

METAHEURÍSTICAS

Tema 2.2: ALGORITMOS GENÉTICOS I. CONCEPTOS BÁSICOS

Huelva, Febrero de 2015 V1.0





Tema 2.2: Algoritmos Genéticos I

Tema 2.2: Algoritmos Genéticos I. Conceptos Básicos

Índice:

1.1 Introducción

1.2 Modelos Generacional vs. Estacionario

1.3 ¿Cómo se construye un AG?

1.4 Utilización de AGs

1.5 Ejemplo

1.6 Conclusiones

1.7 Bibliografía



Tema 2.2: Algoritmos Genéticos I

Tema 2.2: Algoritmos Genéticos I. Conceptos Básicos

Índice:

1.1 Introducción

1.2 Modelos Generacional vs. Estacionario

1.3 ¿Cómo se construye un AG?

1.4 Utilización de AGs

1.5 Ejemplo

1.6 Conclusiones

1.7 Bibliografía



Tema 2.2: Algoritmos Genéticos I

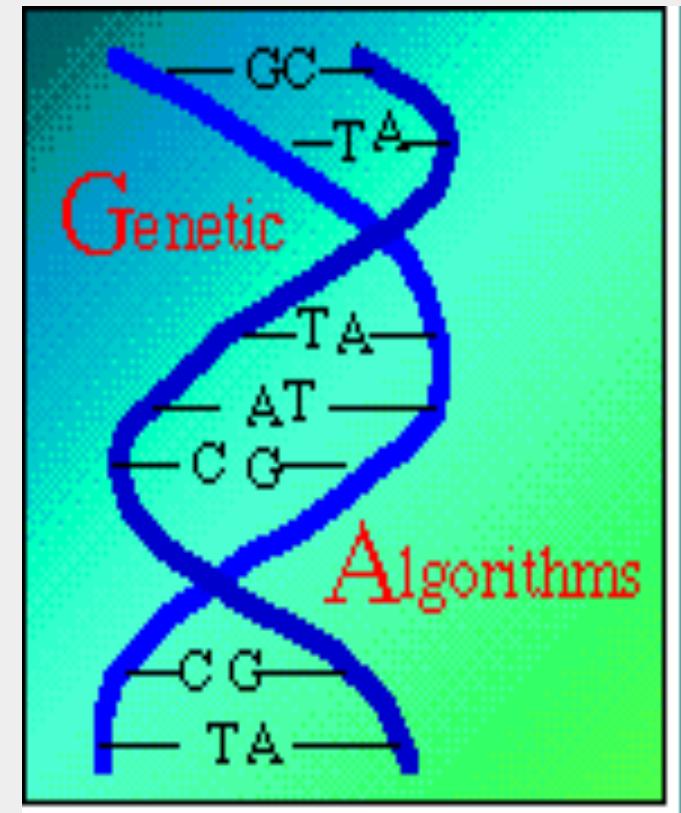
1.1 Introducción

¿Qué es un AG?

Los Algoritmos Genéticos

Son Algoritmos de *Optimización*,
Búsqueda y Aprendizaje
inspirados en los procesos de la

Evolución Natural
y la
Evolución Genética

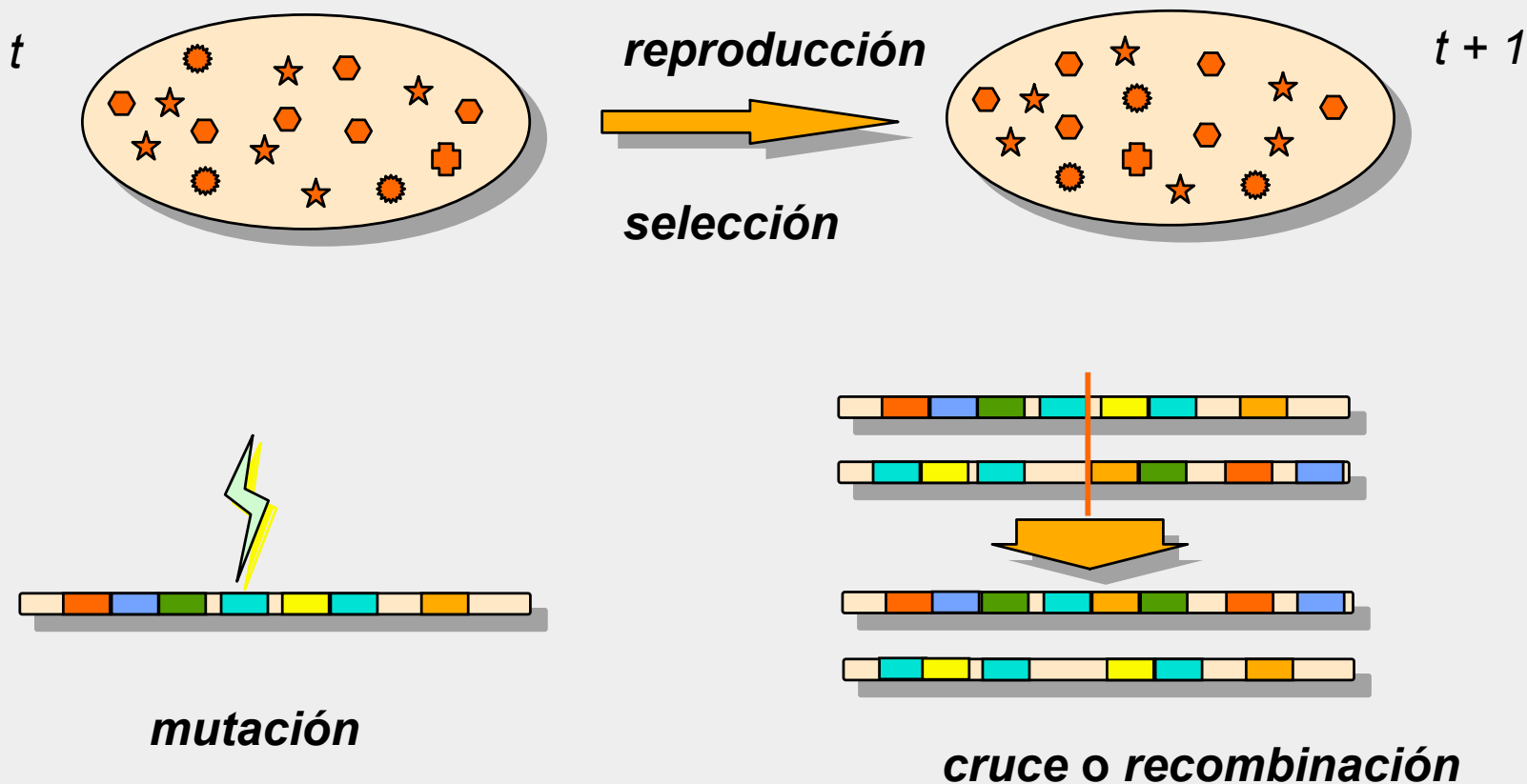




Tema 2.2: Algoritmos Genéticos I

1.1 Introducción

Ingredientes

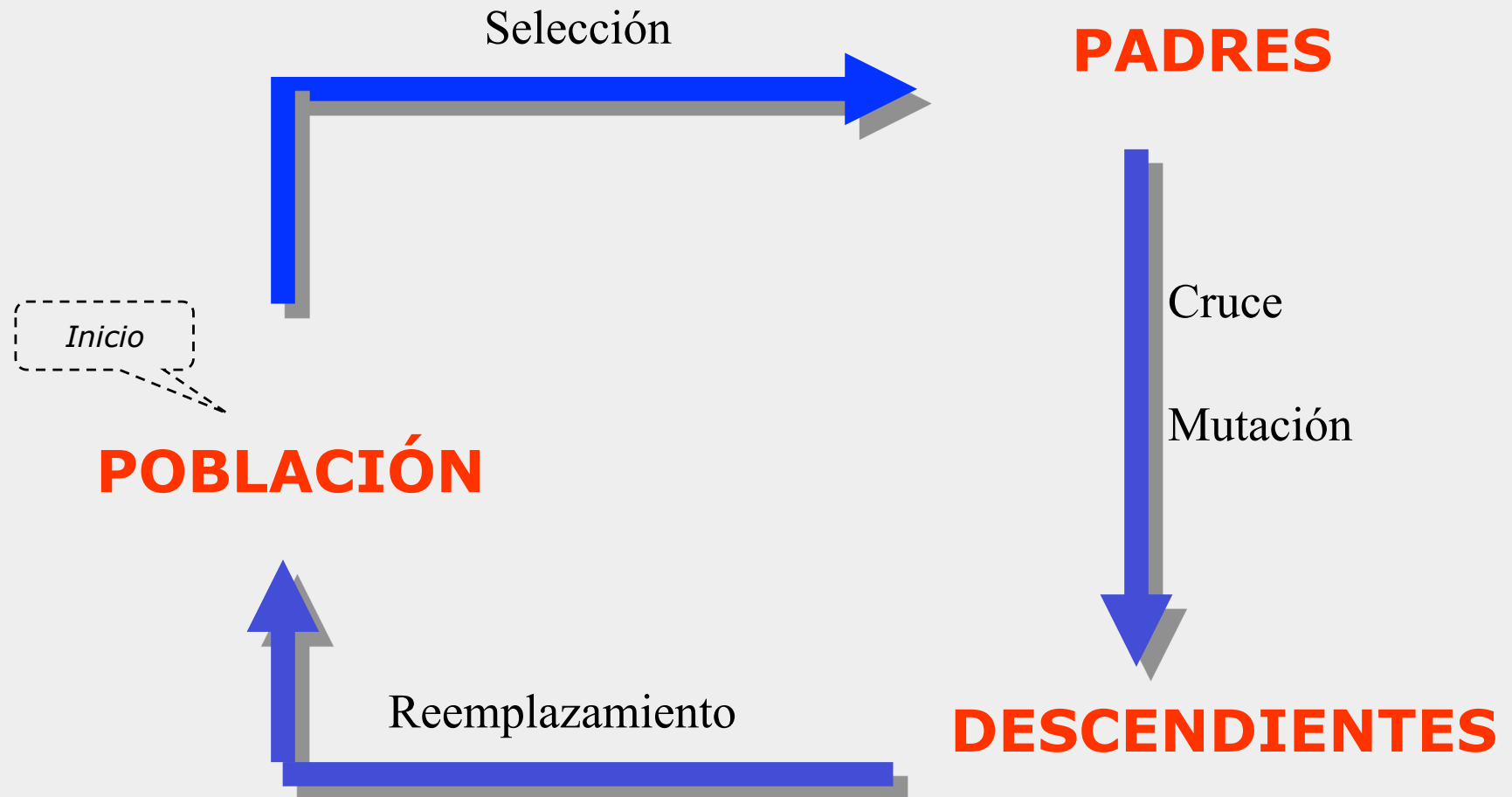




Tema 2.2: Algoritmos Genéticos I

1.1 Introducción

El ciclo de la Evolución





Tema 2.2: Algoritmos Genéticos I

1.1 Introducción

Estructura de un Algoritmo Genético

Procedimiento Algoritmo Genético

Inicio (1)

$t = 0$;

Inicializar $P(t)$;

Evaluar $P(t)$;

Mientras (no se cumpla la condición de parada) hacer

 Inicio(2)

$t = t + 1$

 Seleccionar $P(t)$ desde $P(t-1)$

 Recombinar $P(t)$

 Mutar $P(t)$

 Evaluar $P(t)$

 Final(2)

Final(1)



Tema 2.2: Algoritmos Genéticos I

Tema 2.2: Algoritmos Genéticos I. Conceptos Básicos

Índice:

1.1 Introducción

1.2 Modelos Generacional vs. Estacionario

1.3 ¿Cómo se construye un AG?

1.4 Utilización de AGs

1.5 Ejemplo

1.6 Conclusiones

1.7 Bibliografía



Tema 2.2: Algoritmos Genéticos I

1.2 Modelos: Generacional vs. Estacionario

Modelo Generacional : Durante cada iteración se crea una población completa con nuevos individuos.

La nueva población reemplaza directamente a la antigua.

Modelo Estacionario : Durante cada iteración se escogen dos padres de la población (diferentes mecanismos de muestreo) y se les aplican los operadores genéticos.

El/los descendiente/s reemplazan a uno/dos cromosoma/s de la población inicial.

El modelo estacionario produce una **presión selectiva alta** (convergencia rápida) cuando se reemplazan los peores cromosomas de la población.



Tema 2.2: Algoritmos Genéticos I

Tema 2.2: Algoritmos Genéticos I. Conceptos Básicos

Índice:

1.1 Introducción

1.2 Modelos Generacional vs. Estacionario

1.3 ¿Cómo se construye un AG?

1.4 Utilización de AGs

1.5 Ejemplo

1.6 Conclusiones

1.7 Bibliografía



Tema 2.2: Algoritmos Genéticos I

1.3 ¿Cómo se construye un AG?

