacion (t)

OPERADOR DE REPRODUCCIÓN

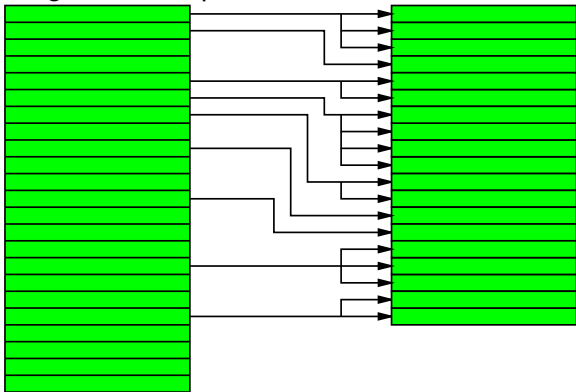
Se genera una población intermedia del mismo tamaño que la original:



Poblacion (t)

OPERADOR DE REPRODUCCIÓN

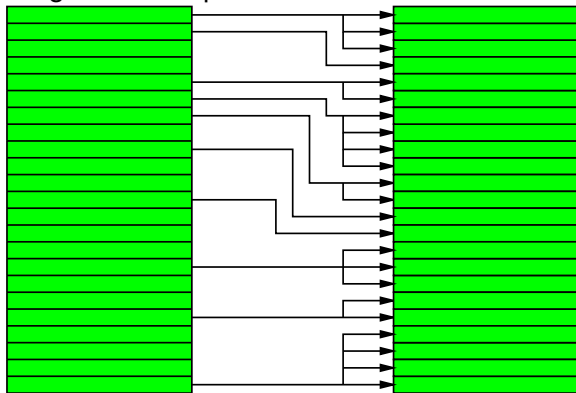
Se genera una población intermedia del mismo tamaño que la original:



Poblacion (t)

OPERADOR DE REPRODUCCIÓN

Se genera una población intermedia del mismo tamaño que la original:

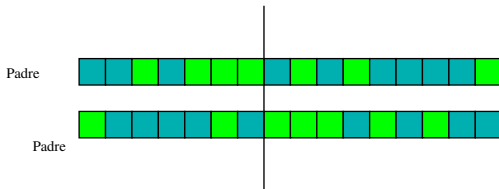


Poblacion (t)

