Seminario Permanente de Formación en Inteligencia Artificial Aplicada a la Defensa

Optimización Inteligente I

Daniel Molina Cabrera

Instituto Andaluz de Investigación en Data Science and Computational Intelligence (DaSCI)

Dpto. Ciencias de la Computación e I.A. Universidad de Granada

dmolina@decsai.ugr.es

http://sci2s.ugr.es

DaSCI

UNIVERSIDAD DE GRANADA

Optimización Inteligente (I)

1. Concepto de Optimización y Metaheurísticas
2. Ejemplos Reales de uso de Metaheurísticas
3. Algoritmos Evolutivos y Genéticos
4. Caso de Estudio (Aplicando el AG para optimizar)

Optimización Inteligente (II)

5 Swarm Intelligence

6 Tipos de Optimización

***7 Frameworks* de Algoritmos Evolutivos**

8 Caso de Estudio (Problemas continuos)

Optimización Inteligente (I)

1 Concepto de Optimización y Metaheurísticas

2 Ejemplos Reales de uso de Metaheurísticas

3 Algoritmos Evolutivos y Genéticos

4 Caso de Estudio (Aplicando el AG para optimizar)

Concepto de Optimización

Optimizar: Buscar la mejor manera de realizar una actividad

- Contexto científico: la optimización es el proceso de tratar de encontrar la **mejor solución posible** para un determinado problema.
- Nivel Profesional:
  - Reducir los costes.
  - Mejorar la Experiencia del cliente.

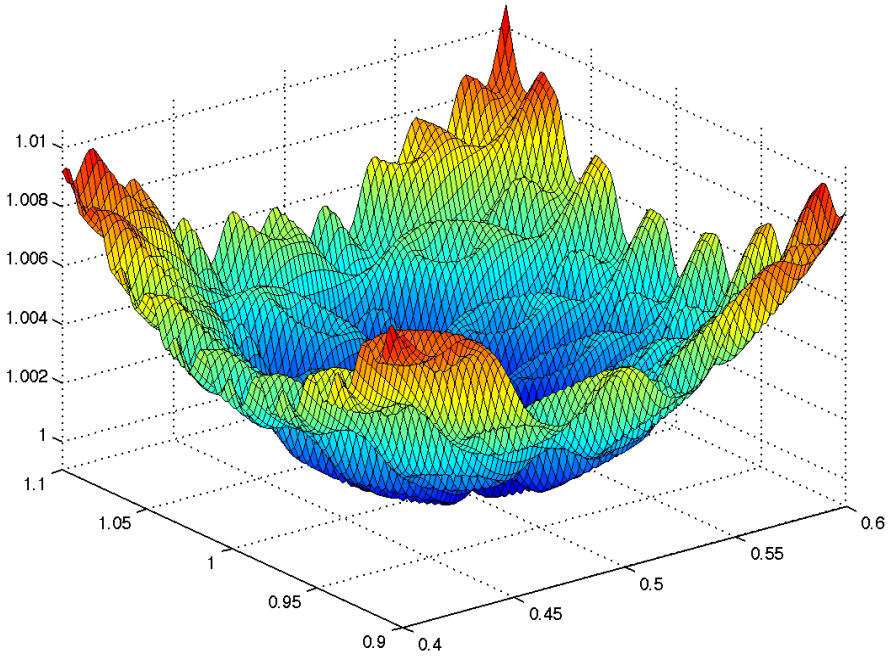
Concepto de Optimización

- Problema de optimización:
  - Diferentes **soluciones**, un criterio para **discriminar** entre ellas y el **objetivo** es encontrar la mejor
  - *Encontrar el valor de unas variables de decisión (sujeto a restricciones) para los que una determinada función objetivo alcanza su valor máximo o mínimo*

Tipos de Problemas

Combinatorios

Continuos



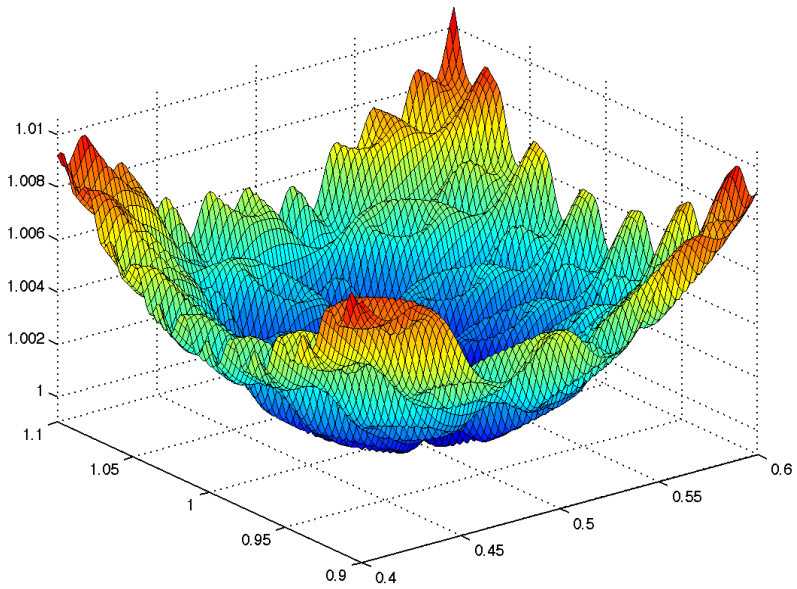
Problemas combinatorios

- Problemas de explosión combinatorios.
- Pocos modelos teóricos.

Optimización continua

- Optimizar un vector de variables reales, cada una con un rango.

$$\text{Global Optima } f(x^*) \leq f(x) \quad \forall x \in \mathit{Domain}$$
$$\text{Real-parameter Optimization } \mathit{Domain} \subseteq \Re^D,$$
$$x^* = [x_1, x_2, \cdots, x_D]$$



Problemas de Técnicas Clásicas de Optimización

1. A menudo no existe una técnica fácil de optimizar un problema.

2. Otras veces si existe pero es demasiado costoso.

3. Con frecuencia se usan simulaciones para poder estudiar la influencia de los parámetros de entrada.

4. No suelen abordar bien múltiples objetivos.

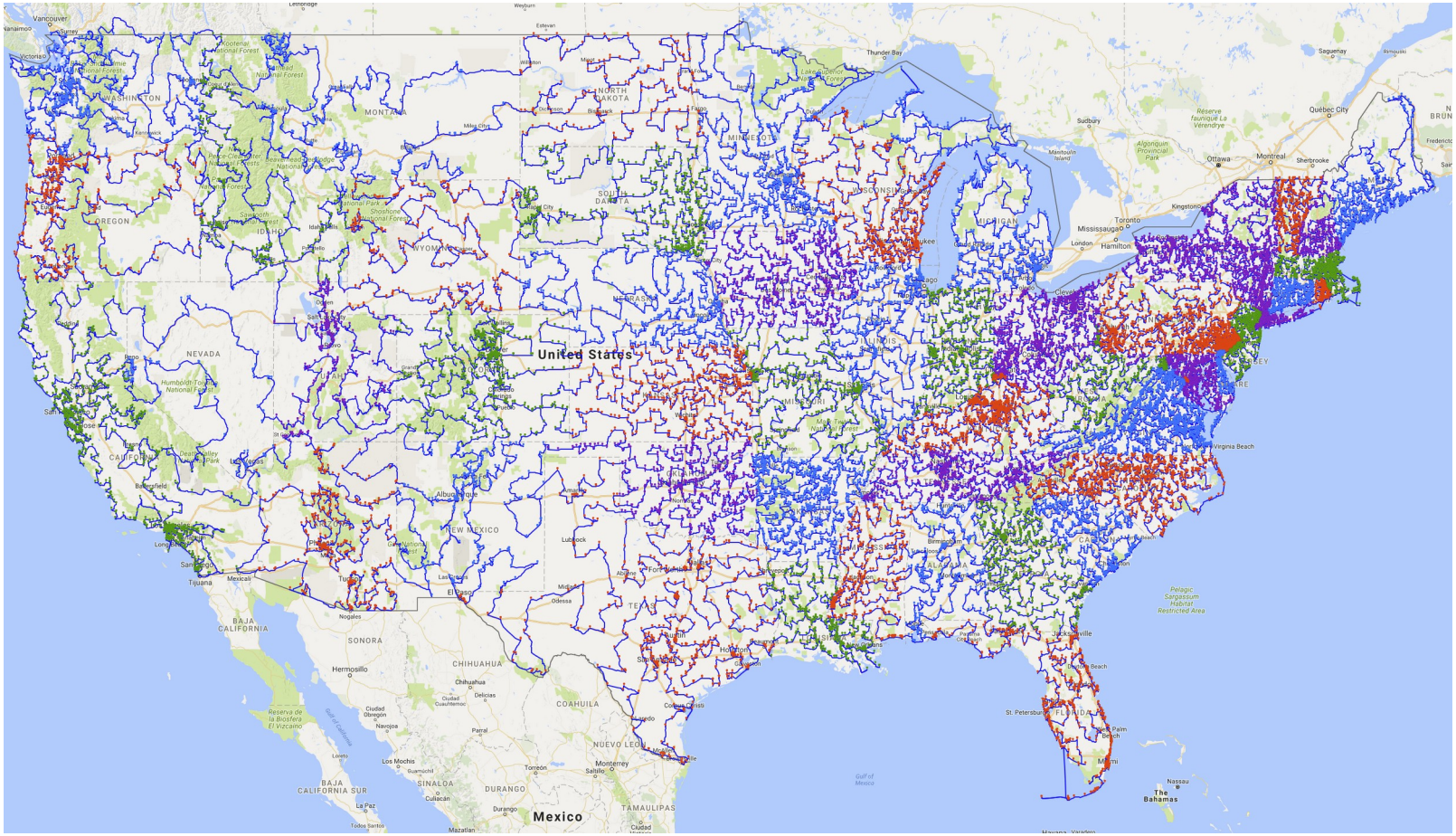
Ejemplo: El problema del viajante de comercio (TSP)

1. Problema del Viajante de Comercio:

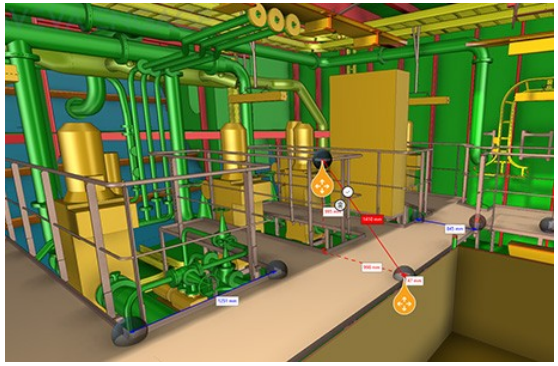
   Travelling Salesman Problem.

2. Problema:
   - Encontrar la ruta más rápida entre N ciudades.
   - Por cada ciudad se pasa una única vez.
   - Se debe de volver a la ciudad origen.

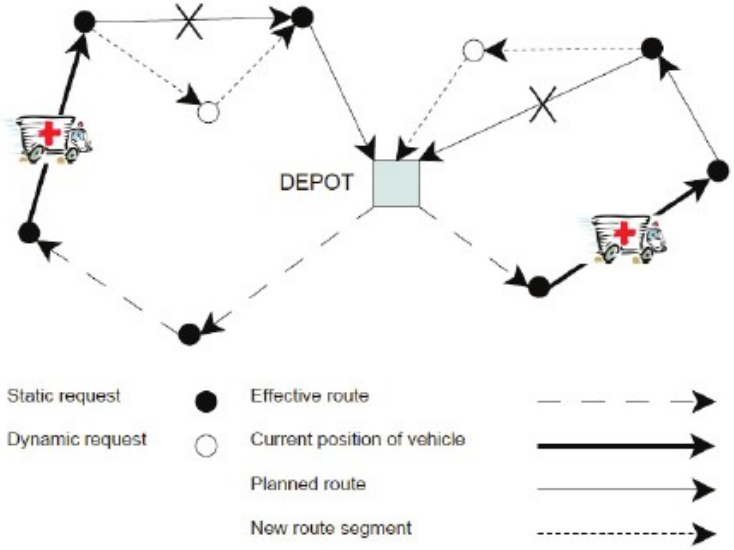
Ejemplo: El problema del viajante de comercio

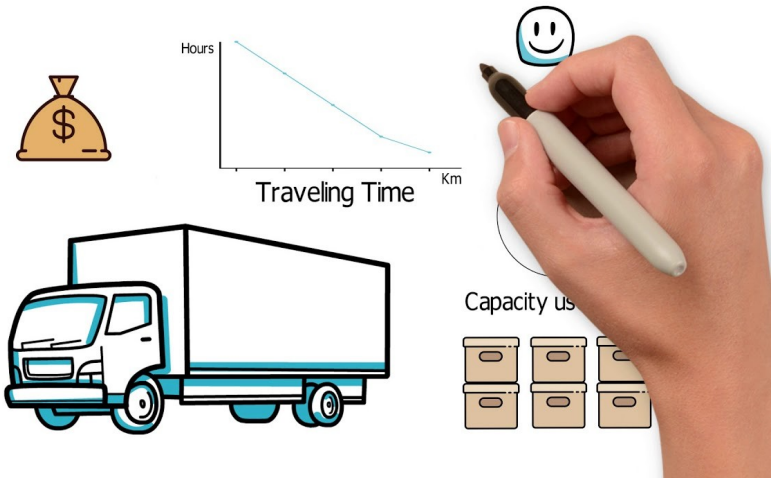


¿Para qué sirve?



- Muchas Aplicaciones
  - Diseño chips: Rutas más cortas.
  - Rutas aéreas: Entre aeropuestos.
  - Reparto almacén: Optimizar entregas.
  - Cableado (eléctrico): Recorrido mínimo.





Tiempos

Ciudades (N)	Fuerza Bruta	Algoritmo Held-Karp
10	2 segundos	0.1 segundo
11	22 segundos	0.2 segundos
14	13 horas	3 segundos
16	200 días	11 segundos
25	270000 años	4 horas
50	$5\ 10^{50}$ años	58000 años

No es abordable con ningún algoritmo tradicional.

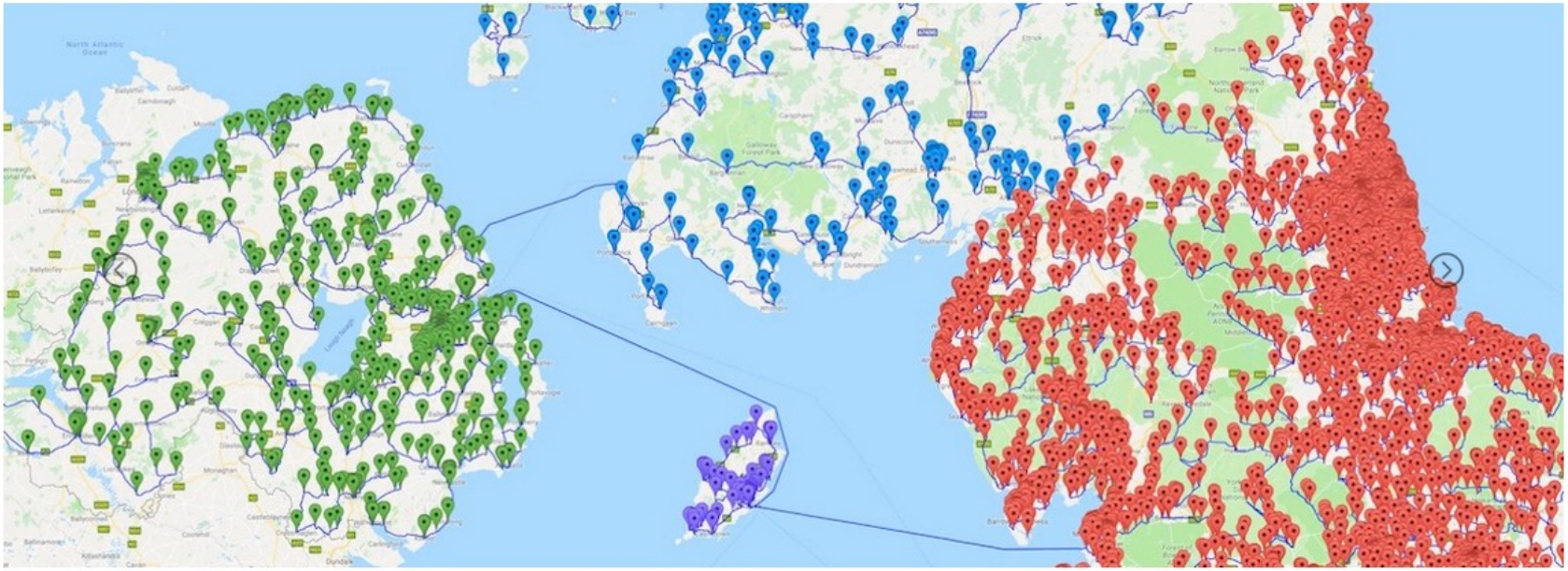
¡Necesitamos buenos algoritmos y eficientes!

Algoritmos que proporcionen una buena solución en un tiempo razonable

¿Es suficiente?

UK49687

Shortest possible tour to nearly every pub in the United Kingdom.



Optimal 49,687-stop pub crawl. Click.

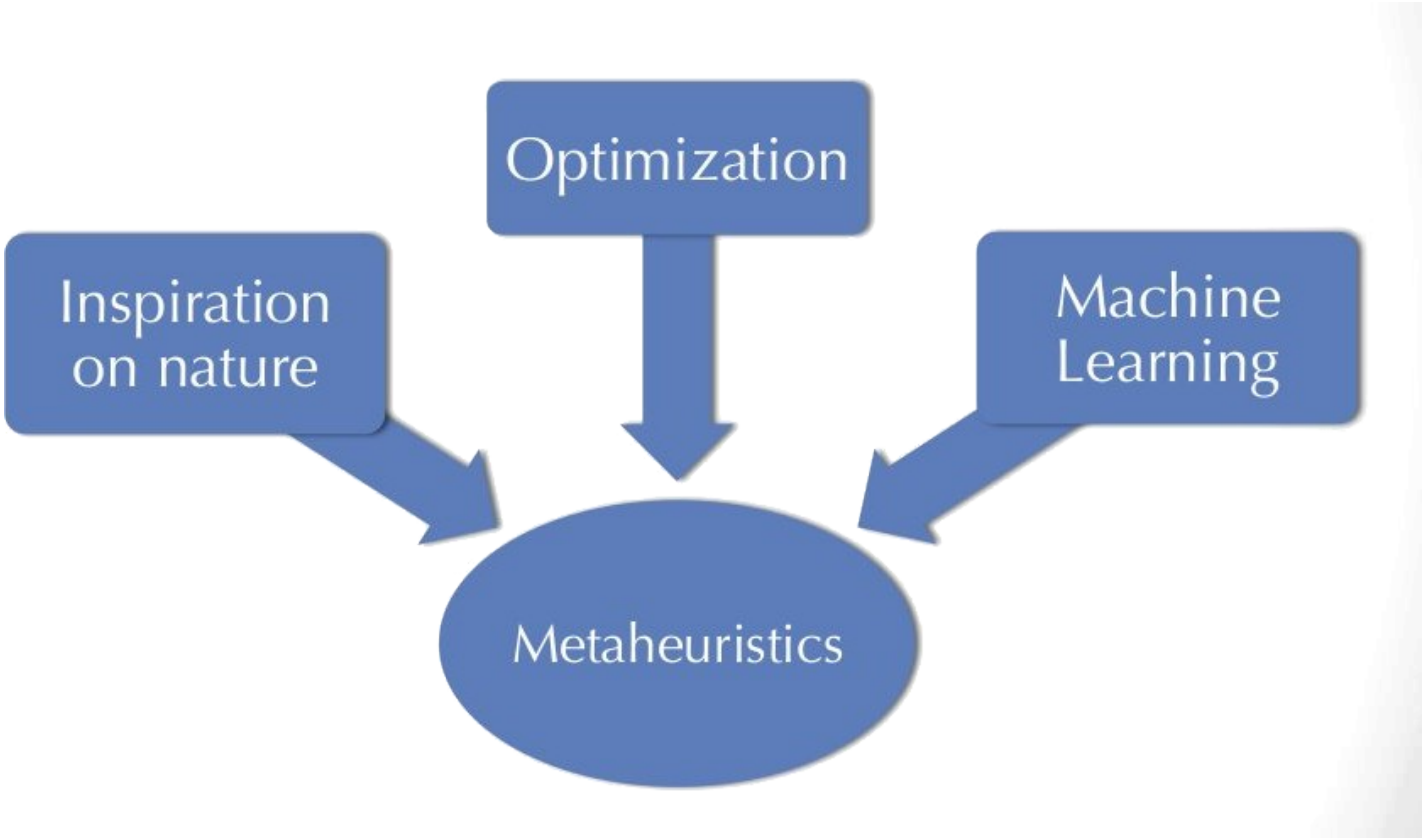
Terminología

- Heurística: Criterio para guiar la búsqueda, no seguro (ej: Distancia euclídea para rutas).
- Metaheurísticas: Algoritmos no exactos, aproximativos que usan heurísticas (como que combinar soluciones buenas puede conducir a mejores).
- Algoritmos Bioinspirados: Algoritmos inspirados en la naturaleza.
- Algoritmos Evolutivos: Algoritmos inspirados en mantener una población de soluciones que evoluciona a mejor.