Pasos para Construir un AG

- Diseñar una representación
- Decidir cómo reiniciar una población
- Diseñar una correspondencia entre genotipo y fenotipo
- Diseñar una forma de evaluar un individuo
- Diseñar un operador de mutación adecuado
- Diseñar un operador de cruce adecuado
- Decidir cómo seleccionar los individuos para ser padres
- Decidir cómo reemplazar a los individuos
- Decidir la condición de parada

DEPENDE DEL
PROBLEMA

COMPONENTES
DEL ALGORITMO



Tema 2.2: Algoritmos Genéticos I

1.3 ¿Cómo se construye un AG?

Diseñar una representación

- Debemos disponer de un **mecanismo para codificar un individuo** como un **genotipo** (conjunto de genes).
- Existen **muchas maneras** de hacer esto y se ha de **elegir la más relevante** para el problema en cuestión.
- Una vez elegida una representación, hemos de tener en mente como los genotipos (codificación) serán evaluados y qué **operadores genéticos** hay que utilizar.

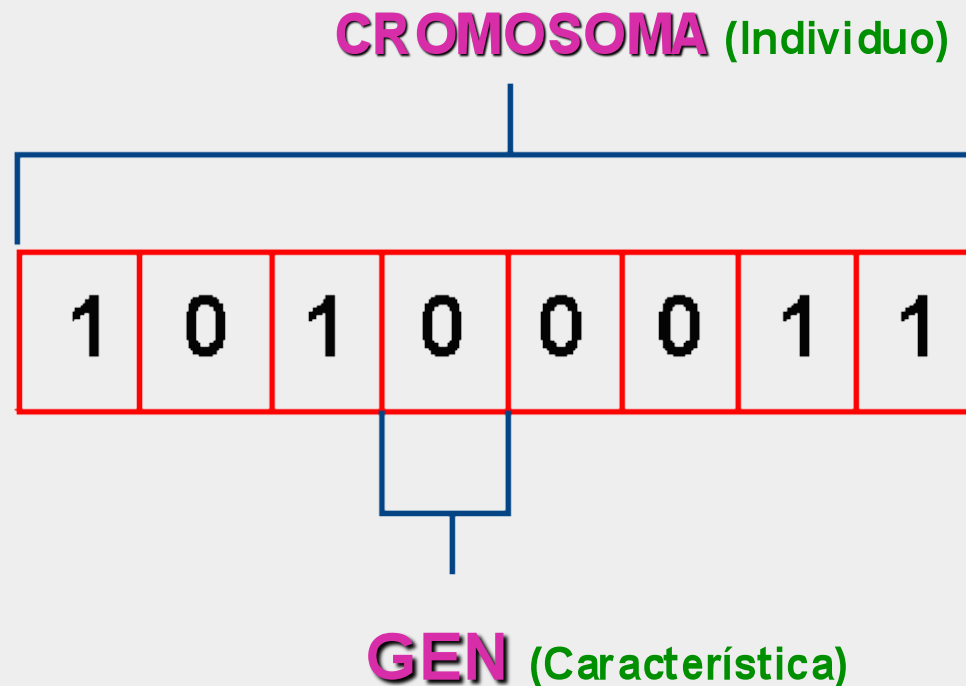
Tema 2.2: Algoritmos Genéticos I

1.3 ¿Cómo se construye un AG?

Diseñar una representación (II)

Representación Binaria

- La representación de un individuo se puede hacer mediante una codificación discreta, y en particular binaria.





Tema 2.2: Algoritmos Genéticos I

1.3 ¿Cómo se construye un AG?

Diseñar una representación (III)

Representación Binaria (II)

Límites de la Metáfora:

- ❑ **Biología:**
 - ❑ **Genotipo o genoma:** Conjunto de genes que caracterizan a una especie.
 - ❑ **Fenotipo:** Un individuo real de una especie concreta, es decir, valores para los genes de un Genotipo + Caracteres morfológicos no ubicados en los genes + conductas aprendidas, etc.
- ❑ **Algoritmos Genéticos:**
 - ❑ Genotipo: **Mecanismo para codificar un individuo**, ejemplo: *usar 8 bits*.
 - ❑ Fenotipo: **Lo que se puede codificar en un conjunto de genes**, ejemplo: un entero, un real, etc.
 - ❑ Un mismo Genotipo, puede tener distintos Fenotipos.
- ❑ **Conclusión:**
 - ❑ **Metáfora un tanto forzada en cuanto a la nomenclatura de fenotipos.**

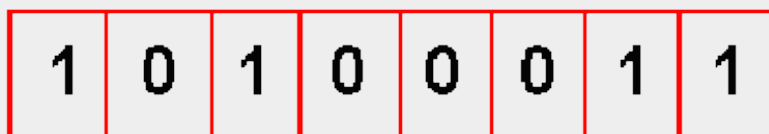
Tema 2.2: Algoritmos Genéticos I

1.3 ¿Cómo se construye un AG?

Diseñar una representación (IV)

Representación Binaria (III)

8 bits Genotipo



Fenotipo

- Entero
- Número real
- Secuencia
- ...
- ¿Cualquier otra?



Tema 2.2: Algoritmos Genéticos I

1.3 ¿Cómo se construye un AG?

Diseñar una representación (V)

Representación Binaria (IV)

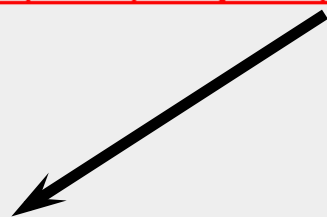
- El fenotipo pueden ser números enteros

Genotipo:

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

Fenotipo:

= 163



$$1 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 =$$
$$128 + 32 + 2 + 1 = 163$$



Tema 2.2: Algoritmos Genéticos I

1.3 ¿Cómo se construye un AG?

Diseñar una representación (VI)

Representación Binaria (V)

- El fenotipo también pueden ser números reales.
Ejemplo: un número entre 1.5 y 20.5 utilizando 8 dígitos binarios

Genotipo:

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

Fenotipo:

= 13.9609


$$x = 2.5 + \frac{163}{256} (20.5 - 2.5) = 13.9609$$



Tema 2.2: Algoritmos Genéticos I

1.3 ¿Cómo se construye un AG?

Diseñar una representación (VII)

Representación Real

- Una forma natural de codificar una solución es utilizando valores reales como genes.
- Aunque originalmente los AGs se concibieron con codificación binaria, muchas aplicaciones tienen esta forma natural de codificación.
- Los individuos se representan como vectores de valores reales:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, x_i \in R$$

- La función de evaluación asocia a un vector un valor real de evaluación:

$$f : R^n \rightarrow R$$



Tema 2.2: Algoritmos Genéticos I

1.3 ¿Cómo se construye un AG?

Diseñar una representación (VIII)

Representación de Orden

- Los individuos se representan como permutaciones.

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

- Se utilizan para problemas de secuenciación.
 - Ejemplo famoso: Viajante de Comercio, donde cada ciudad tiene asignado un único número entre 1 y n.
- Necesita operadores especiales para garantizar que el resultado de aplicar un operador sigue siendo una permutación.





Tema 2.2: Algoritmos Genéticos I

1.3 ¿Cómo se construye un AG?

Decidir cómo reiniciar (*inicializar*) una población

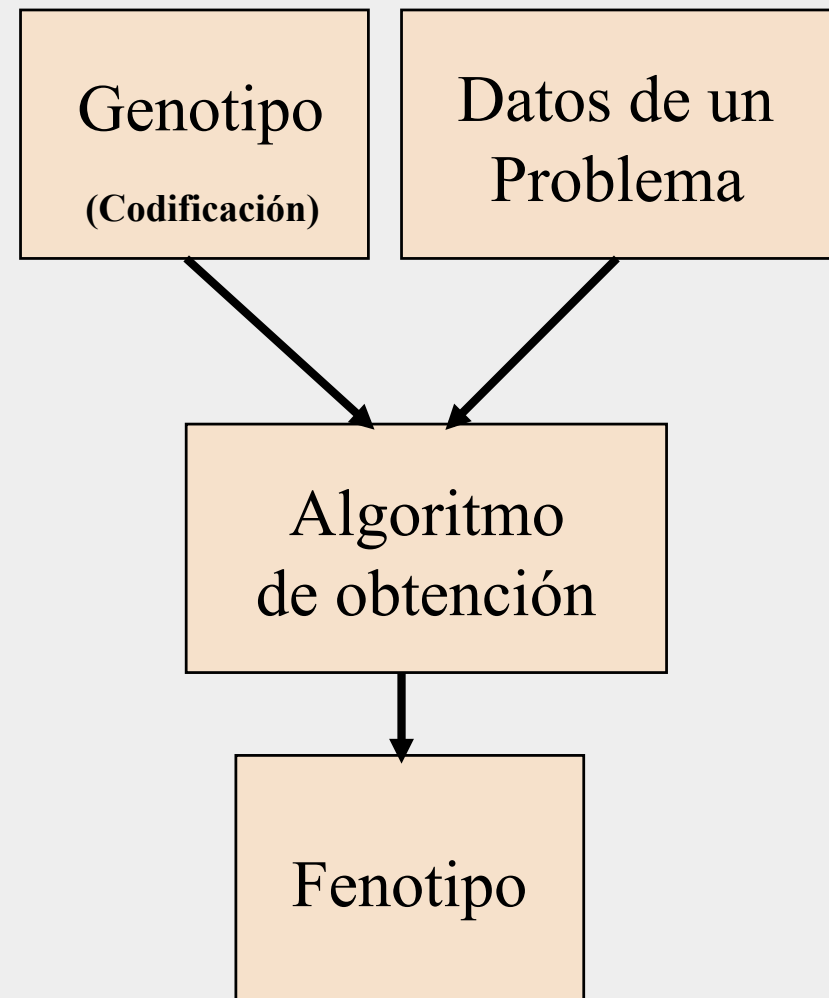
- A **ser posible**, elegida de modo uniforme sobre el espacio de búsqueda:
 - Representación Binaria: 0 ó 1 con probabilidad 0.5
 - Representación Real: uniforme sobre un intervalo dado (para valores acotados).
- Elegir la población (al menos **algún individuo**) a partir de los **resultados de una heurística previa**.

Tema 2.2: Algoritmos Genéticos I

1.3 ¿Cómo se construye un AG?

Diseñar una correspondencia entre genotipo y fenotipo

- Algunas veces la obtención del fenotipo a partir del genotipo es un proceso obvio.
- En otras ocasiones el genotipo puede ser un conjunto de parámetros para algún algoritmo, el cual trabaja sobre los datos de un problema para obtener un fenotipo.





Tema 2.2: Algoritmos Genéticos I

1.3 ¿Cómo se construye un AG?

Diseñar una forma de evaluar un individuo

- Este es el paso más **costoso** (computacionalmente) para una aplicación real.
- Puede ser una **subrutina**, un **simulador**, o **cualquier proceso externo** (ej. experimentos en un robot,)
- Se pueden utilizar **funciones aproximadas** para reducir el costo de evaluación.
- Cuando hay **restricciones**, éstas se pueden **introducir en el costo** como **penalización**.
- Si hay **múltiples objetivos** se busca una **solución de compromiso**.

Tema 2.2: Algoritmos Genéticos I

1.3 ¿Cómo se construye un AG?

Decidir cómo seleccionar los individuos para ser padres





Tema 2.2: Algoritmos Genéticos I

1.3 ¿Cómo se construye un AG?

Decidir cómo seleccionar los individuos para ser padres (II)

- Debemos garantizar que los mejores individuos tienen una mayor posibilidad de ser padres (reproducirse) frente a los individuos menos buenos.
- Debemos ser cuidadosos para dar una oportunidad de reproducirse a los individuos menos buenos. Éstos pueden incluir material genético útil en el proceso de reproducción.
- Esta idea nos define la **presión selectiva** que determina en qué grado la reproducción está dirigida por los mejores individuos.