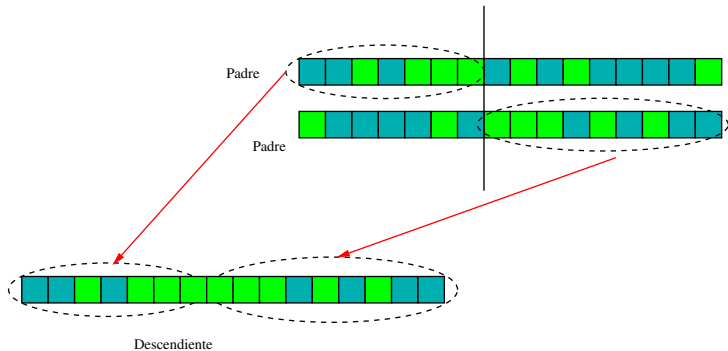
Operador de seleccion

Poblacion
intermedia

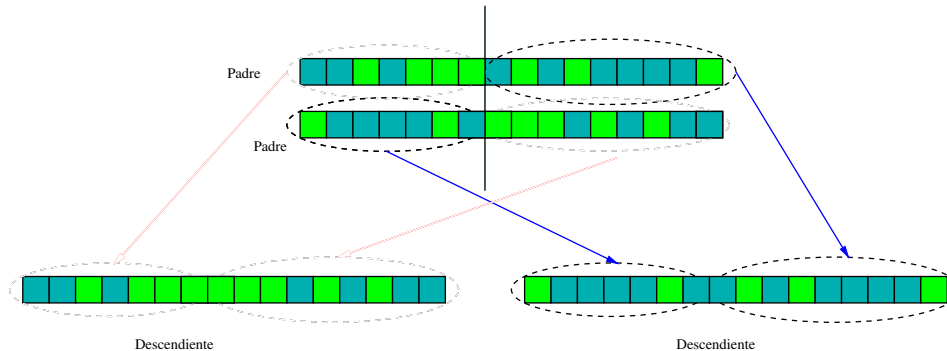
CRUCE SIMPLE



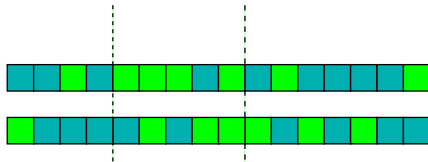
CRUCE SIMPLE



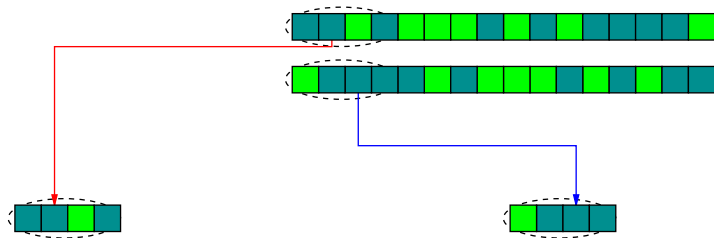
CRUCE SIMPLE



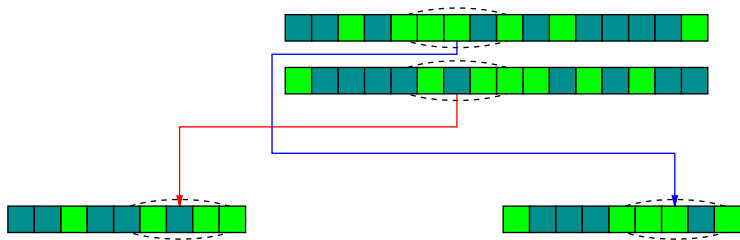
CRUCE MULTIPUNTO



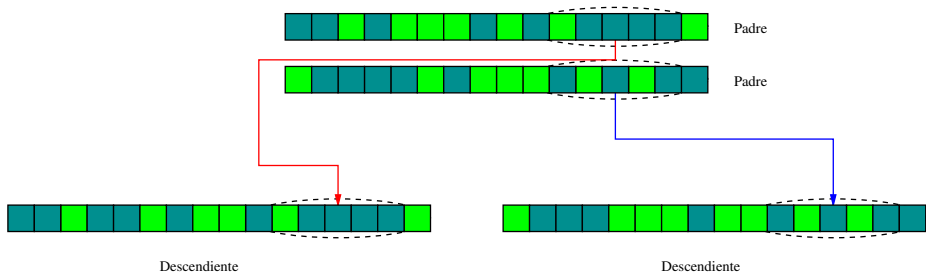
CRUCE MULTIPUNTO



CRUCE MULTIPUNTO



CRUCE MULTIPUNTO



CRUCE UNIFORME

Padre



Padre



Descendiente

Descendiente

CRUCE UNIFORME

Padre



Descendiente

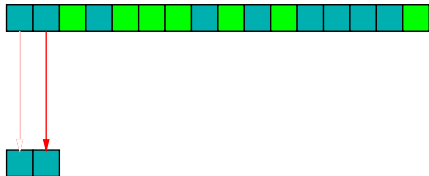