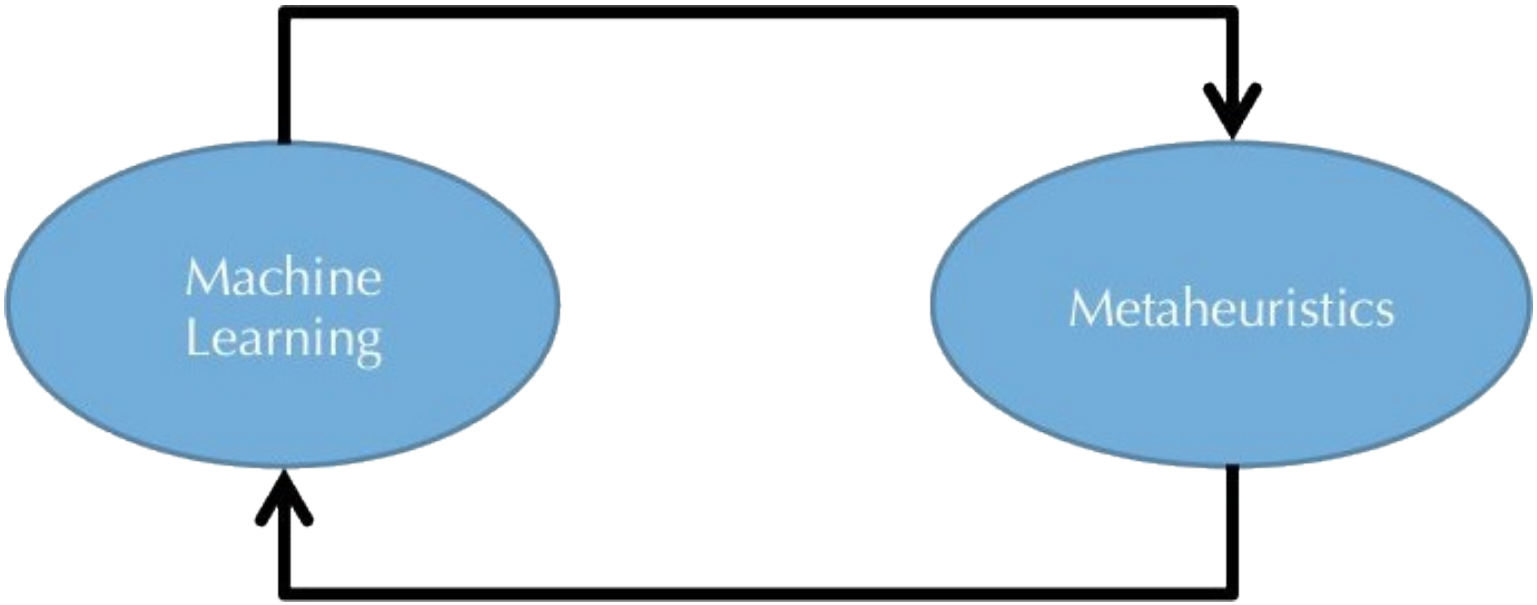
¿Cuándo se pueden aplicar?

- No requiere información del problema.
  - Pero ayuda (Mejora soluciones).
- Son métodos sin información:
  - No conoce la función, ni sus características: derivable, continua, ...
- Para aplicarla sólo es necesario:
  - Poder representar una solución.
  - Poder evaluar cada solución (función *fitness*), para comparar soluciones entre sí.

Metaheurísticas



Metaheurísticas y Aprendizaje



- El uso de metaheurísticas se pueden aplicar a técnicas de aprendizaje automático y viceversa.

Optimización Inteligente (I)

1 Concepto de Optimización y Metaheurísticas

2 Ejemplos Reales de uso de Metaheurísticas

3 Algoritmos Evolutivos y Genéticos

4 Caso de Estudio (Aplicando el AG para optimizar)

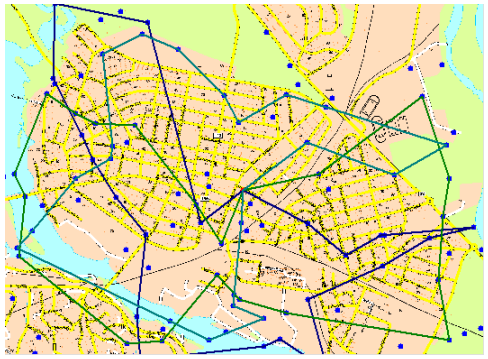
Ejemplos Exitosos

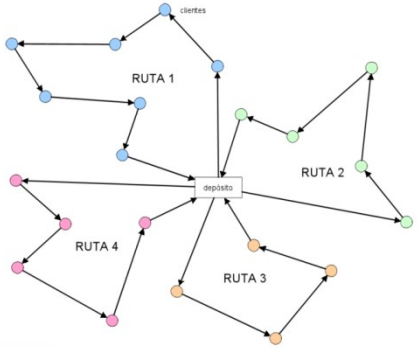
1. Planificación de Rutas para el transporte de mercancías
2. Identificación Forense de Personas Desaparecidas
3. Ayuda al diseño
4. Neuroevolución (Aprendizaje de Juegos)
5. Otros

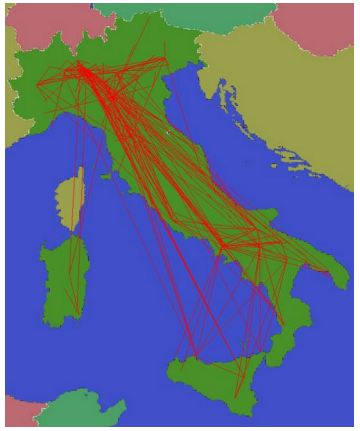
Planificación de Rutas para Transporte de Mercancías

- Hoy en día es difícil encontrar empresas que gestionen las operaciones de logística sin la ayuda del ordenador

- El problema típico es diseñar las rutas más adecuadas de transporte/recogida de productos entre un almacén central y unos destinos dispersos geográficamente







- Su resolución de forma adecuada puede suponer ahorros muy significativos para la empresa

Planificación de Rutas para Transporte de Mercancías

- Esta tarea se lleva a cabo empleando una flota de vehículos pertenecientes o no a la empresa

- Un sistema de planificación de vehículos debe proporcionar un conjunto de rutas de reparto a los conductores

- Las mercancías deben ser entregadas cuándo y donde se requieran, con el mínimo coste posible y verificando todas las restricciones legales y políticas de la empresa

- Los algoritmos de hormigas (AntRoute) son una herramienta muy potente para la planificación de rutas

Planificación de Rutas para Transporte de Mercancías

- AntRoute planifica diariamente las rutas de reparto desde el almacén central de Migros, una gran cadena suiza con 600 supermercados, localizado en Suhr (AG), a toda Suiza

- Migros dispone de una flota de entre 150 y 200 vehículos con tres tamaños: camiones (capacidad de 17 palés), trailers (35 palés) y unidades tractoras (33 palés)

- Esto provoca restricciones de acceso a los almacenes de los supermercados, restricciones de uso de ciertas carreteras, …

- Los repartos tienen de realizarse a horas específicas, todos ellos en un solo día (productos perecederos) y el último tiene que hacerse lo más lejos posible del almacén (servicios extra)

Planificación de Rutas para Transporte de Mercancías

- Por ejemplo, en un reparto de 52000 palés a 6800 clientes en un periodo de 20 días, AntRoute obtuvo el diseño diario de rutas en menos de 5 minutos en un PC estándar

- Los expertos de la empresa necesitaron tres horas...

- Las soluciones de AntRoute fueron de mucha mejor calidad en cuanto al número de rutas necesario, la distancia total recorrida y al aprovechamiento de los vehículos:

	Human Planner	AR-RegTW	AR-Free
Total number of tours	2056	1807	1614
Total km	147271	143983	126258
Average truck loading	76.91%	87.35%	97.81%

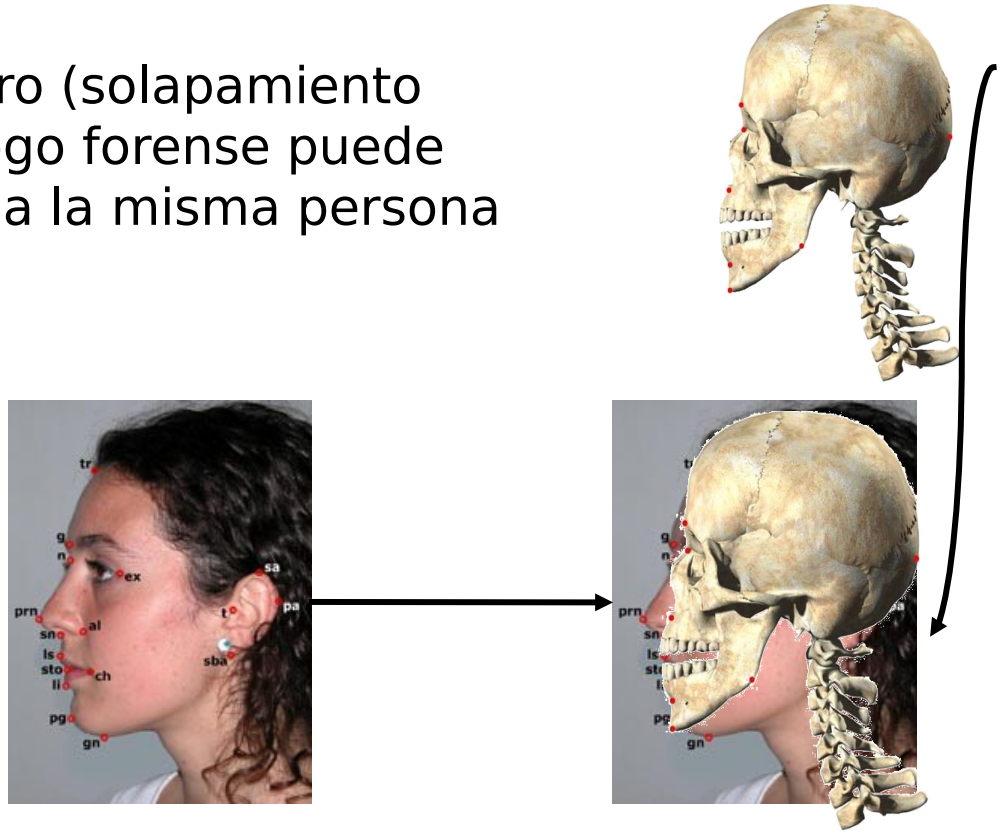
Identificación Forense de Personas Desaparecidas

CSI: crime scene investigation

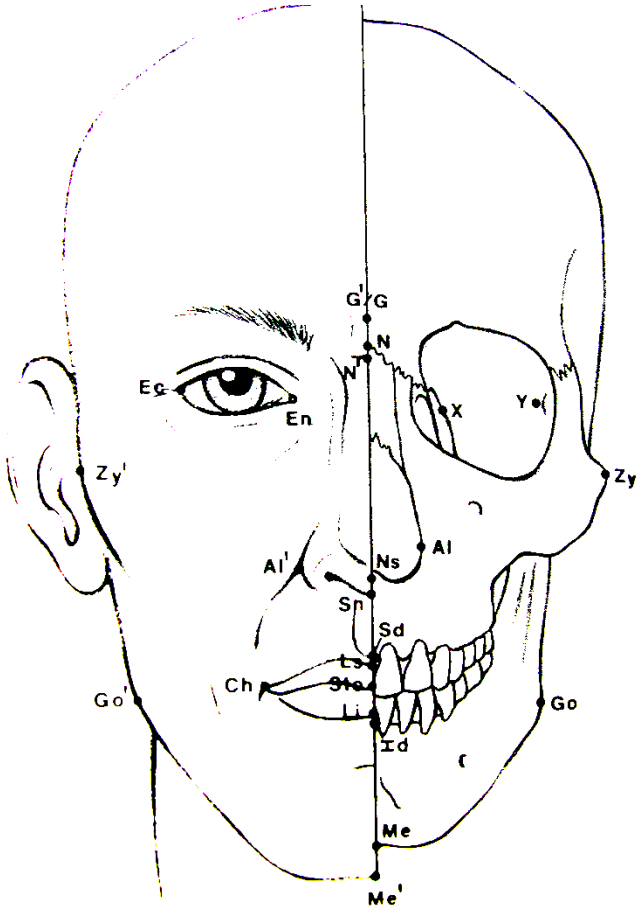
Identificación Forense de Personas Desaparecidas

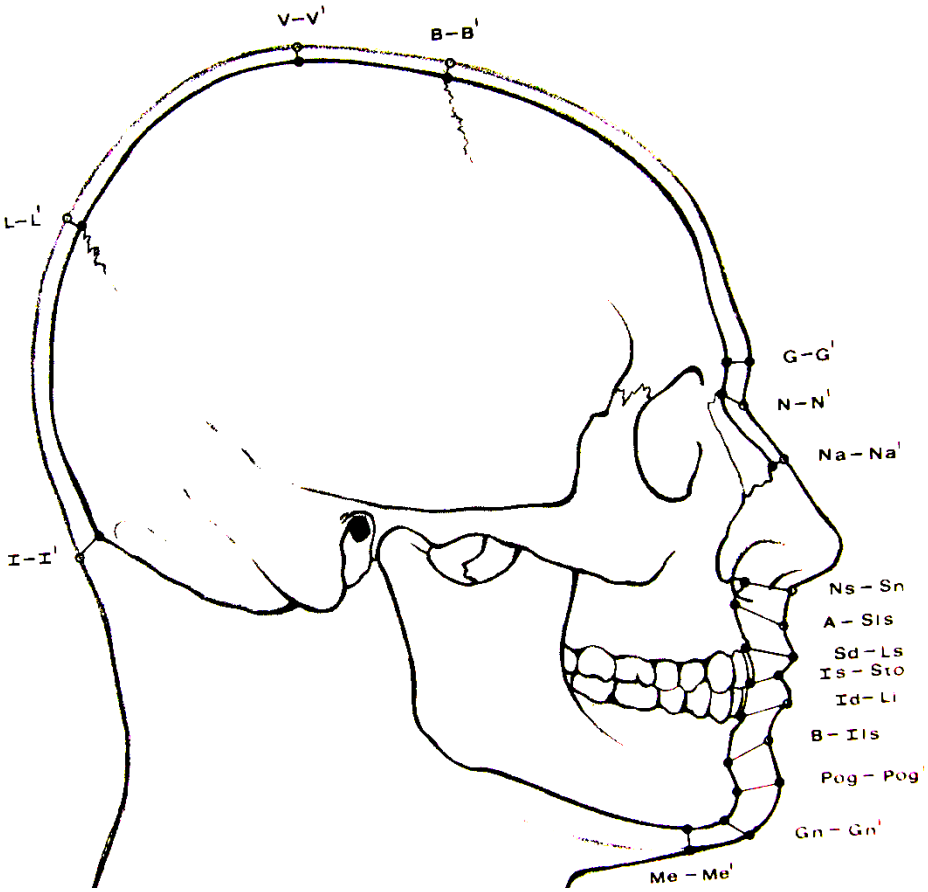
- La superposición craniofacial es una técnica de identificación forense basada en la comparación de un “modelo” del cráneo encontrado y una foto de una persona desaparecida

- Proyectando uno sobre otro (solapamiento cráneo-cara), el antropólogo forense puede determinar si pertenecen a la misma persona



Identificación Forense de Personas Desaparecidas



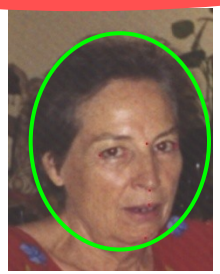


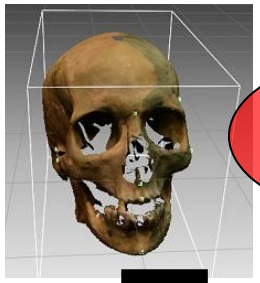
Correlación entre los puntos craneométricos y cefalométricos

Identificación Forense de Personas Desaparecidas

1. Modelado del cráneo y mejora facial

Identificación {Positiva/negativa/ probable positiva/probable negativa/ indeterminada}



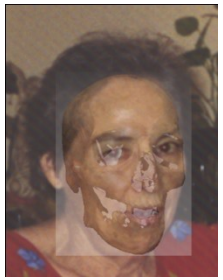


Reconstrucción del modelo 3D

Procesado de imagen y localización de landmarks

2. Solapamiento cráneo-cara

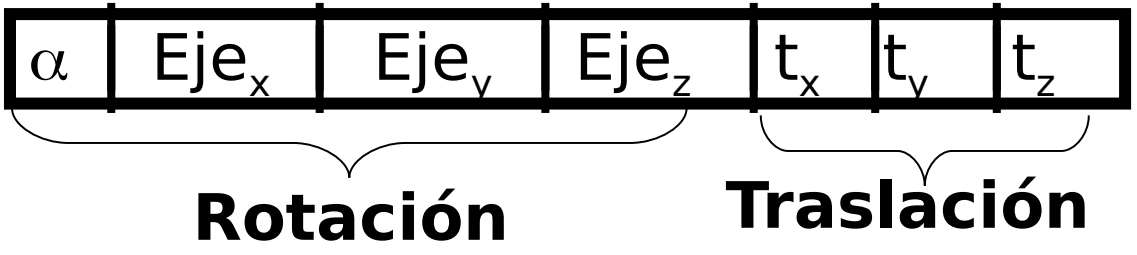
RI 3D-2D: traslación, rotación, escalado, y proyección 2D