Tema 2.2: Algoritmos Genéticos I

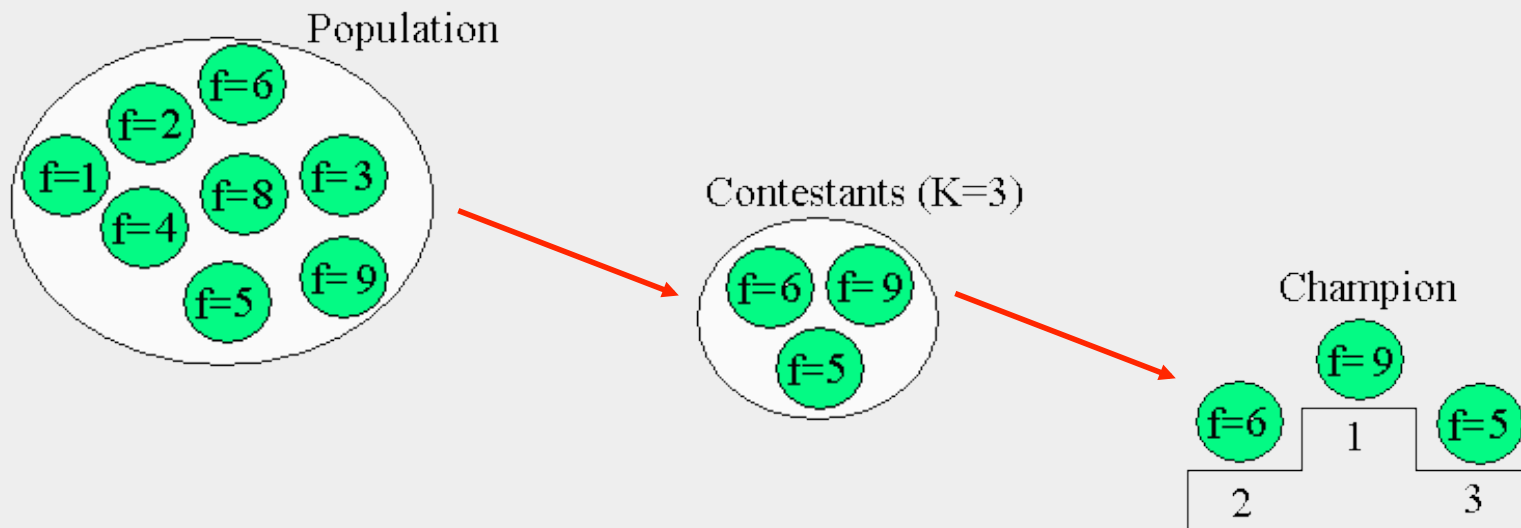
1.3 ¿Cómo se construye un AG?

Decidir cómo seleccionar los individuos para ser padres (III)

Selección por Torneo: Método para regular la presión selectiva

- Escoger **aleatoriamente** k individuos, (con reemplazo, generalmente) de una población.
- Seleccionar el mejor de ellos (o los L mejores de ellos)
- Repitiendo el proceso, se genera la población

(k se denomina **tamaño del torneo**. A mayor k , mayor **presión selectiva** y viceversa; usualmente $k=2,3,\dots$)





Tema 2.2: Algoritmos Genéticos I

1.3 ¿Cómo se construye un AG?

Decidir cómo seleccionar los individuos para ser padres (y IV)

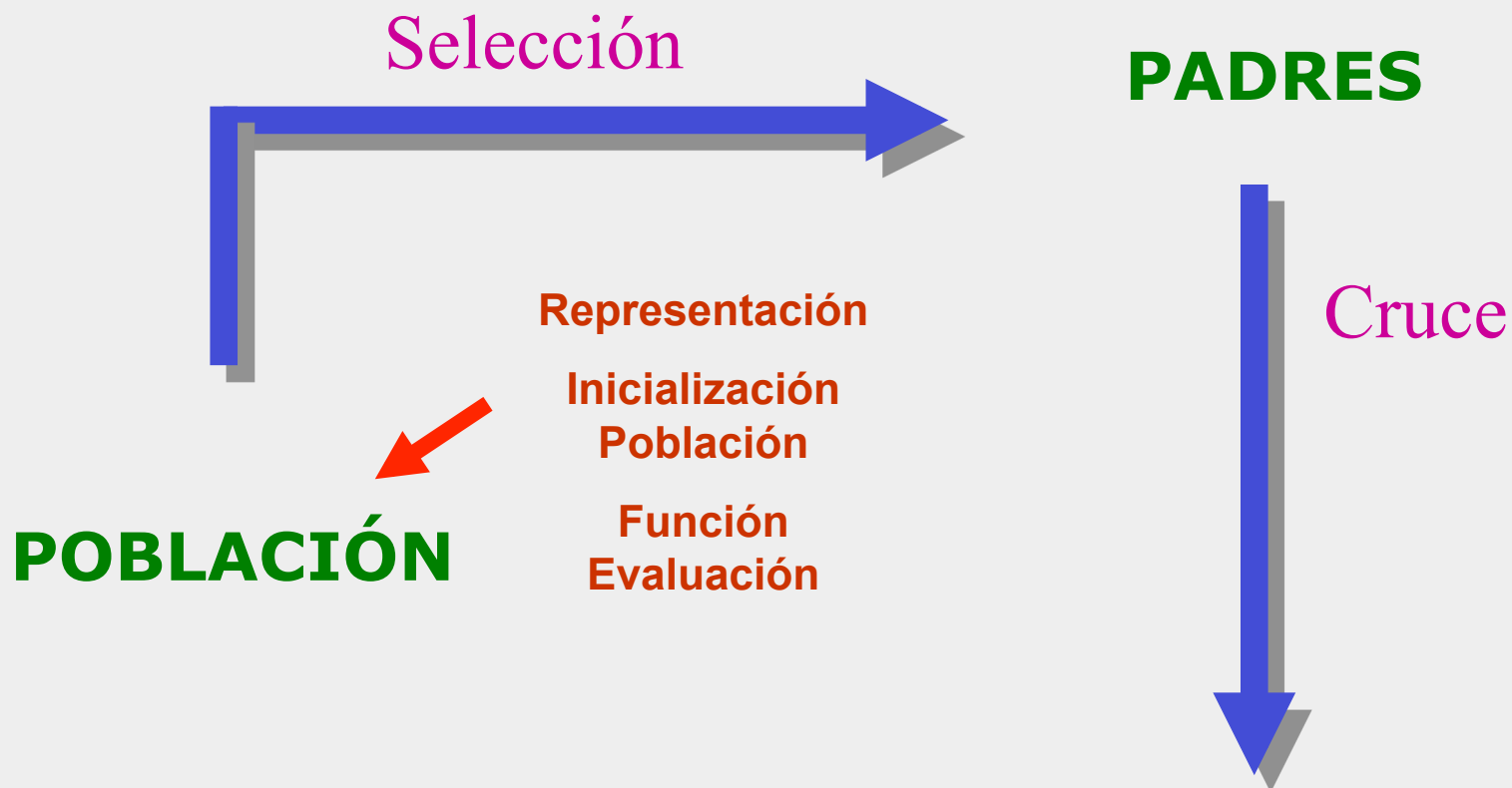
Distintos Esquemas de Selección:

- **Selección por Torneo (TS):** Escoge al mejor *fitness* de entre un grupo de tamaño 2, 3, ... elegido aleatoriamente.
- **Orden Lineal (LR):** Usa una probabilidad de selección en función de su *fitness*.
- **Selección Aleatoria (RS).**
- **Emparejamiento Variado Inverso (NAM):** Un padre se escoge aleatoriamente; para el otro selecciona N_{nam} padres y escoge el más lejano al primero, ($N_{nam}=3,5, \dots$). Está orientado a generar diversidad.

Tema 2.2: Algoritmos Genéticos I

1.3 ¿Cómo se construye un AG?

Diseñar un operador de cruce adecuado





Tema 2.2: Algoritmos Genéticos I

1.3 ¿Cómo se construye un AG?

Diseñar un operador de cruce adecuado (II)

- Podríamos tener uno o más operadores de cruce para nuestra representación.
- Algunos aspectos importantes a tener en cuenta son:
 - Los hijos deberían heredar algunas características de cada padre. Si éste no es el caso, entonces estamos ante un operador de mutación.
 - Se debe diseñar de acuerdo a la representación.
 - La recombinación debe producir cromosomas válidos.
 - Se utiliza con una probabilidad alta de actuación sobre cada pareja de padres a cruzar (P_c entre 0.6 y 0.9), si no actúa, los padres son los descendientes del proceso de recombinación de la pareja.

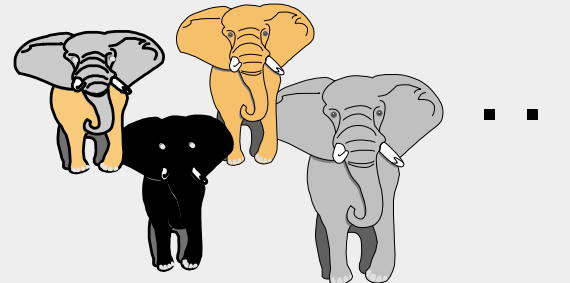
Tema 2.2: Algoritmos Genéticos I

1.3 ¿Cómo se construye un AG?

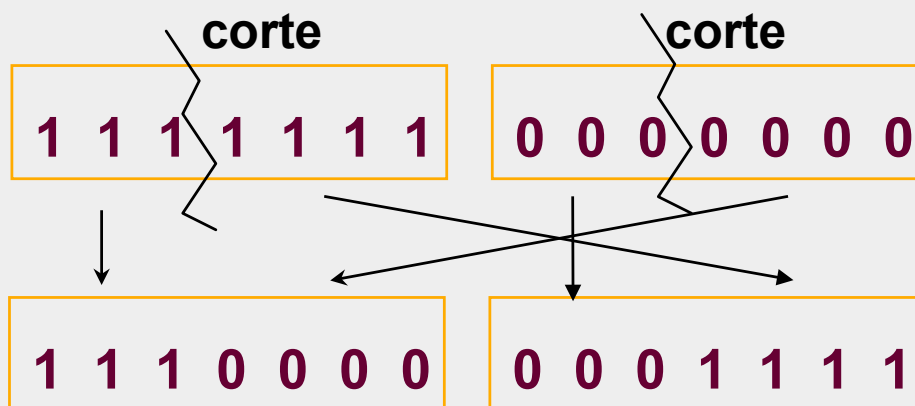
Diseñar un operador de cruce adecuado (III)

Ejemplo Operador de Cruce para Representación Binaria

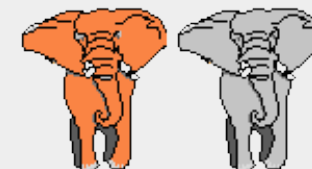
Población:



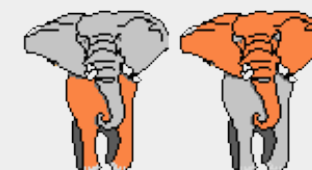
Cada cromosoma se corta en n partes que son recombinadas.
(Ejemplo para $n = 1$).



padres



descendientes



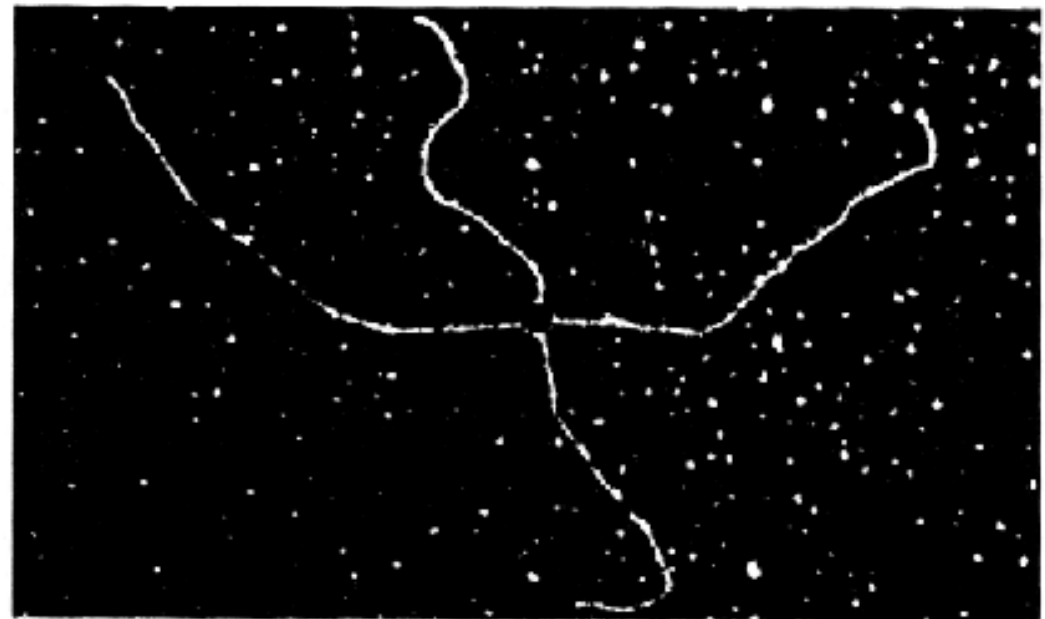
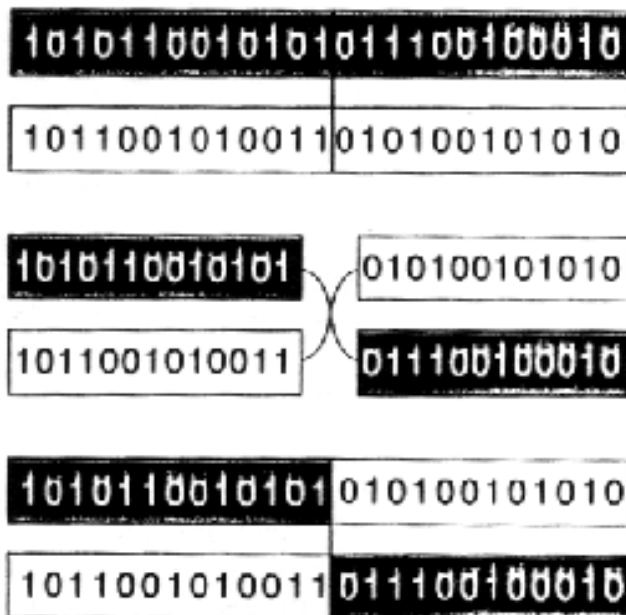


Tema 2.2: Algoritmos Genéticos I

1.3 ¿Cómo se construye un AG?

