Algoritmos Genéticos

Francisco Herrera

Grupo de Investigación
"Soft Computing y Sistemas de Información Inteligentes"

Dpto. Ciencias de la Computación e I.A.
Universidad de Granada
18071 – ESPAÑA
herrera@decsai.ugr.es
http://sci2s.ugr.es

DECSAI
Universidad de Granada

ALGORITMOS GENÉTICOS

1. ALGORITMOS GENÉTICOS. INTRODUCCIÓN
2. ¿CÓMO SE CONSTRUYE?
3. SOBRE SU UTILIZACIÓN
4. DOMINIOS DE APLICACIÓN
5. EJEMPLO: VIAJANTE DE COMERCIO
6. APLICACIONES
7. ALGUNAS CUESTIONES: DIVERSIDAD, EXPLORACIÓN vs EXPLOTACIÓN
8. MODELOS: GENERACIONAL vs ESTACIONARIO
9. ALGUNAS EXTENSIONES

1. INTRODUCCIÓN A LOS ALGORITMOS GENÉTICOS

- ¿QUÉ ES UN ALGORITMO GENÉTICO?
- EVOLUCIÓN NATURAL. EVOLUCIÓN ARTIFICIAL
- LOS INGREDIENTES
- El CICLO DE LA EVOLUCIÓN
- ESTRUCTURA DE UN ALGORITMO GENÉTICO

¿Qué es un Algoritmo Genético?

Los Algoritmos Genéticos

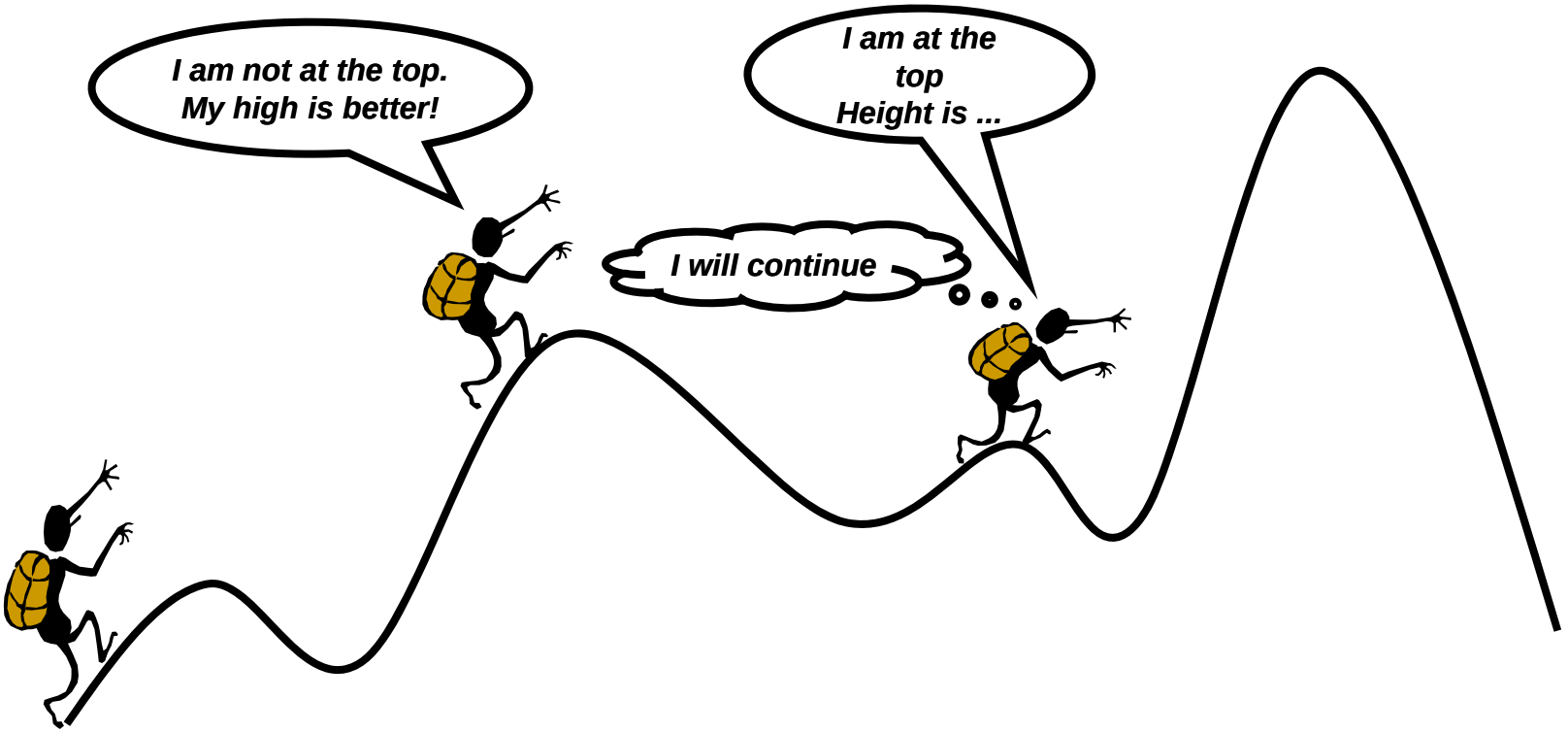
son algoritmos de optimización,
búsqueda
y aprendizaje
inspirados en los procesos de

Evolución Natural

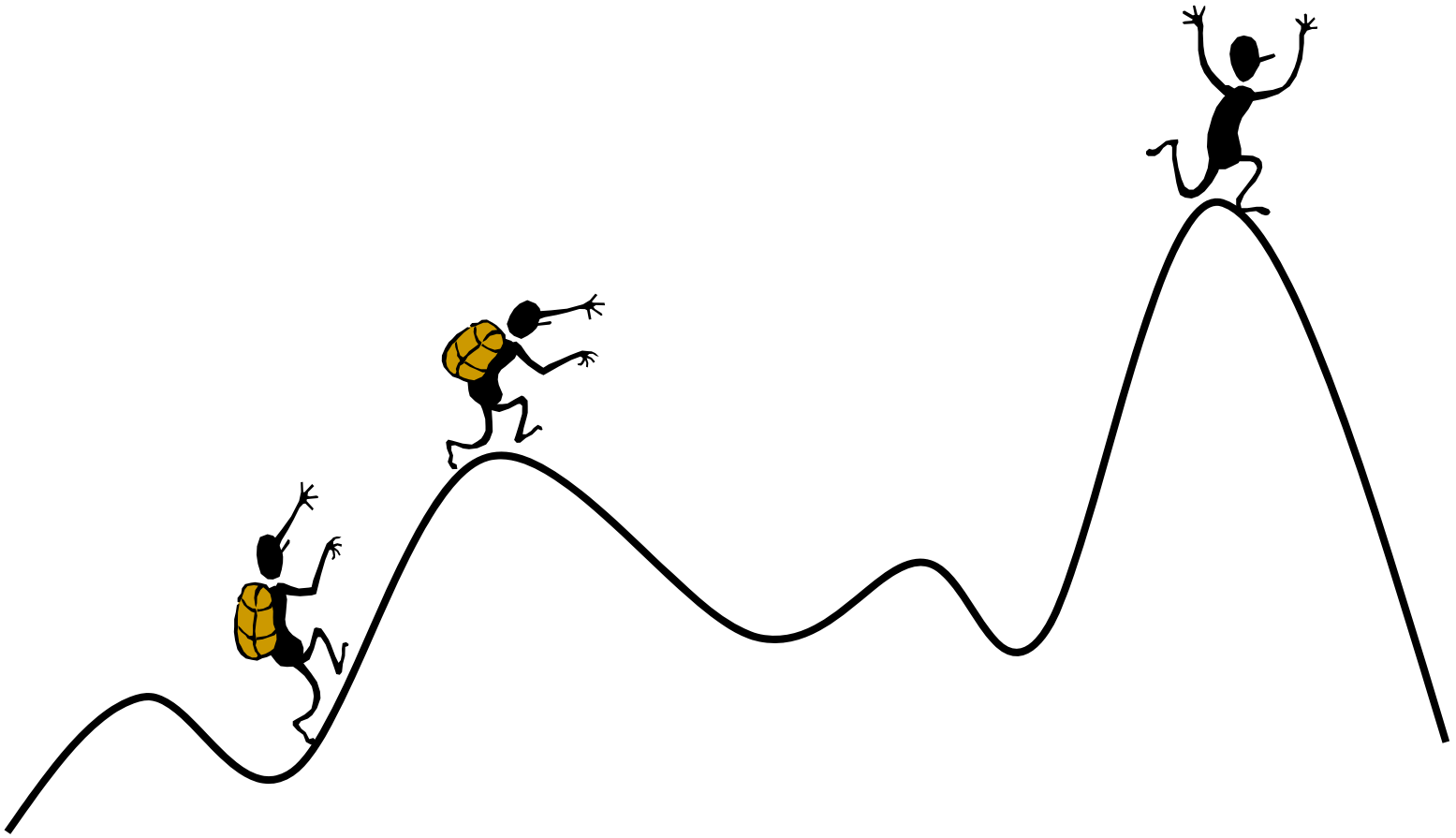
y

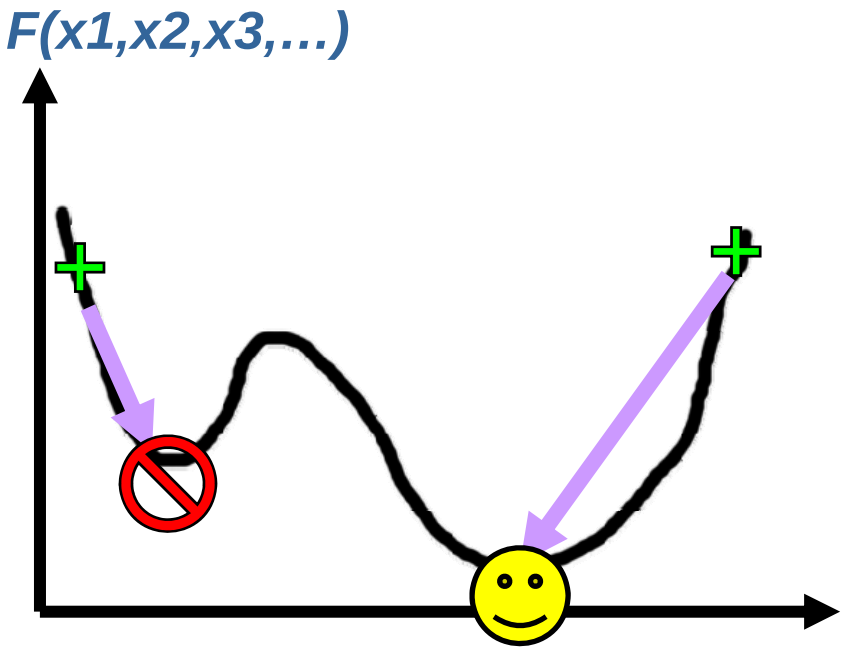
Evolución Genética

¿Qué es un Algoritmo Genético?



¿Qué es un Algoritmo Genético?





Los algoritmos evolutivos no tiene este problema!!!

Evolución Natural

En la naturaleza, los procesos evolutivos ocurren cuando se satisfacen las siguientes condiciones:

Una entidad o individuo tiene la habilidad de reproducirse.

Hay una población de tales individuos que son capaces de reproducirse.

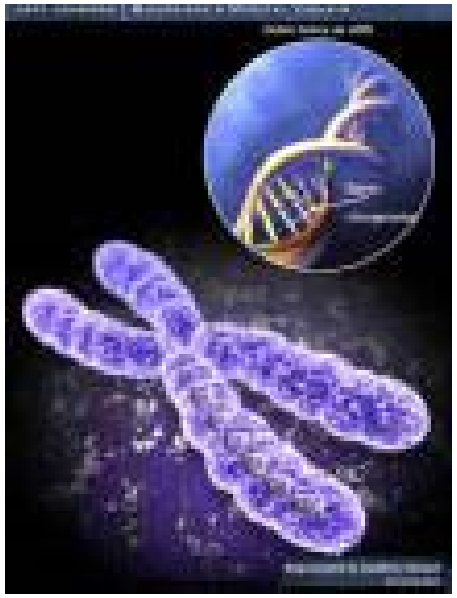
Existe alguna variedad, diferencia, entre los individuos que se reproducen.

Algunas diferencias en la habilidad para sobrevivir en el entorno están asociadas con esa variedad.

Evolución Natural

Los mecanismos que conducen esta evolución no son totalmente conocidos, pero sí algunas de sus características, que son ampliamente aceptadas:

La evolución es un proceso que opera sobre los cromosomas más que sobre las estructuras de la vida que están codificadas en ellos.



Evolución Natural

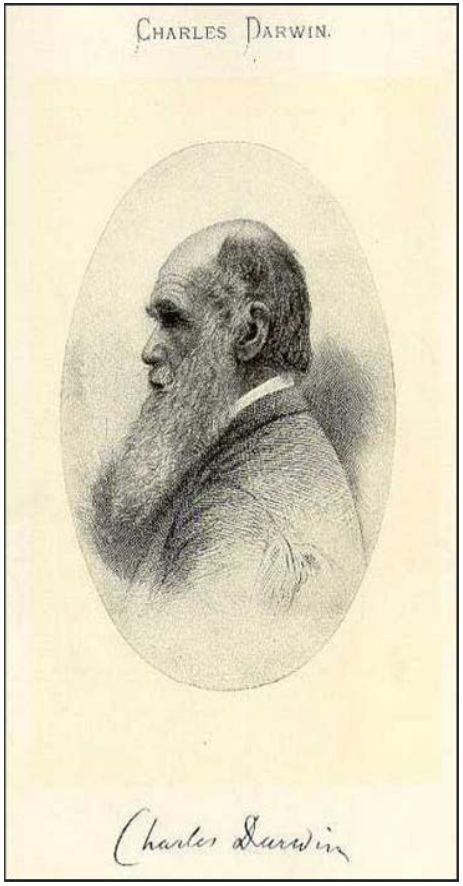
La selección natural es el enlace entre los cromosomas y la actuación de sus estructuras decodificadas.

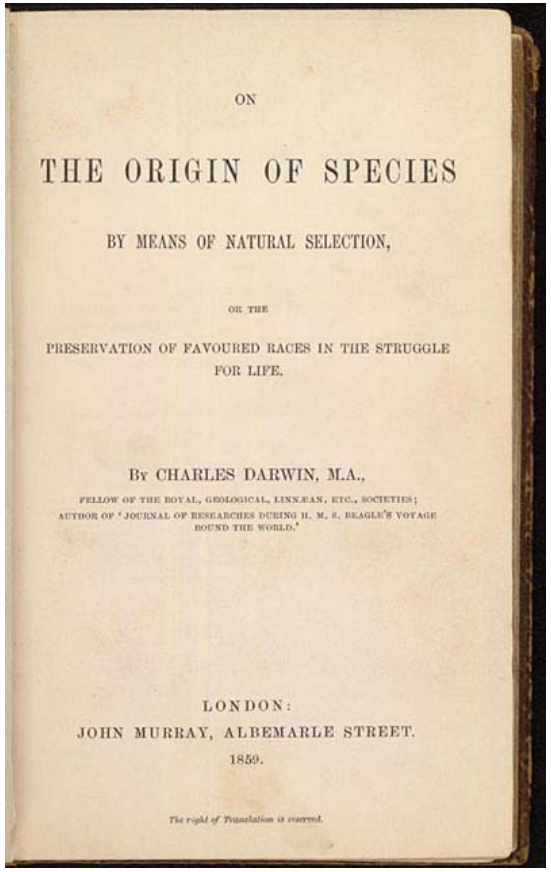
El proceso de reproducción es el punto en el cual la evolución toma parte, actúa.

La evolución biológica no tiene memoria.

Darwin, C. (1859). *On the Origin of Species by Means of Natural Selection or the Preservations of Favored Races in the Struggle for Life.* London: John Murray.

Evolución Natural




ON

THE ORIGIN OF SPECIES

BY MEANS OF NATURAL SELECTION,

OR THE

PRESERVATION OF FAVOURED RACES IN THE STRUGGLE FOR LIFE.

By CHARLES DARWIN, M.A.,

FELLOW OF THE ROYAL, GEOLOGICAL, LINNÆAN, ETC., SOCIETIES;
AUTHOR OF 'JOURNAL OF RESEARCHES DURING H. M. S. BEAGLE'S VOYAGE ROUND THE WORLD.'

LONDON:
JOHN MURRAY, ALBEMARLE STREET.
1859.

Evolución Artificial

COMPUTACIÓN EVOLUTIVA

Está compuesta por modelos de evolución basados en poblaciones cuyos elementos representan soluciones a problemas

La simulación de este proceso en un ordenador resulta ser una técnica de optimización probabilística, que con frecuencia mejora a otros métodos clásicos en problemas difíciles

Evolución Artificial

Existen cuatro paradigmas básicos:

John Holland
Inventor of genetic algorithms
Professor of CS and Psychology at the U. of Michigan.