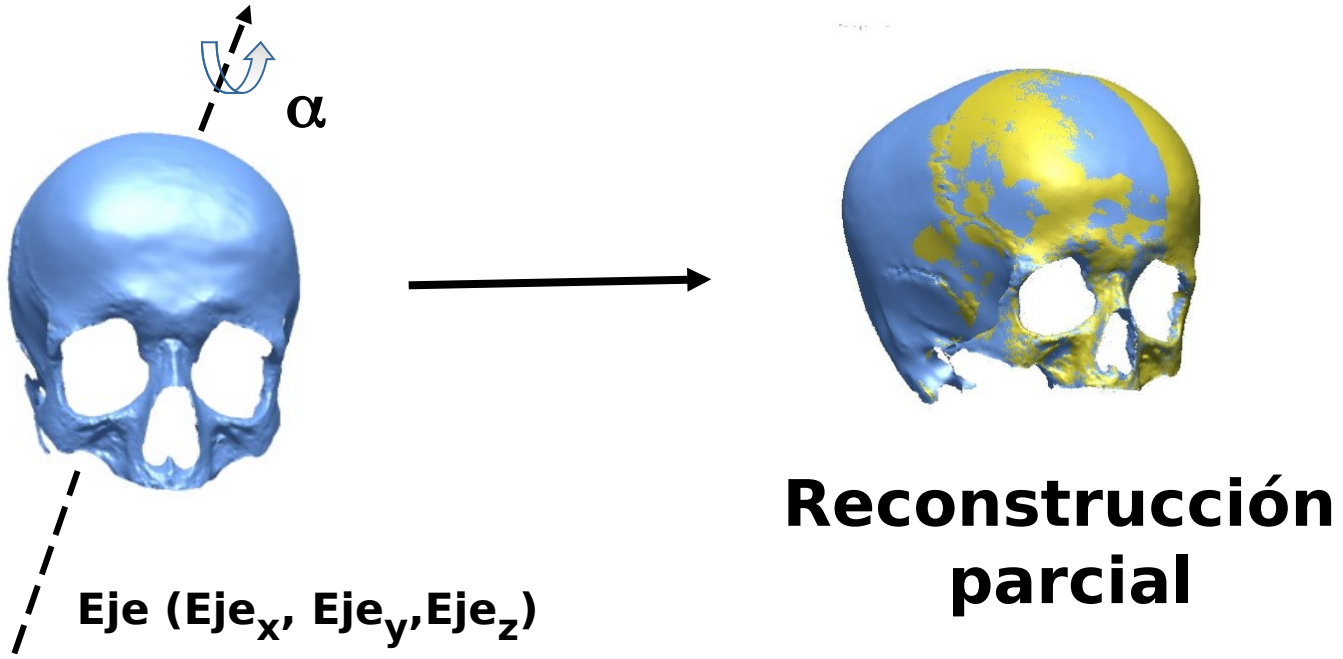


3. Toma de decisiones

Identificación Forense de Personas Desaparecidas

- **Algoritmos Meméticos con codificación real para el modelado 3D de cráneos. Representación de una solución a este problema:**

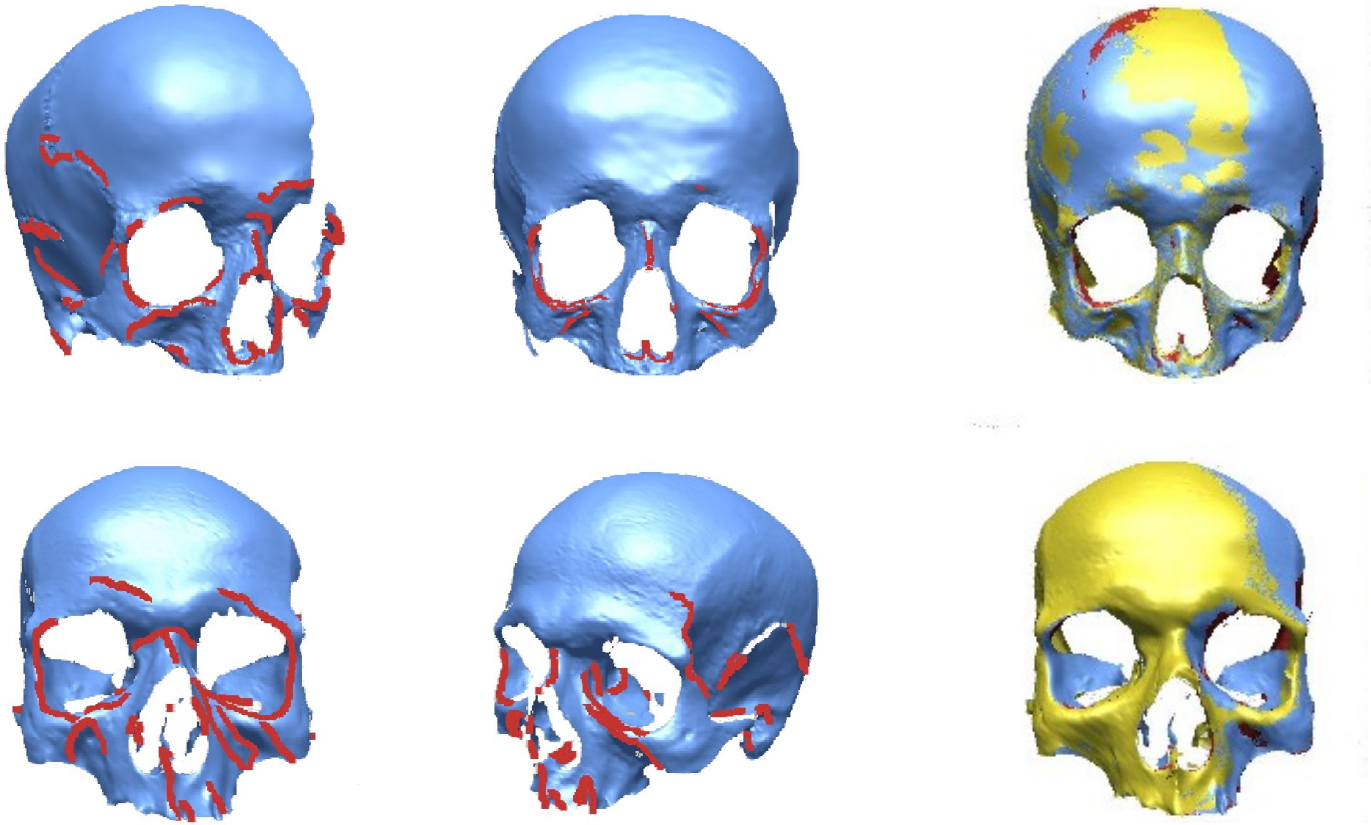




Identificación Forense de Personas Desaparecidas

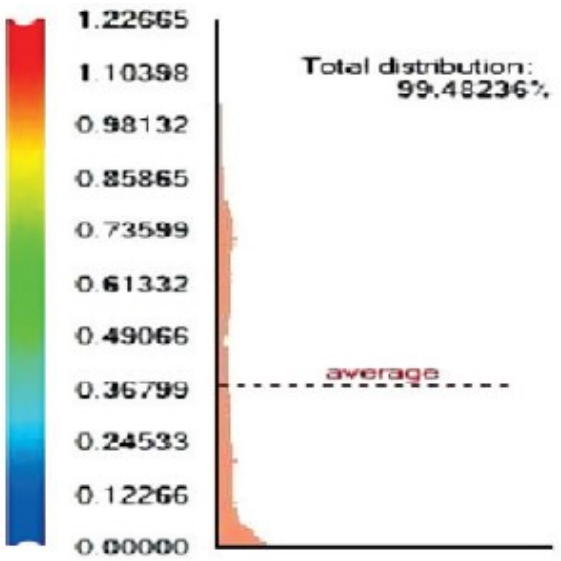
Entrada: vistas 3D

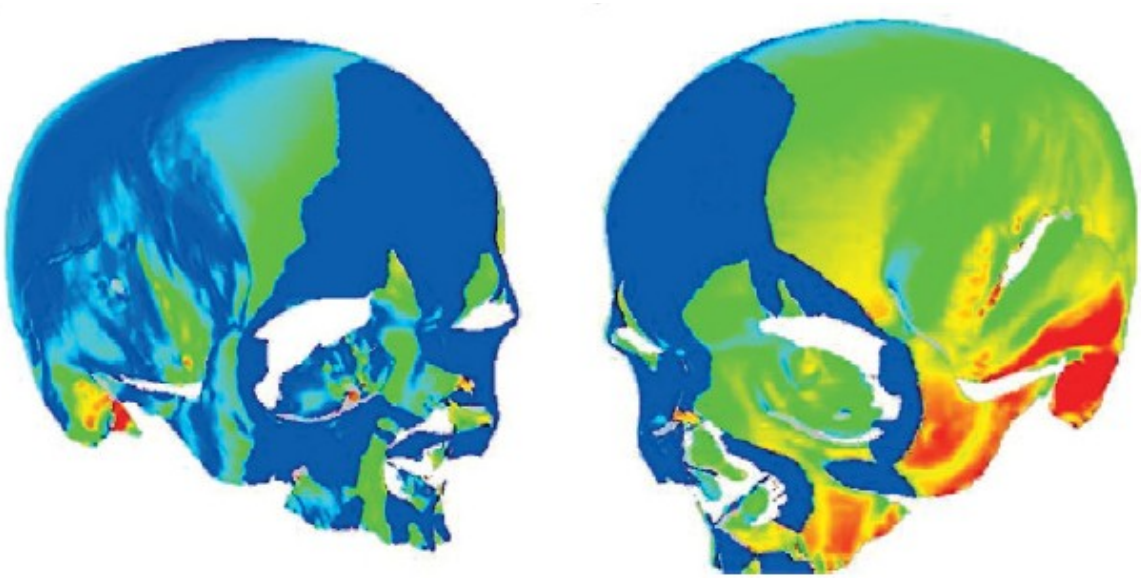
Reconstrucción



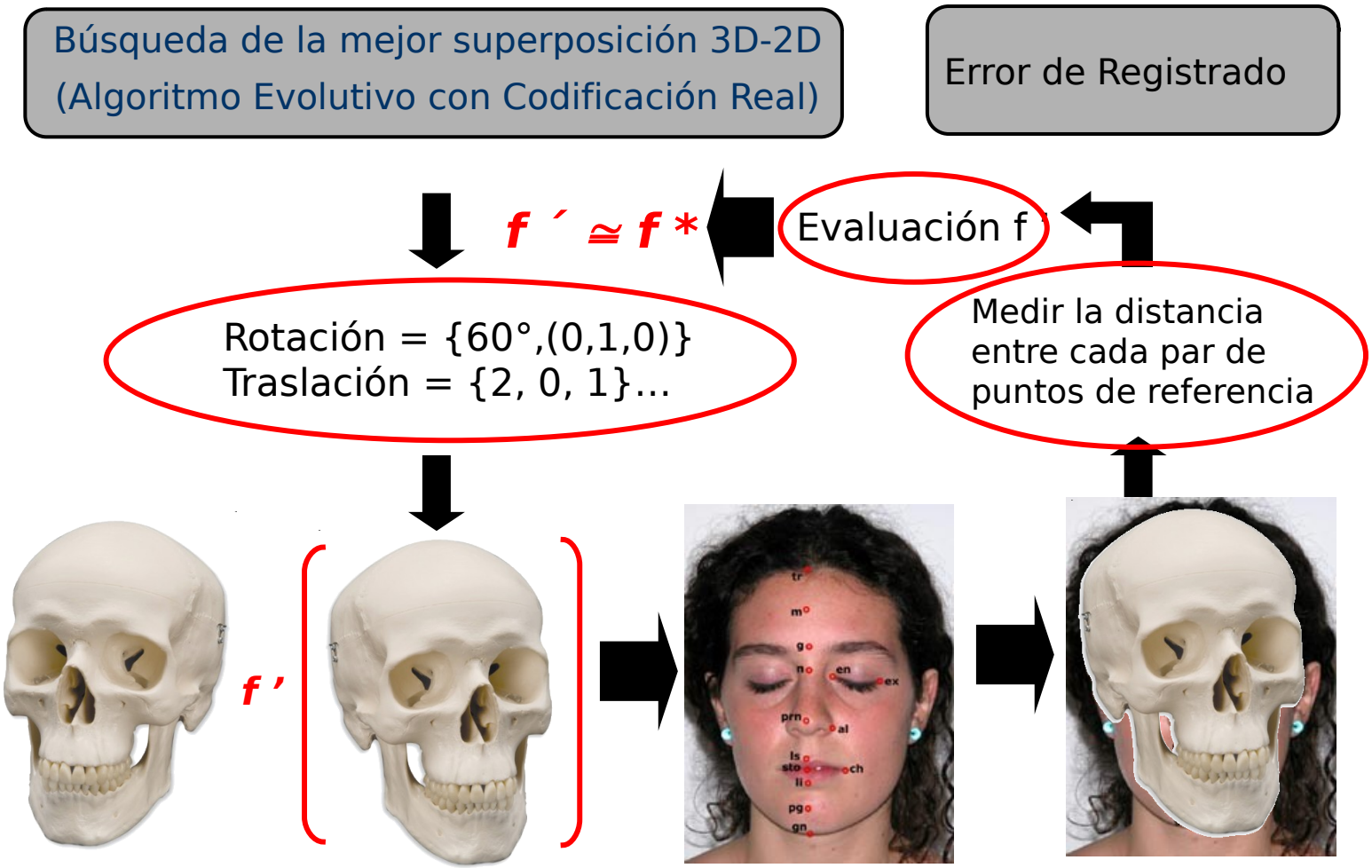
Identificación Forense de Personas Desaparecidas

- **Error global del modelo 3D: menor de 1 milímetro**
- **Tiempo de reconstrucción 3D: 2 minutos**
- **Robustez del método: baja desviación típica en 30 ejecuciones distintas**



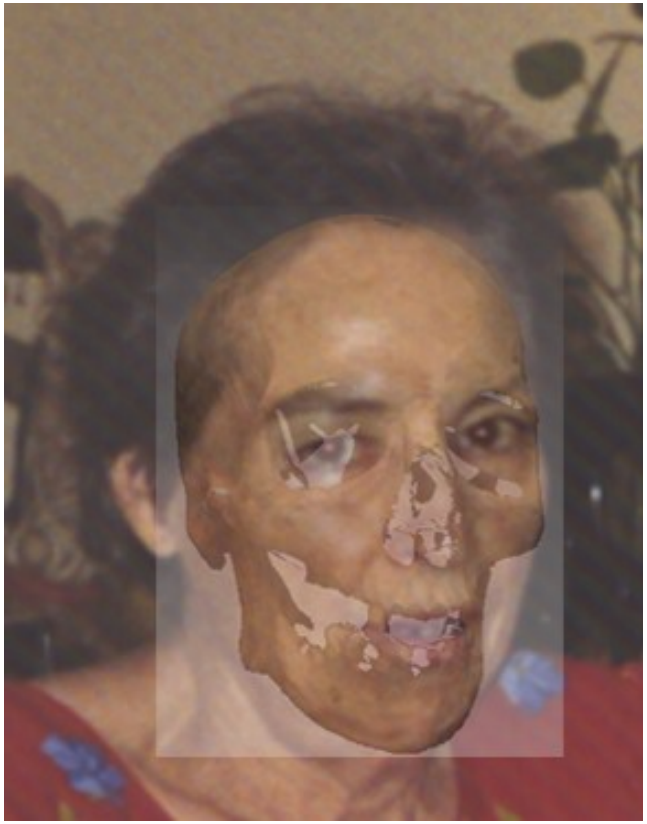


Identificación Forense de Personas Desaparecidas



Identificación Forense de Personas Desaparecidas

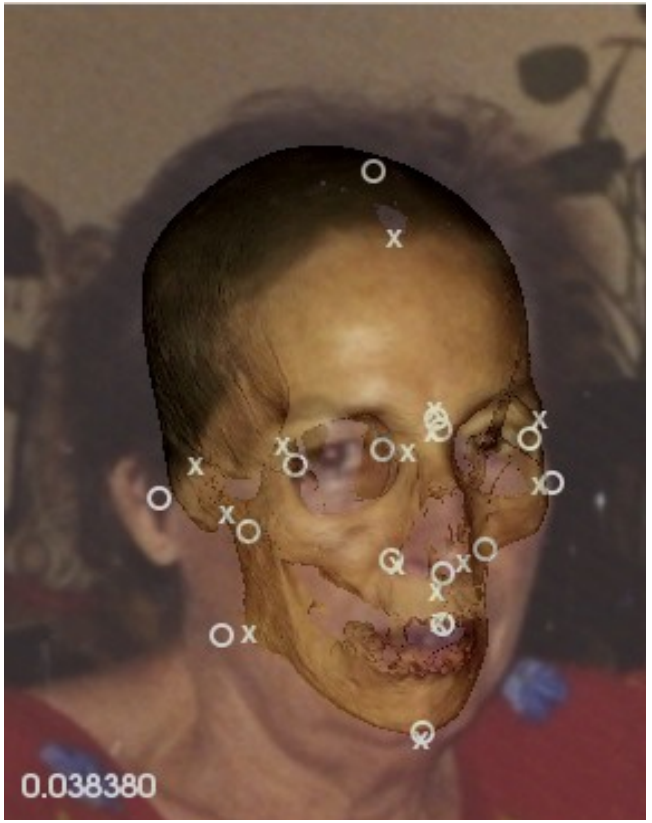
Manual



Area deviation error: 34.70%

varias horas

Fuzzy AE



Area deviation error: 13.23%

2-4 minutos

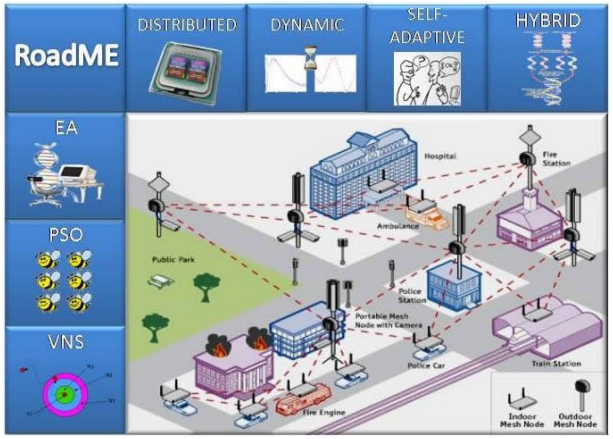
Resolver problemas de Diseño

- Líneas de Montaje
- Colocación y temporización de señales de tráfico
- Diseño industrial (canalizaciones)