Diseñar un operador de cruce adecuado (IV)

Imagen clásica (John Holland) que introduce el operador de cruce



CROSSOVER is the fundamental mechanism of genetic rearrangement for both real organisms and genetic algorithms.

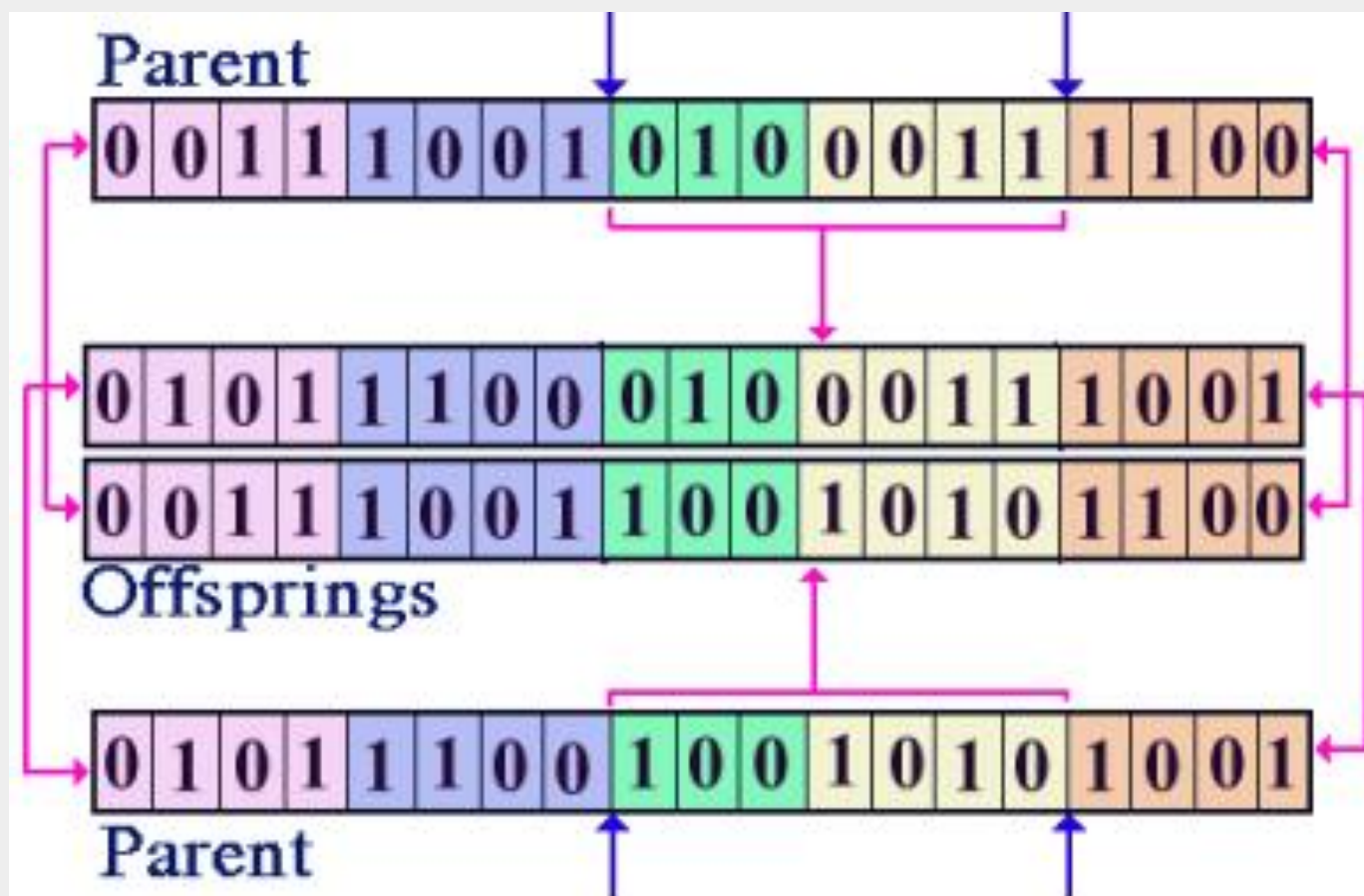
Chromosomes line up and then swap the portions of their genetic code beyond the crossover point.

Tema 2.2: Algoritmos Genéticos I

1.3 ¿Cómo se construye un AG?

Diseñar un operador de cruce adecuado (V)

Operador de Cruce en dos puntos para representación binaria



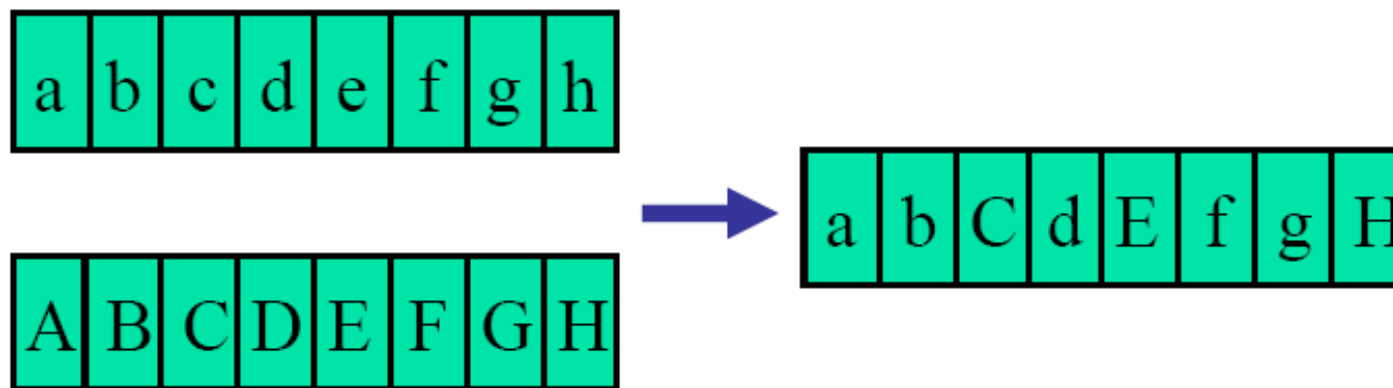
Tema 2.2: Algoritmos Genéticos I

1.3 ¿Cómo se construye un AG?

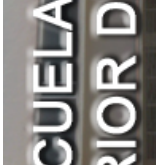
Diseñar un operador de cruce adecuado (VI)

Operador de Cruce para Representación Real

Recombinación discreta (**cruce uniforme**): dados 2 padres se crea un descendiente como sigue:



Existe muchos operadores específicos para la codificación real.



Tema 2.2: Algoritmos Genéticos I

1.3 ¿Cómo se construye un AG?

Diseñar un operador de cruce adecuado (VII)

Operador de Cruce para Representación Real (II)

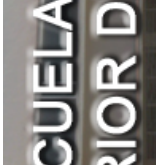
Recombinación aritmética (**cruce aritmético**):

a	b	c	d	e	f
---	---	---	---	---	---

A	B	C	D	E	F
---	---	---	---	---	---



$(a+A)/2$	$(b+B)/2$	$(c+C)/2$	$(d+D)/2$	$(e+E)/2$	$(f+F)/2$
-----------	-----------	-----------	-----------	-----------	-----------



Tema 2.2: Algoritmos Genéticos I

1.3 ¿Cómo se construye un AG?

Diseñar un operador de cruce adecuado (VIII)

Operador de Cruce para Representación Real (III)

Cruce BLX- α :

- Dados 2 cromosomas

$$C_1 = (c_{11}, \dots, c_{1n}) \text{ y } C_2 = (c_{21}, \dots, c_{2n}) ,$$

- BLX- α genera dos descendientes

$$H_k = (h_{k1}, \dots, h_{ki}, \dots, h_{kn}) , k = 1, 2$$

donde h_{ki} se genera aleatoriamente en el intervalo:

$$[C_{\min} - l \cdot \alpha, C_{\max} + l \cdot \alpha]$$

- $C_{\max} = \max \{c_{1i}, c_{2i}\}$
- $C_{\min} = \min \{c_{1i}, c_{2i}\}$
- $l = C_{\max} - C_{\min} , \alpha \in [0, 1]$



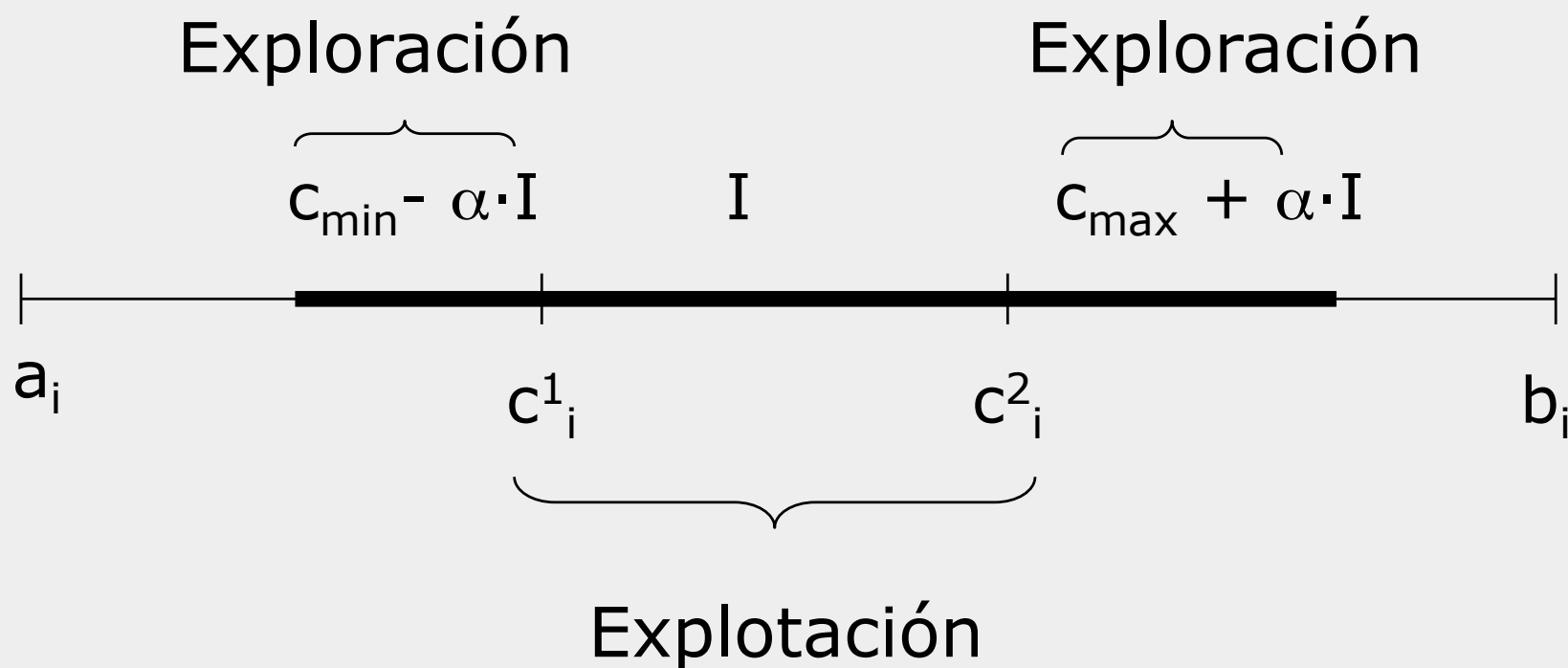
Tema 2.2: Algoritmos Genéticos I

1.3 ¿Cómo se construye un AG?