Padre



Descendiente

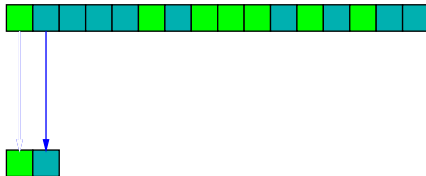
CRUCE UNIFORME

Padre



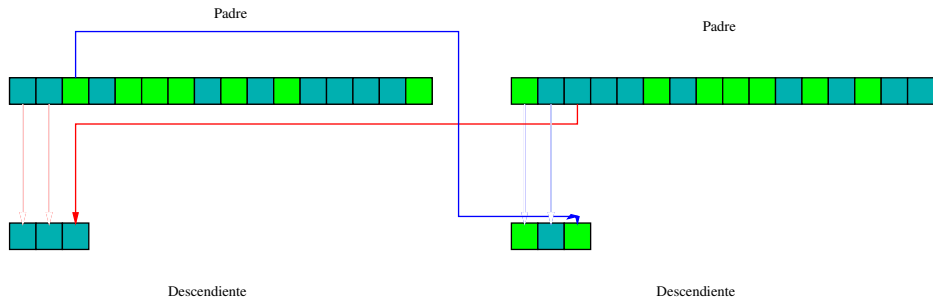
Descendiente

Padre

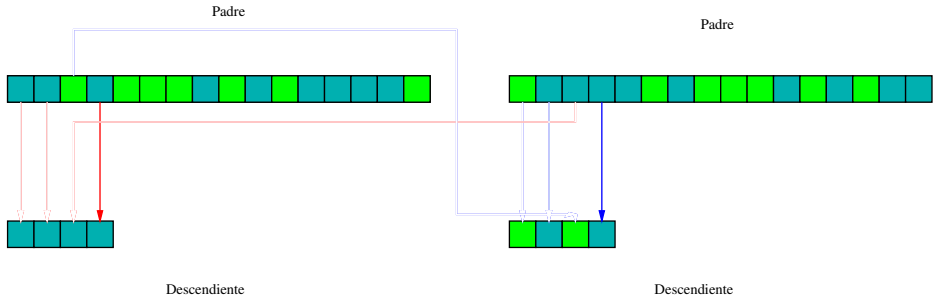


Descendiente

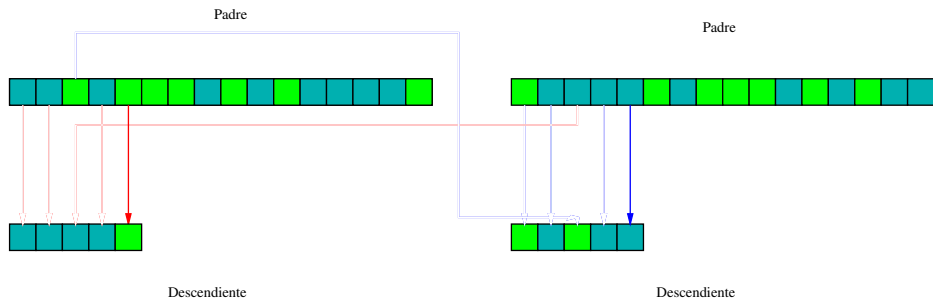
CRUCE UNIFORME



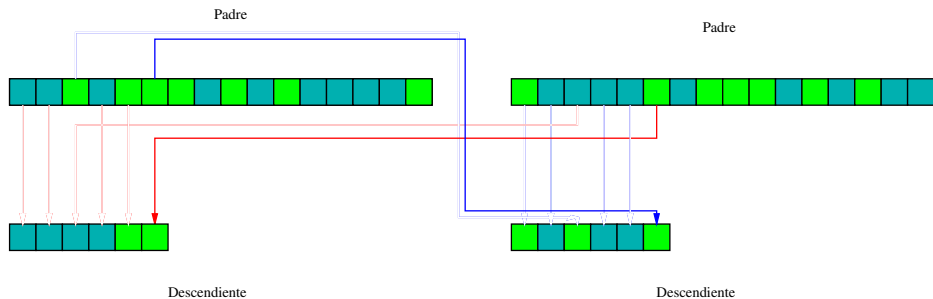
CRUCE UNIFORME



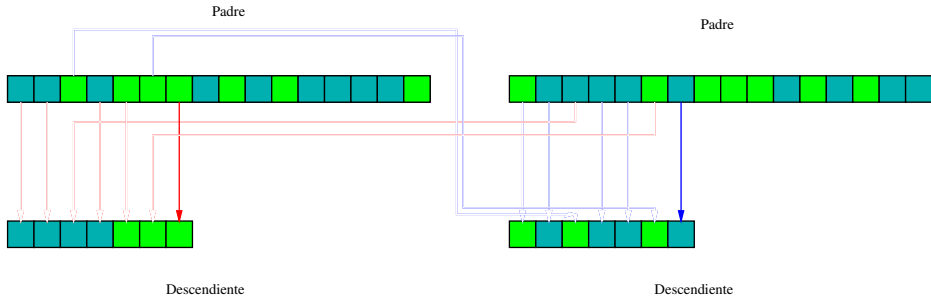
CRUCE UNIFORME



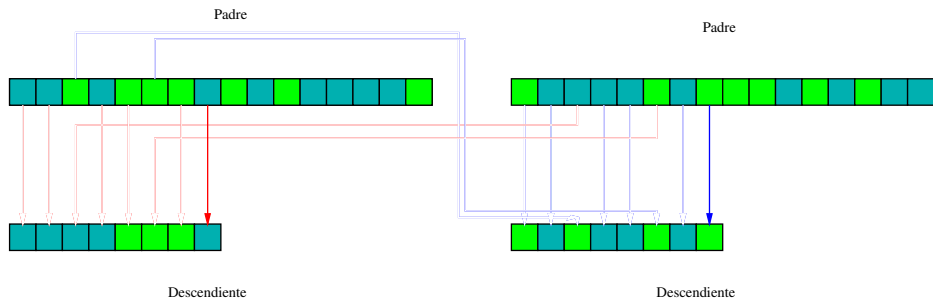
CRUCE UNIFORME



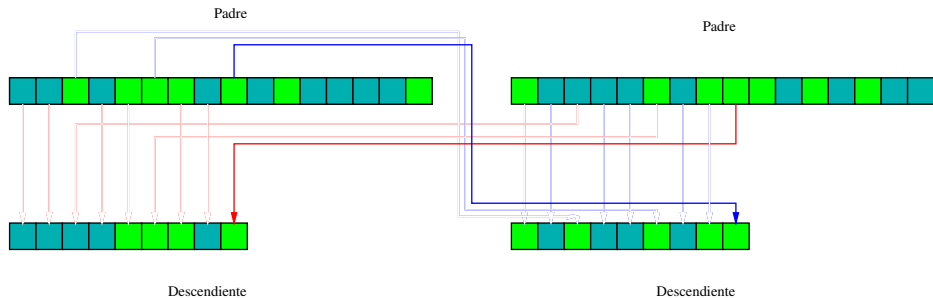
CRUCE UNIFORME



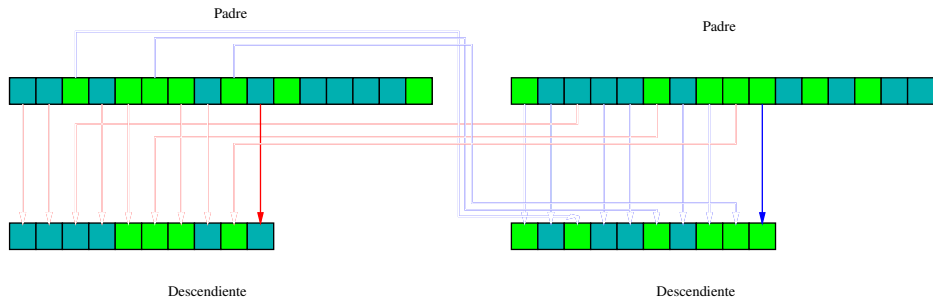