Líneas de Montaje en Nissan

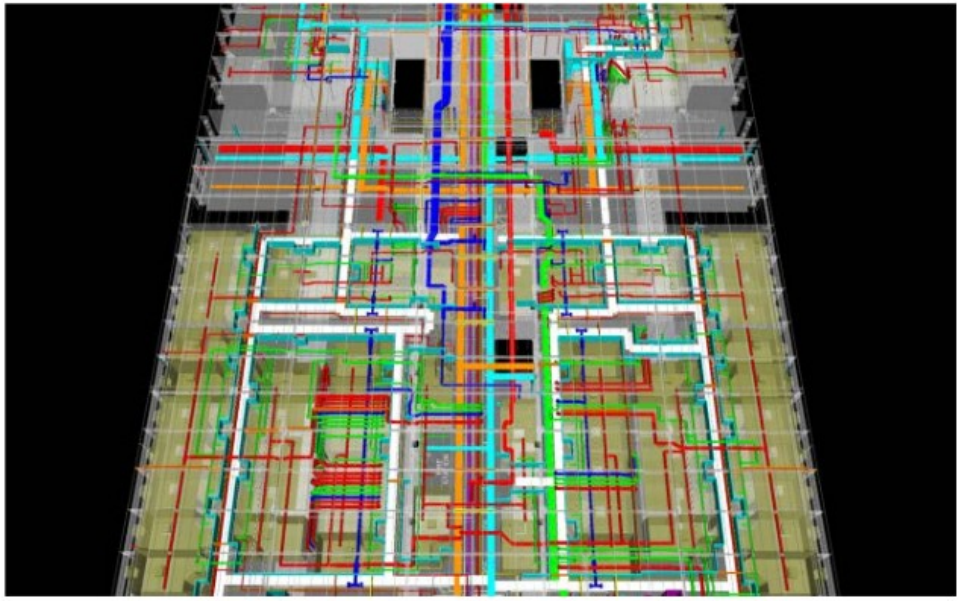
- La mayoría de los sistemas productivos actuales se basan en líneas de montaje
- La producción de un ítem se divide en un conjunto de tareas que tienen que llevarse a cabo según un orden concreto y respetando una serie de precedencias
- Cada tarea necesita un tiempo dado (más un área de trabajo) y tiene asociada un conjunto de predecesores directos
- El diseño (equilibrado) de la línea requiere agrupar de forma eficiente las tareas necesarias en estaciones de trabajo para maximizar la producción y reducir tiempos muertos

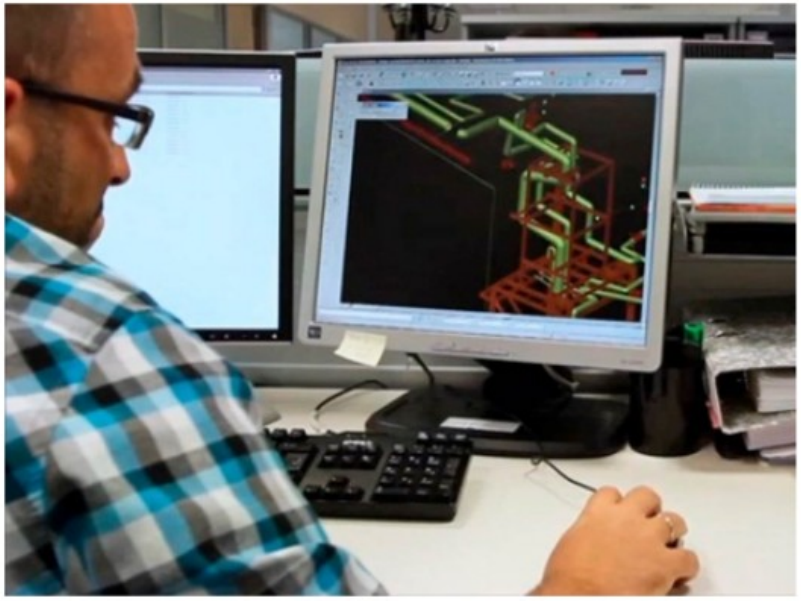
Problema real: Tráfico



http://roadme.lcc.uma.es/index.html

Ayuda al Diseño

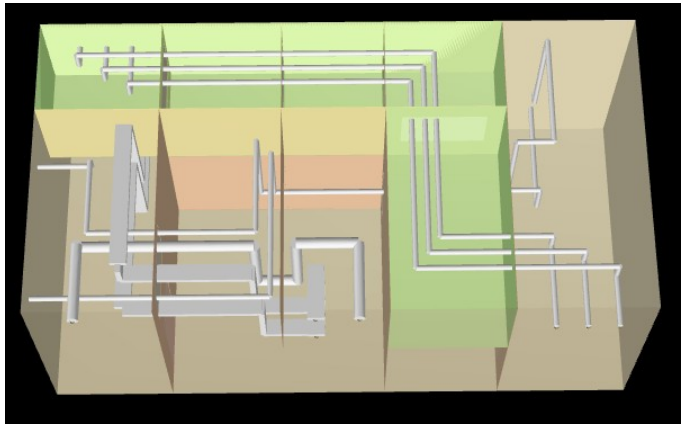




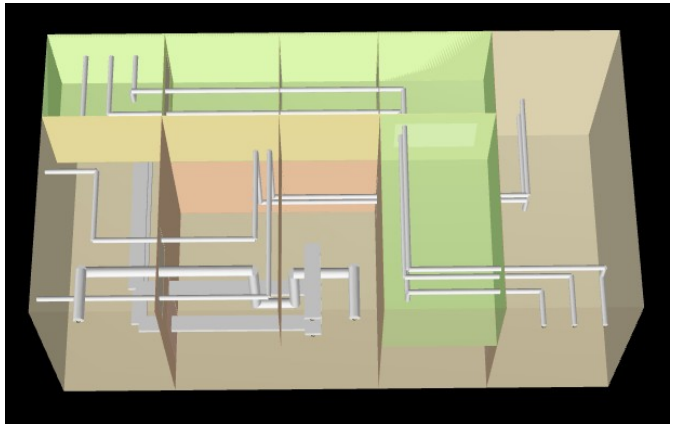
- Tareas de diseño que antes hacía un diseñador
- El humano puede revisar y adaptar

Ayuda al Diseño

Peor Solución



Mejor Solución



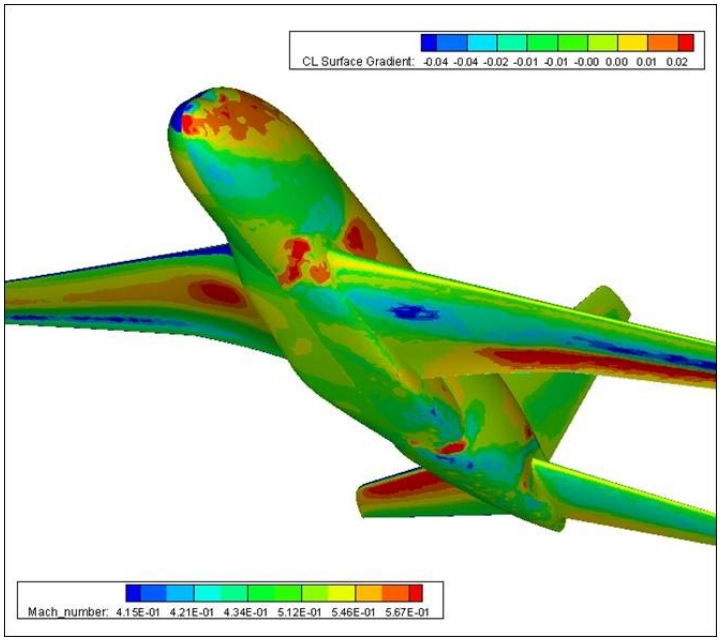
- Sistema Automático mediante un proceso automático.
- Dependía de distintos parámetros.
  - Reales: Problema Real.
  - Discretos: Problema Combinatorio.
- Se optimizaron dichos parámetros mediante algoritmos evolutivos.

Diseño Aeronáutico

- No se trata sólo de una cuestión estética, sino de aerodinámica, con objeto de optimizar el rendimiento de los vehículos o los deportistas
- Hoy en día, casi todo el proceso de optimización se realiza mediante el empleo del simulaciones
- La optimización es encontrar la aeronave aerodinámicamente más eficiente, dentro de unas determinadas restricciones estructurales

Diseño Aeronáutico

- Se simulan el comportamiento del avión objeto sólido (el avión) en interacción con un fluido (el aire), según la Dinámica Computacional de Fluidos.



- Usan métodos de optimización para obtener la forma óptima del avión que minimiza la resistencia bajo los requisitos geométricos y físicos
- Los diseños prometedores mediante la simulación computacional son validados en el túnel de viento

Diseño Aeronáutico

- El problema de los algoritmos de optimización numérica clásicos es su lentitud
- En casos reales, cada ejecución puede requerir meses de cómputo en máquinas de alto rendimiento al tener que realizar una gran cantidad de simulaciones
- Por ello, se han empleado los algoritmos evolutivos para esta tarea, que son capaces de proporcionar diseños de buena calidad en un tiempo mucho más reducido
- Además, como optimizadores multiobjetivo, pueden optimizar varios criterios a la vez (velocidad, estabilidad o gasto de combustible, por ejemplo)

Optimización Inteligente (I)

1 Concepto de Optimización y Metaheurísticas

2 Ejemplos Reales de uso de Metaheurísticas

3 Algoritmos Evolutivos y Genéticos

4 Caso de Estudio (Aplicando el AG para optimizar)

¿Cómo se pueden resolver?

- ¿Cómo lo hace la naturaleza?
  - Búsqueda de comida (hormigas, abejas, ...).

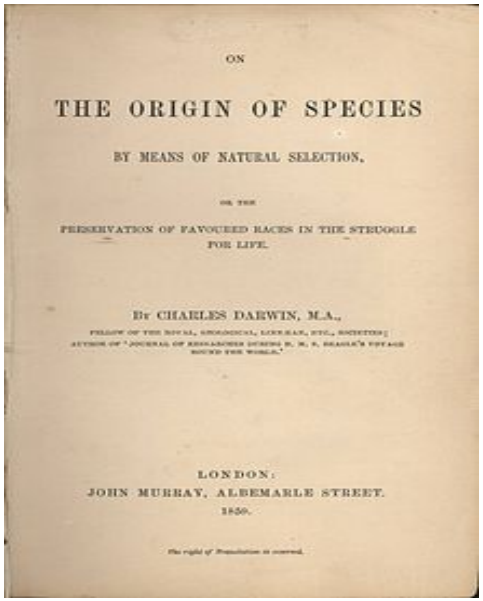


¿Cómo se pueden resolver?

- ¿Cómo lo hace la naturaleza?
  - Búsqueda de comida (hormigas, abejas, …).
  - Movimiento de bandadas (aves, peces, …).

¿Cómo se pueden resolver?

- ¿Cómo lo hace la naturaleza?
  - Búsqueda de comida (hormigas, abejas, …).
  - Movimiento de bandadas (aves, peces, …).
  - Adaptación de especies.


ON

THE ORIGIN OF SPECIES

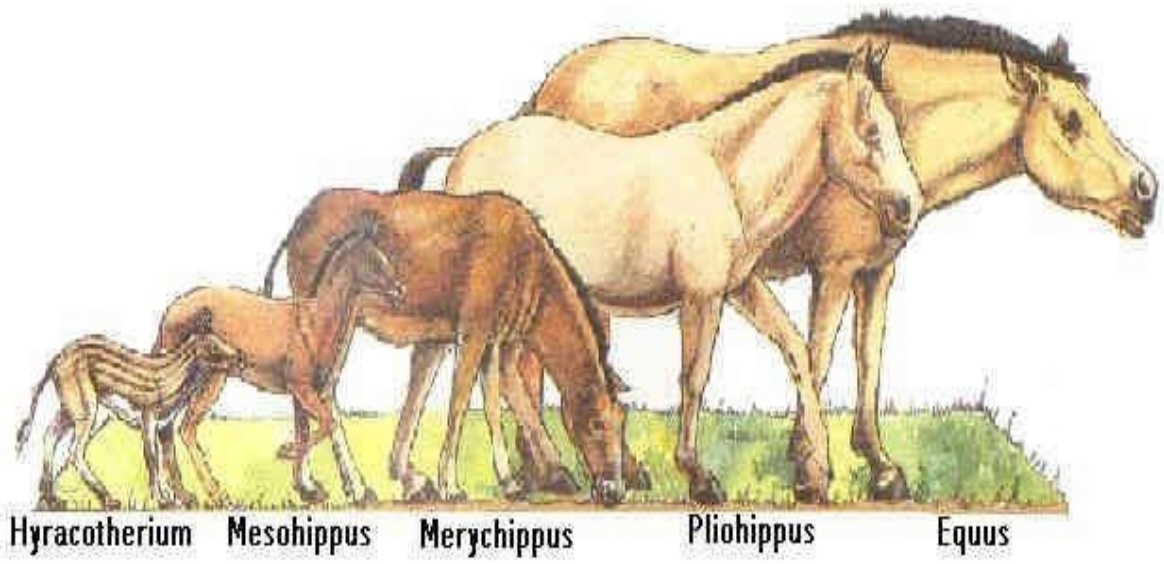
BY MEANS OF NATURAL SELECTION,

OR THE

PRESERVATION OF FAVOURED RACES IN THE STRUGGLE FOR LIFE.

By CHARLES DARWIN, M.A.,

LONDON:
JOHN MURRAY, ALBEMARLE STREET.
1859.

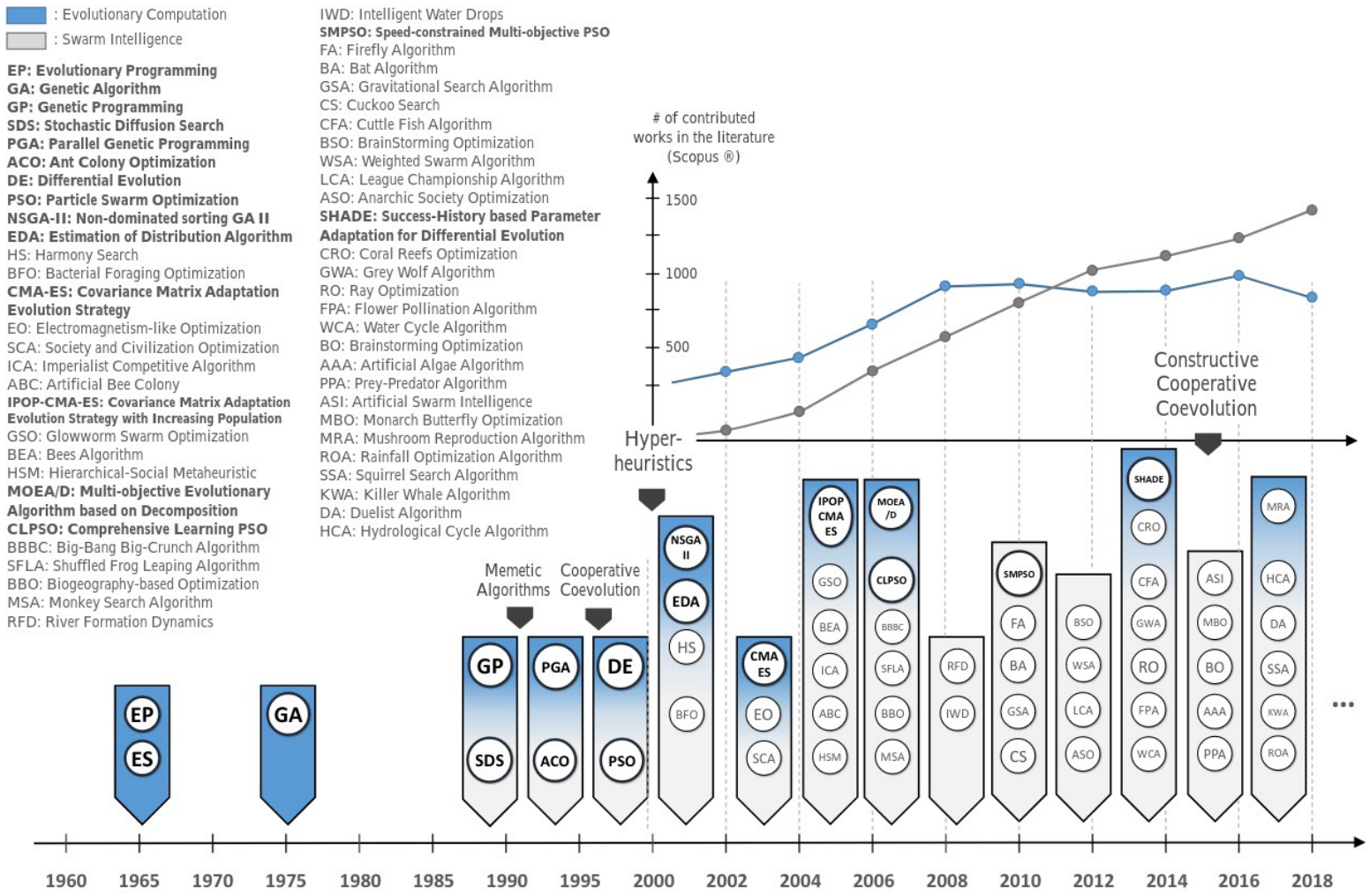


Algoritmos Bioinspirados

- Se basan en analogías con sistemas naturales como los anteriores.
- Los algoritmos bioinspirados simulan el comportamiento de sistemas naturales para el diseño de métodos heurísticos no determinísticos de *búsqueda*…

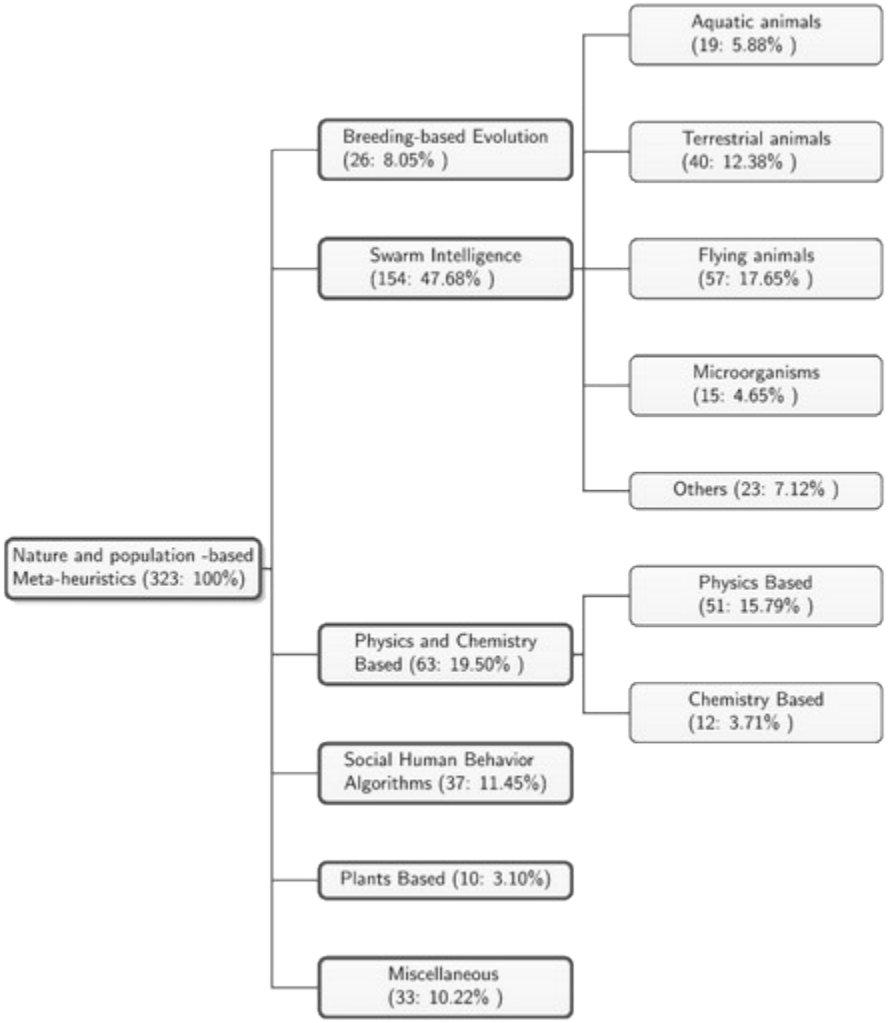
- Características:
  - Son no determinísticos
  - Presentan a menudo una estructura paralela.
  - Son adaptativos (utilizan realimentación con el entorno para modificar el modelo y los parámetros).