Diseñar un operador de cruce adecuado (IX)

Operador de Cruce para Representación Real (y IV)

Cruce BLX- α : (y II)

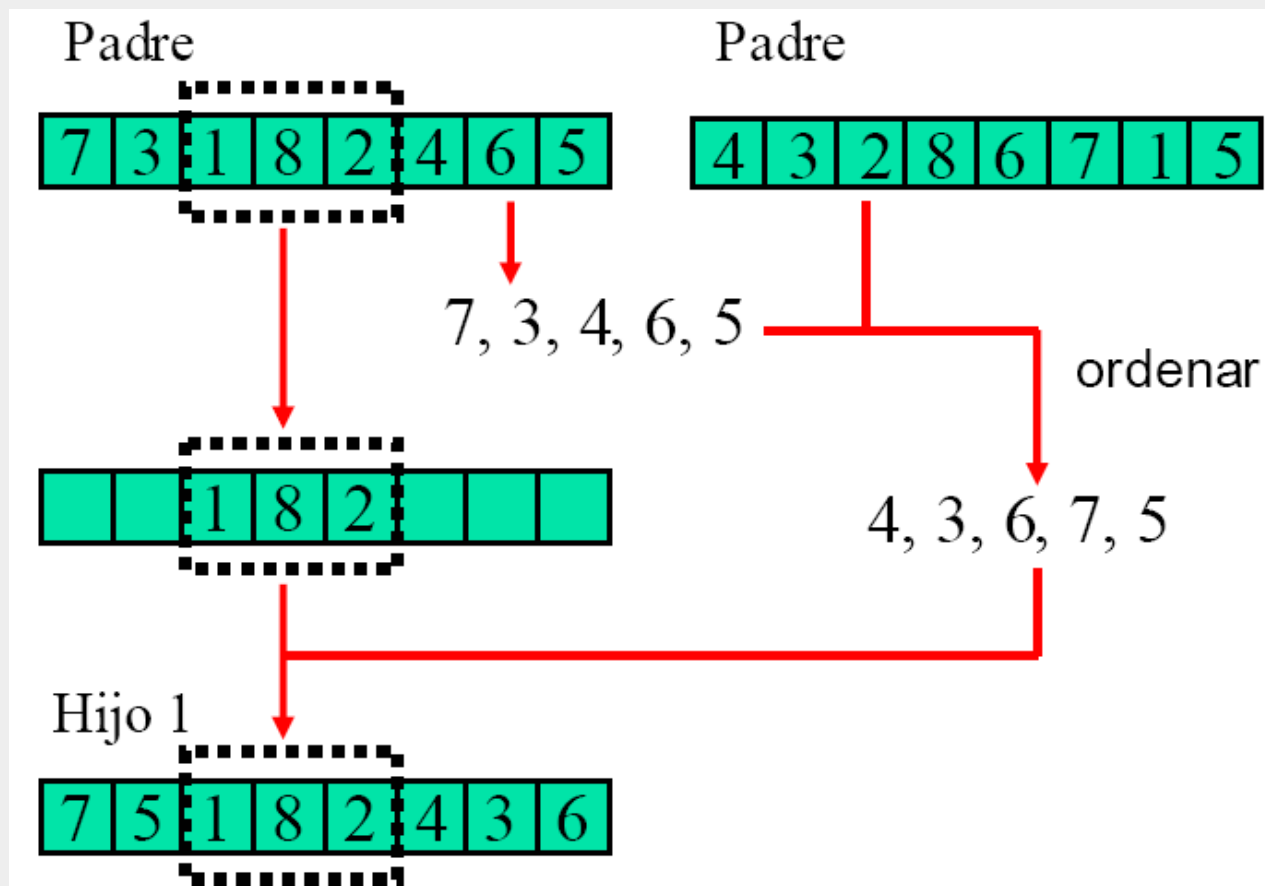


Tema 2.2: Algoritmos Genéticos I

1.3 ¿Cómo se construye un AG?

Diseñar un operador de cruce adecuado (y X)

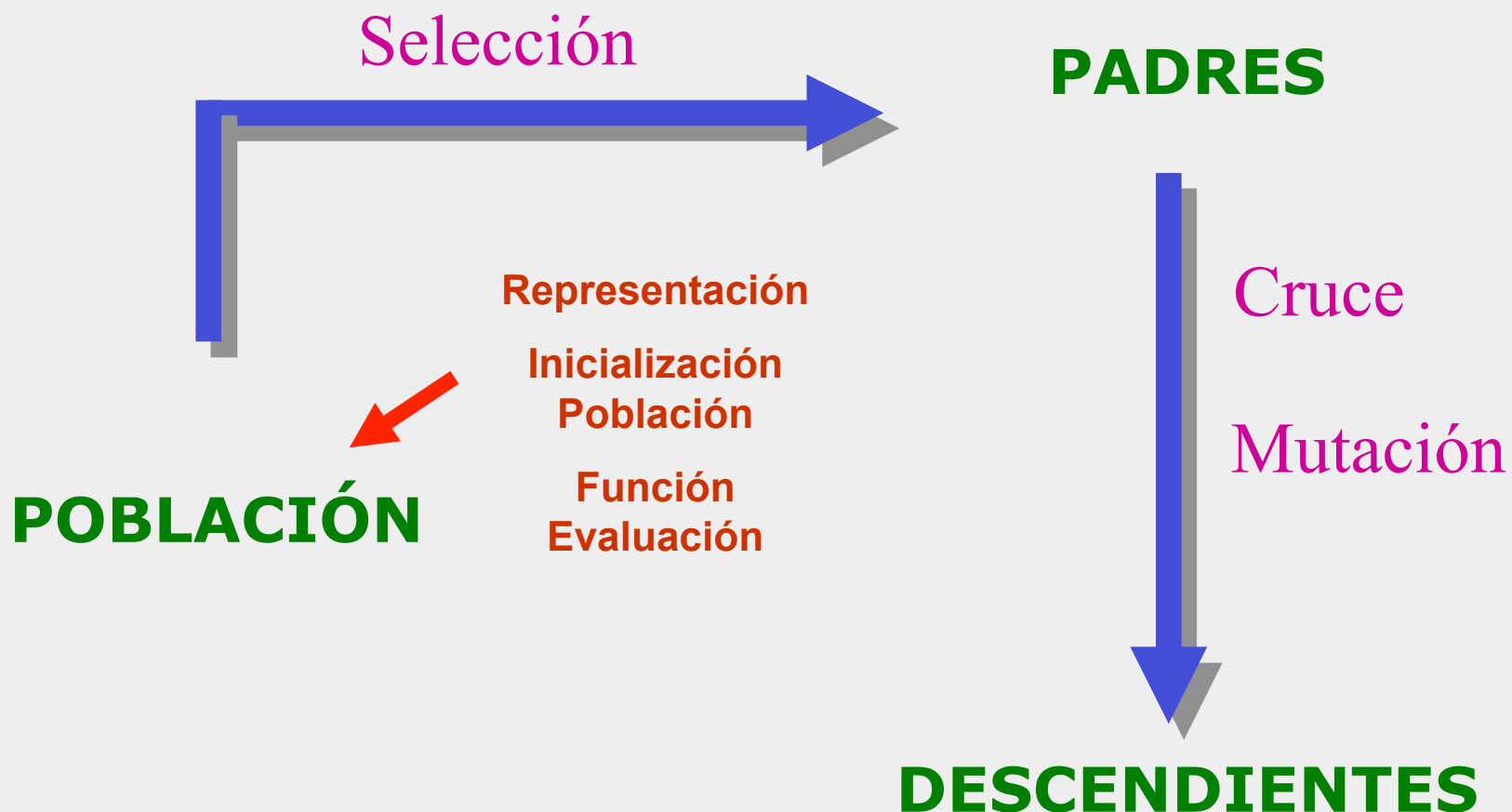
Operador de Cruce para Representación de Orden (OX)



Tema 2.2: Algoritmos Genéticos I

1.3 ¿Cómo se construye un AG?

Diseñar un operador de mutación adecuado





Tema 2.2: Algoritmos Genéticos I

1.3 ¿Cómo se construye un AG?

Diseñar un operador de mutación adecuado (II)

- Podemos tener uno o más operadores de mutación para nuestra representación.
- Algunos aspectos importantes a tener en cuenta son:
 - Debe permitir alcanzar cualquier parte del espacio de búsqueda.
 - El tamaño de la mutación debe ser controlado.
 - Debe producir cromosomas válidos.
 - Se aplica con una probabilidad muy baja de actuación sobre cada descendiente obtenido tras aplicar el operador de cruce (incluidos los descendientes que coinciden con los padres porque el operador de cruce no actúa).

Tema 2.2: Algoritmos Genéticos I

1.3 ¿Cómo se construye un AG?

Diseñar un operador de mutación adecuado (III)

Mutación para Representación Binaria

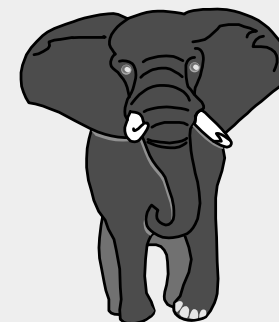
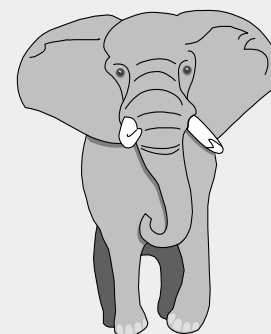
antes

1 1 1 1 1 1 1

después

1 1 1 0 1 1 1

gen mutado



La mutación ocurre con una probabilidad p_m para cada gen



Tema 2.2: Algoritmos Genéticos I

1.3 ¿Cómo se construye un AG?

Diseñar un operador de mutación adecuado (IV)

Mutación para Representación Real

- Perturbación del valor del i -ésimo mediante la adición de un valor aleatorio.
- Frecuentemente, mediante una distribución Gaussiana / Normal $N(\delta, \sigma)$, donde:
 - δ es la media
 - σ es la desviación típica

$$x'_i = x_i + N(\delta, \sigma_i)$$

para cada parámetro.



Tema 2.2: Algoritmos Genéticos I

1.3 ¿Cómo se construye un AG?

Diseñar un operador de mutación adecuado (y V)

Mutación para Representación de Orden

Selección aleatoria de dos genes e intercambio de ambos.

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



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

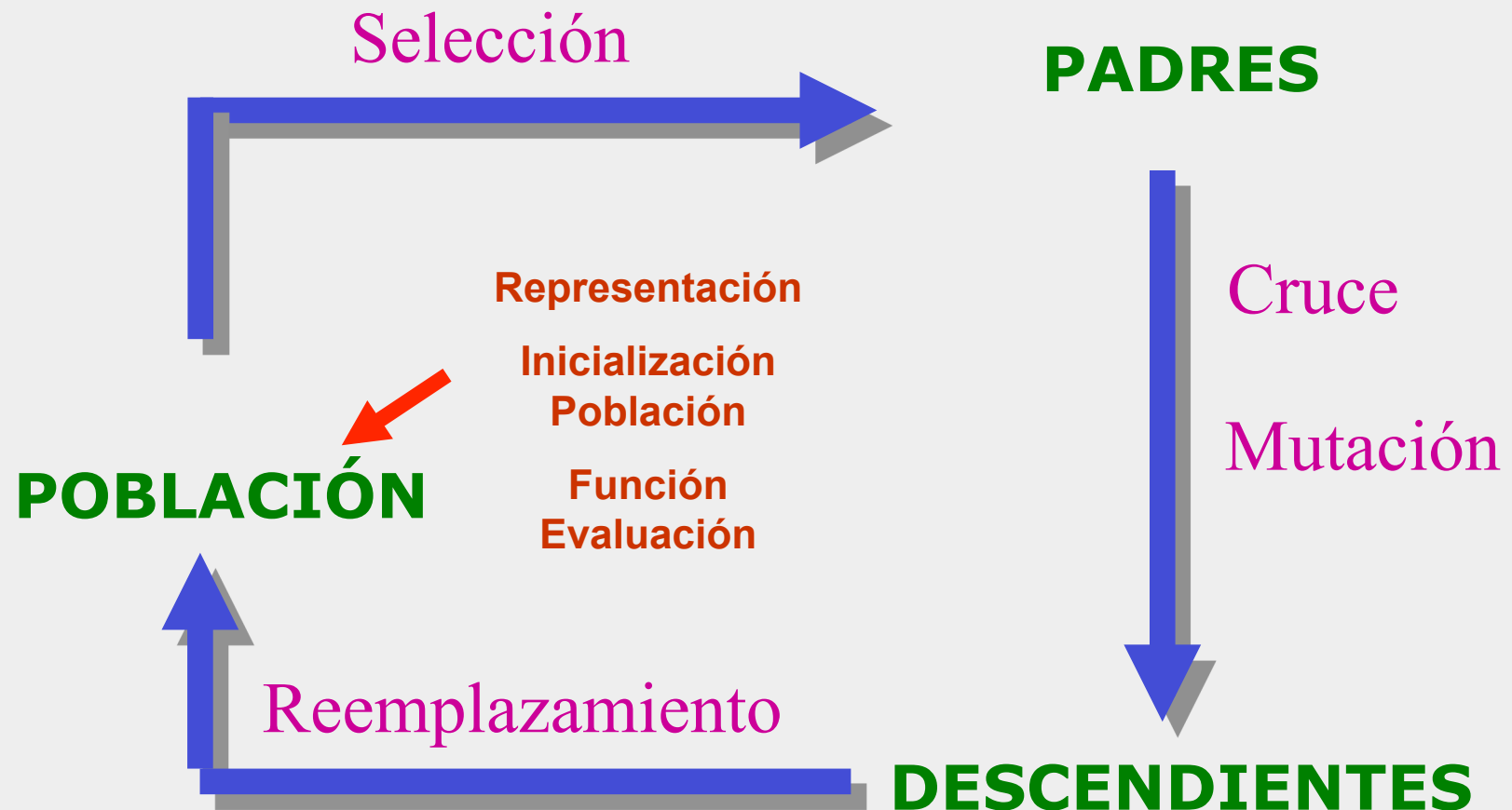




Tema 2.2: Algoritmos Genéticos I

1.3 ¿Cómo se construye un AG?