CRUCE UNIFORME



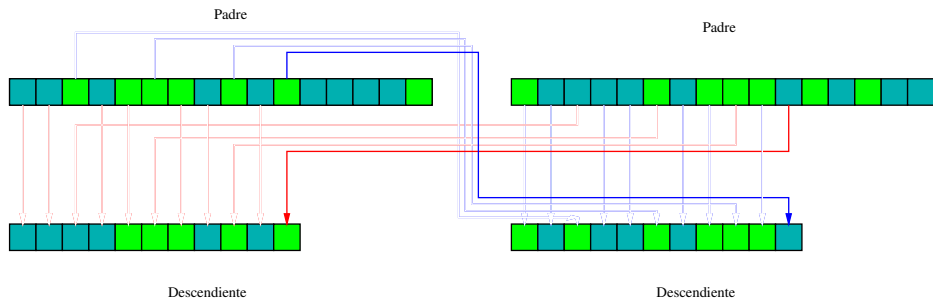
CRUCE UNIFORME



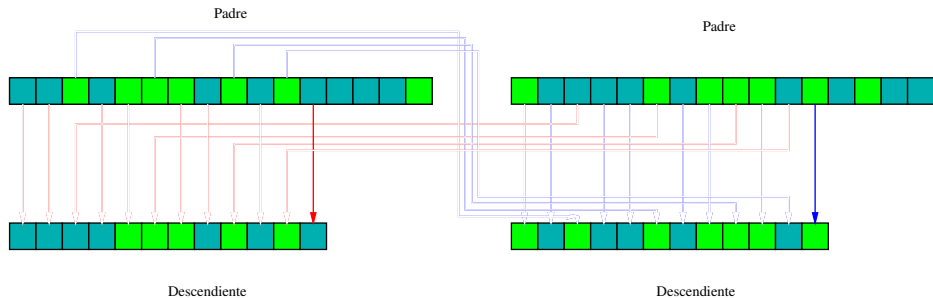
CRUCE UNIFORME



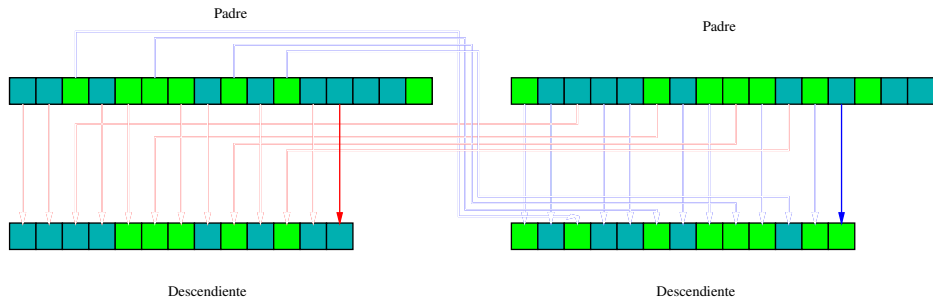
CRUCE UNIFORME



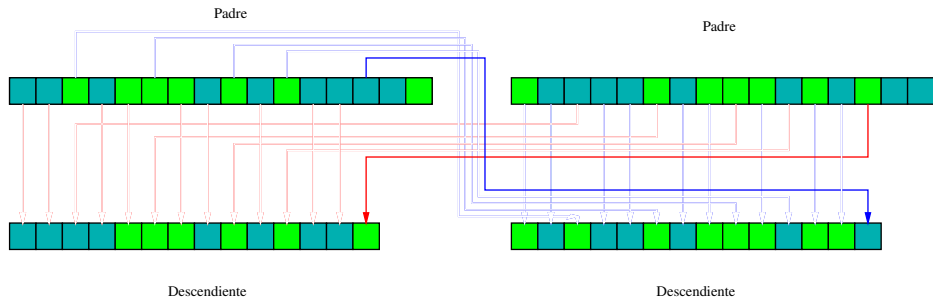
CRUCE UNIFORME



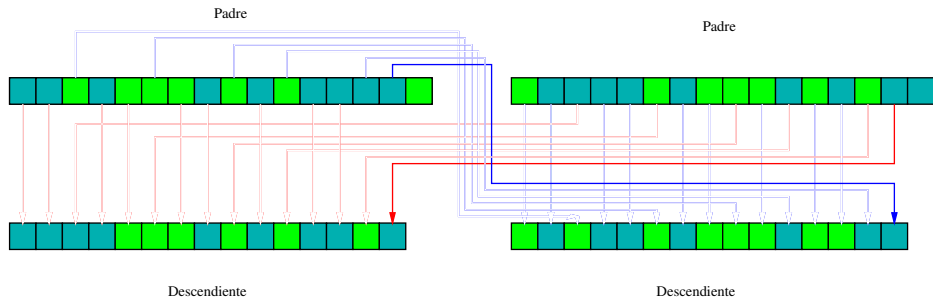
CRUCE UNIFORME



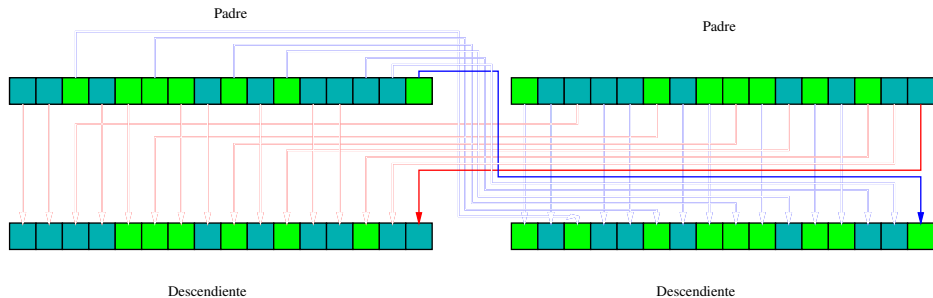
CRUCE UNIFORME



CRUCE UNIFORME

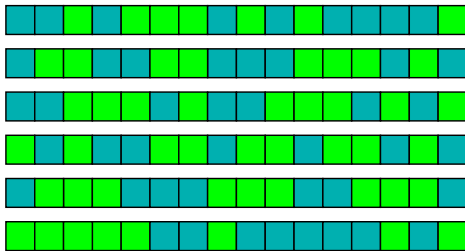


CRUCE UNIFORME



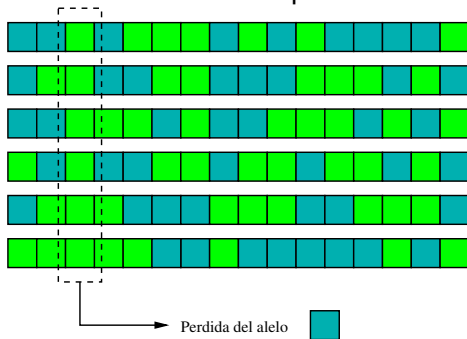
EL PROBLEMA DE LOS INCESTOS