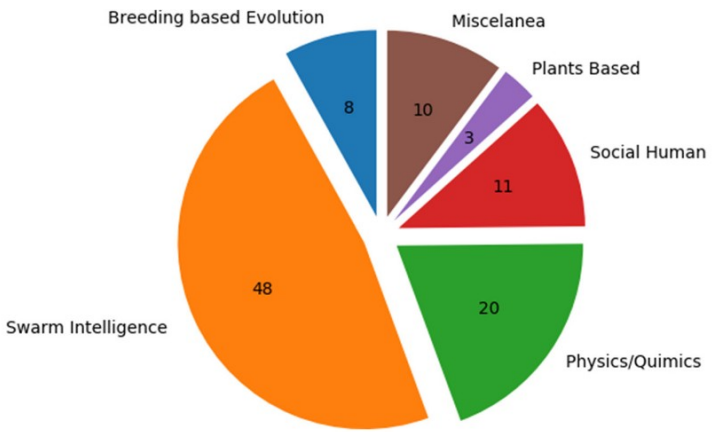
Algoritmos Bioinspirados



J. Del Ser, E. Osaba, D. Molina, Xin-She Yang, S. Salcedo-Sanz, D. Camacho, S. Das, P. N.Suganthan, C. A.Coello, F. Herrera. Bio-inspired computation: Where we stand and what's next. Swarm and Evolutionary Computation 48 (2019) 220-250. doi: 10.1016/j.swevo.2019.04.008

Algoritmos Bioinspirados

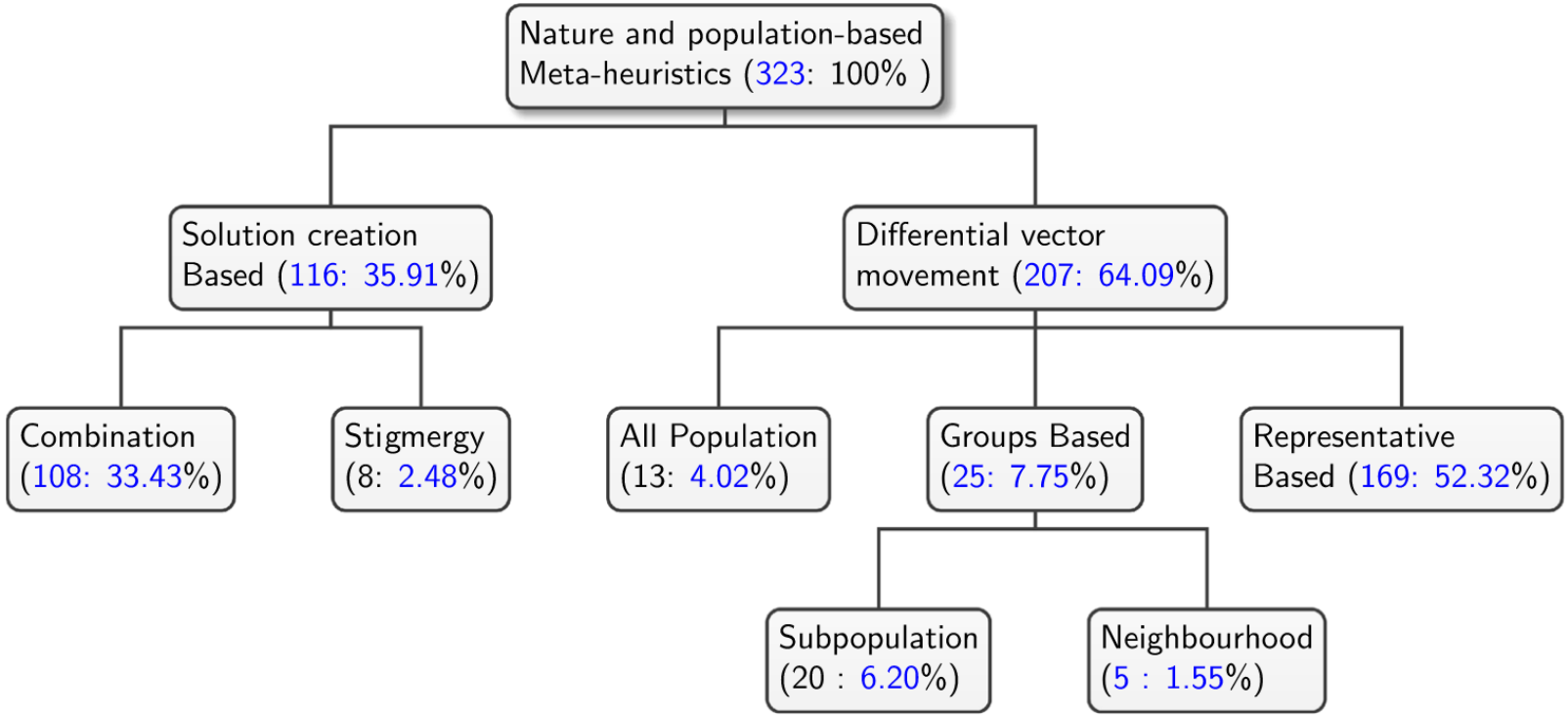




- Muchos algoritmos.
- Muchas Metáforas, pocas ideas.

D. Molina, J. Poyatos, J.D. Del Ser, S. García, A. Hussain, F. Herrera. Comprehensive Taxonomies of Nature-and Bio-inspired Optimization: Inspiration Versus Algorithmic Behavior, Critical Analysis Recommendations. Cognitive Computation 12:5 (2020) 897-939. doi: 10.1007/s12559-020-09730-8

Algoritmos Bioinspirados



- Muchos algoritmos muy parecidos.
- Lo importante son las nuevas ideas, no las metáforas.

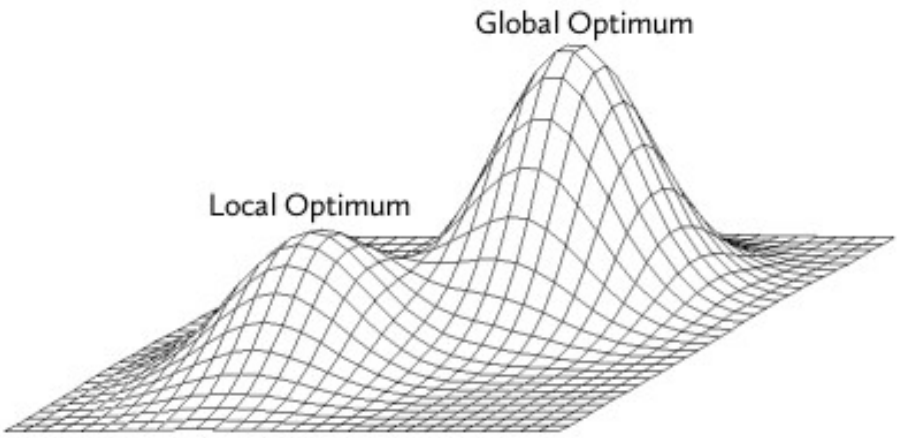
D. Molina, J. Poyatos, J.D. Del Ser, S. García, A. Hussain, F. Herrera. Comprehensive Taxonomies of Nature-and Bio-inspired Optimization: Inspiration Versus Algorithmic Behavior, Critical Analysis Recommendations. Cognitive Computation 12:5 (2020) 897-939. doi: 10.1007/s12559-020-09730-8

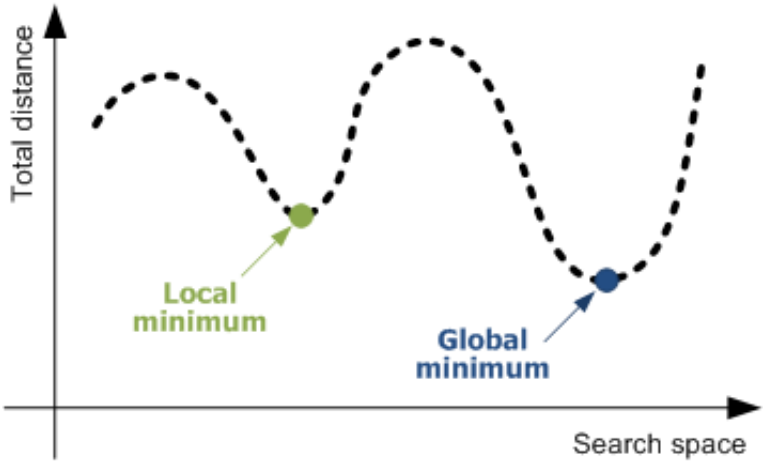
¿Qué suelen tener en común?

- Exploran usando varias soluciones.
- La población de soluciones evoluciona:
  - Creando nuevas combinando existentes.
  - Transformando existentes.
- Devuelve la mejor encontrada
- A más tiempo/soluciones generadas:
  - Más posibilidad de encontrar buena solución.
  - Más tiempo tarda.

Problema: óptimo local

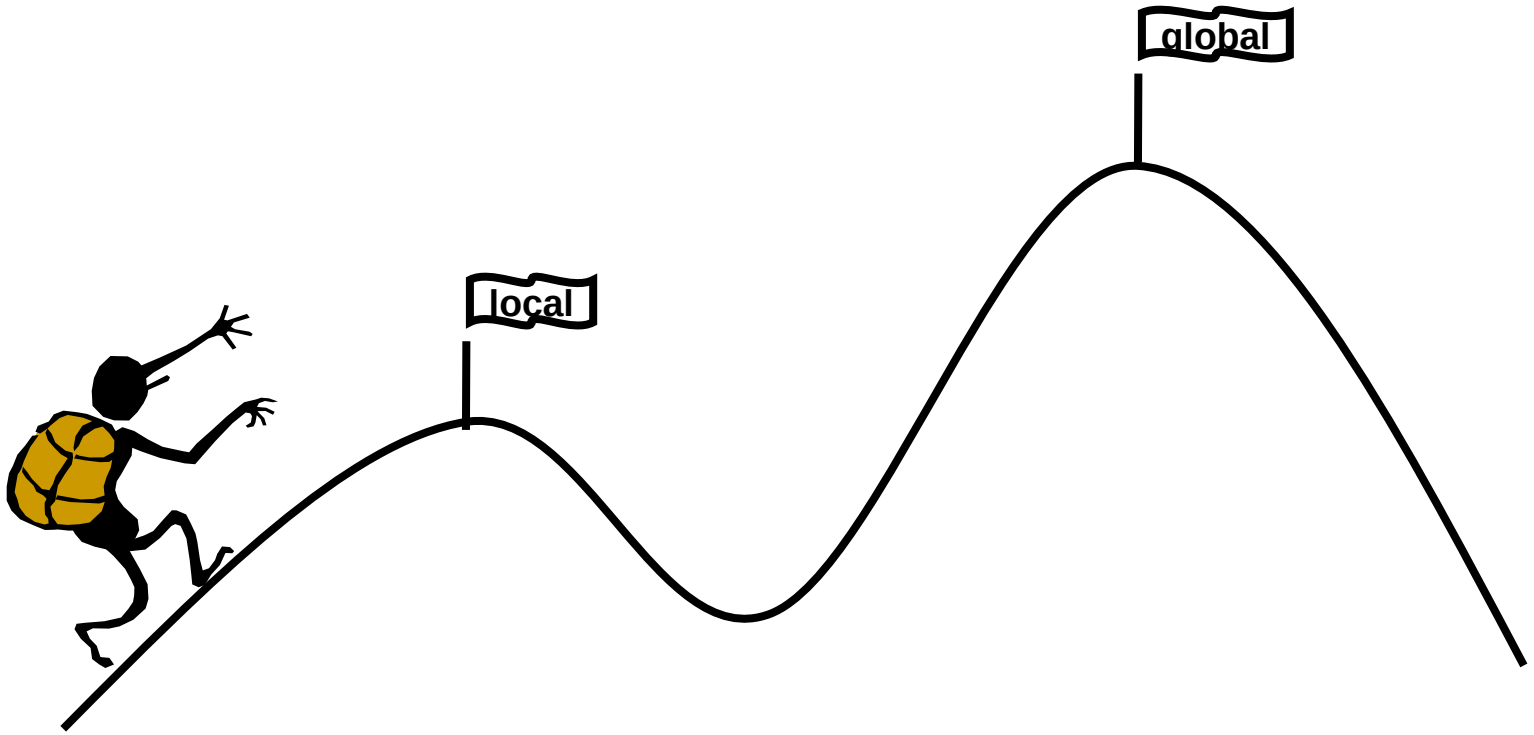
- Tomar una acción que mejora a corto plazo pero inadecuada a largo plazo.