Decidir cómo reemplazar a los individuos





Tema 2.2: Algoritmos Genéticos I

1.3 ¿Cómo se construye un AG?

Decidir cómo reemplazar a los individuos

Estrategia de Reemplazamiento

- La **presión selectiva** se ve también afectada por la forma en que los cromosomas de la población son reemplazados por los nuevos descendientes.
- Podemos utilizar métodos de **reemplazamiento aleatorios**, o **determinísticos**.
- Podemos decidir **no reemplazar al mejor cromosoma** de la población: **Elitismo** (que consiste en mantener a los mejores).



Tema 2.2: Algoritmos Genéticos I

1.3 ¿Cómo se construye un AG?

Decidir cómo reemplazar a los individuos

Algunas Estrategias de Reemplazo para AGs Estacionarios:

- **Reemplazar el peor de la población (RW).** Genera Alta presión selectiva.
- **Torneo Restringido (RTS):** Reemplaza el más parecido de entre w ($w=3$).
Objetivo: Mantener una cierta diversidad.
- **Peor entre semejantes (WAMS):** Busca equilibrio entre diversidad y presión selectiva.
- **Algoritmo de Crowding Determinístico (DC):** Reemplaza a su padre más parecido. Mantiene diversidad.



Tema 2.2: Algoritmos Genéticos I

1.3 ¿Cómo se construye un AG?

Estudio Comparativo de Algunos Modelos Estacionarios

- Algoritmo Genético Estacionario
- Para cada combinación selección-reemplazo se ha definido un AG.
- Las características comunes de todos ellos son:
 - Cruce aplicado es BLX-0.5.
 - Mutación no uniforme aplicado con $p_{\text{Mut}} = 1/8$ (ind.)
 - Tamaño de la población de 60 individuos.
 - 100.000 evaluaciones por ejecución



Tema 2.2: Algoritmos Genéticos I

1.3 ¿Cómo se construye un AG?

Estudio Comparativo de Algunos Modelos Estacionarios

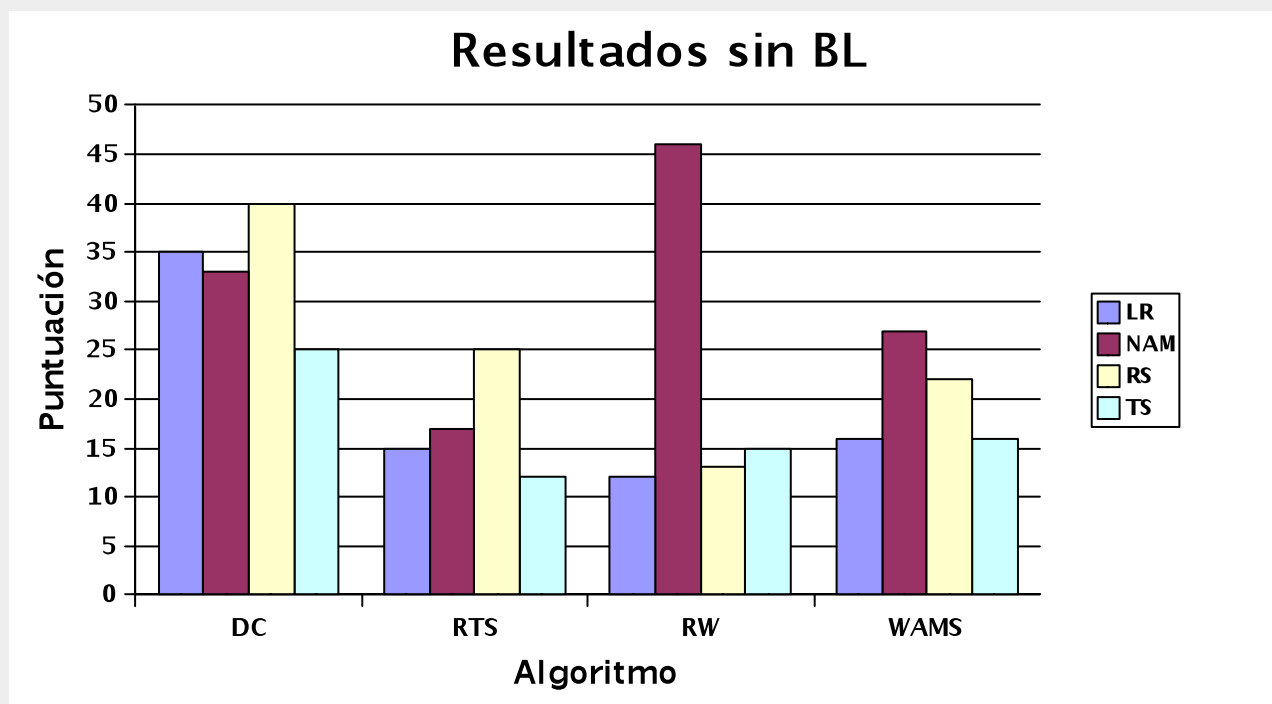
- Se evalúan sobre un conjunto de 13 problemas (2 reales y 11 funciones clásicas de distinta dificultad).
- Para cada combinación se muestra una puntuación obtenida así:
 - Media para 50 ejecuciones.
 - Se aplica el *t-test de Student* con $p=0.05$.
 - *Para cada función se asigna a cada algoritmo una función. 5 al mejor, 4 al siguiente, Los equivalentes entre sí (mediante t-student) reciben igual puntuación.*
 - *Se suma los resultados para las 13 funciones.*



Tema 2.2: Algoritmos Genéticos I

1.3 ¿Cómo se construye un AG?

Estudio Comparativo de Algunos Modelos Estacionarios



Reemplazar al peor de la población (RW)
Torneo Restringido (RTS)
Peor entre semejantes (WAMS)
Algoritmo de Crowding Determinístico (DC)

Selección por Torneo (TS)
Orden Lineal (LR)
Selección Aleatoria (RS).
Emparejamiento Variado Inverso (NAM)



Tema 2.2: Algoritmos Genéticos I

1.3 ¿Cómo se construye un AG?

Decidir la condición de parada

- Ideal: Cuando se alcanza el óptimo!
- Recursos limitados de CPU:
 - Fijar el máximo número de evaluaciones
- Límite sobre la paciencia del usuario: Después de algunas iteraciones sin mejora.



Tema 2.2: Algoritmos Genéticos I

Tema 2.2: Algoritmos Genéticos I. Conceptos Básicos

Índice:

1.1 Introducción

1.2 Modelos Generacional vs. Estacionario

1.3 ¿Cómo se construye un AG?

1.4 Utilización de AGs

1.5 Ejemplo

1.6 Conclusiones

1.7 Bibliografía



Tema 2.2: Algoritmos Genéticos I

1.4 Utilización de AGs

- **Nunca** se deben sacar conclusiones de una única ejecución
 - utilizar medidas estadísticas (medias, medianas, ...)
 - con un número suficiente de ejecuciones independientes
- No se debe ajustar/comprobar la actuación de un algoritmo sobre ejemplos simples si se desea trabajar con casos reales.
- Existe un comentario genérico en el uso de los Algoritmos no determinísticos:

“Se puede obtener lo que se desea en una experimentación de acuerdo a la dificultad de los casos utilizados”

(Se encuentran propuestas en las que basta encontrar un caso adecuado para un algoritmo para afirmar que es muy bueno, pero esta afirmación no significa que sea extensible a otros casos: es el error en el que incurren algunos autores).



Tema 2.2: Algoritmos Genéticos I

Tema 2.2: Algoritmos Genéticos I. Conceptos Básicos

Índice:

1.1 Introducción

1.2 Modelos Generacional vs. Estacionario

1.3 ¿Cómo se construye un AG?

1.4 Utilización de AGs

1.5 Ejemplo

1.6 Conclusiones

1.7 Bibliografía



Tema 2.2: Algoritmos Genéticos I

1.5 Ejemplo: Viajante de Comercio

Representación de orden

(3 5 1 13 6 15 8 2 17 11 14 4 7 9 10 12 16)

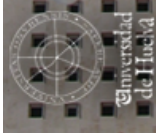
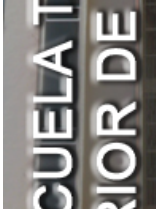
17 ciudades

Objetivo: Suma de la distancia entre las ciudades.

Población: 61 cromosomas - Elitismo

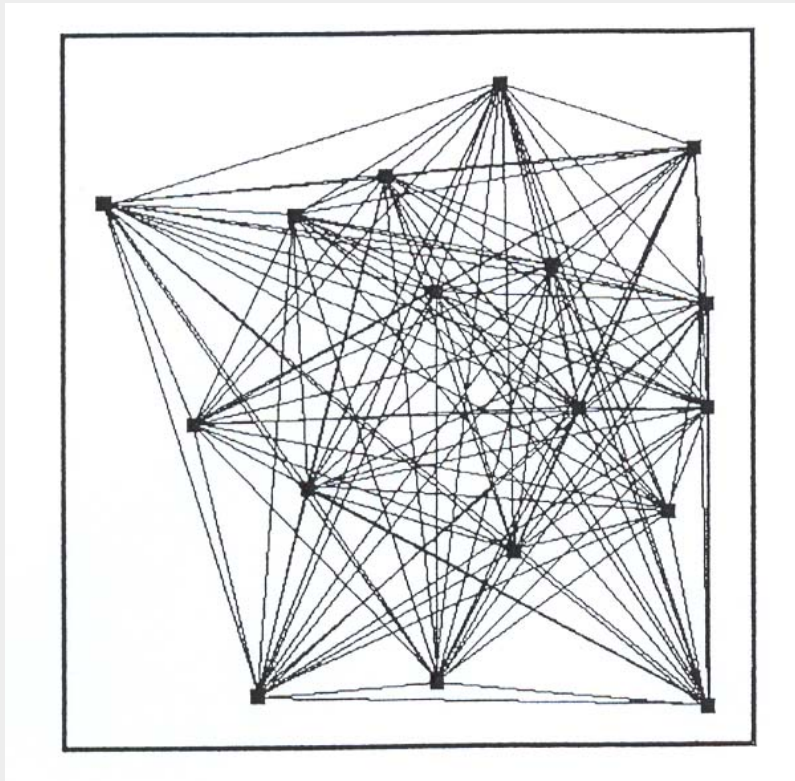
Cruce: OX ($P_c = 0.6$)

Mutación: Inversión de una lista ($P_m = 0.01$ – cromosoma)