Algoritmos Genéticos que utilizan operadores genéticos sobre cromosomas. 1975, Michigan University

Inventors of Evolution Strategies

Estrategias de Evolución que enfatizan los cambios de comportamiento al nivel de los individuos. 1964, Technische Universität Berlin

Hans-Paul Schwefel
Universität Dortmund

Ing. Ingo Rechenberg
Bionics & Evolutiontechnique
Technical University Berlin
http://www.bionik.tu-berlin.de/

Programación Evolutiva que enfatizan los cambios de comportamiento al nivel de las especies. 1960-1966, Florida

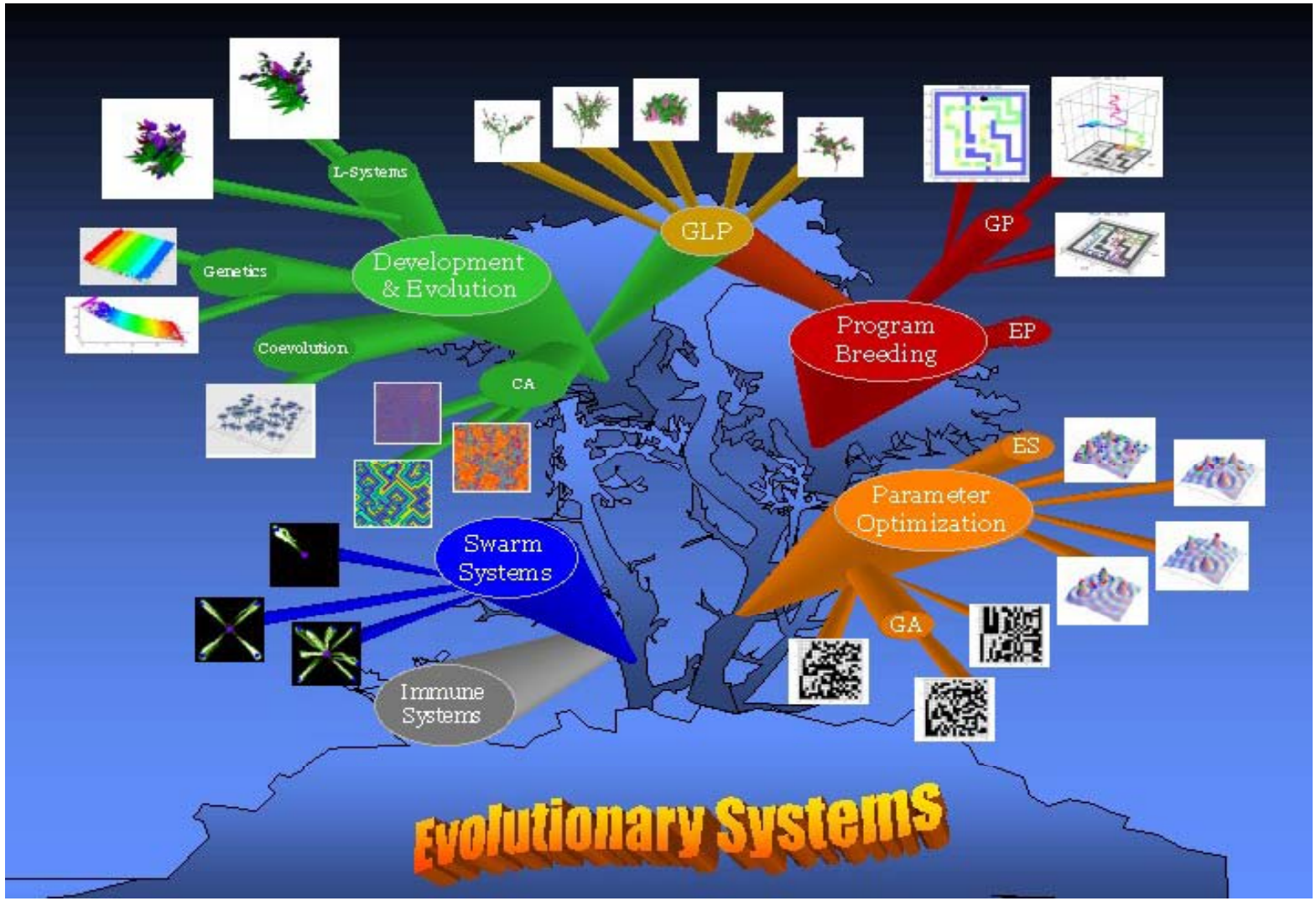
Lawrence J. Fogel,
Natural Selection, Inc.
Inventor of Evolutionary Programming

Programación Genética que evoluciona expresiones representadas como árboles.
1989, Stanford University

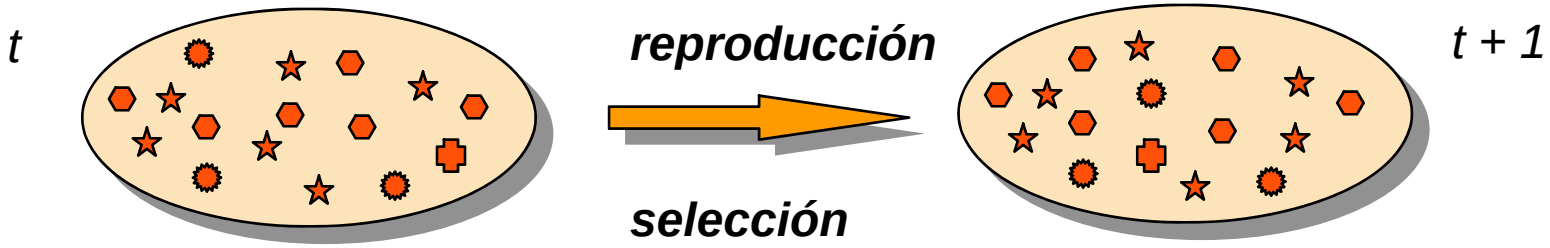
John Koza
Stanford University.
Inventor of Genetic Programming

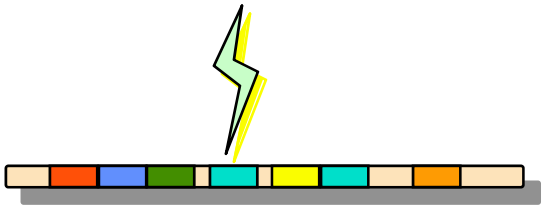
Existen otros múltiples Modelos de Evolución de Poblaciones

Evolución Artificial

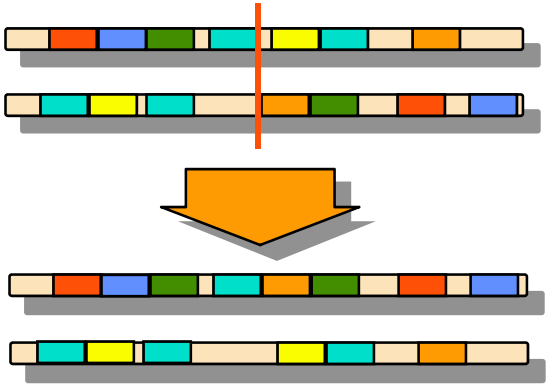


Los Ingredientes



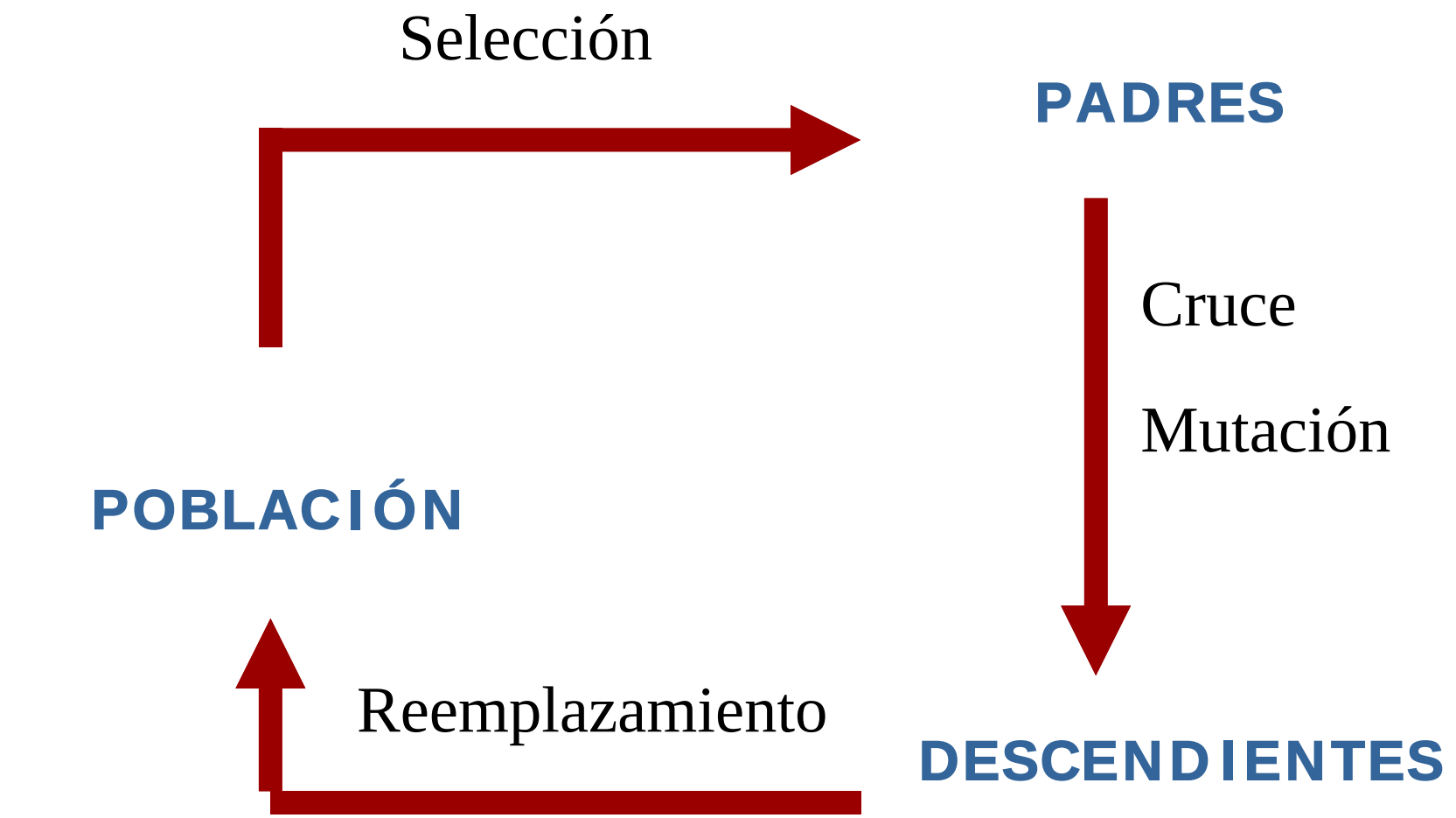


mutación



Cruce
(o recombinación)

El ciclo de la Evolución



Estructura de un Algoritmo Genético

Algoritmo Genético Básico

```
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)
          P''(t) ← cruce P'(t)
          P'''(t) ← mutación P''(t)
          P(t) ← reemplazamiento (P(t-1),P'''(t))
          evaluar P(t)
    Final(2)

Final(1)
```

2. ¿CÓMO SE CONSTRUYE UN AG?

Los pasos para construir un Algoritmo Genético

- Diseñar una representación
- Decidir cómo inicializar una población
- Diseñar una forma de evaluar un individuo
- Decidir cómo seleccionar los individuos para ser padres
- Diseñar un operador de cruce adecuado
- Diseñar un operador de mutación adecuado
- Decidir cómo reemplazar a los individuos
- Decidir la condición de parada

DEPENDE DEL PROBLEMA

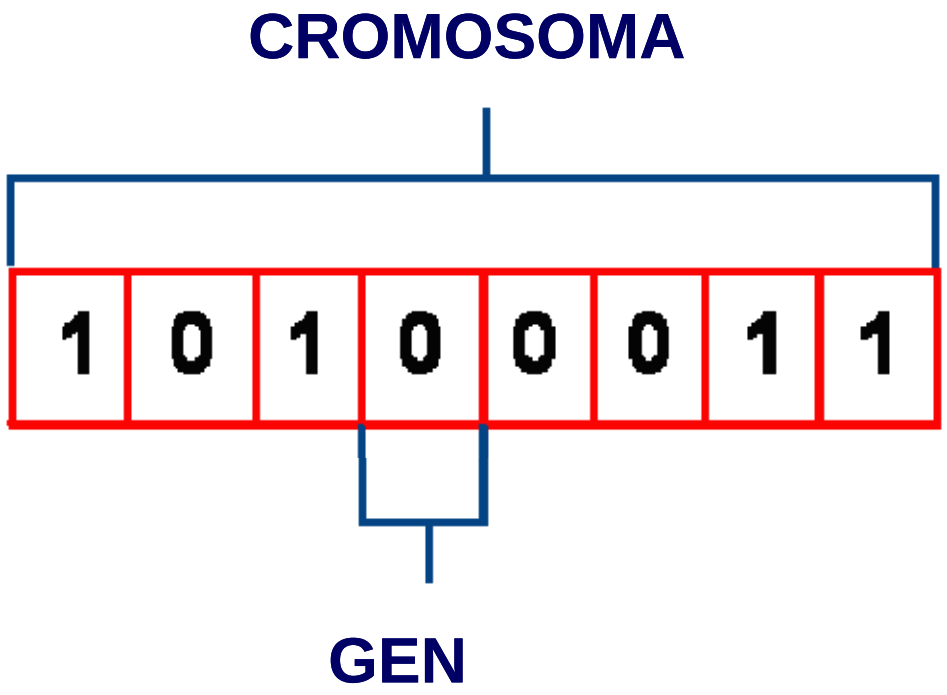
COMPONENTES DEL ALGORITMO

Representación

- Debemos disponer de un mecanismo para codificar un individuo como un genotipo
- Existen muchas maneras de hacer esto y se debe elegir la más relevante para el problema en cuestión
- Una vez elegida una representación, tenemos que tener en mente cómo se evaluarán los genotipos (codificación) y qué operadores genéticos habrá que utilizar

Ejemplo: Representación binaria

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



Ejemplo: Representación binaria

8 bits Genotipo

Fenotipo

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

Ejemplo: Representación binaria

El fenotipo puede ser uno o varios números enteros

Genotipo:

Fenotipo:

= 163

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

$$128 + 32 + 2 + 1 = 163$$

Ejemplo: Representación binaria

También puede corresponder a números reales

Ejemplo: un número entre 2.5 y 20.5 utilizando 8 dígitos binarios

Genotipo: **Fenotipo:**

= 13,9609

$$x = 2{,}5 + \frac{163}{256}\left(20{,}5 - 2{,}5\right) = 13{,}9609$$

Ejemplo: Representación Real

- Una forma natural de codificar una solución es utilizando valores reales como genes

- Muchas aplicaciones tienen esta forma natural de codificación

Ejemplo: Representación Real

- 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:

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

Ejemplo: Representación de orden

- Los individuos se representan como permutaciones
- 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

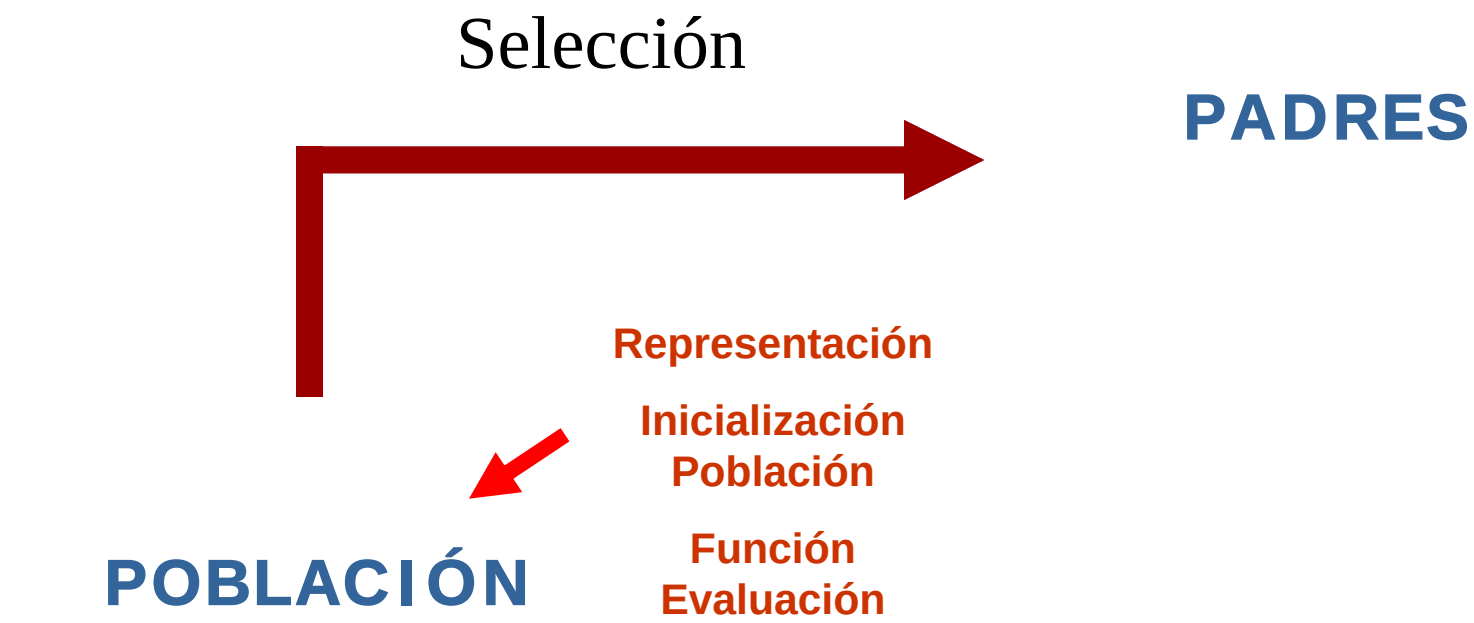
Inicialización

- Uniforme sobre el espacio de búsqueda ... (si es posible)
  - Cadena binaria: 0 ó 1 con probabilidad 0.5
  - Representación real: uniforme sobre un intervalo dado (para valores acotados)
- Elegir la población a partir de los resultados de una heurística previa

Evaluación de un individuo: *fitness* o valor de adecuación

- Este es el paso más costoso 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 la función de fitness como penalización del costo
- Con múltiples objetivos se busca una solución de compromiso

¿CÓMO SE CONSTRUYE UN AG?



Estrategia de Selección

Debemos de garantizar que los mejores individuos tienen una mayor posibilidad de ser padres (reproducirse) frente a los individuos menos buenos.

Debemos de 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.

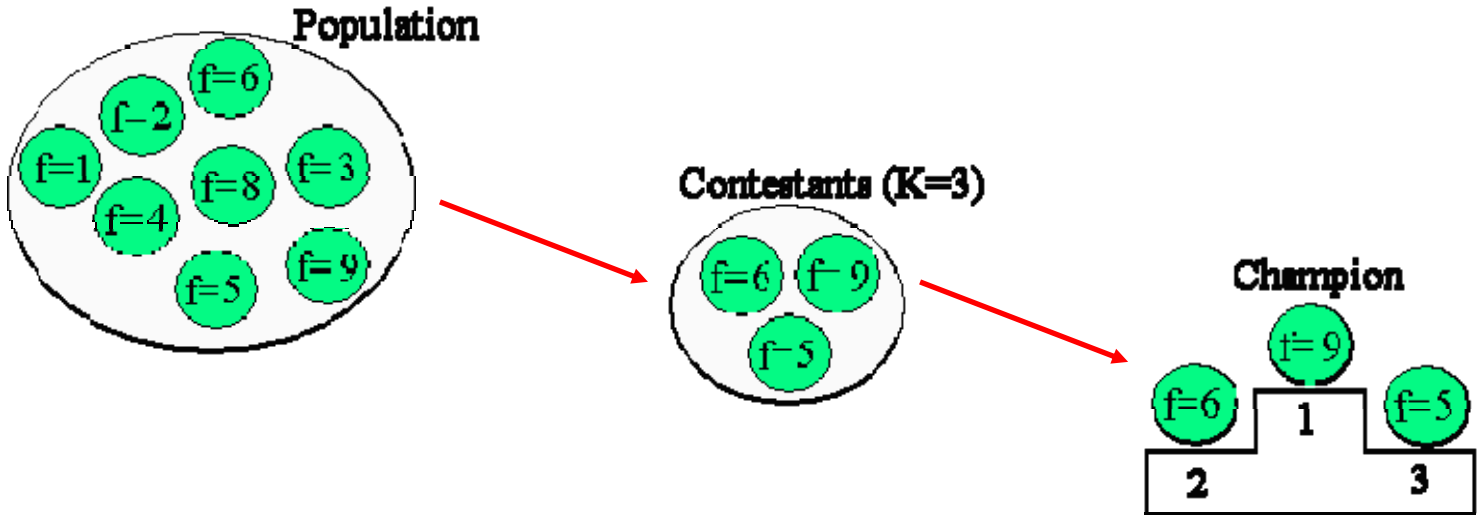
Estrategia de Selección

Selección por torneo

Para cada padre a seleccionar:

- Escoger aleatoriamente k individuos, con reemplazamiento
- Seleccionar el mejor de ellos

k se denomina tamaño del torneo. A mayor k, mayor presión selectiva y viceversa.