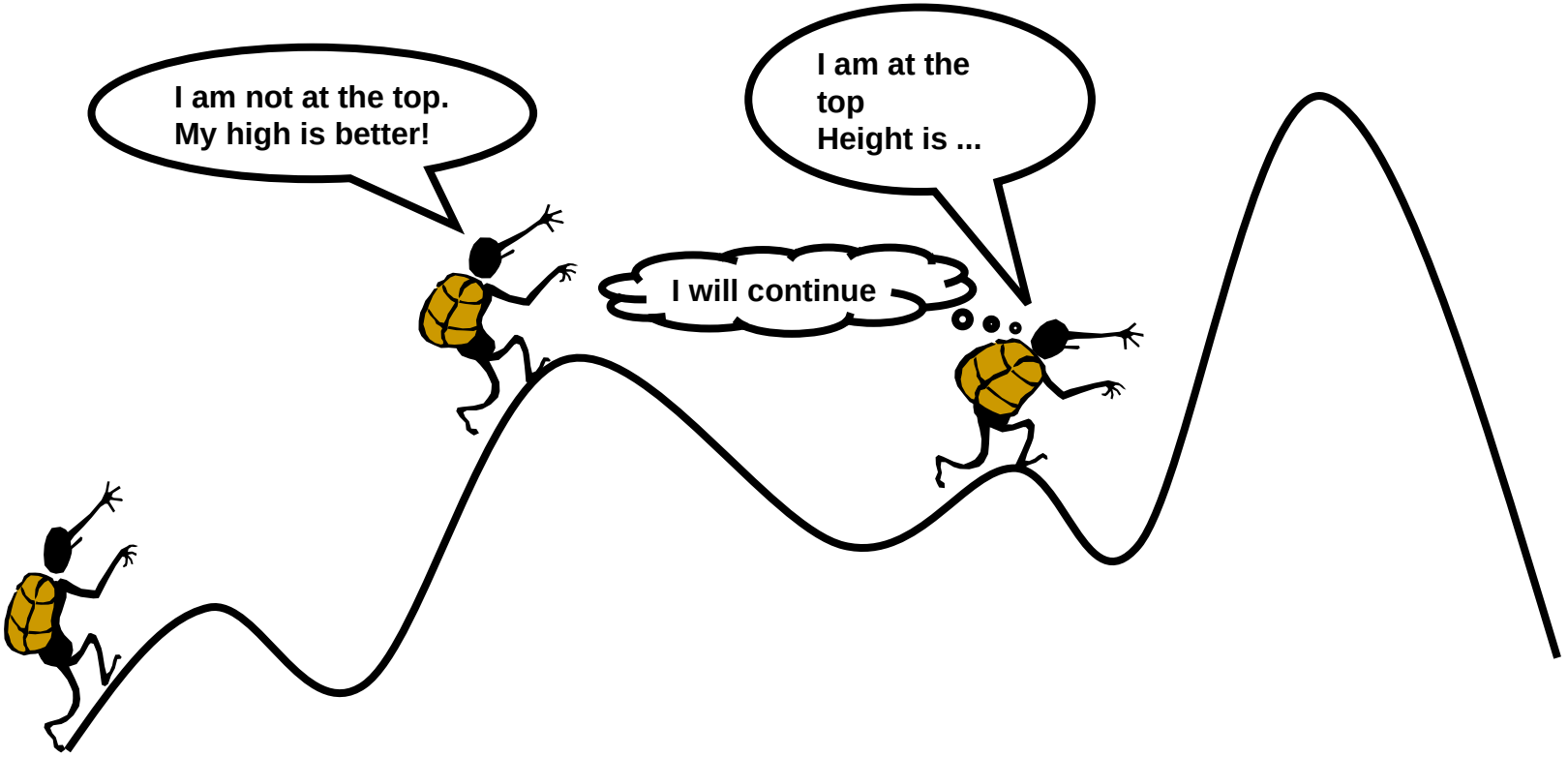
Problema: óptimo local

- **Si se usase un único individuo**



Problema: óptimo local

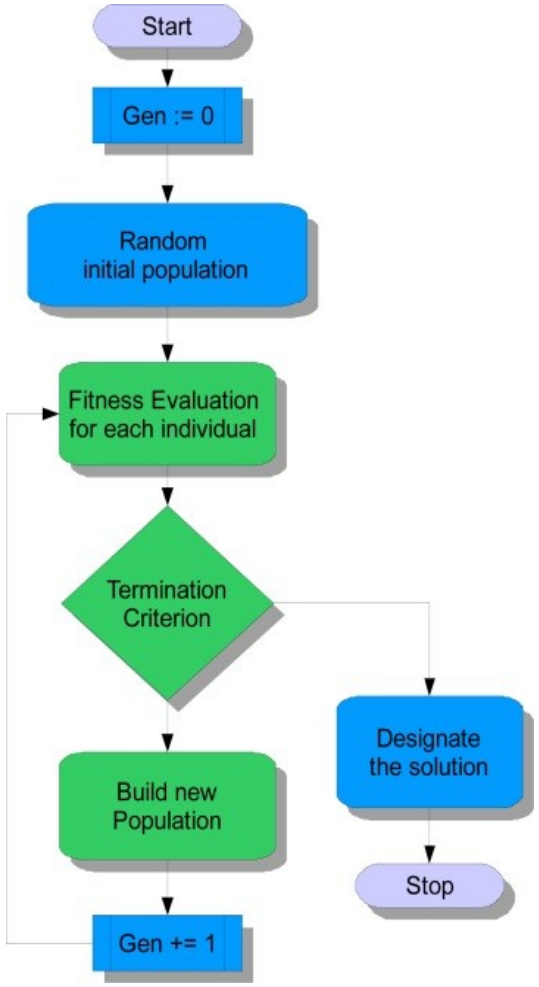
- **Búsqueda basada en poblaciones**

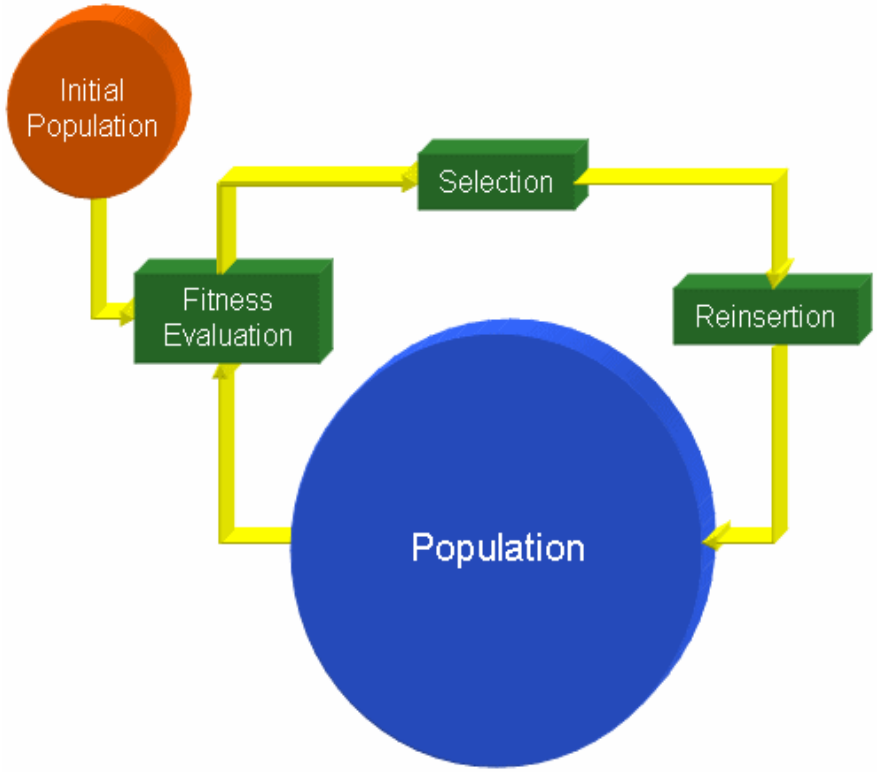


Problema: óptimo local

- **Búsqueda basada en poblaciones**

Algoritmo Poblacional





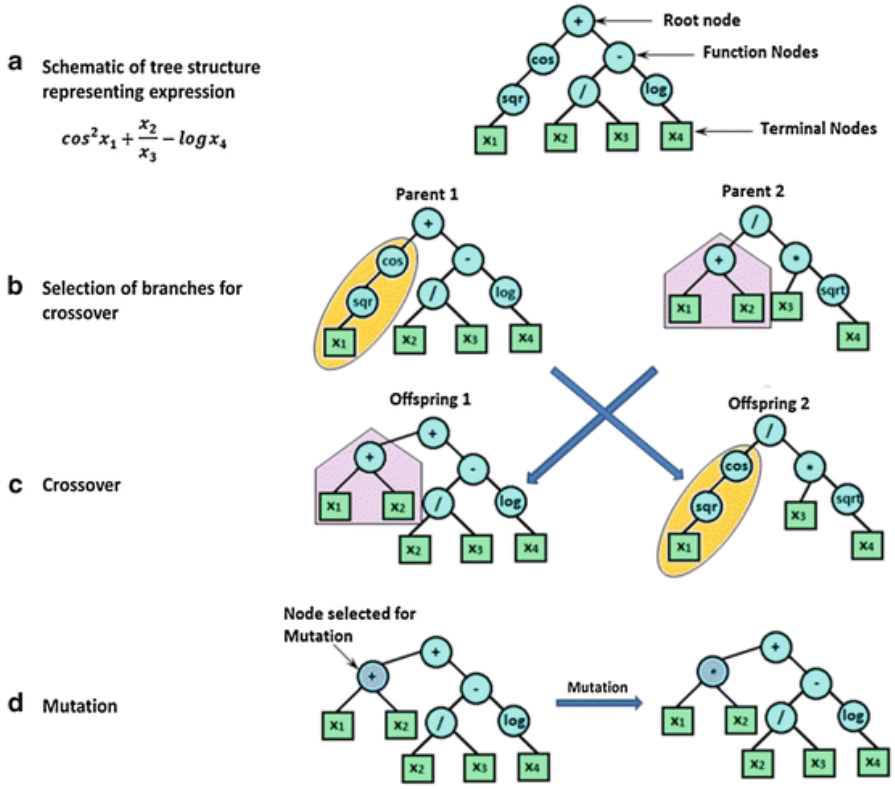
Se podría tener un modelo que seleccione soluciones, opere con ellas y puedan ser reinsertadas en la población dando lugar a una nueva población.

Algoritmos Genéticos

- Basados en los principios darwinianos de la Evolución natural.
- Conjunto de soluciones (población) que se combinan para generar mejores.
- Modelo más flexible, admite muchos tipos de representación.

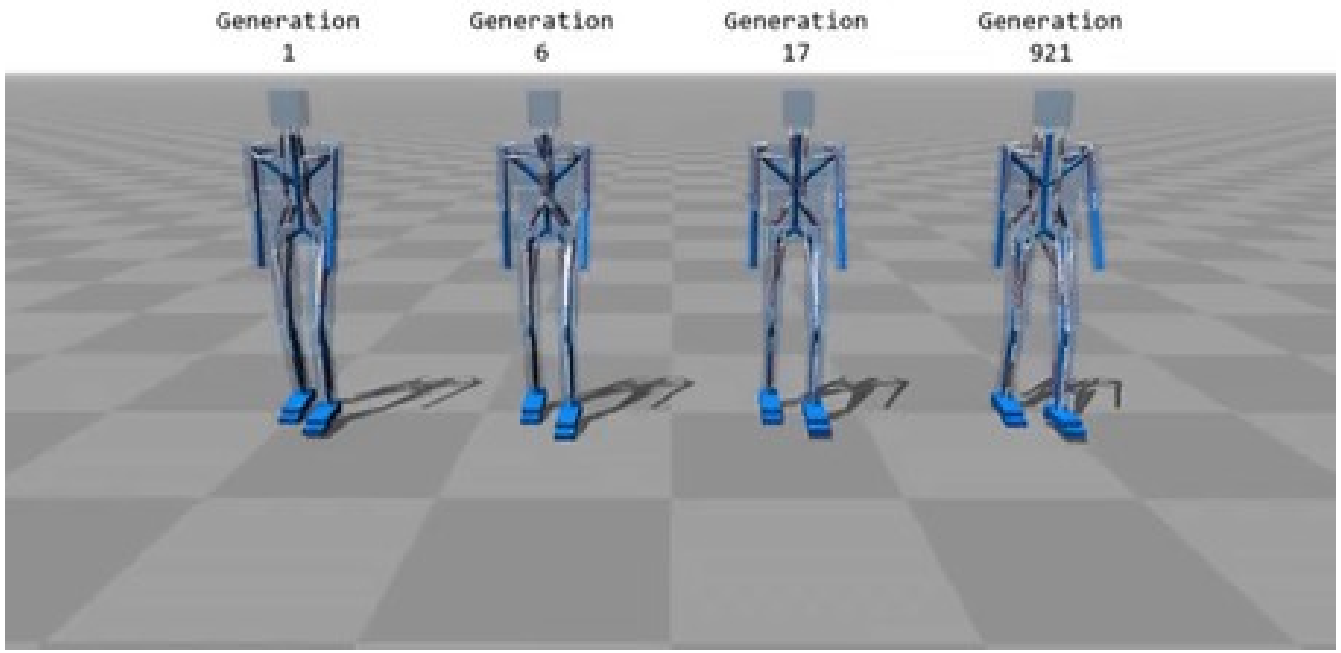
Ventajas de AG

- Cada solución representa un árbol de expresión.
  - Programación Genética (GP).



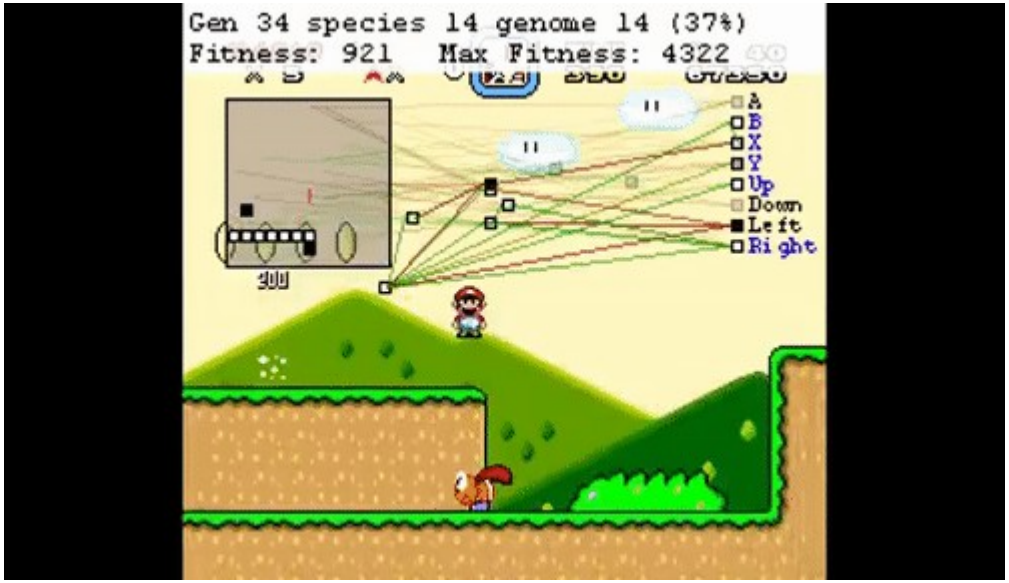
Ventajas de AG

- Muy flexible.



Neurocomputación

- Mejora la Red Neuronal para aprendizaje.
- Bastante usado en juegos.



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.

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

LA METÁFORA

EVOLUCIÓN		RESOLUCIÓN DE PROBLEMAS
		Solución
Individuo	⟷	Candidata
Adaptación	⟷	Calidad
Entorno	⟷	Problema

Evolución Natural

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

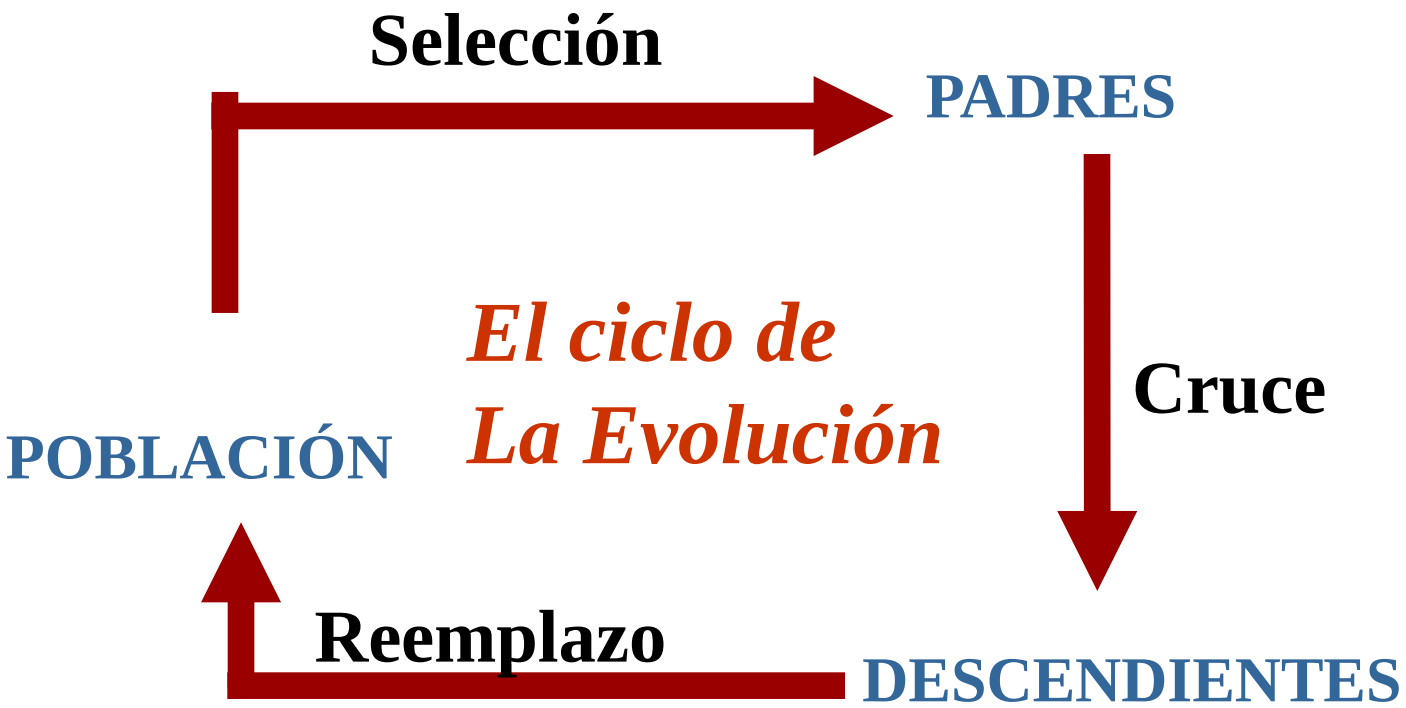
- Una entidad o individuo tiene la habilidad de reproducirse.
- Hay una población de tales individuos que son capaces de reproducirse.
- Existe alguna variedad, diferencia, entre

los individuos que se reproducen, que

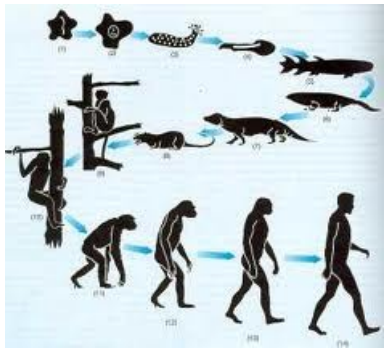
Influyen en la habilidad para sobrevivir.

Algoritmos Genéticos

Los algoritmos genéticos son algoritmos de optimización, búsqueda y aprendizaje inspirados en los procesos de Evolución Natural y Evolución Genética



Proceso en detalle



Algoritmo Genético

- PADRES

Cruce

Mutación

- DESCENDIENTES

Reemplazamiento

- POBLACIÓN

Modelos Genéticos

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 anterior.

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

Modelo Generacional

$P_{actual}(t)$
C_1
C_2
...
C_M

SELECCIÓN

(los C' son copias de los C)

P_{padres}
C'_1
C'_2
...
C'_M

CRUCE

con prob P_c

$P_{intermedia}$
C''_1
C'_2
...
C''_M

MUTACIÓN con prob. P_m

P_{hijos}
$H_1 = C''^m_1$
$H_2 = C'^m_2$
...
$H_M = C''_M$

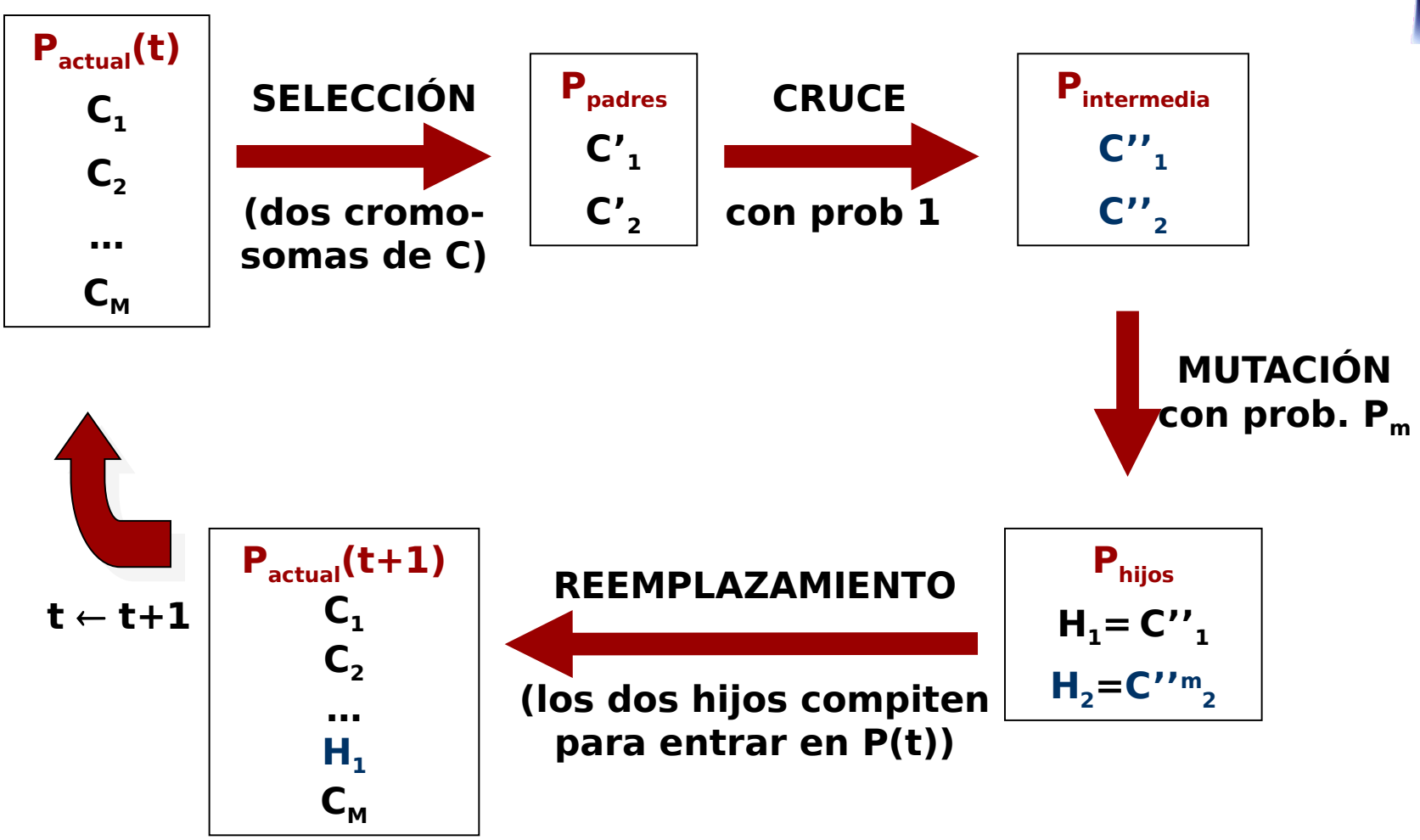
REEMPLAZAMIENTO

con elitismo (se mantiene el mejor de P(t))

$P_{actual}(t+1)$
H_1
H_2
...
H_{M-1}
C_{mejor}

$t \leftarrow t+1$

Modelo Estacionario



Creación de un AG

Los pasos para construir un Algoritmo Genético

- Diseñar una representación
- Decidir cómo inicializar 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

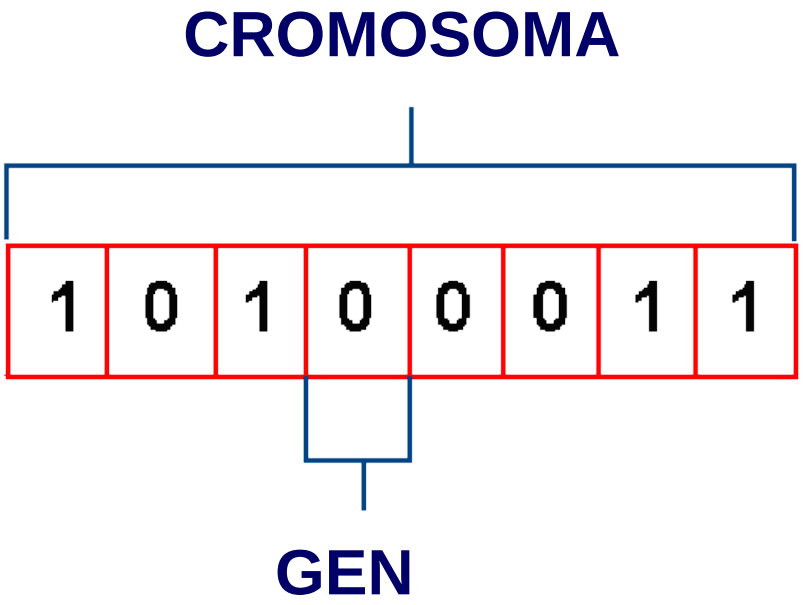
DEPENDE DEL PROBLEMA

COMPONENTES DEL ALGORITMO

Representación

- **Debemos disponer de un mecanismo para codificar un individuo como un genotipo.**
- **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.**

Representación Binaria



- Permiten representar valor lógicos.
- Como los ordenadores todo es binario sirve para representar cualquier tipo de información.
- Primera representación.