- Si se hace sólo cruce puede ocurrir que se pierdan alelos:



EL PROBLEMA DE LOS INCESTOS

- Si se hace sólo cruce puede ocurrir que se pierdan alelos:



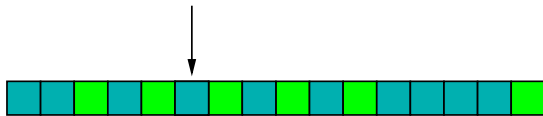
- Se pierde rápidamente variabilidad genética y se produce convergencia prematura
- Hay que introducir un nuevo operador: Mutación

MUTACIÓN E INVERSIÓN



Mutacion

MUTACIÓN E INVERSIÓN

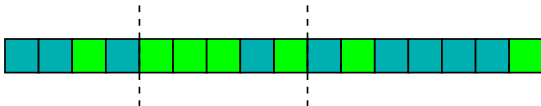


Mutacion

MUTACIÓN E INVERSIÓN

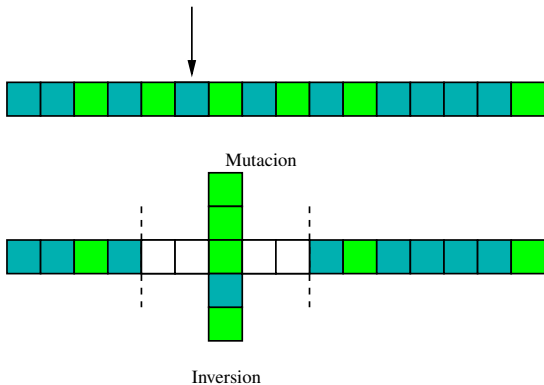


Mutación

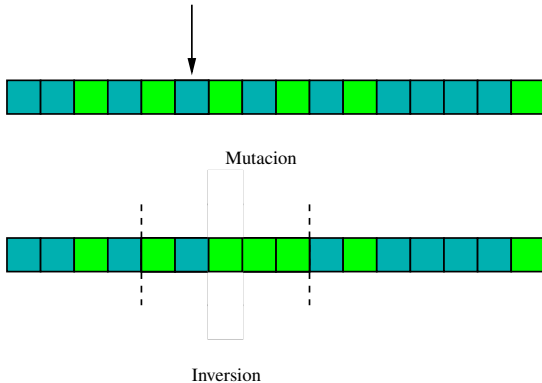


Inversion

MUTACIÓN E INVERSIÓN



MUTACIÓN E INVERSIÓN



- Evita la pérdida de alelos
- Aleatoriza la búsqueda. Probabilidades altas hace que la búsqueda se convierta en ciega
- Realiza exploración, frente a cruce que realiza explotación
- Al principio es necesaria la exploración, a medida que se obtienen soluciones buenas hay que cambiar a explotación
- Al principio mutación alta, al final baja
- Valores bajos también hacen caer en mínimos locales de los que es difícil salir
- Hay que ajustar bien la probabilidad para evitar todo esto