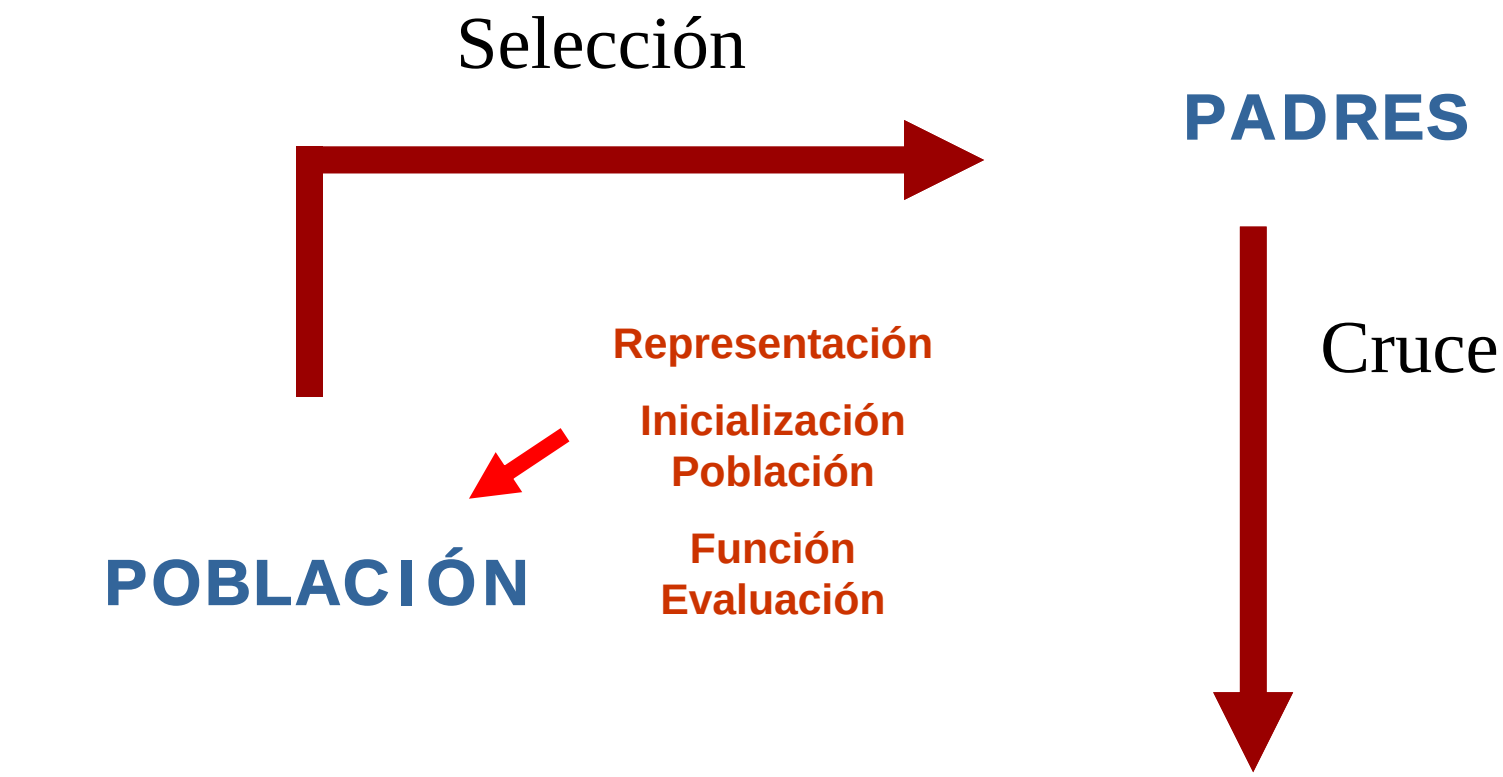


Estrategia de Selección

Algunos esquemas de selección

- Selección por Torneo (TS): Escoge al individuo de mejor fitness de entre Nts individuos seleccionados aleatoriamente (Nts=2,3,...).
- Orden Lineal (LR): La población se ordena en función de su fitness y se asocia una probabilidad de selección a cada individuo que depende de su orden.
- Selección Aleatoria (RS).
- Emparejamiento Variado Inverso (NAM): Un padre lo escoge aleatoriamente, para el otro selecciona Nnam padres y escoge el más lejano al primer (Nnam=3,5, ….). Está orientado a generar diversidad.
- Selección por Ruleta: Se asigna una probabilidad de selección proporcional al valor del fitness del cromosoma. (Modelo clásico)

¿CÓMO SE CONSTRUYE UN AG?



Operador de Cruce

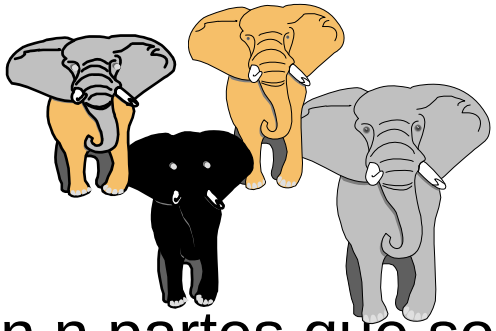
Podríamos tener uno o más operadores de cruce para nuestra representación.

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 (Pc entre 0.6 y 0.9), si no actua los padres son los descendientes del proceso de recombinación de la pareja.

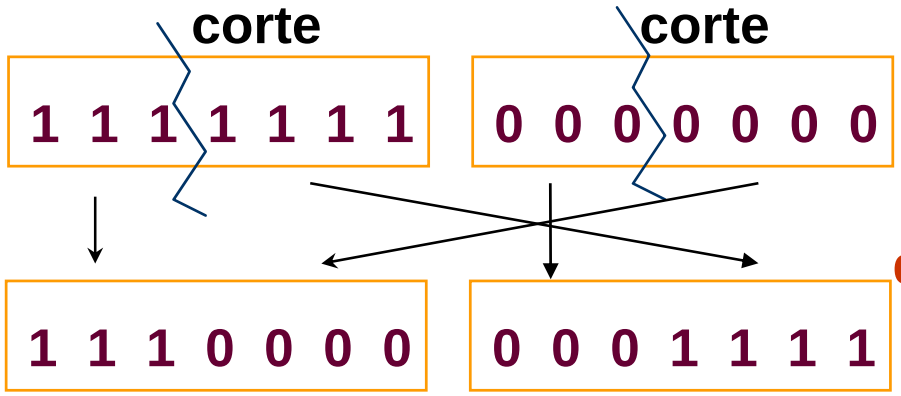
Ejemplo: Operador de cruce para representación binaria

Población:

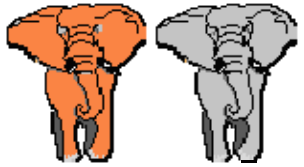


. . .

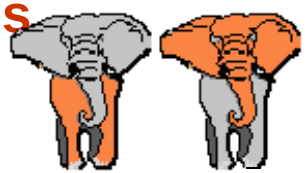
Cada cromosoma se corta en n partes que son recombinadas. (Ejemplo para n = 1).



padres

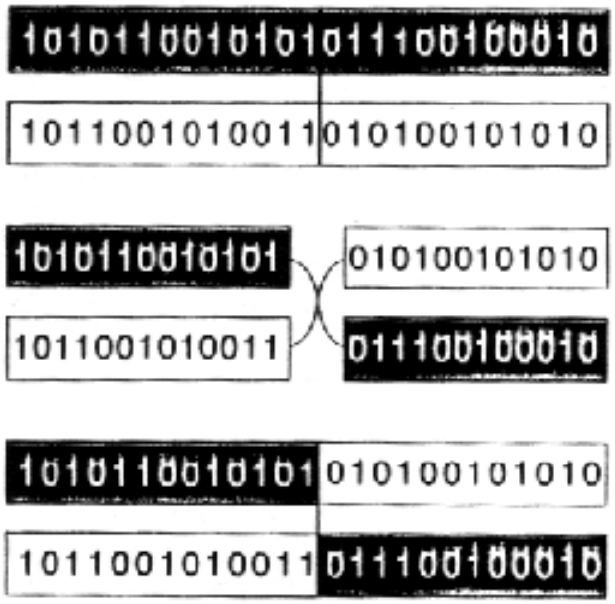


descendientes



Operador de Cruce

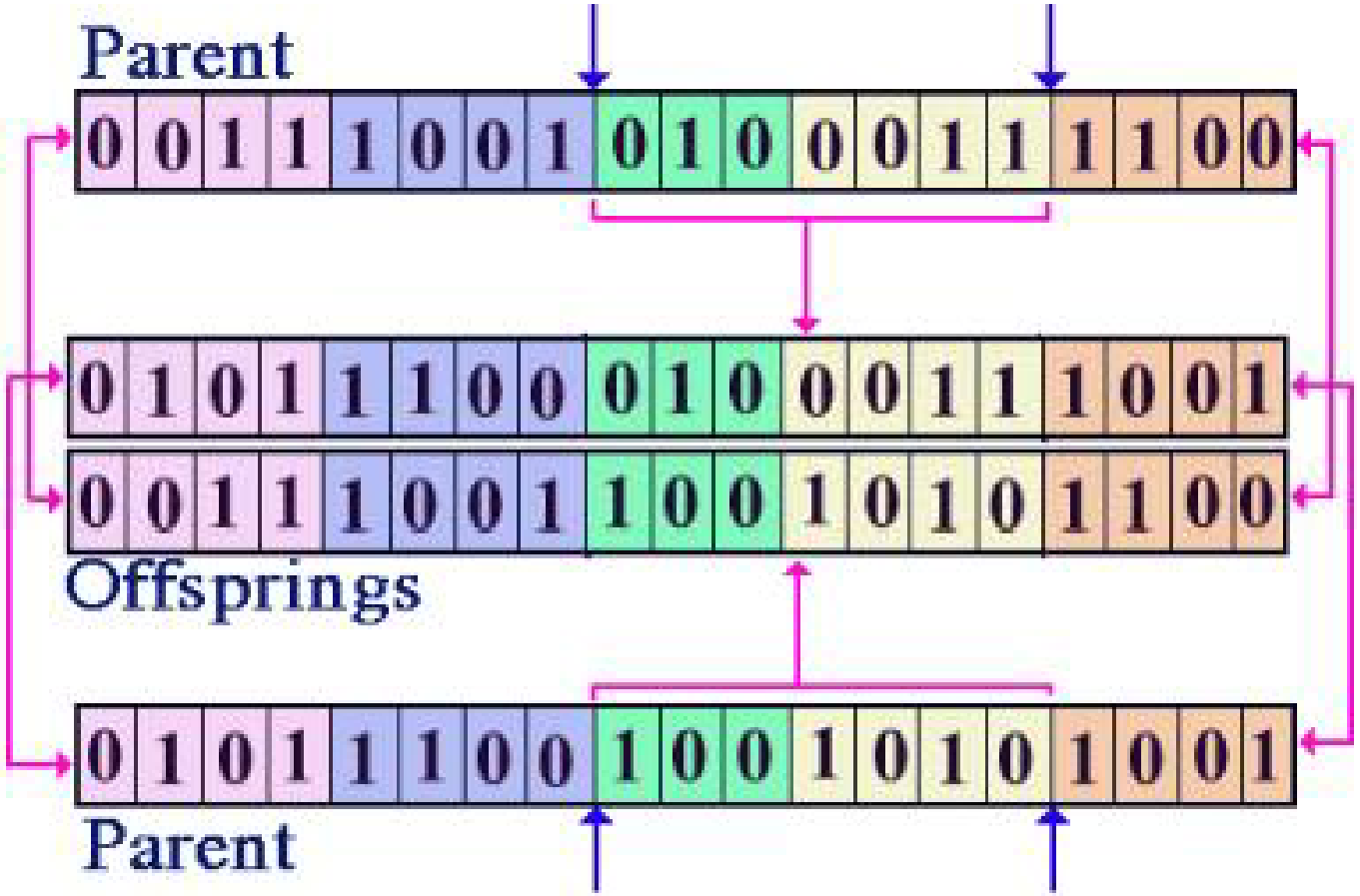
Imagen clásica (John Holland) que introduce el operador de cruce



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

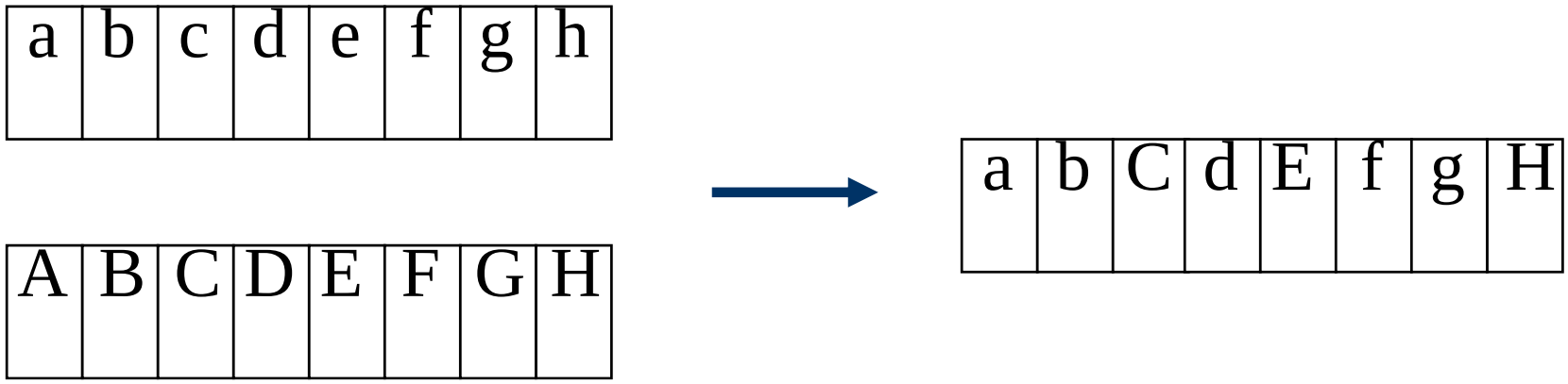
Chromosomes line up and then swap the portions of their genetic code beyond the crossover point.

Ejemplo: Operador de cruce en dos puntos para representación binaria



Ejemplo: Operador de cruce para representación real

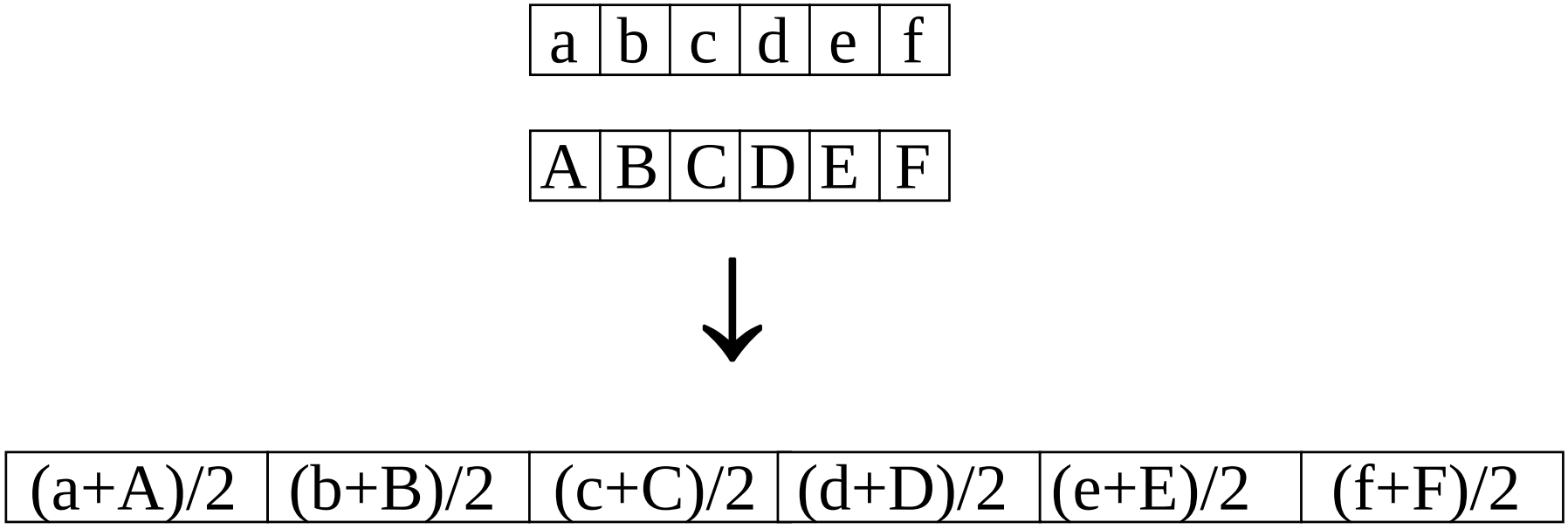
Recombinación discreta (cruce uniforme): dados 2 padres se crea un descendiente como sigue:



Existe muchos operadores específicos para la codificación real.