Representación Binaria

8 bits Genotipo

1	0	1	0	0	0	1	1

Fenotipo

- **Entero**
- **Número real**
- **secuencia**
- ...
- **Cualquier otra?**

Representación Real

- **Una forma natural de codificar una solución es utilizando valores reales como genes**
- **Muchas aplicaciones tienen esta forma natural de codificación**
- **Los individuos se representan como vectores de valores reales:**

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

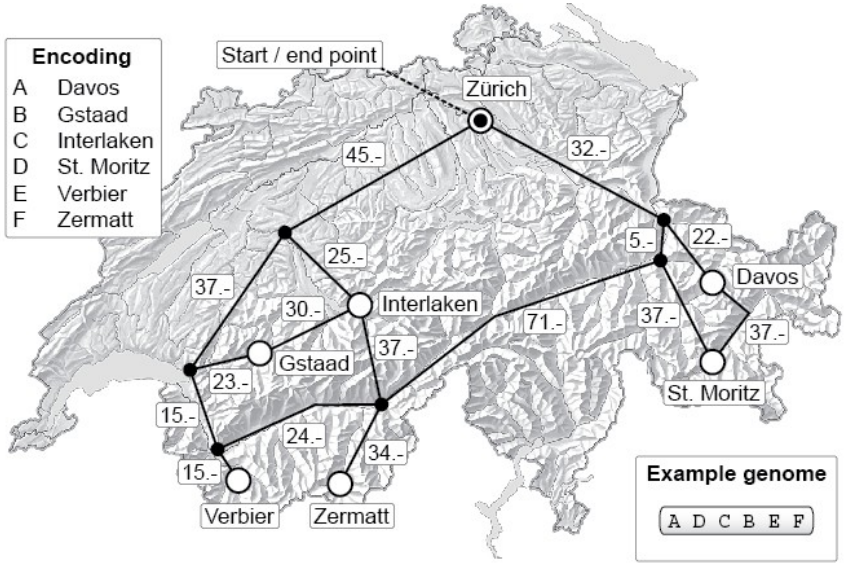
Permutación

- **Los individuos se representan como permutaciones.**

7	3	6	8	2	4	1	5

- **Se utilizan para problemas de secuenciación.**

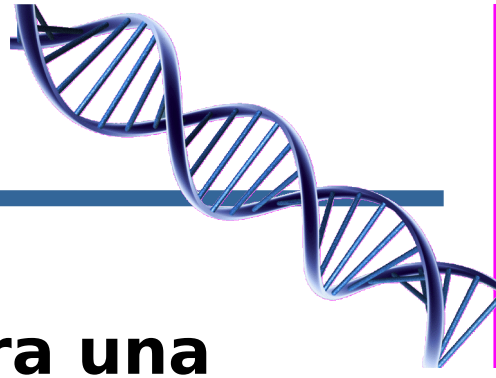
Ejemplo: Viajante de Comercio, donde cada ciudad tiene asignado un único número entre 1 y n. Necesita operadores especiales para garantizar que el resultado de aplicar un operador sigue siendo una permutación.



Inicialización

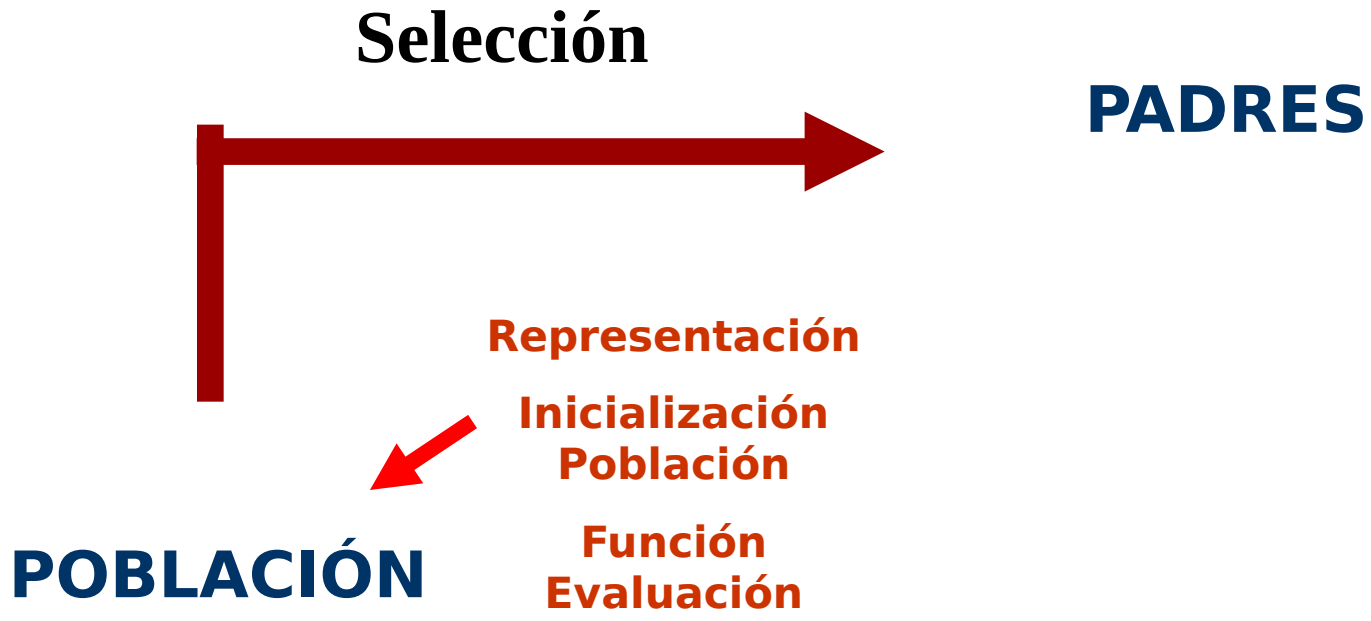
- **Uniforme sobre el espacio de búsqueda …**
  (si es posible)
  - **Cadena binaria: 0 ó 1 con probabilidad 0.5**
  - **Representación real: uniforme sobre un intervalo dado (para valores acotados)**
- **Elegir la población a partir de los resultados de una heurística previa.**

Evaluación de solución



- **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 el costo como penalización.**
- **Con múltiples objetivos se busca una solución de compromiso.**

Proceso de AG



Criterio 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.

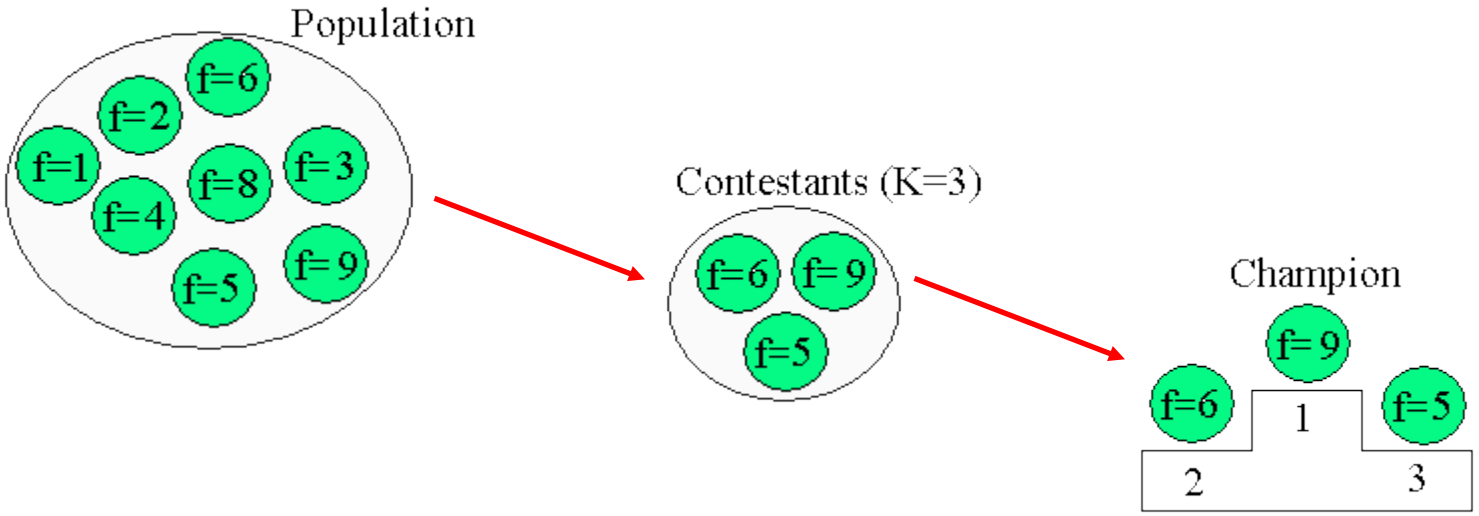
Esta idea nos define la presión selectiva que determina en qué grado la reproducción está dirigida por los mejores individuos.

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.



Otras selecciones

- **Selección por Torneo (TS): Escoge al individuo de mejor fitness de entre Nts individuos seleccionados aleatoriamente (Nts=2,3,…).**

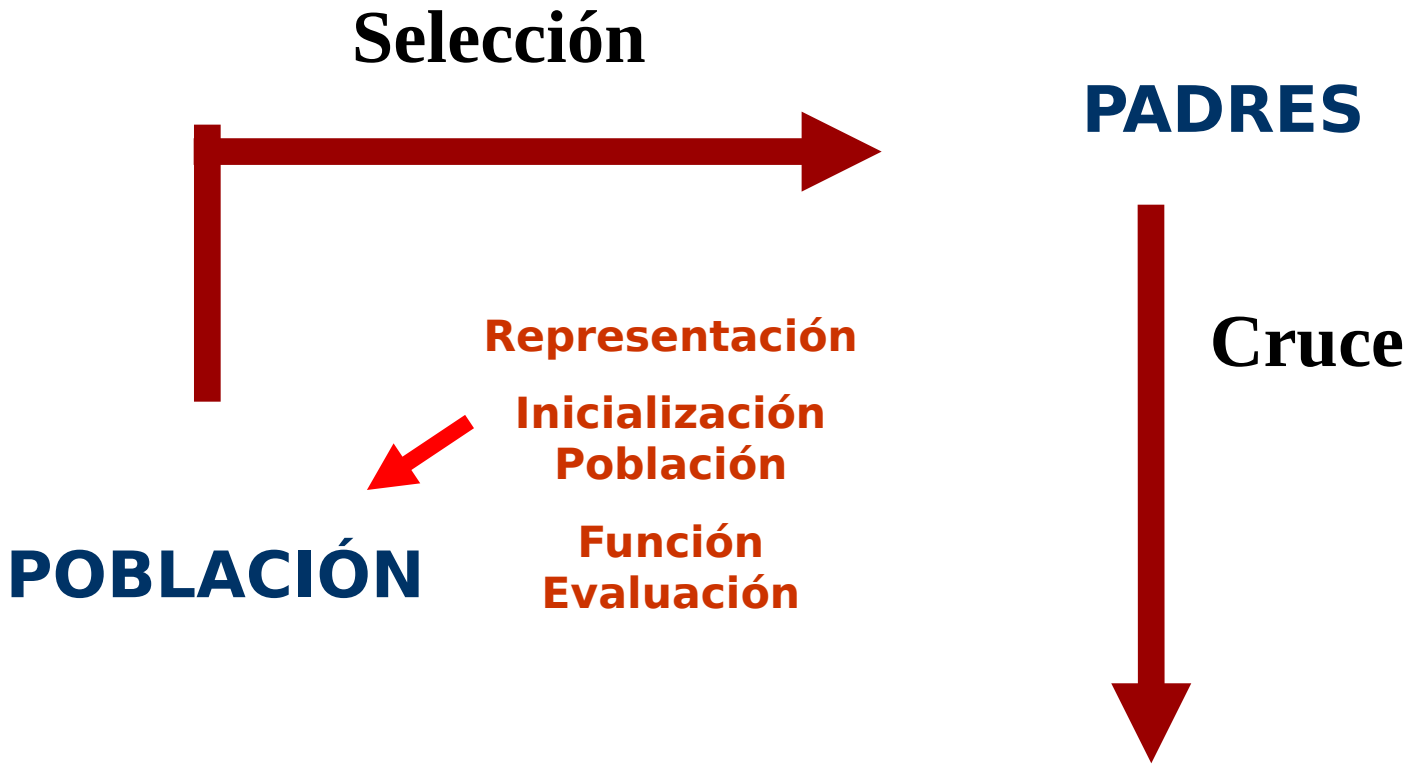
- **Orden Lineal (LR): La población se ordena en función de su fitness y se asocia una probabilidad de selección a cada individuo que depende de su orden.**

- **Selección Aleatoria (RS).**

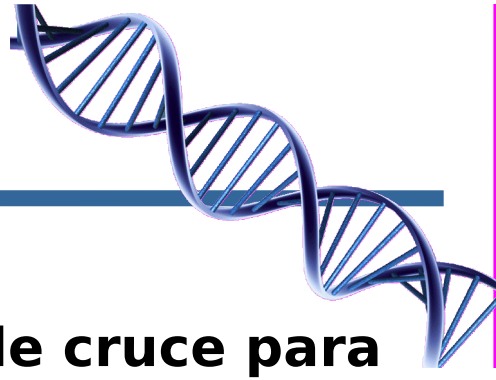
- **Emparejamiento Variado Inverso (NAM): Un padre lo escoge aleatoriamente, para el otro selecciona Nnam padres y escoge el más lejano al primer (Nnam=3,5, ….). Está orientado a generar diversidad.**

- **Selección por Ruleta: Se asigna una probabilidad de selección proporcional al valor del fitness del cromosoma.**

Proceso de AG



Operador de Cruce



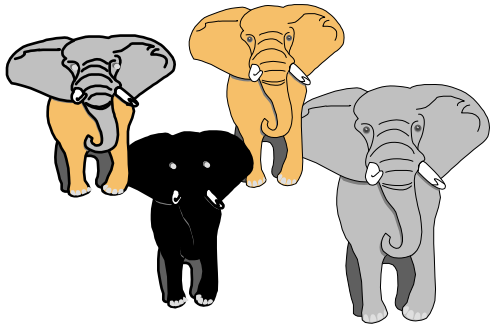
Podemos tener uno o más operadores de cruce para nuestra representación.

Algunos aspectos importantes a tener en cuenta son:

- **Los hijos deberían heredar algunas características de cada padre.**
- **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.**

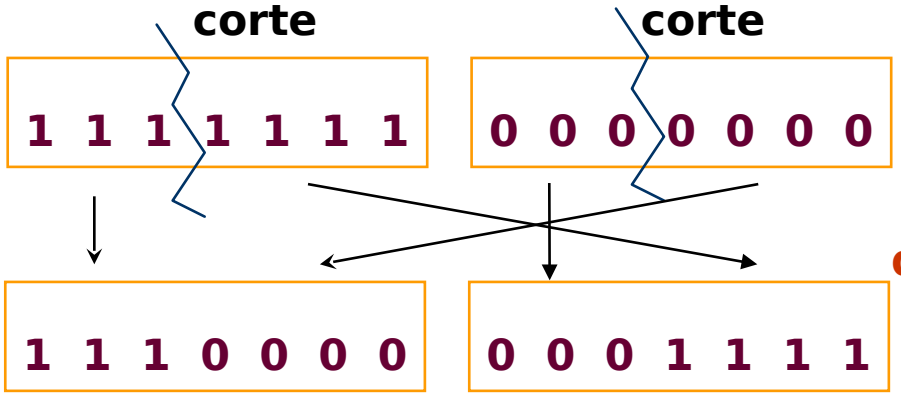
Cruce binario en 1 punto

Población:

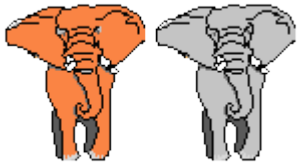


. . .

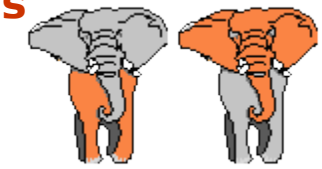
Cada cromosoma se corta en n partes que son recombinadas. (Ejemplo para n = 1).



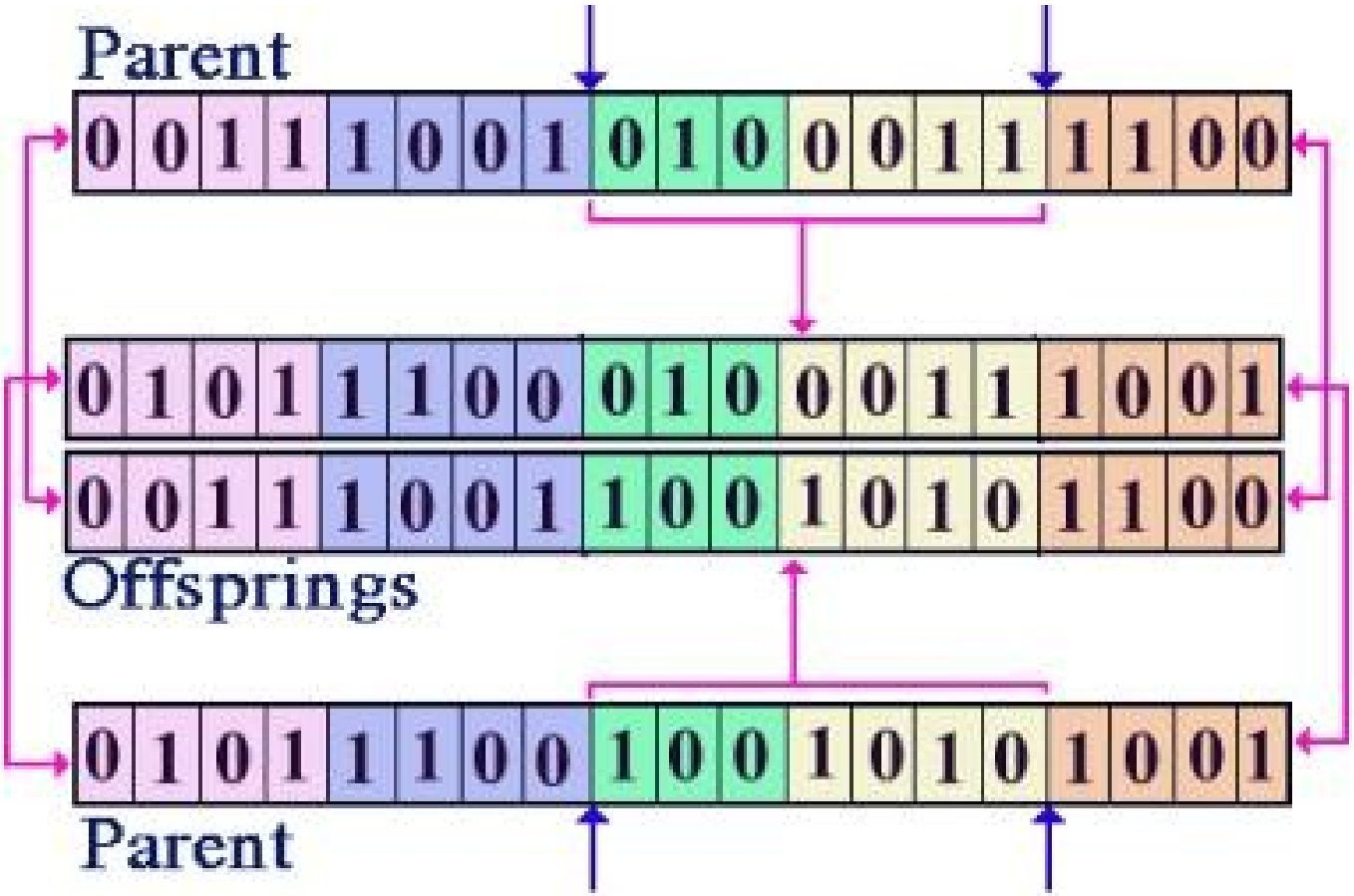
padres



descendientes



Cruce binario en 2 puntos



Cruce Aritmético

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

Existe muchos operadores específicos para la codificación real.