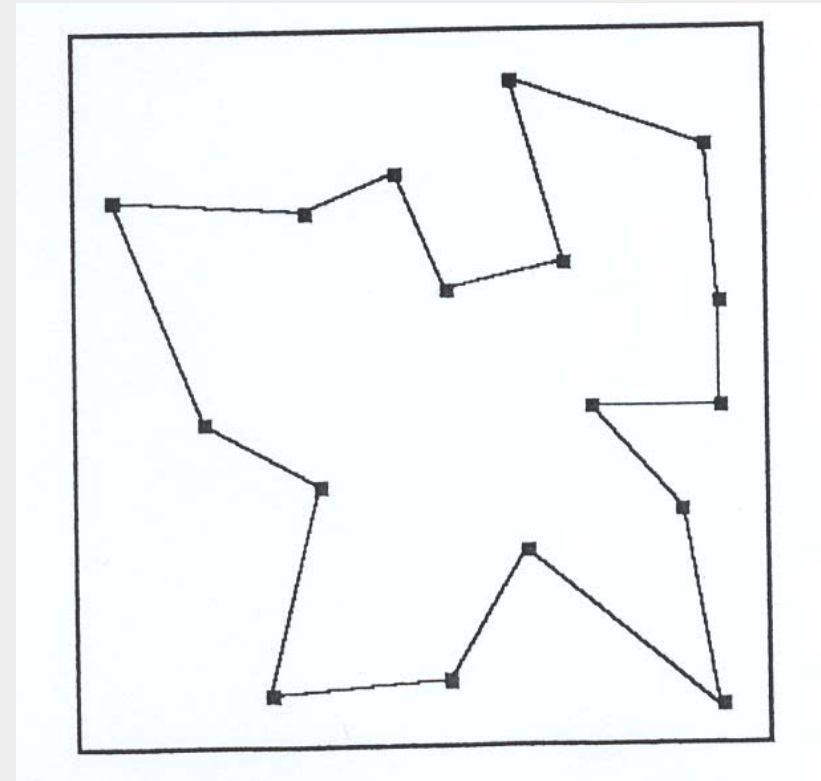


Tema 2.2: Algoritmos Genéticos I

1.5 Ejemplo: Viajante de Comercio



$17! = 3.5568743 \times 10^{14}$ recorridos posibles

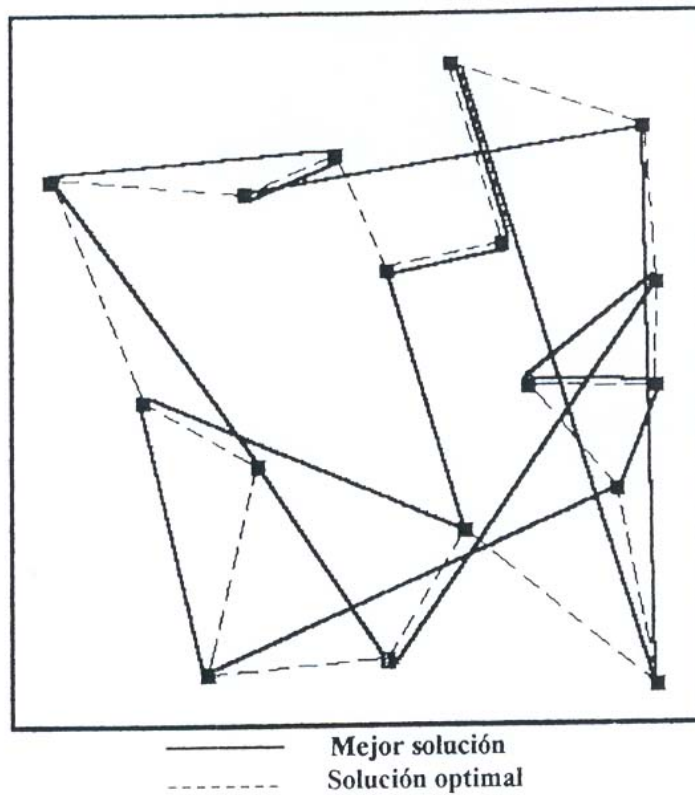


Solución óptima: 221.64

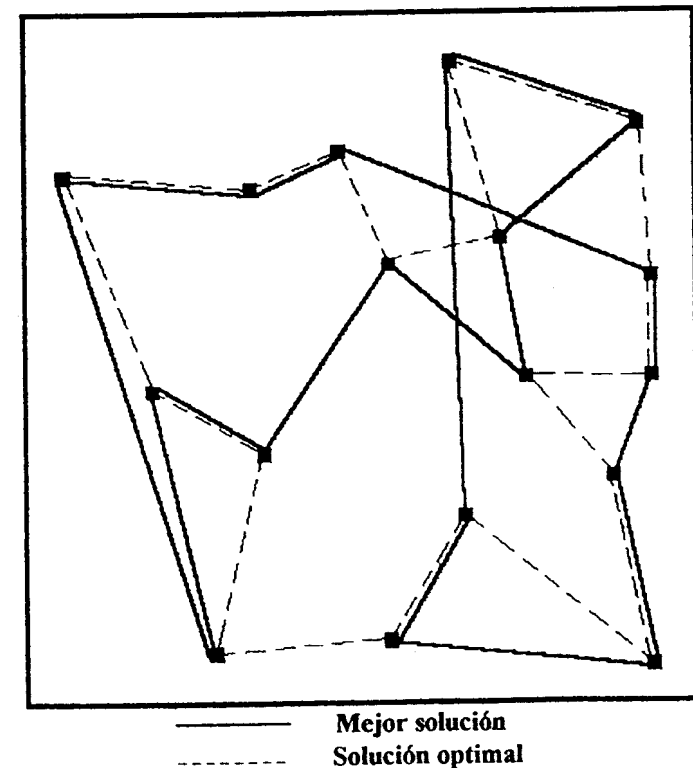


Tema 2.2: Algoritmos Genéticos I

1.5 Ejemplo: Viajante de Comercio



Iteración: 0 Costo: 403.7



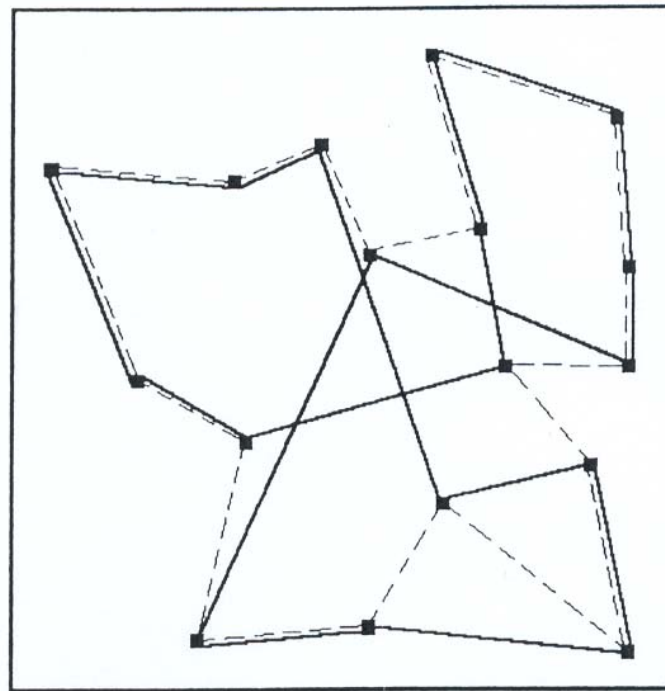
Iteración: 25 Costo: 303.86

Solución óptima: 221.64



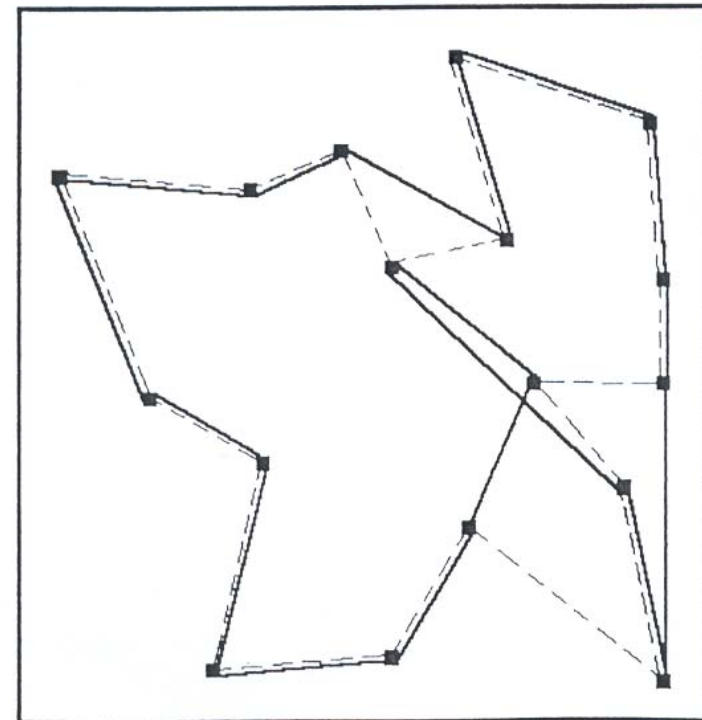
Tema 2.2: Algoritmos Genéticos I

1.5 Ejemplo: Viajante de Comercio



— Mejor solución
- - - Solución óptima

Iteración: 50 Costo: 293.6



— Mejor solución
- - - Solución óptima

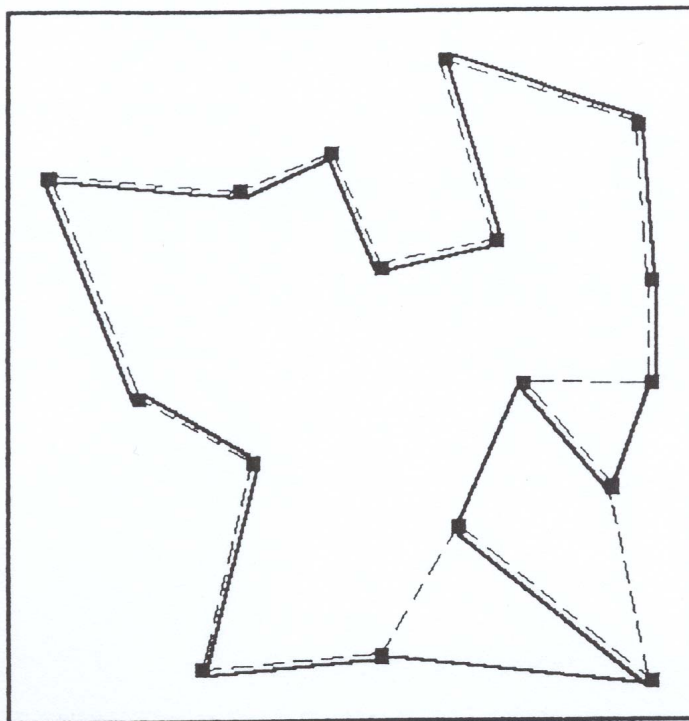
Iteración: 100 Costo: 251.55

Solución óptima: 221.64



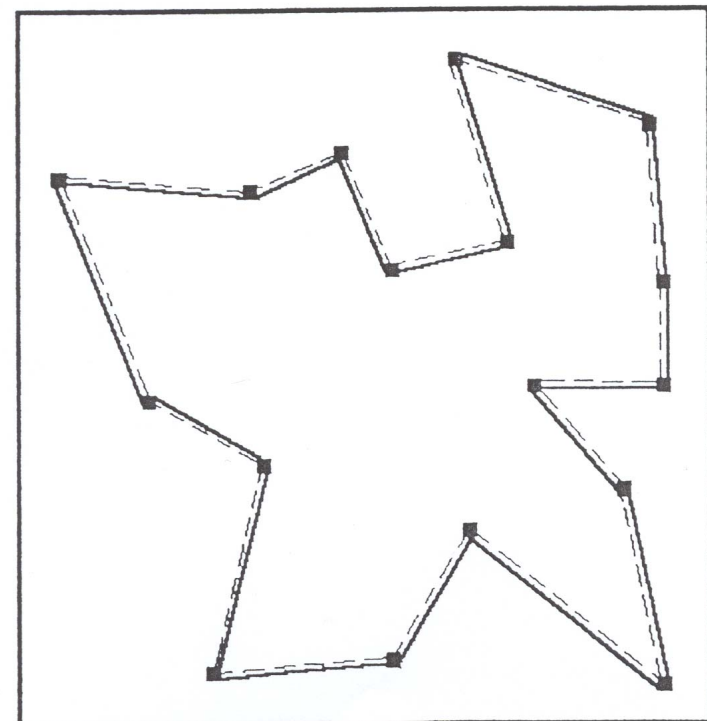
Tema 2.2: Algoritmos Genéticos I

1.5 Ejemplo: Viajante de Comercio



— Mejor solución
- - - Solución optimal

Iteración: 200 Costo: 231.4

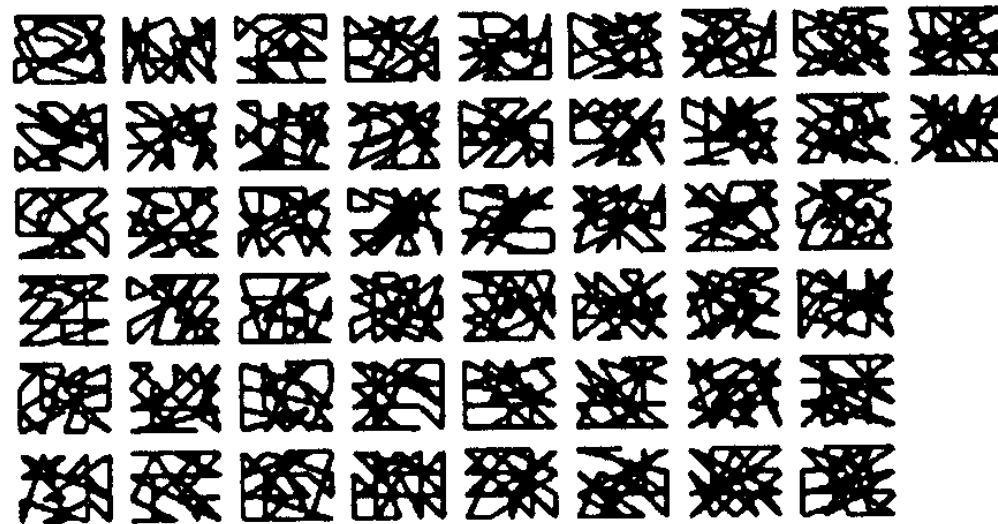


— Mejor solución
- - - Solución optimal

Iteración: 250 Solución
óptima: 221.64

Tema 2.2: Algoritmos Genéticos I

1.5 Ejemplo: Viajante de Comercio



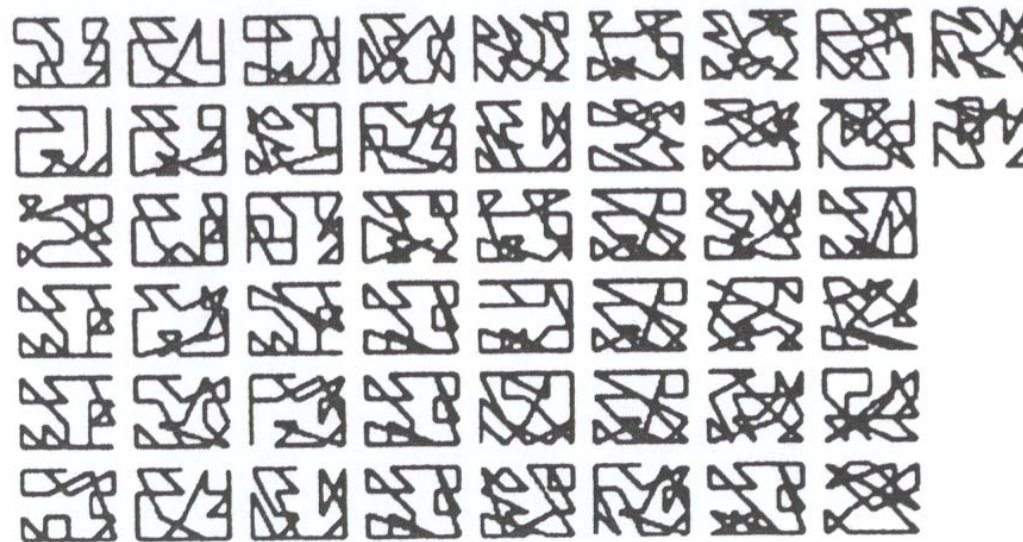
(0)