Ejemplo: Operador de cruce para representación real

Recombinación aritmética (cruce aritmético):



Ejemplo: Operador de cruce para representación real: **BLX-α**

- Dados 2 cromosomas

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

- BLX- α genera dos descendientes

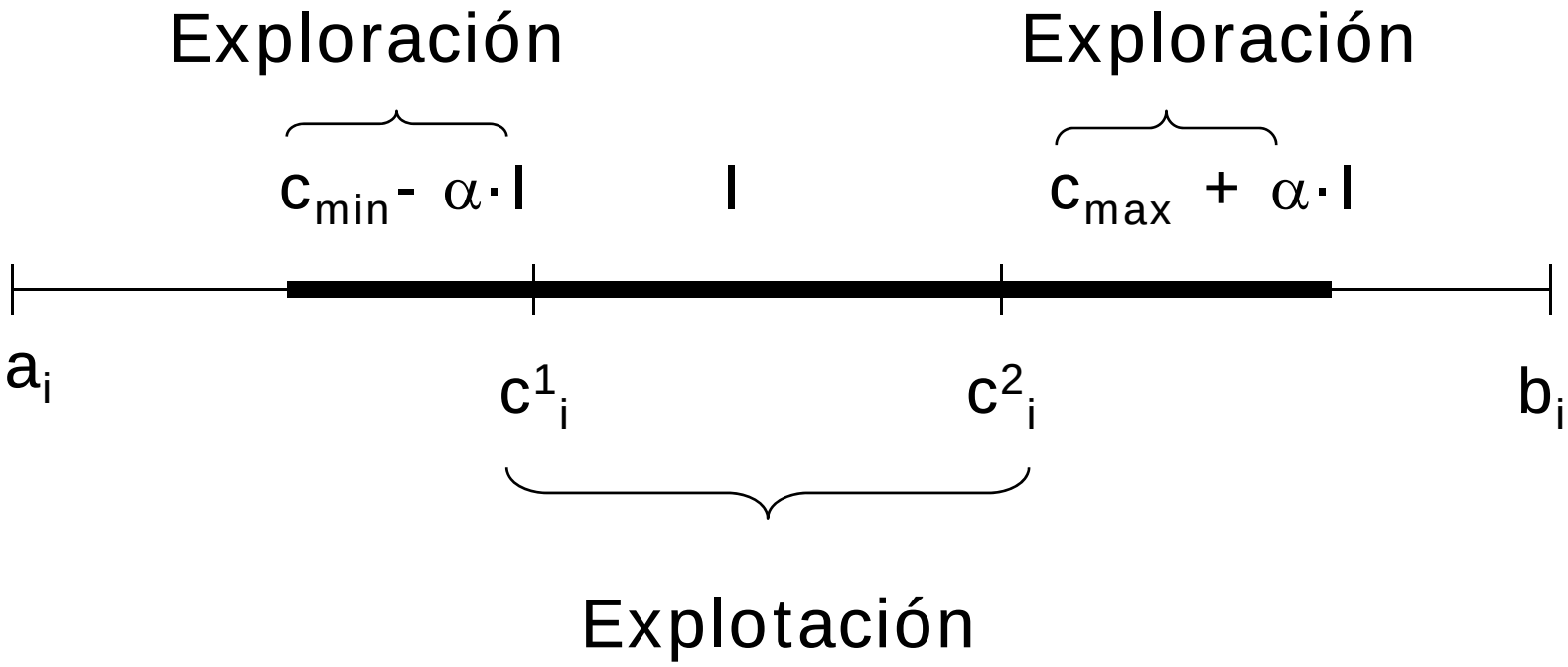
$$H_k = (h_{k1},\ldots, h_{ki},\ldots, h_{kn}) , k = 1,2$$

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

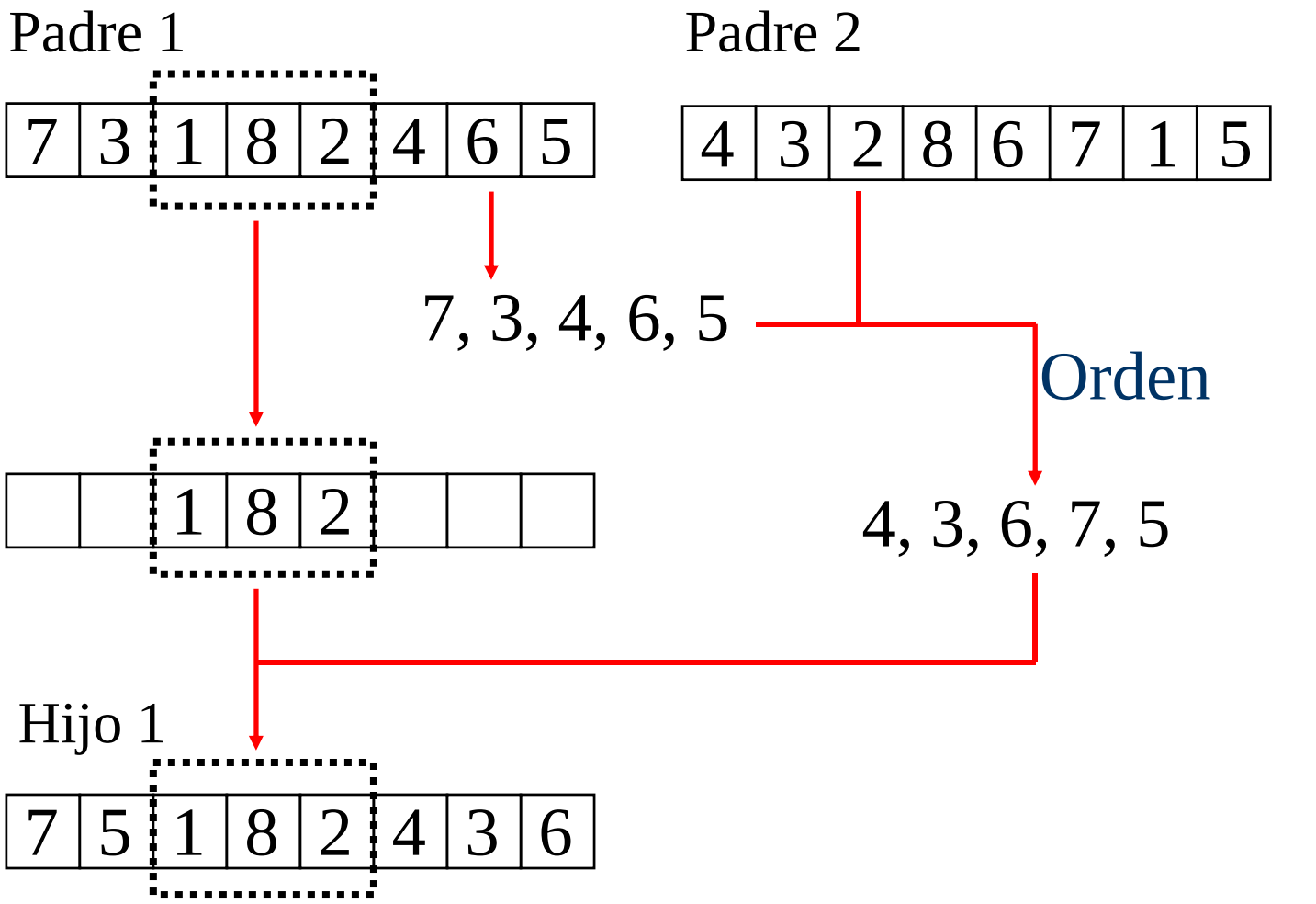
$$[C_{min} - I\cdot\alpha, C_{max} + I\cdot\alpha]$$

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

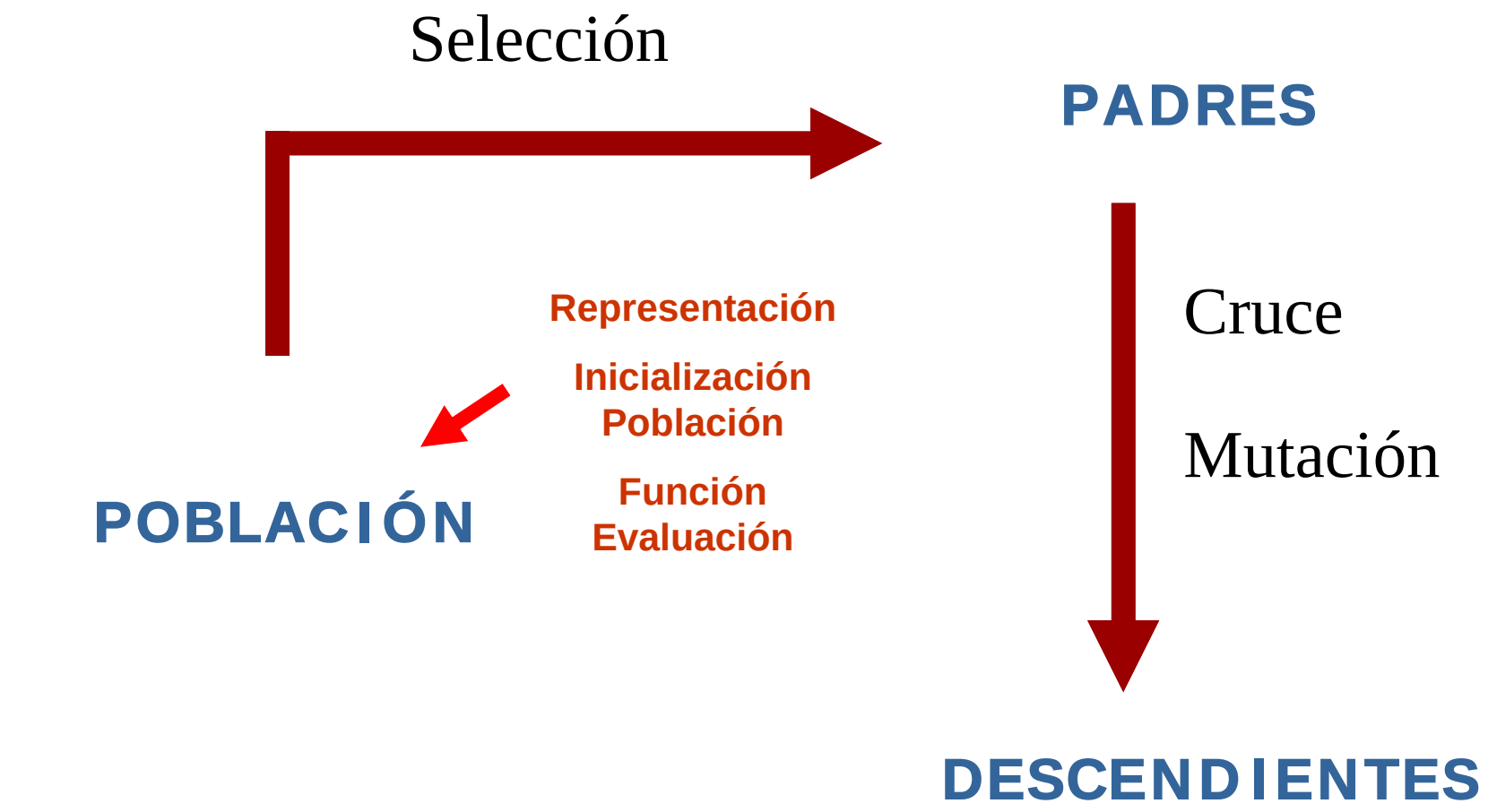
Ejemplo: Operador de cruce para representación real: **BLX-α**



Ejemplo: Operador de cruce para representación de orden OX



¿CÓMO SE CONSTRUYE UN AG?



Operador de mutación

Podemos tener uno o más operadores de mutación para nuestra representación.

Algunos aspectos importantes a tener en cuenta son:

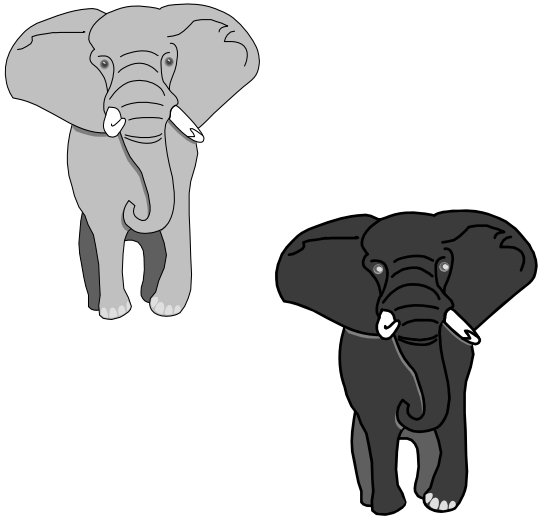
- Debe permitir alcanzar cualquier parte del espacio de búsqueda.
- El tamaño de la mutación debe ser controlado.
- Debe producir cromosomas válidos.
- Se aplica con una probabilidad muy baja de actuación sobre cada descendiente obtenido tras aplicar el operador de cruce (incluidos los descendientes que coinciden con los padres porque el operador de cruce no actúa).

Ejemplo: Mutación para representación discreta binaria

antes 1 1 1 1 1 1 1

después 1 1 1 0 1 1 1

gen mutado



La mutación ocurre con una probabiliad p_m para cada gen

Ejemplo: Mutación para representación real

Perturbación de los valores mediante un valor aleatorio.

Frecuentemente, mediante una distribución Gaussiana/normal $N(0,\sigma)$, donde

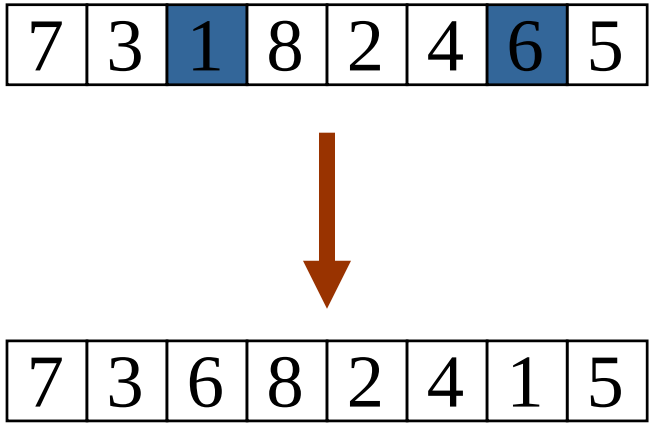
- 0 es la media
- σ es la desviación típica

$$x'_i = x_i + N(0,\sigma_i)$$

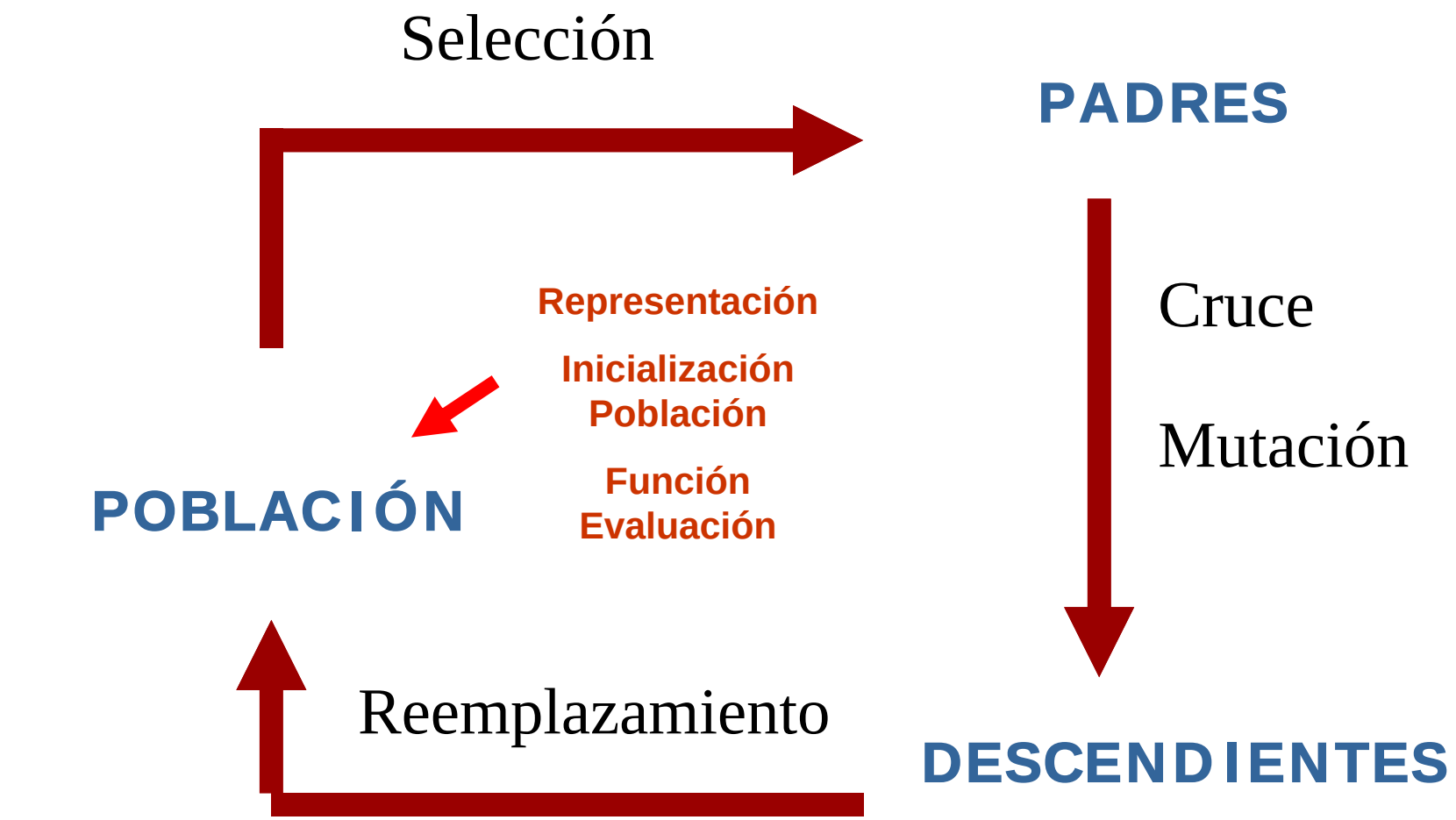
para cada parámetro.

Ejemplo: Mutación para representación de orden

Seleción aleatoria de dos genes e intercambio de ambos.



¿CÓMO SE CONSTRUYE UN AG?



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**

(el uso del Elitismo es aconsejado en los modelos generacionales para no perder la mejor solución encontrada).

Un modelo con alto grado de elitismo consiste en utilizar una población intermedia con todos los padres (N) y todos los descendientes y seleccionar los N mejores. Esto se combina con otras componentes con alto grado de diversidad.

Estrategia de Reemplazamiento

Algunas estrategias de reemplazo para AG estacionarios

Cuando se considera un modelo estacionario (en el que se reemplazan solo uno o dos padres, frente al modelo generacional en el que se reemplaza la población completa), nos encontramos con diferentes propuestas.