- Si hay que minimizar ($f(i_1) > f(i_2) \Rightarrow i_1$ mejor adaptado que i_2).-
Se modifica la función de fitness por una complementaria: $f' = -f$
o si $0 \leq f' \leq 1 \Rightarrow f = 1 - f'$
- Si hay valores negativos

$$f(i) = \begin{cases} F_{max} - f'(i) & \text{cuando } f'(i) < F_{max} \\ 0 & \text{en caso contrario} \end{cases}$$

- Si hay que minimizar y hay valores negativos

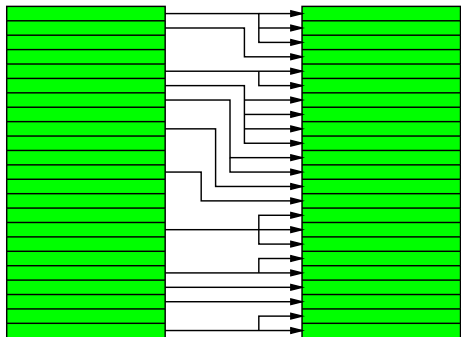
$$f(i) = \begin{cases} f(i) + F_{min} & \text{cuando } f(i) + F_{min} > 0 \\ 0 & \text{en caso contrario} \end{cases}$$

ESQUEMA DE FUNCIONAMIENTO



Poblacion (t)

ESQUEMA DE FUNCIONAMIENTO

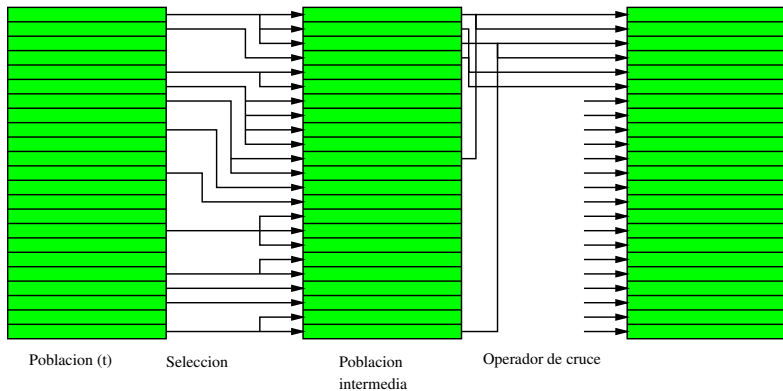


Poblacion (t)