Visualización de la evolución de una población de 50 cromosomas
y 70 iteraciones



Tema 2.2: Algoritmos Genéticos I

1.5 Ejemplo: Viajante de Comercio



(10)

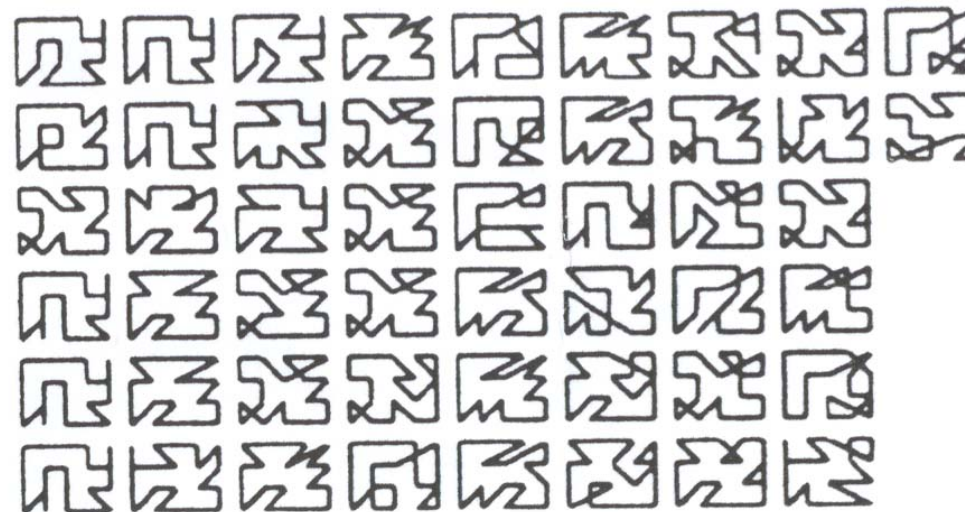
Visualización de la evolución de una población de 50 cromosomas y 70 iteraciones





Tema 2.2: Algoritmos Genéticos I

1.5 Ejemplo: Viajante de Comercio



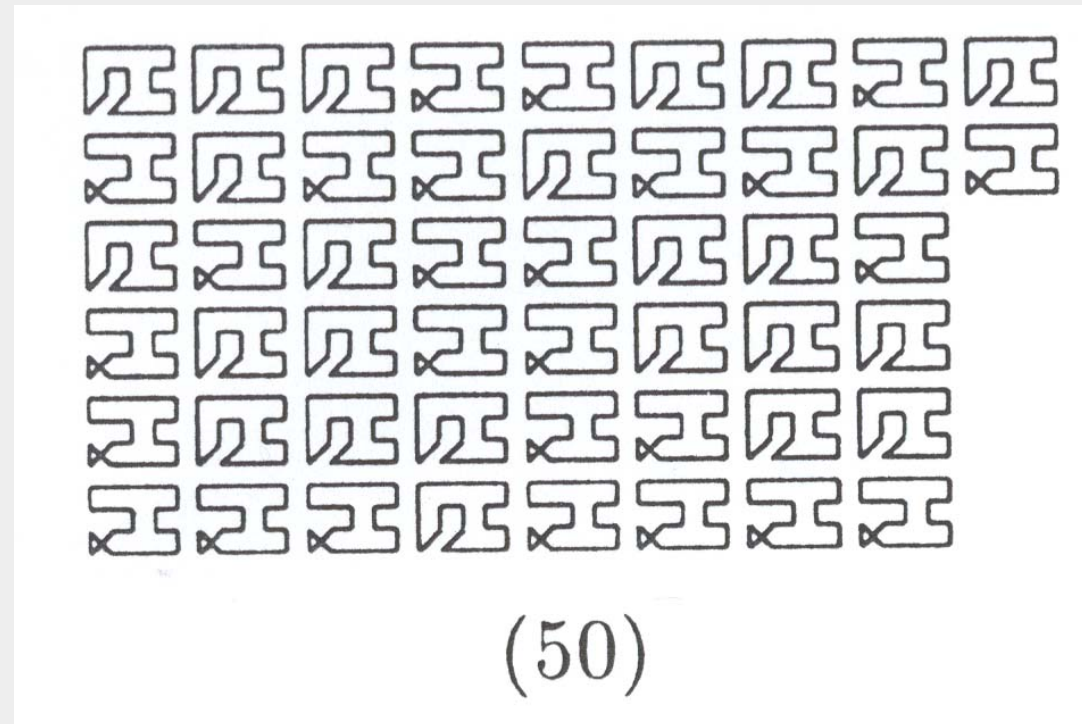
(30)

Visualización de la evolución de una población de 50 cromosomas y 70 iteraciones

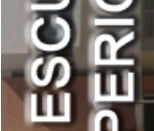
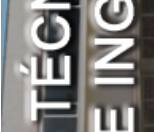


Tema 2.2: Algoritmos Genéticos I

1.5 Ejemplo: Viajante de Comercio

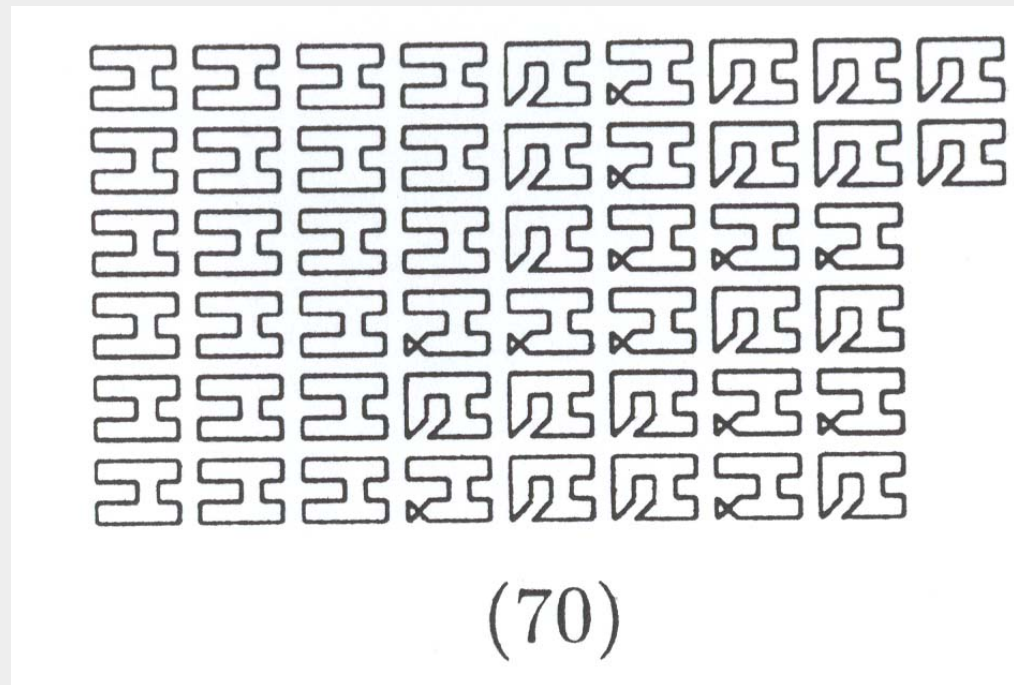


Visualización de la evolución de una población de 50 cromosomas y 70 iteraciones



Tema 2.2: Algoritmos Genéticos I

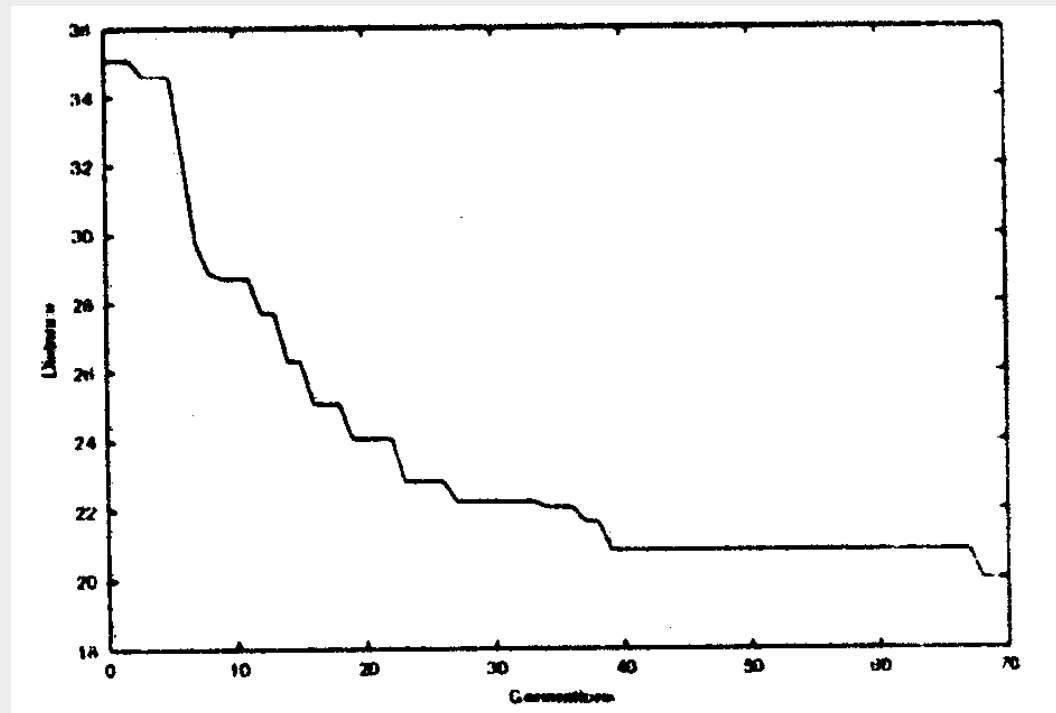
1.5 Ejemplo: Viajante de Comercio



Visualización de la evolución de una población de 50 cromosomas y 70 iteraciones

Tema 2.2: Algoritmos Genéticos I

1.5 Ejemplo: Viajante de Comercio



Visualización de la evolución de una población de 50 cromosomas y 70 iteraciones



Tema 2.2: Algoritmos Genéticos I

Tema 2.2: Algoritmos Genéticos I. Conceptos Básicos

Índice:

1.1 Introducción

1.2 Modelos Generacional vs. Estacionario

1.3 ¿Cómo se construye un AG?

1.4 Utilización de AGs

1.5 Ejemplo

1.6 Conclusiones

1.7 Bibliografía



Tema 2.2: Algoritmos Genéticos I

1.6 Conclusiones

Los Algoritmos Genéticos:

- *Están basados en una metáfora biológica: evolución .*
- *Tienen gran potencialidad de aplicación.*
- *Son muy populares en muchos campos.*
- *Son muy potentes en diversas aplicaciones.*
- *Ofrecen altas prestaciones a bajo costo.*

***SON ATRACTIVOS DESDE UN
PUNTO DE VISTA COMPUTACIONAL***



Tema 2.2: Algoritmos Genéticos I

Tema 2.2: Algoritmos Genéticos I. Conceptos Básicos

Índice:

1.1 Introducción

1.2 Modelos Generacional vs. Estacionario

1.3 ¿Cómo se construye un AG?

1.4 Utilización de AGs

1.5 Ejemplo

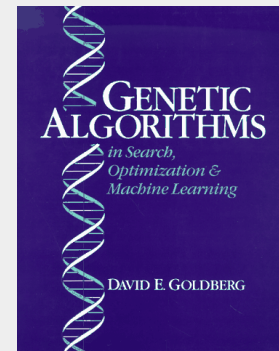
1.6 Conclusiones

1.7 Bibliografía

Tema 2.2: Algoritmos Genéticos I

1.7 Bibliografía

- **D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning.** Addison Wesley, 1989.
- **Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs.** Springer Verlag, 1991.
- **T. Bäck, D.B. Fogel, Z. Michalewicz, Handbook of Evolutionary Computation,** Institute of Physics Publishers, 1997.
- **A.E. Eiben, J.E. Smith. Introduction to Evolutionary Computing.** Springer, 2003.



Clásico en AG