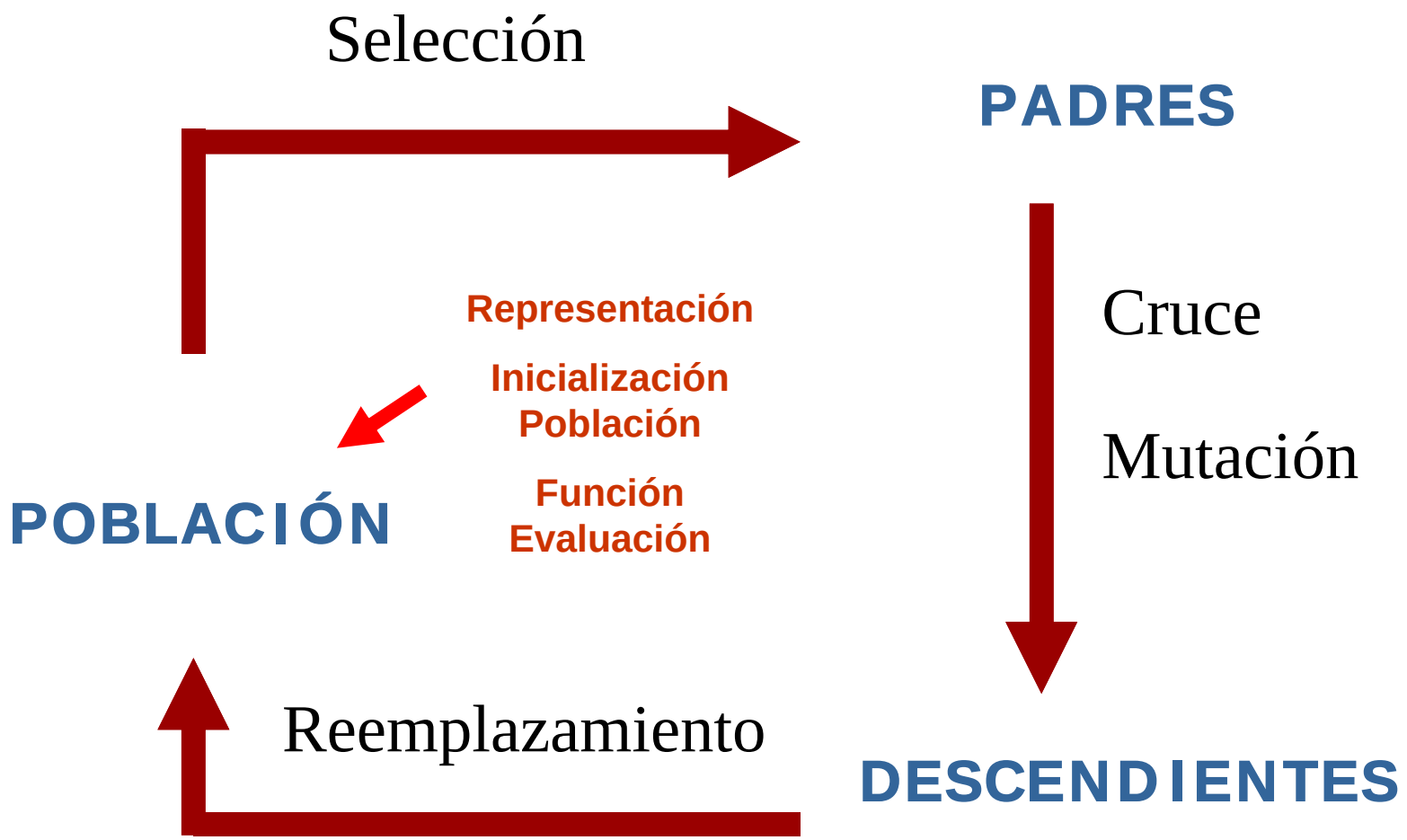
A continuación presentamos algunas posibilidades:

- Reemplazar al peor de la población (RW). Genera alta presión selectiva.
- Torneo Restringido (RTS): Se reemplaza al mas parecido de entre *w (w=3, ...)*. Mantiene una cierta diversidad.
- Peor entre semejantes (WAMS): Se reemplaza el peor cromosoma del conjunto de los *w (w=3, ...)* padres más parecidos al descendiente generado (seleccionados de toda la población). Busca equilibrio entre diversidad y presión selectiva.
- Algoritmo de Crowding Determinístico (DC): El hijo reemplaza a su padre más parecido. Mantiene diversidad.

Criterio de parada

- ¡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

Componentes

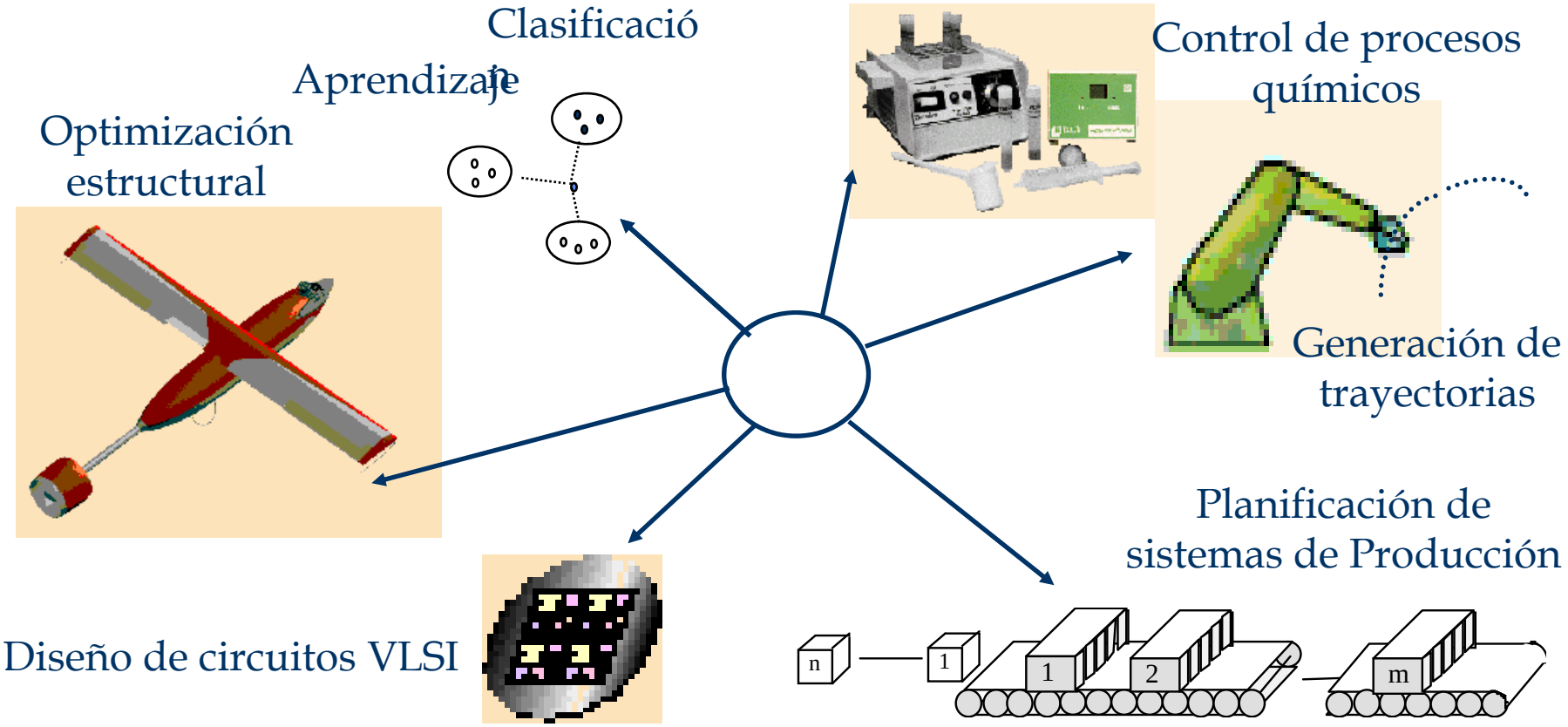


3. SOBRE SU UTILIZACIÓN

- Nunca sacar conclusiones de una única ejecución
  - utilizar medidas estadísticas (medias, medianas, ...)
  - con un número suficiente de ejecuciones independientes
- "Se puede obtener lo que se desea en una experimentación de acuerdo a la dificultad de los casos utilizados" – No se debe ajustar/chequear la actuación de un algoritmo sobre ejemplos simples si se desea trabajar con casos reales
- Desde el punto de vista de las aplicaciones:

  doble enfoque y diferente diseño
  - **Encontrar una solución muy buena al menos una vez**
  - **Encontrar al menos una solución muy buena en cada ejecución**

4. DOMINIOS DE APLICACIÓN



DOMINIOS DE APLICACIÓN

- Optimización combinatoria y en dominios reales
- Modelado e identificación de sistemas
- Planificación y control
- Ingeniería
- Vida artificial
- Aprendizaje y minería de datos
- Visión por Computador e Informática Gráfica
- Internet y Sistemas de Recuperación de Información
- ...

T. Bäck, D.B. Fogel, Z. Michalewicz, Handbook of Evolutionary Computation. Oxford Univ. Press, 1997

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

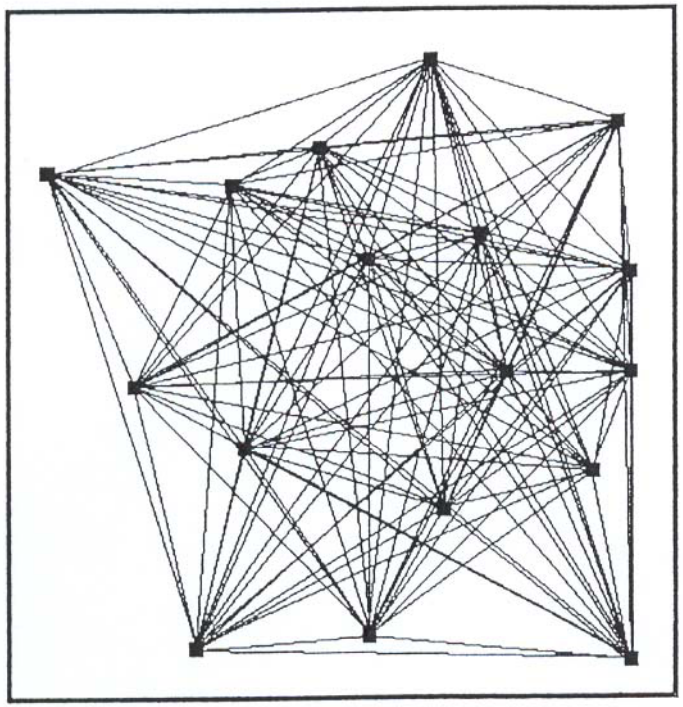
Objetivo: Suma de la distancia entre las ciudades.

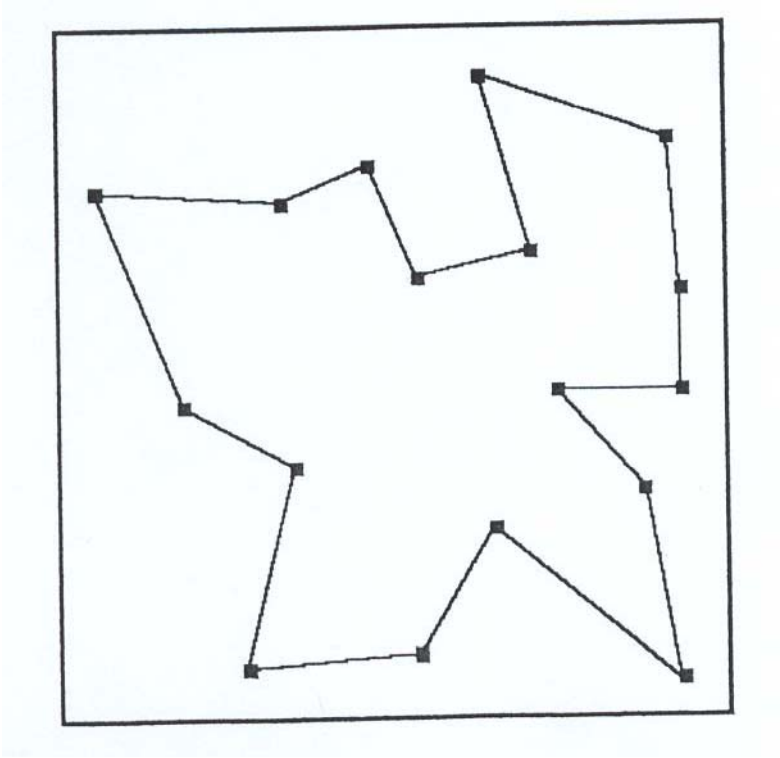
Población: 61 cromosomas - Elitismo

Cruce: OX (P_c = 0,6)

Mutación: Inversión de una lista (P_m = 0,01 – cromosoma)