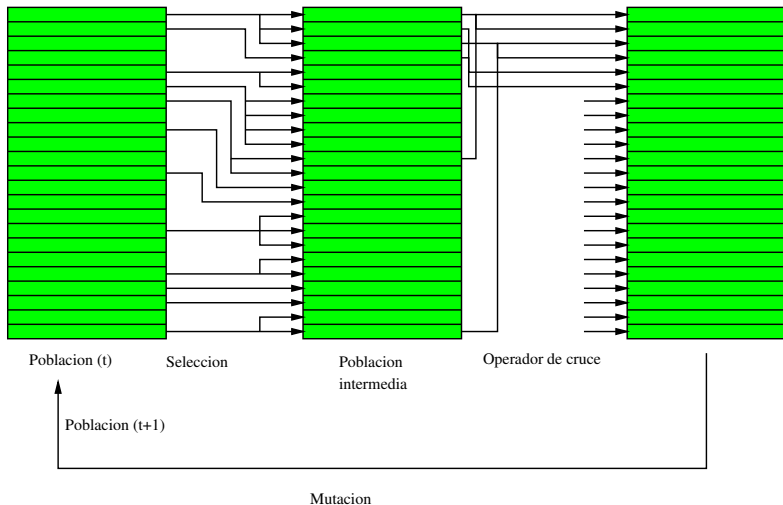
Seleccion

Poblacion
intermedia

ESQUEMA DE FUNCIONAMIENTO



ESQUEMA DE FUNCIONAMIENTO



1 INTRODUCCIÓN

2 ALGORITMOS GENÉTICOS

- Introducción a los Algoritmos Genéticos
- Algoritmos Généticos Canónicos
- **Ejemplo de Algoritmo Genético**
- Propiedades de los Algoritmos Genéticos
- Métodos de compartición y soluciones múltiples
- Manejo de restricciones en Algoritmos Genéticos
- Fundamentos de los Algoritmos Genéticos

3 COMPUTACIÓN EVOLUTIVA

4 COMPUTACIÓN CON INSPIRACIÓN BIOLÓGICA

5 BIOLOGÍA Y COMPUTACIÓN

1 INTRODUCCIÓN

2 ALGORITMOS GENÉTICOS

- Introducción a los Algoritmos Genéticos
- Algoritmos Génticos Canónicos
- Ejemplo de Algoritmo Genético
- **Propiedades de los Algoritmos Genéticos**
- Métodos de compartición y soluciones múltiples
- Manejo de restricciones en Algoritmos Genéticos
- Fundamentos de los Algoritmos Genéticos

3 COMPUTACIÓN EVOLUTIVA

4 COMPUTACIÓN CON INSPIRACIÓN BIOLÓGICA

5 BIOLOGÍA Y COMPUTACIÓN

1 INTRODUCCIÓN

2 ALGORITMOS GENÉTICOS

- Introducción a los Algoritmos Genéticos
- Algoritmos Génticos Canónicos
- Ejemplo de Algoritmo Genético
- Propiedades de los Algoritmos Genéticos
- **Métodos de compartición y soluciones múltiples**
- Manejo de restricciones en Algoritmos Genéticos
- Fundamentos de los Algoritmos Genéticos

3 COMPUTACIÓN EVOLUTIVA

4 COMPUTACIÓN CON INSPIRACIÓN BIOLÓGICA

5 BIOLOGÍA Y COMPUTACIÓN

1 INTRODUCCIÓN

2 ALGORITMOS GENÉTICOS

- Introducción a los Algoritmos Genéticos
- Algoritmos Génticos Canónicos
- Ejemplo de Algoritmo Genético
- Propiedades de los Algoritmos Genéticos
- Métodos de compartición y soluciones múltiples
- **Manejo de restricciones en Algoritmos Genéticos**
- Fundamentos de los Algoritmos Genéticos

3 COMPUTACIÓN EVOLUTIVA

4 COMPUTACIÓN CON INSPIRACIÓN BIOLÓGICA

5 BIOLOGÍA Y COMPUTACIÓN

1 INTRODUCCIÓN

2 ALGORITMOS GENÉTICOS

- Introducción a los Algoritmos Genéticos
- Algoritmos Genéticos Canónicos
- Ejemplo de Algoritmo Genético
- Propiedades de los Algoritmos Genéticos
- Métodos de compartición y soluciones múltiples
- Manejo de restricciones en Algoritmos Genéticos
- **Fundamentos de los Algoritmos Genéticos**

3 COMPUTACIÓN EVOLUTIVA

4 COMPUTACIÓN CON INSPIRACIÓN BIOLÓGICA

5 BIOLOGÍA Y COMPUTACIÓN

1 INTRODUCCIÓN

2 ALGORITMOS GENÉTICOS

- Introducción a los Algoritmos Genéticos
- Algoritmos Génticos Canónicos
- Ejemplo de Algoritmo Genético
- Propiedades de los Algoritmos Genéticos
- Métodos de compartición y soluciones múltiples
- Manejo de restricciones en Algoritmos Genéticos
- Fundamentos de los Algoritmos Genéticos

3 COMPUTACIÓN EVOLUTIVA

4 COMPUTACIÓN CON INSPIRACIÓN BIOLÓGICA

5 BIOLOGÍA Y COMPUTACIÓN