Viajante de Comercio

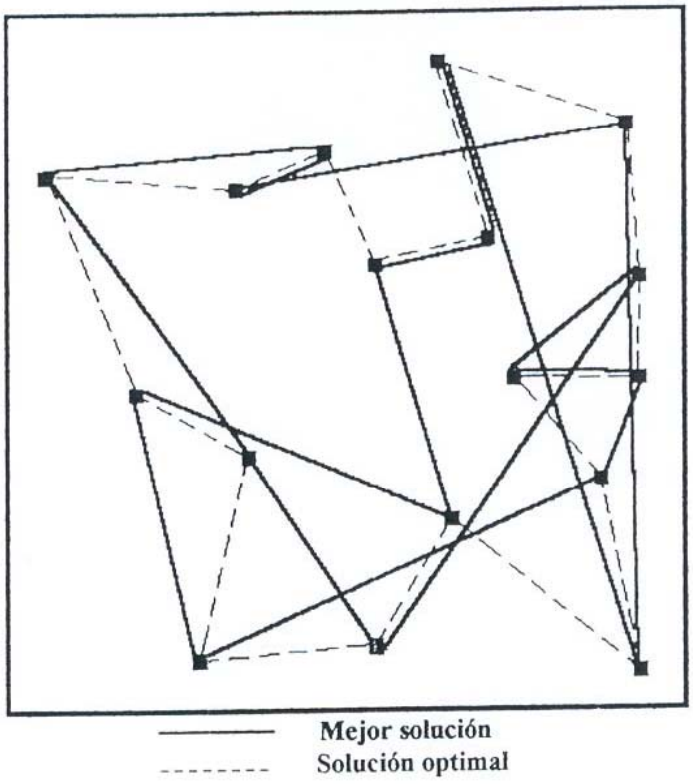


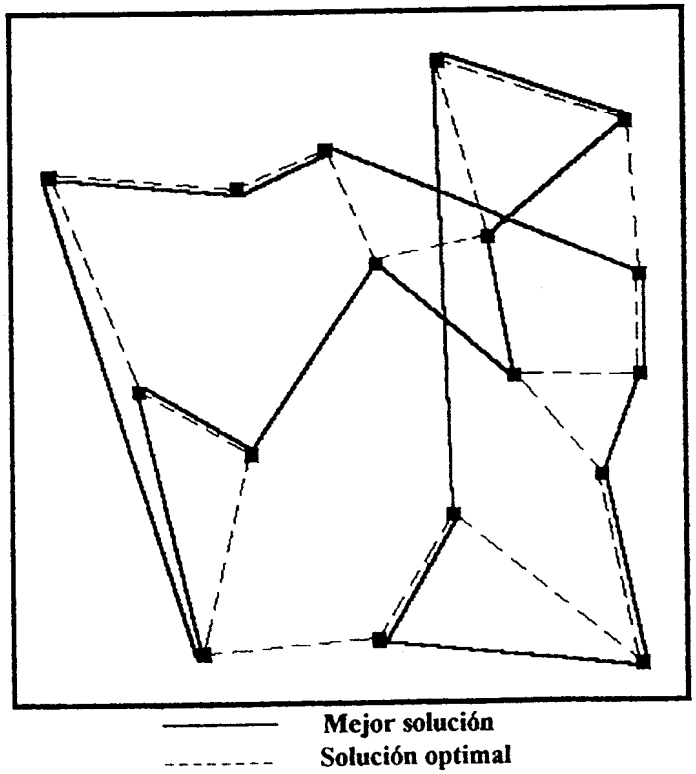


17! = 3.5568743 e14 recorridos posibles

Solución óptima: 226.64

Viajante de Comercio



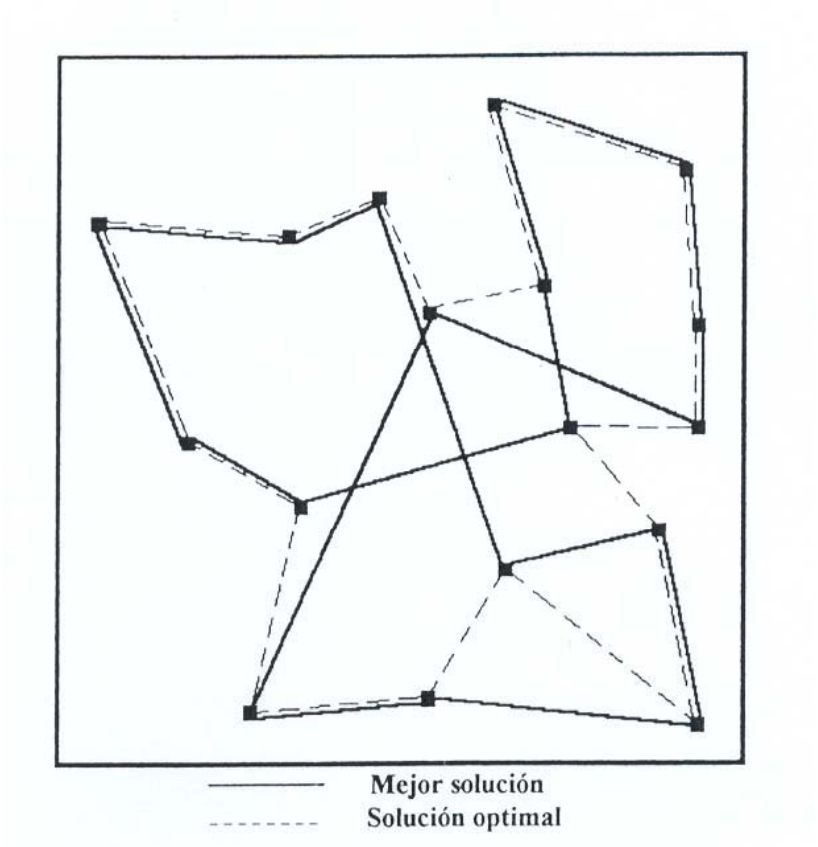


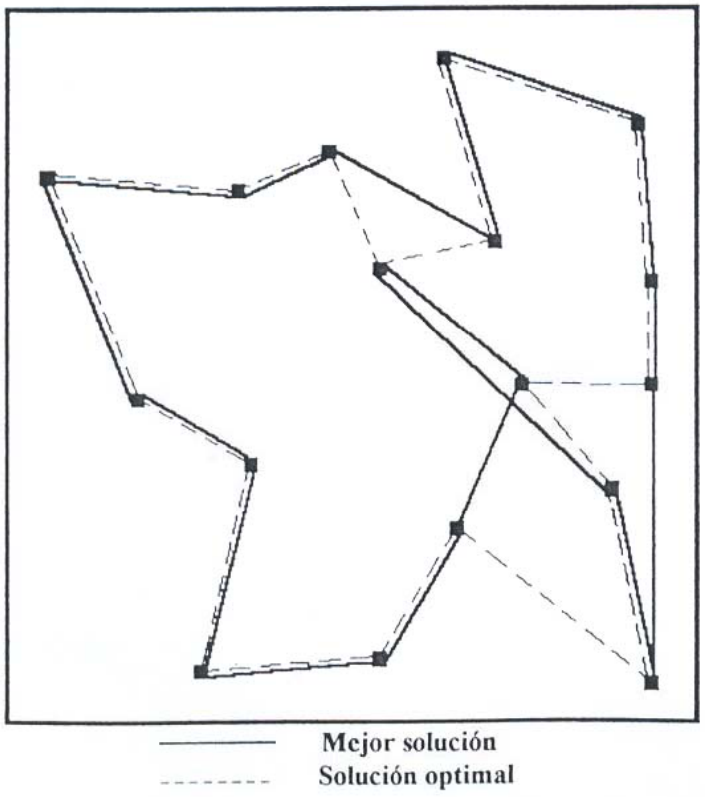
Iteración: 0 Costo: 403.7

Iteración: 25 Costo: 303.86

Solución óptima: 226.64

Viajante de Comercio



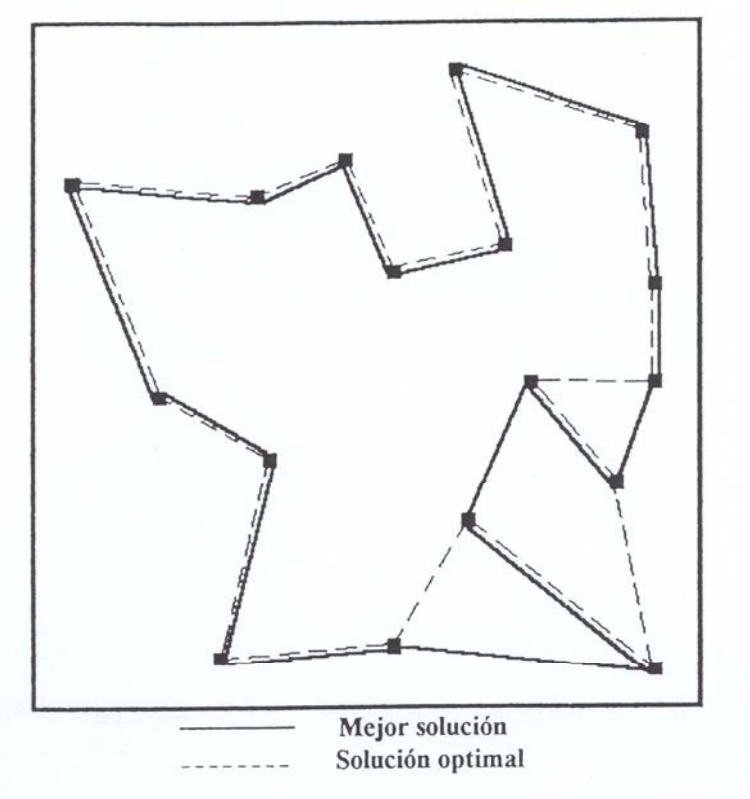


Iteración: 50 Costo: 293,6

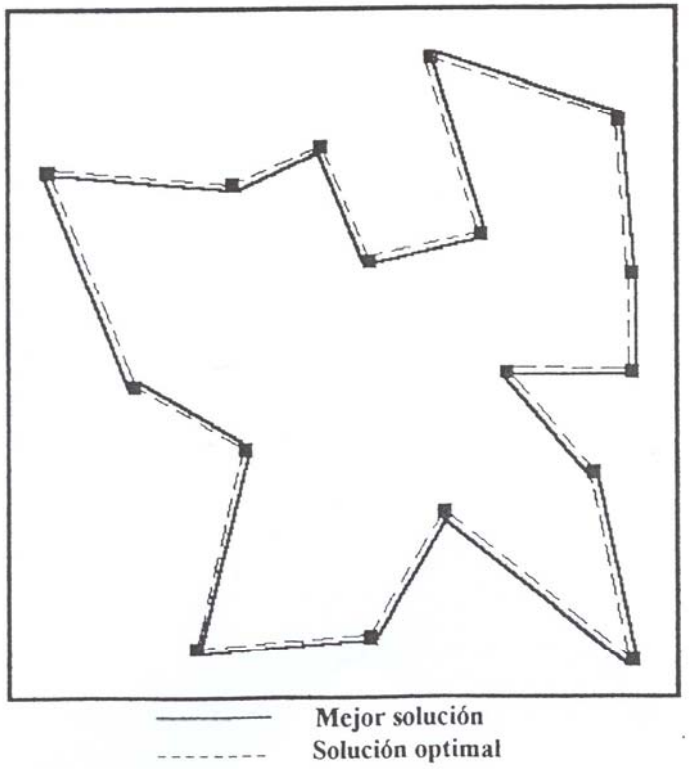
Iteración: 100 Costo: 256,55

Solución óptima: 226,64

Viajante de Comercio

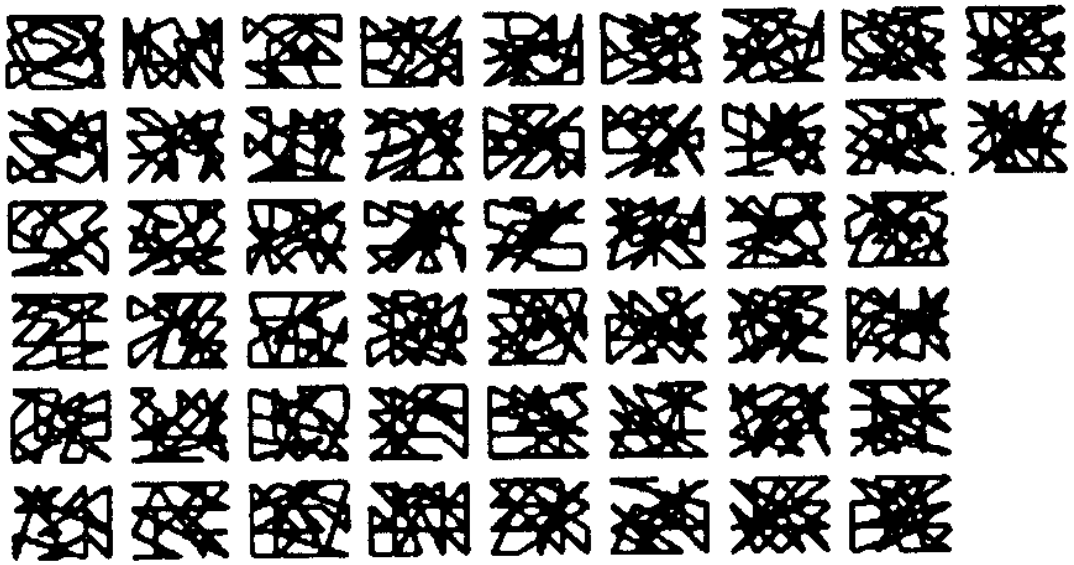


Iteración: 200 Costo: 231,4



Iteración: 250 Solución óptima: 226,64

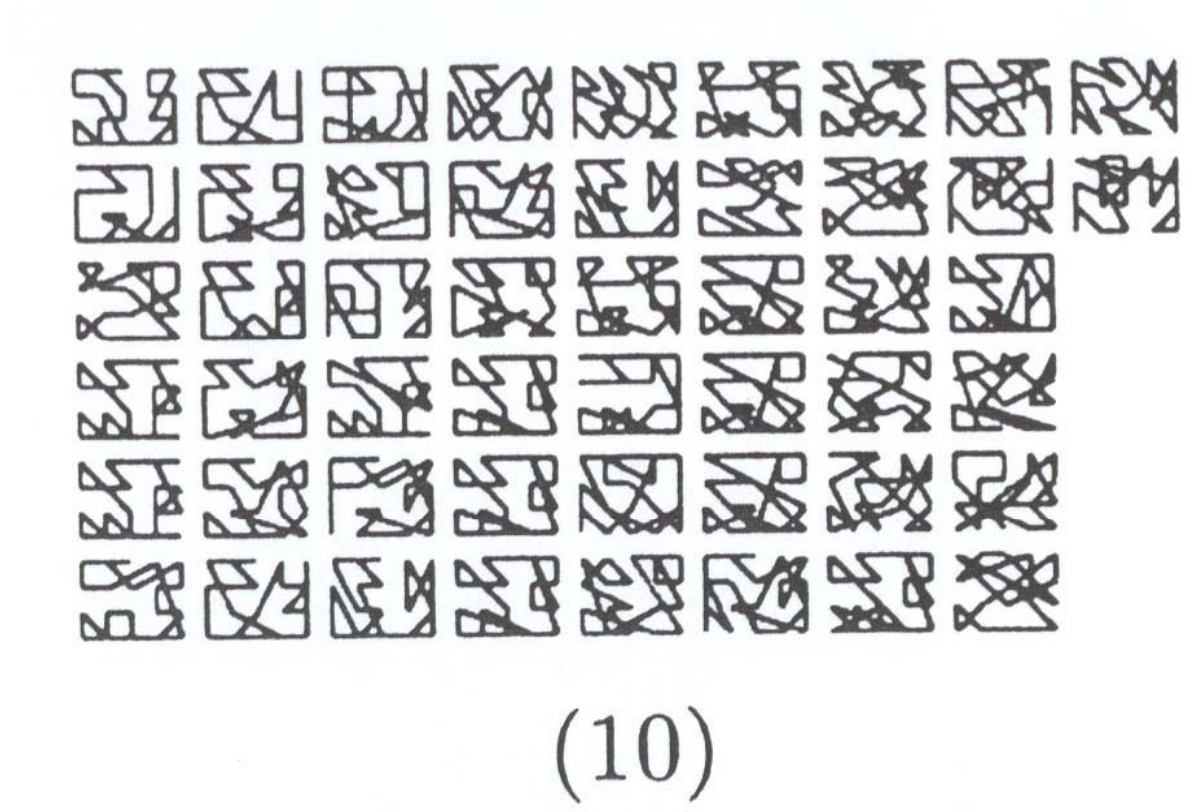
Viajante de Comercio



(0)

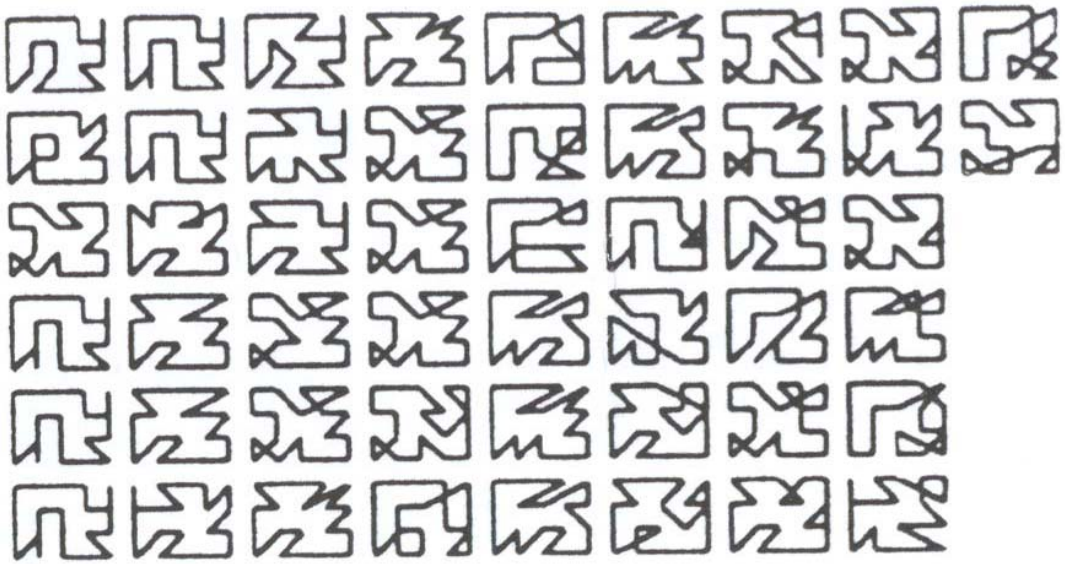
Visualización de la evolución de una población de 50 cromosomas y 70 iteraciones

Viajante de Comercio



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

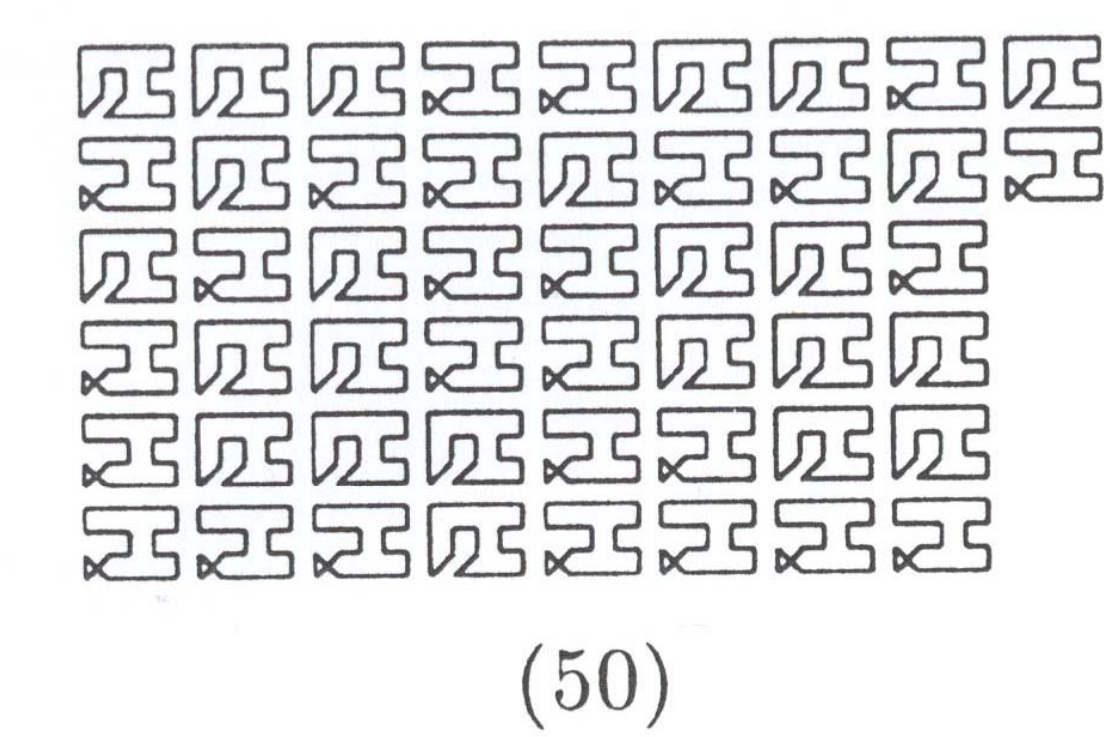
Viajante de Comercio



(30)

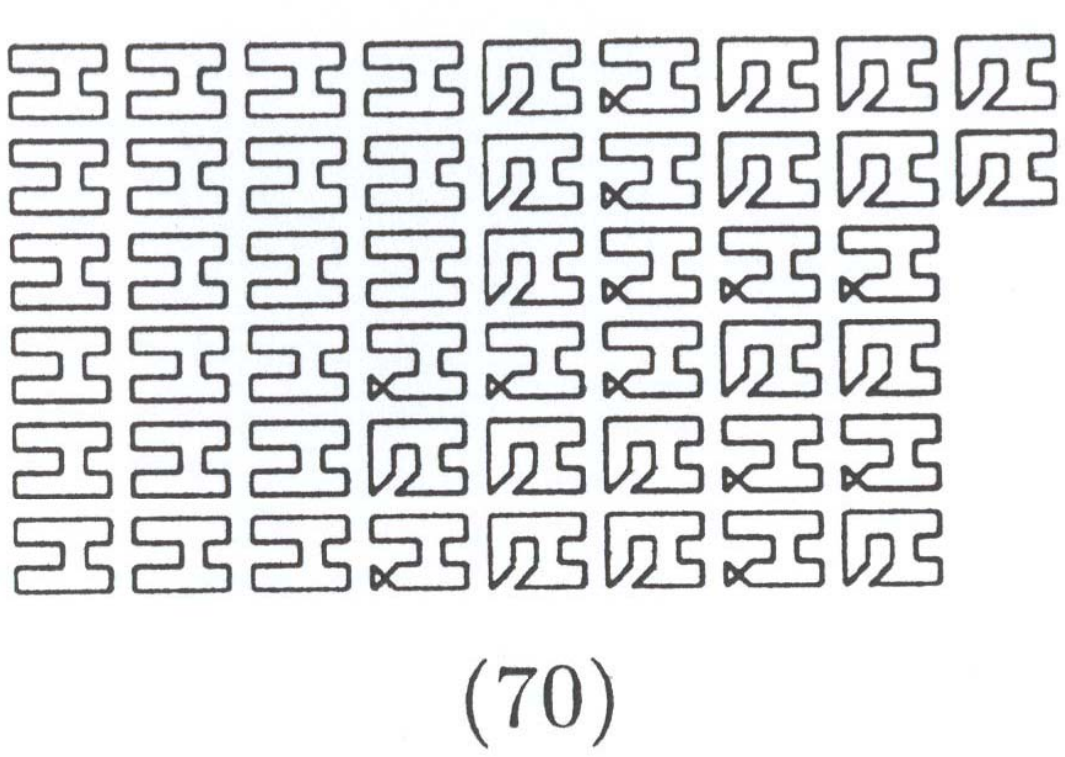
Visualización de la evolución de una población de 50 cromosomas y 70 iteraciones

Viajante de Comercio



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

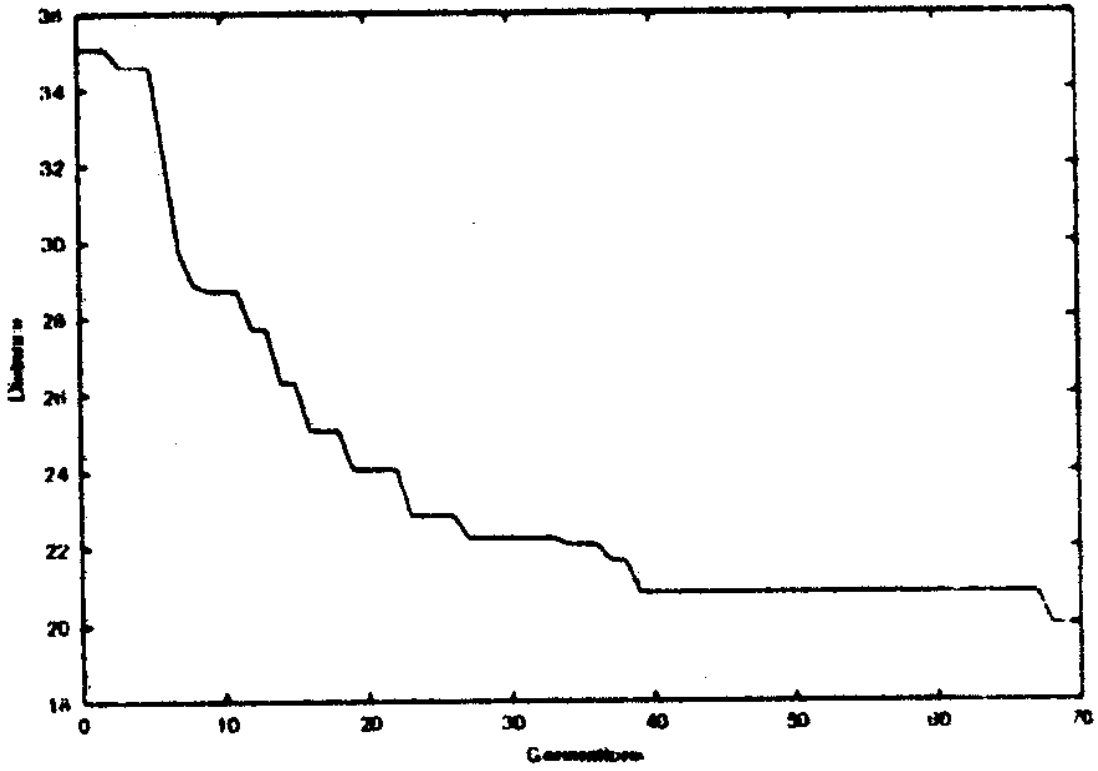
Viajante de Comercio



(70)

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

Viajante de Comercio

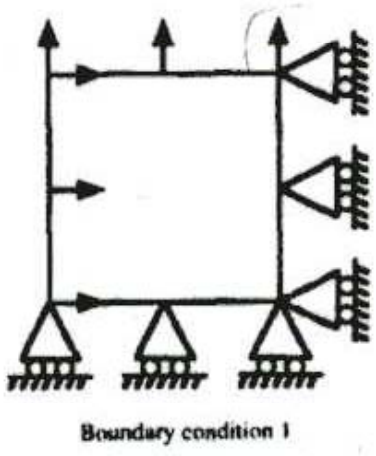


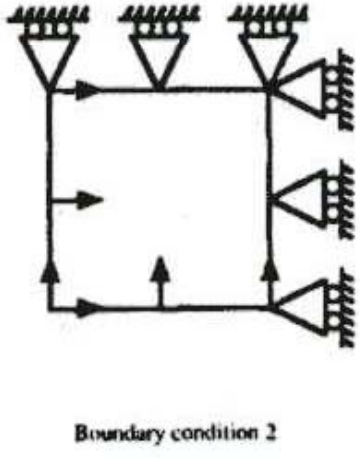
Visualización de la evolución de una población de 50 cromosomas y 70 iteraciones

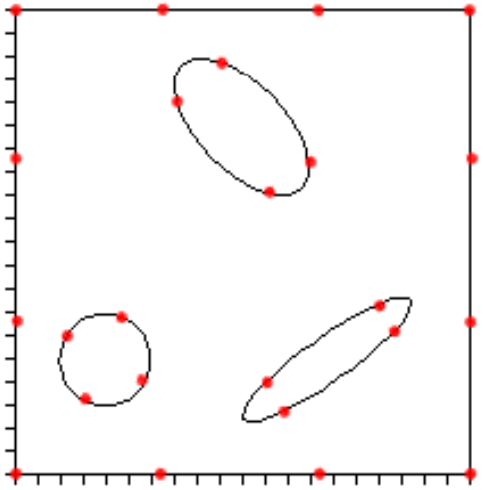
6. APLICACIONES

Ejemplo: Identificación de defectos en una placa cuadrada con análisis de contornos

- En ingeniería civil, el desarrollo de métodos no destructivos de detección de defectos en materiales ha tenido un fuerte desarrollo en los últimos años en orden a estimar la seguridad y resto de la vida útil de la estructura



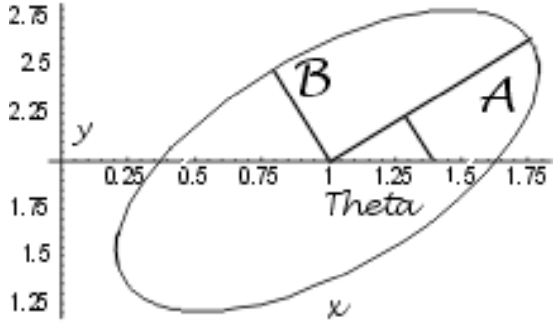




6. APLICACIONES

Ejemplo: Identificación de defectos en una placa cuadrada con análisis de contornos

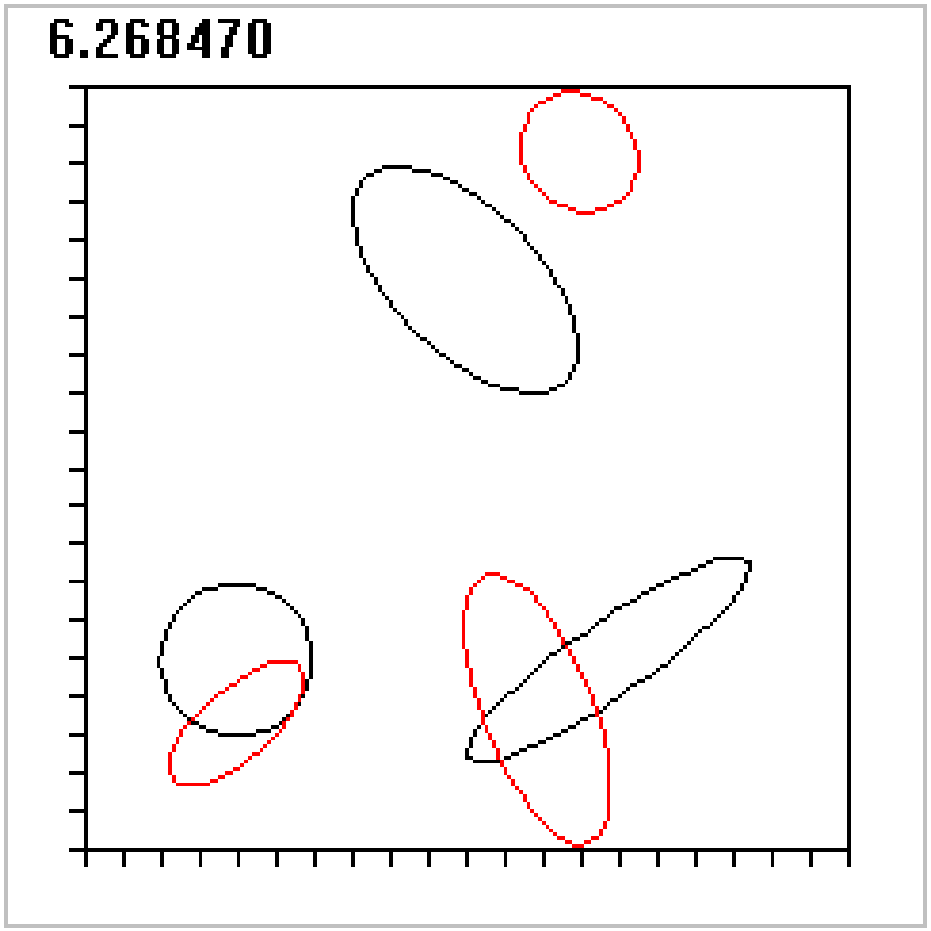
- Alterando los parámetros de cada elipse podemos generar soluciones al problema



- Para decidir la calidad de una solución se analiza la distancia euclídea legítima ponderada, entre los valores de las variables de tensión y desplazamiento simuladas y las observadas

$$Fitness = 10 - 1000\sqrt{\sum_{i=1}^{n} (u_x^i - \bar{u}_x^i)^2 + (t_x^i - \bar{t}_x^i)^2 + (u_y^i - \bar{u}_y^i)^2 + (t_y^i - \bar{t}_y^i)^2}$$

6. APLICACIONES



6. APLICACIONES

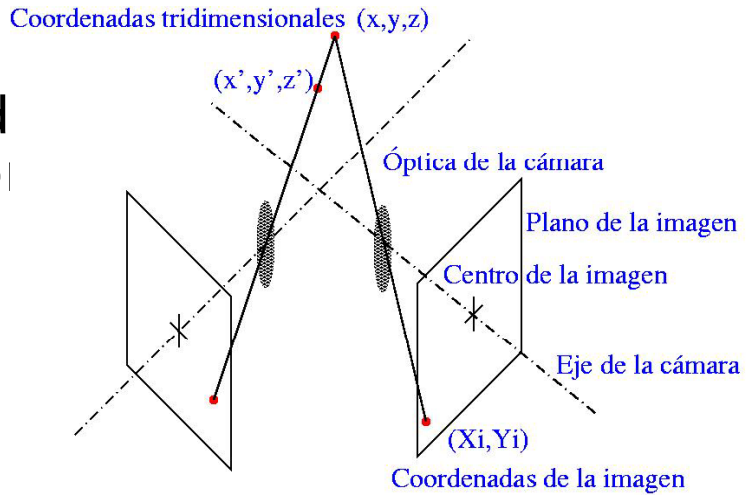
Ejemplo Real: Sistema de Bajo Coste para la Medición de Cotas de Vehículos en la ITV

- La medición de cotas (ancho, largo, alto, distancia entre ejes, ...) es una de las tareas del reglamento de la ITV
- Implica la atención de dos mecánicos inspectores, por lo que es interesante automatizarla. El presupuesto de los talleres requiere que se haga a bajo coste
- En la estación de la ITV de Pruvia, Asturias, se planteó la instalación de un sistema basado en cuatro cámaras, una tarjeta capturadora y un ordenador

6. APLICACIONES

Ejemplo Real: Sistema de Bajo Coste para la Medición de Cotas de Vehículos en la ITV

- Las cámaras trabajan a pares. Con la imagen de dos, se pued calcular la posición tridimensio de un punto



- Restricción: Las cámaras han de situarse de forma que todos los puntos de referencia en la medida de una cota sean visibles desde, al menos, dos de ellas
- El operario marca los puntos a medir en la pantalla del ordenador y el sistema calcula las distancias entre los puntos indicados

6. APLICACIONES

Ejemplo Real: Sistema de Bajo Coste para la Medición de Cotas de Vehículos en la ITV

6. APLICACIONES

Ejemplo Real: Sistema de Bajo Coste para la Medición de Cotas de Vehículos en la ITV

- Se produce un error de medida que depende de la posición de las cámaras. P.e., sólo moviendo una cámara 1 metro a la derecha, se reduce el error de 70.7 a 47.2 mm

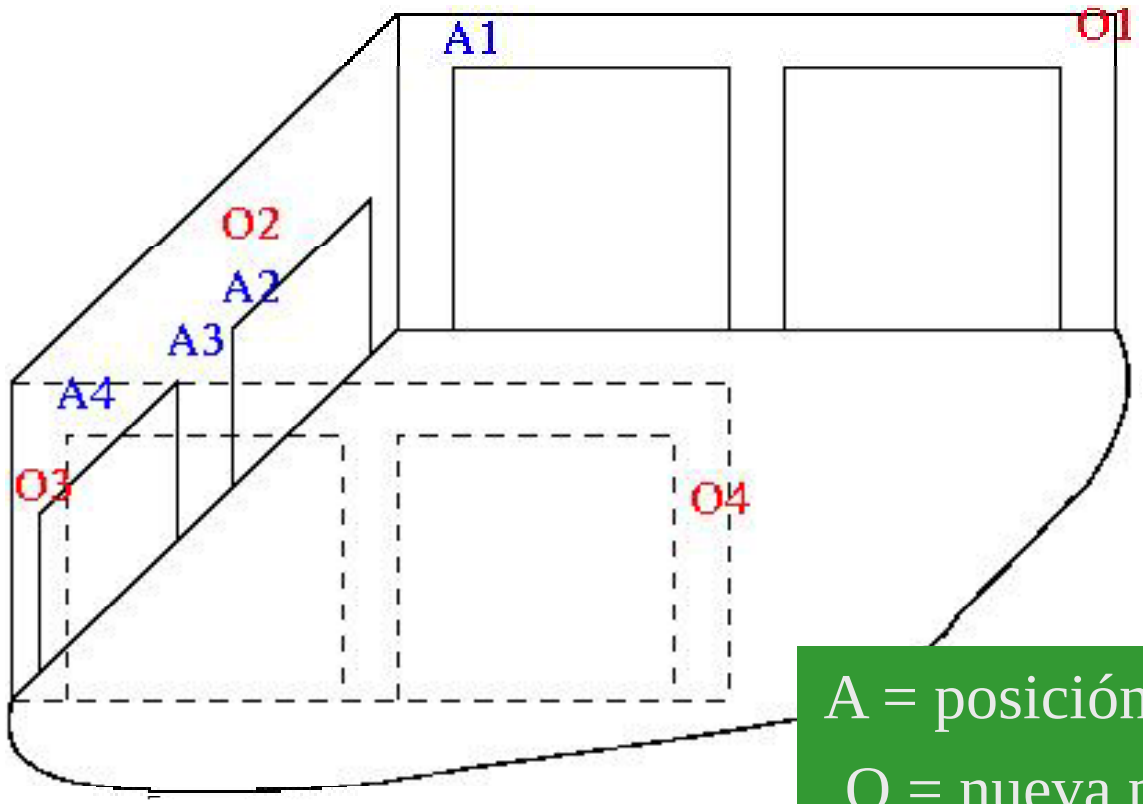
- Solución: Obtener la disposición de las cuatro cámaras resolviendo un problema de optimización con restricciones

- Se aplicó un algoritmo aproximado al problema sobre la planta real de la estación de ITV de Pruvia y se redujo el error medio de 14 a 5 cm

6. APLICACIONES

Ejemplo Real: Sistema de Bajo Coste para la Medición de Cotas de Vehículos en la ITV

PLANTA DE LA ESTACIÓN DE ITV DE PRUVIA



A = posición original

O = nueva posición

7. ALGUNAS CUESTIONES: DIVERSIDAD, EXPLORACIÓN vs EXPLOTACIÓN

Diversidad genética

- Asociada a las diferencias entre los cromosomas en la población
- Falta de diversidad genética = todos los individuos en la población son parecidos
- ***Falta de diversidad*** → ***convergencia al vecino más cercano***
- ***Alta presión selectiva*** → ***falta de diversidad***
- En la práctica es irreversible. **Soluciones:**
  - ***Inclusión de mecanismos de diversidad en la evolución***
  - ***Reinicialización cuando se produce convergencia prematura***

ALGUNAS CUESTIONES: DIVERSIDAD, EXPLORACIÓN vs EXPLOTACIÓN

Exploración vs Explotación

- **Exploración** = muestrear regiones desconocidas

  Excesiva exploración = búsqueda aleatoria, no convergencia

- **Explotación** = trata de mejorar el mejor individuo

  Excesiva explotación = solo búsqueda local … convergencia a un óptimo local

8. MODELOS: GENERACIONAL vs ESTACIONARIO

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

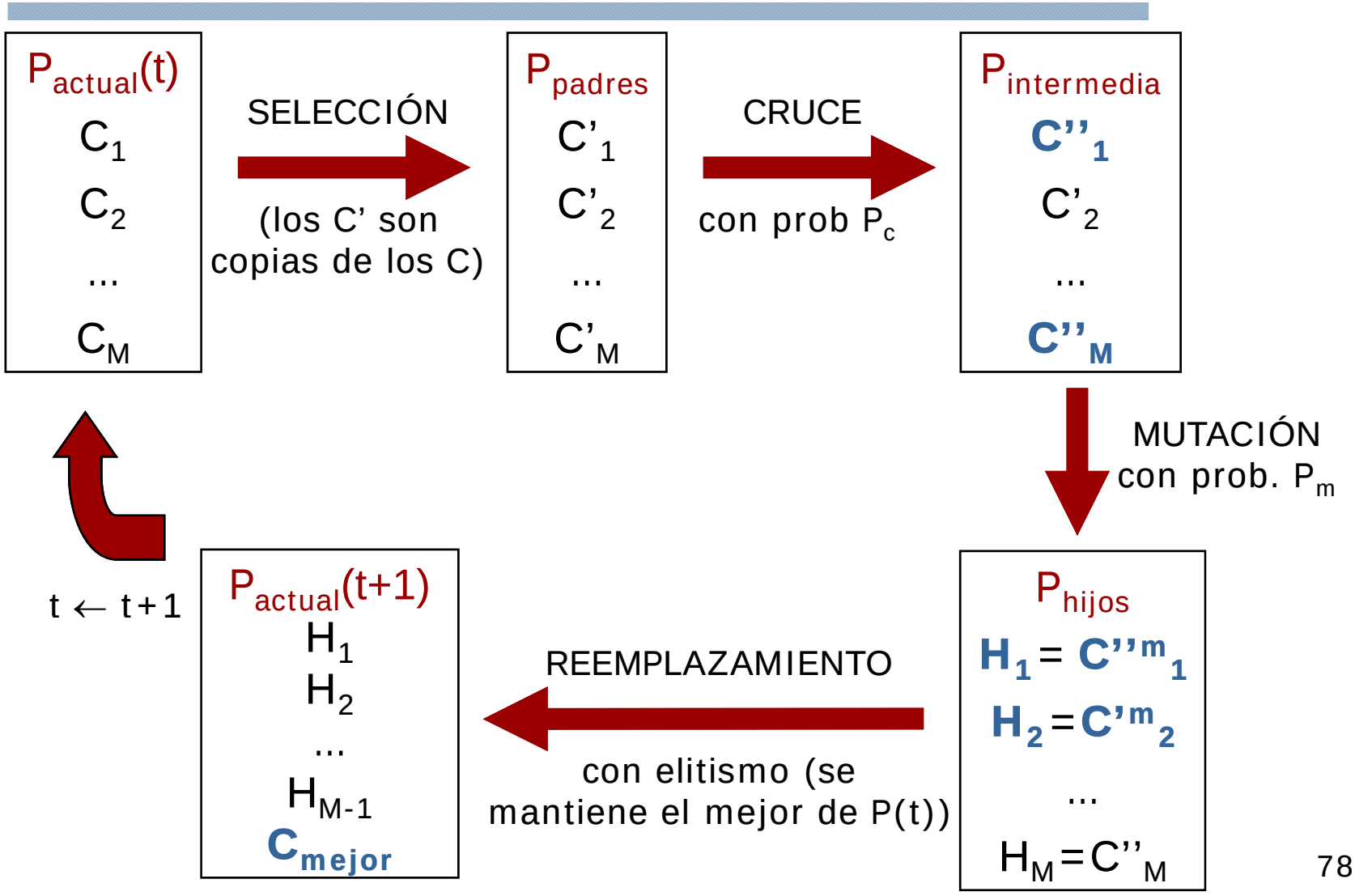
La nueva población reemplaza directamente a la antigua

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

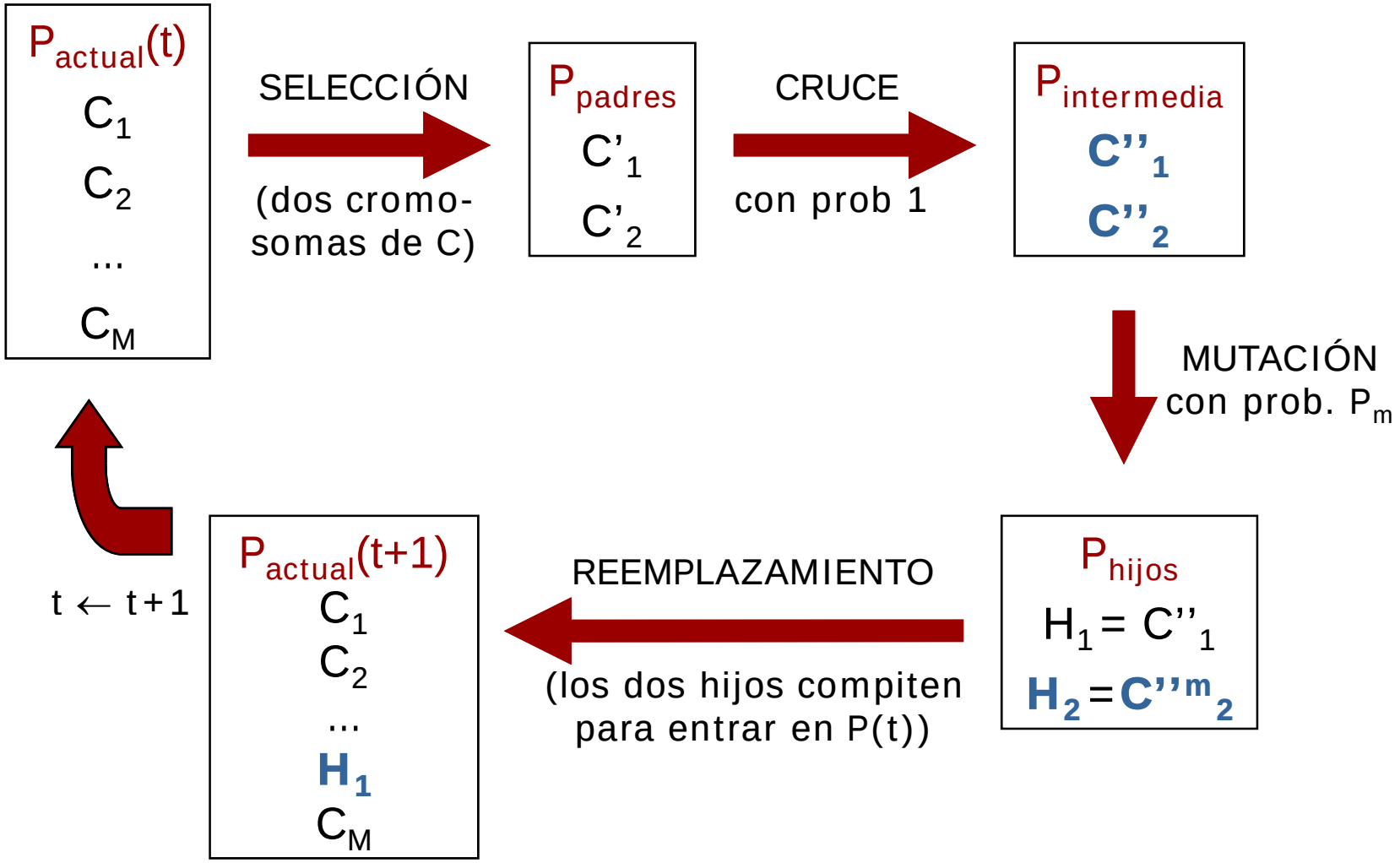
El/los descendiente/s reemplaza/n a uno/dos cromosoma/s de la población inicial

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

Modelo Generacional



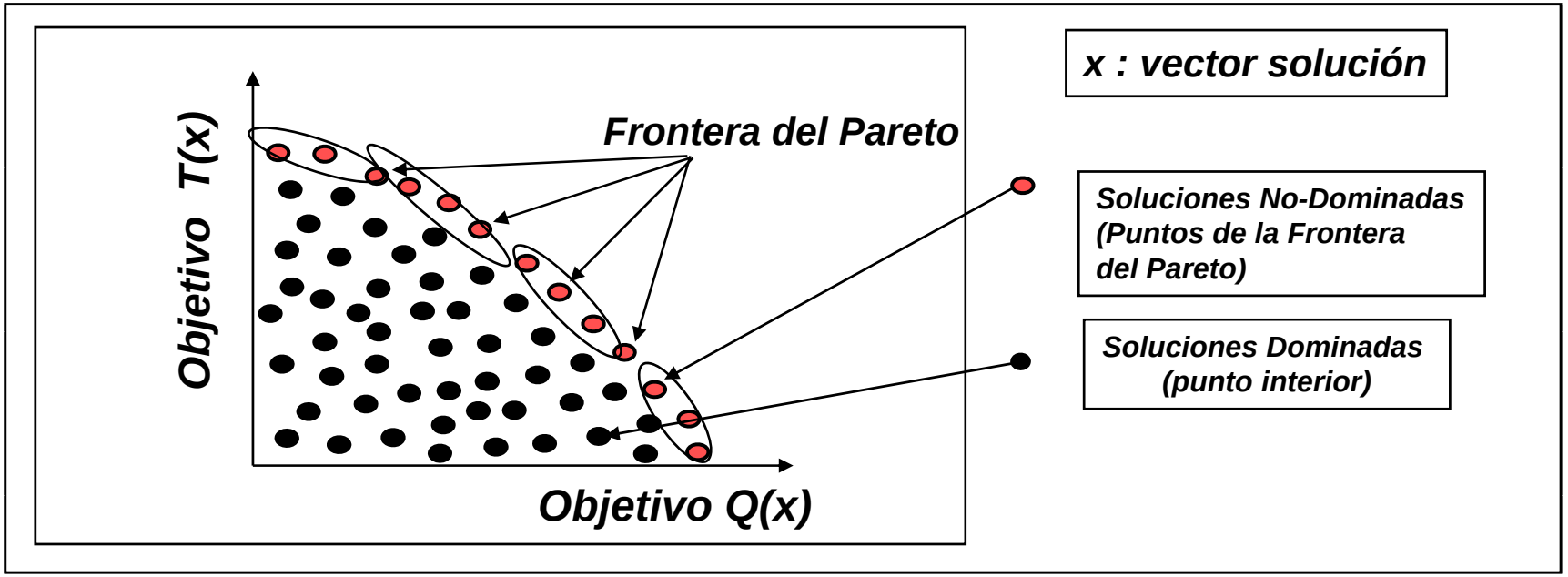
Modelo Estacionario



8. Algunas extensiones

8. Algunas extensiones

Problemas multiobjetivo



ALGORITMOS GENÉTICOS MULTIOBJETIVO

1. ***PROBLEMAS MULTIOBJETIVO***
2. ***EVOLUCIÓN EN PROBLEMAS MULTIOBJETIVO***
3. ***ELITISMO EN LA BÚSQUEDA EVOLUTIVA MULTIOBJETIVO***
4. ***NSGA-II (CONSIDERADO EL MEJOR MODELO)***
5. ***MÉTRICAS DE COMPARACIÓN***

K. Deb, Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons, 2001.

C.A. Coello, D.A. Van Veldhuizen, G.B. Lamont, Evolutionary Algorithms for Solving Multi-Objective Problems. Kluwer Academic Pub., 2002.

1. Problemas Multiobjetivo

- Muchos problemas reales se caracterizan por la existencia de **multiples medidas de actuación**, las cuales deberían ser optimizadas, o al menos ser satisfechas simultáneamente.

Ejemplo: Diseño de un sistema control de aire acondicionado, optimizando el conjunto de parámetros de un sistema de control:

  - Minimizar el consumo de energía
  - Maximizar el confort de los usuarios
  - Maximizar la estabilidad del sistema de control, ...

Problemas Multiobjetivo (2)

Un **Problema Multiobjetivo** consiste en:

$$\text{Max o Min } \; z = f(x) = (f_1(x), f_2(x), ..., f_n(x))$$

Soluciones pareto-optimales o no-dominadas: Se dice que un vector ***a*** domina a otro ***b*** (se nota como $a \prec = b$) si, y sólo si:

$$\forall i \in \{1,2,...,n\} \mid f_i(a) \geq f_i(b) \wedge \exists\, j \in \{1,2,...,n\} \mid f_i(a) > f_i(b)$$

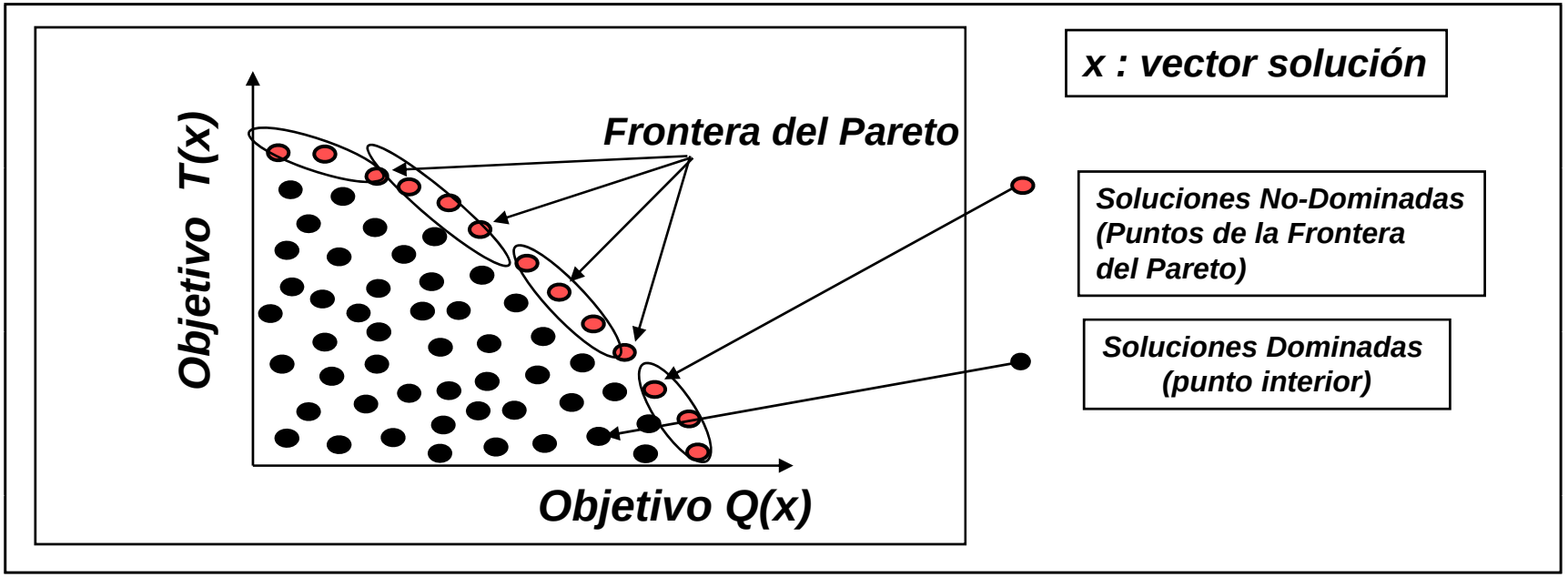
Es decir, una solución domina a otra si es mejor o igual en todos los objetivos y al menos mejor en uno de ellos.

Todos los vectores de decisión que no son dominados por ningún otro vector se llaman **pareto-optimales o no-dominados.**

Problemas Multiobjetivo (3)

- No suele existir una única solución optimal, existe un conjunto (a veces infinito) de soluciones No-Dominadas que forma la Frontera del Pareto
  - Ejemplo:

Identificar la frontera del Pareto para [Max Q(x), Max T(x)]

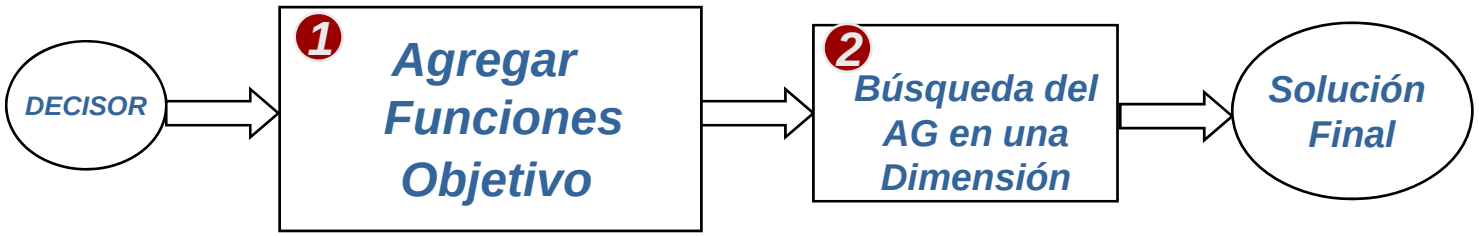


Problemas Multiobjetivo (4)

¿Qué necesitamos para resolver este problema?:

- *Un método de búsqueda basado en los múltiples objetivos.*
- *Una política de equilibrio entre los objetivos.*
- *Un orden para este proceso de optimización.*

Vamos a considerar 2 posibilidades: a) Agregación + búsqueda



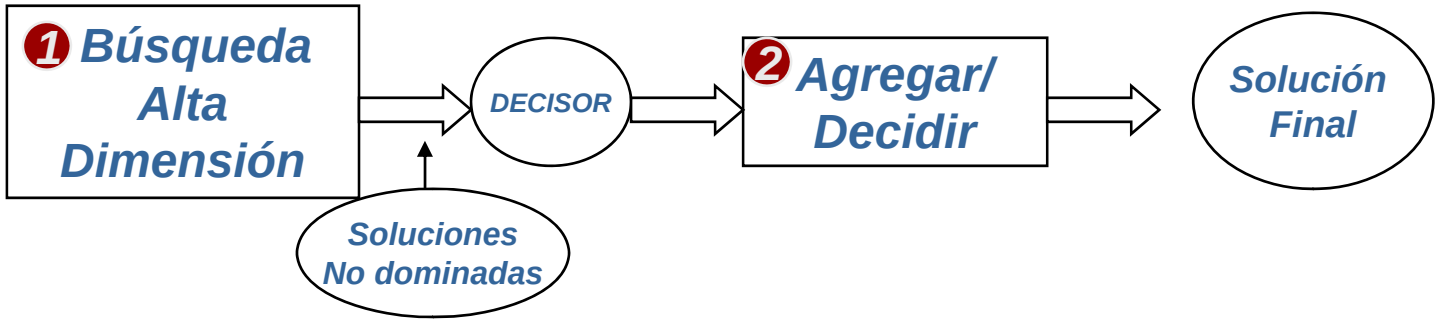
Primero se agregan los objetivos dando lugar a una función de fitness para el AG que se aplica a continuación.

Problemas Multiobjetivo (5)

¿Qué necesitamos para resolver este problema?:

- *Un método de búsqueda basado en los múltiples objetivos,*
- *Una política de equilibrio entre los objetivos,*
- *Un orden para este proceso de optimización.*

Vamos a considerar 2 posibilidades: b) Búsqueda + agregar/decidir



Nota: Se puede considerar una tercera posibilidad híbrida, combinando búsqueda en alta dimensión con búsquedas en dimensiones menores vía agregación parcial de objetivos, como modelos interactivos.

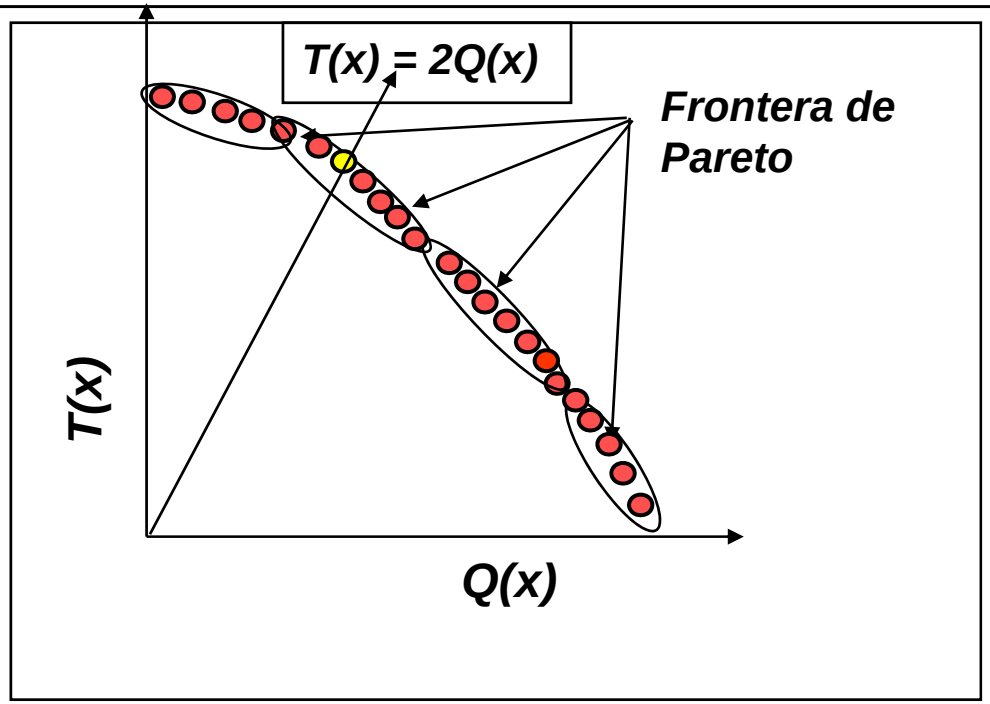
2. Evolución en Problemas Multiobjetivo

- Se evoluciona una población de soluciones al problema.
- Se aplican mecanismos que mantengan diversidad en la población para conseguir un conjunto de soluciones no dominadas lo más grande posible.
- Dos tipos de modelos de acuerdo a las tipologías a) y b):
  - **Modelos evolutivos utilizando pesos para la agregación de los objetivos.**
  - **Modelos evolutivos que generan poblaciones de soluciones no dominadas.**

Modelos Evolutivos utilizando Pesos

- La agregación de los objetivos conduce a la obtención de un único punto de equilibrio en la frontera.
- Ejemplo: *[Max Q(x), Max T(x)]*

Dar a T(x) dos veces la importancia de Q(x), ej: T(x) = 2*Q(x)



La línea T(x) = 2*Q(y) corresponde al vector de pesos W: [1 , 2], cuando se utiliza una función que combina ambos objetivos

F = W * [Q(x), T(X)]
F = [1, 2] * [Q(x), T(X)]
F = Q(x) + 2*T(x)

RESUMEN

Algoritmos Genéticos

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

RESUMEN

Algoritmos Genéticos

FORTRAN Genetic Algorithm (GA) Driver

David L. Carroll

CU Aerospace

60 Hazelwood Drive

Champaign, Illinois 61820

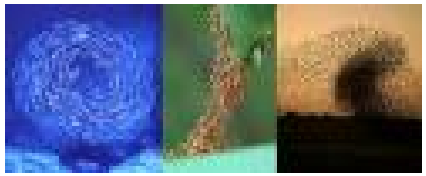
http://cuaerospace.com/carroll/ga.html

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, 1996.

T. Bäck, D.B. Fogel, Z. Michalewicz, Handbook of Evolutionary Computation, Oxford Univ. Press, 1997.



Otras heurísticas bioinspiradas:
Swarm Intelligence

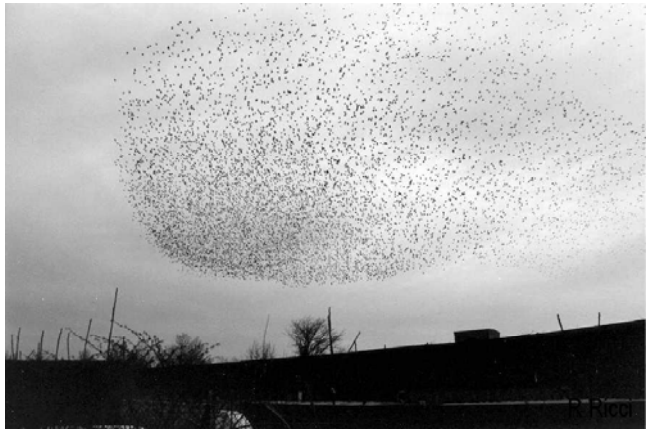
Inteligencia de Enjambre

"La inteligencia colectiva emergente de un grupo de agentes simples"

"The emergent collective intelligence of groups of simple agents"

"Algoritmos o mecanismos distribuidos de resolución de problemas inspirados en el comportamiento colectivo de colonias de insectos sociales u otras sociedades de animales".

(Bonabeau, Dorigo, Theraulaz, 1999)



E. Bonabeau, M. Dorigo, G. Theraulaz
Swarm Intelligence. From Nature to Artificial Systems.
Oxford University Press, 1999.