Cruce BLX-alpha

- **Dados 2 cromosomas**

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

- **BLX-** α **genera dos descendientes**

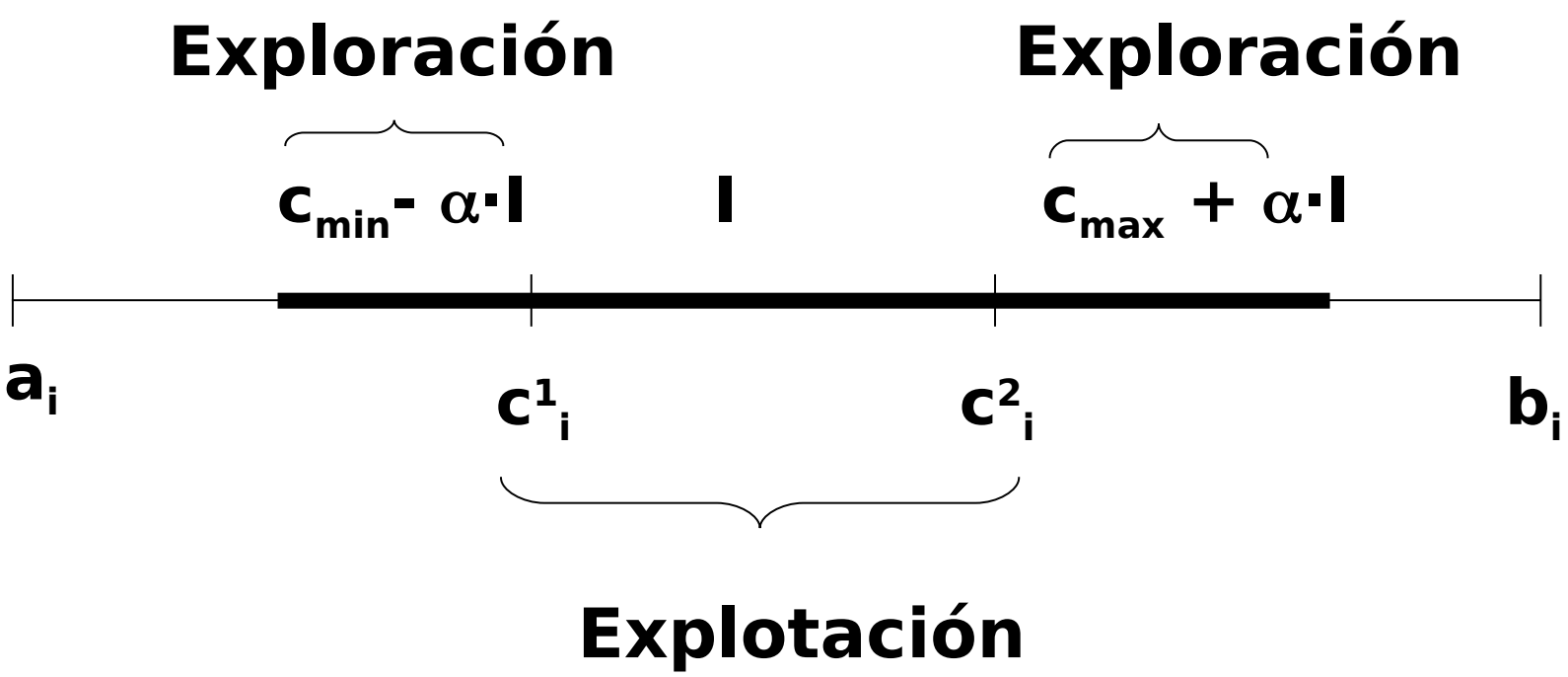
$$H_k = (h_{k1},\ldots, h_{ki},\ldots, h_{kn}) , k = 1,2$$

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

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

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

Cruce BLX-alpha



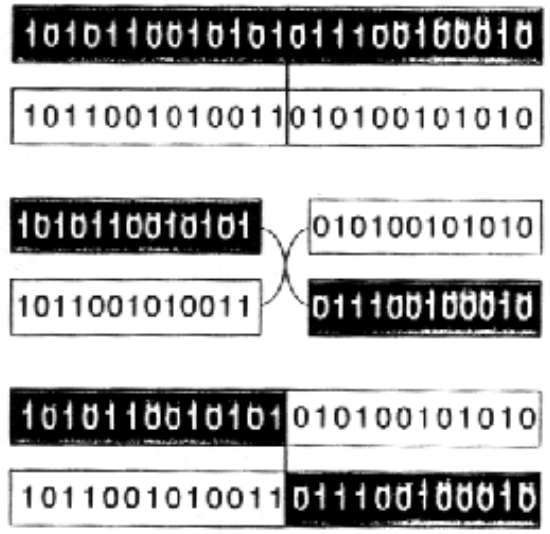
Representación de orden

Padre 1

7	3	1	8	2	4	6	5

Padre 2

4	3	2	8	6	7	1	5

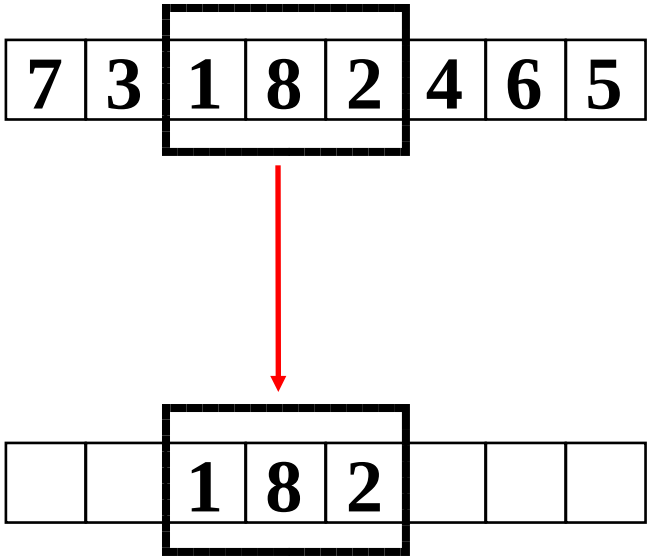


CROSSOVER is the fundamental mechanism of genetic rearrangement for both real organisms and genetic algorithms.

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

Cruce OX

Padre 1



Padre 2

4	3	2	8	6	7	1	5

¿Qué hacemos con las ciudades restantes?

Cruce OX

Padre 1

7	3	1	8	2	4	6	5

Padre 2

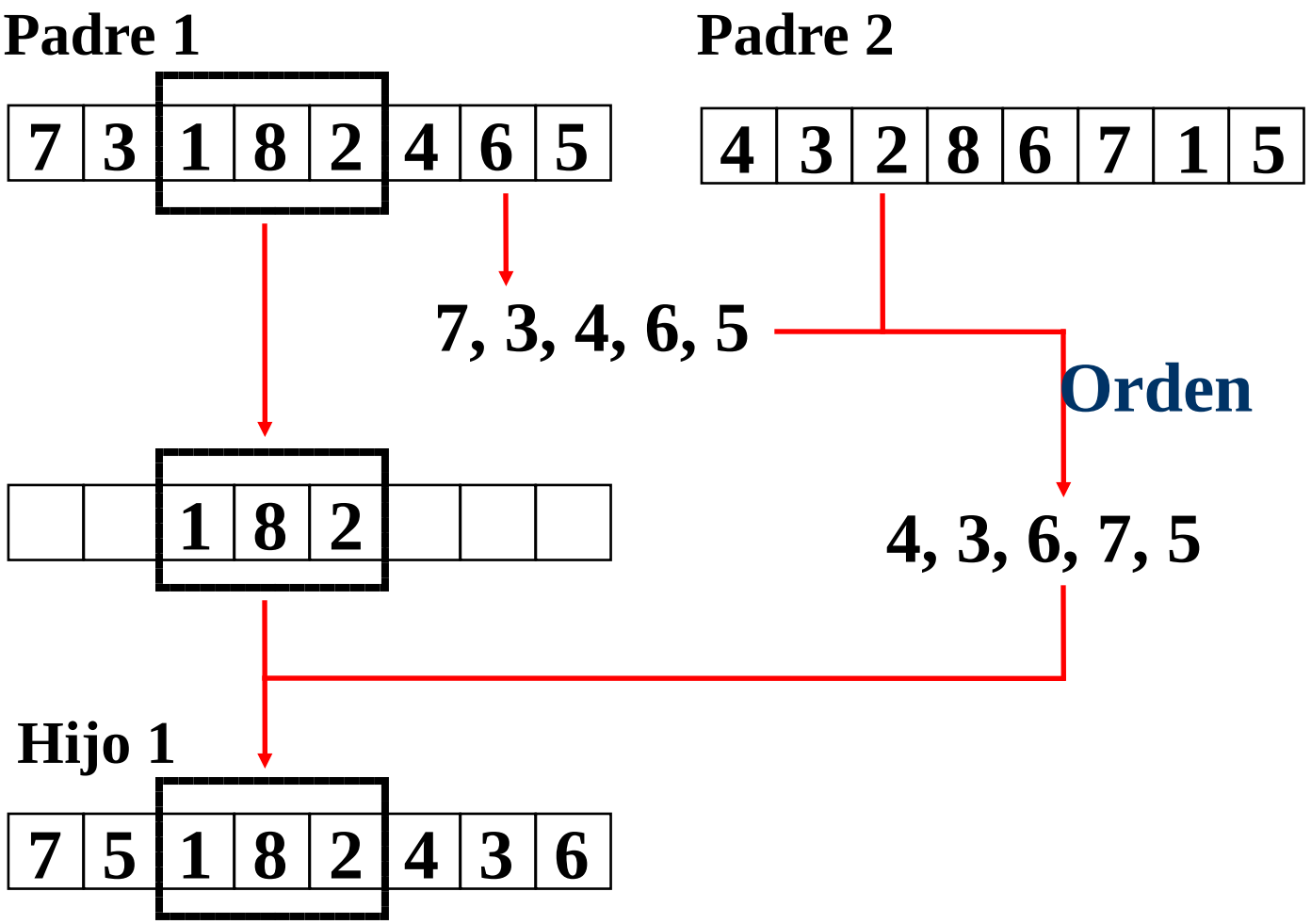
4	3	2	8	6	7	1	5

7, 3, 4, 6, 5

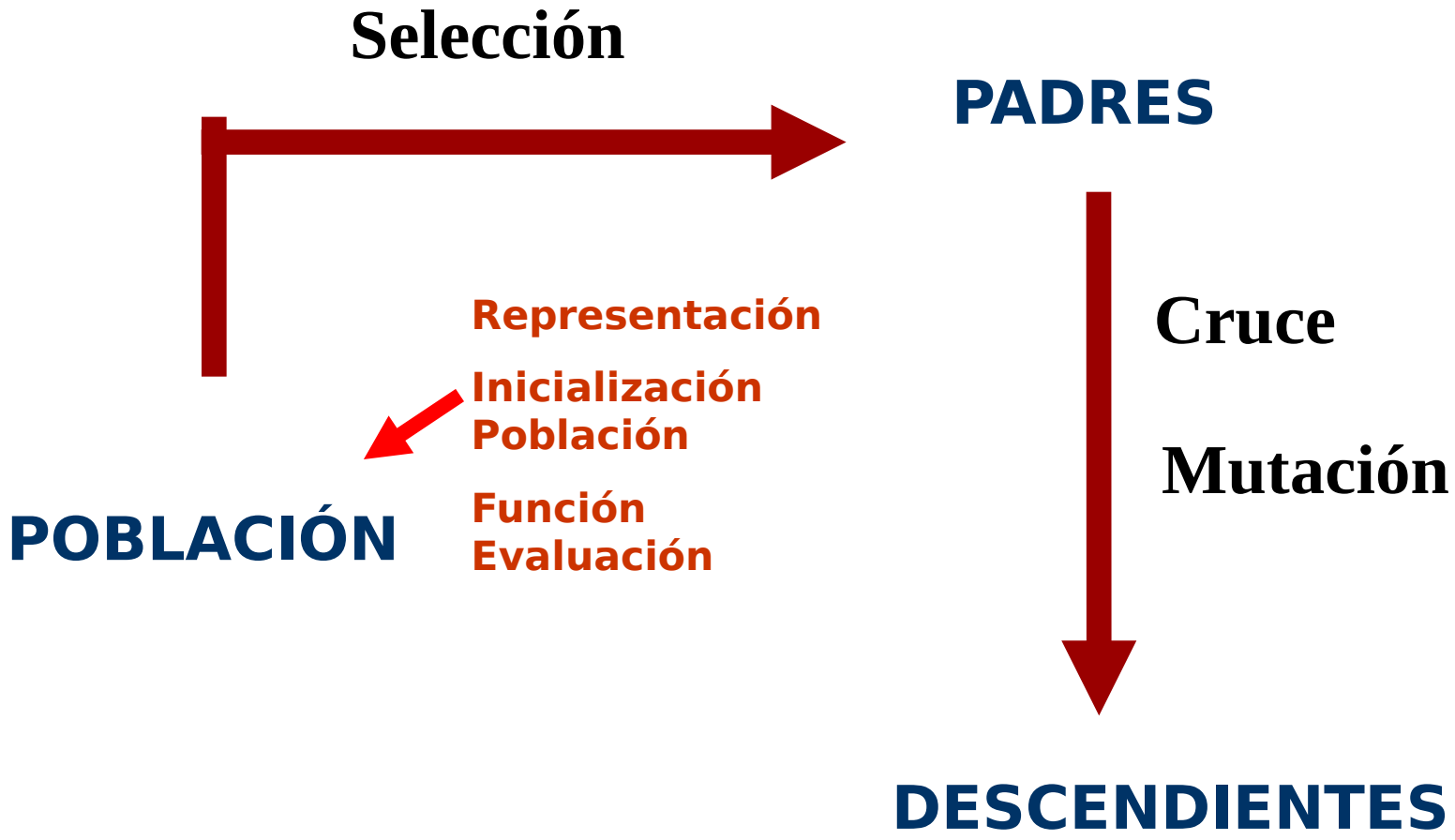
		1	8	2			

¿Qué hacemos con las ciudades restantes?

Cruce OX


Padre 1
7 3 1 8 2 4 6 5
Padre 2
4 3 2 8 6 7 1 5
7, 3, 4, 6, 5
Orden
1 8 2
4, 3, 6, 7, 5
Hijo 1
7 5 1 8 2 4 3 6

Proceso de AG



Operador Mutación

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

Algunos aspectos importantes a tener en cuenta son:

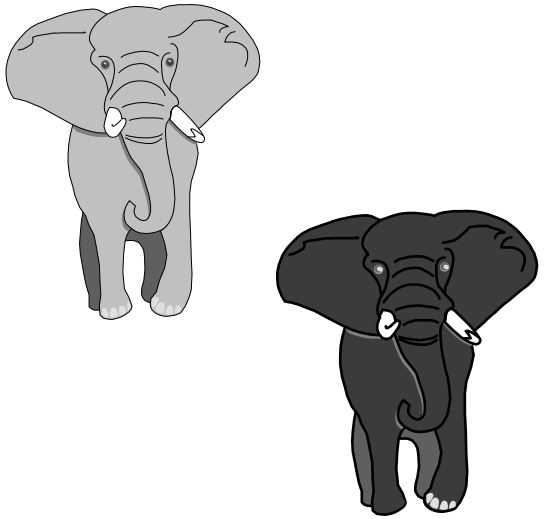
- **Debe permitir alcanzar cualquier parte del espacio de búsqueda.**
- **El tamaño de la mutación debe ser controlado.**
- **Debe producir cromosomas válidos.**
- **Se aplica con una probabilidad muy baja de actuación sobre cada descendiente obtenido tras aplicar el operador de cruce (incluidos los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

Mutación 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

Mutación Real

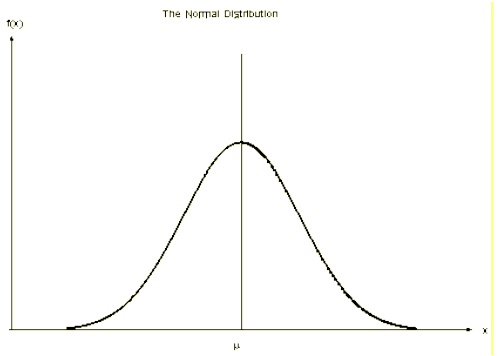
Perturbación de los valores mediante un valor aleatorio.

Frecuentemente, mediante una distribución Gaussiana/normal N(0,σ), donde

- **0 es la media**
- **σ es la desviación típica**

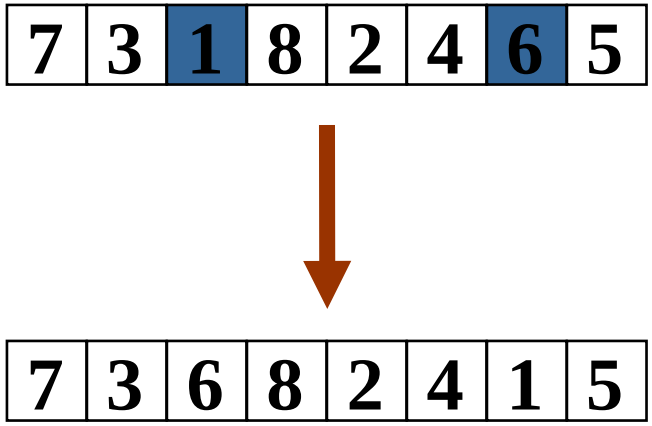
$$x'_i = x_i + N(0,\sigma_i)$$

para cada parámetro.

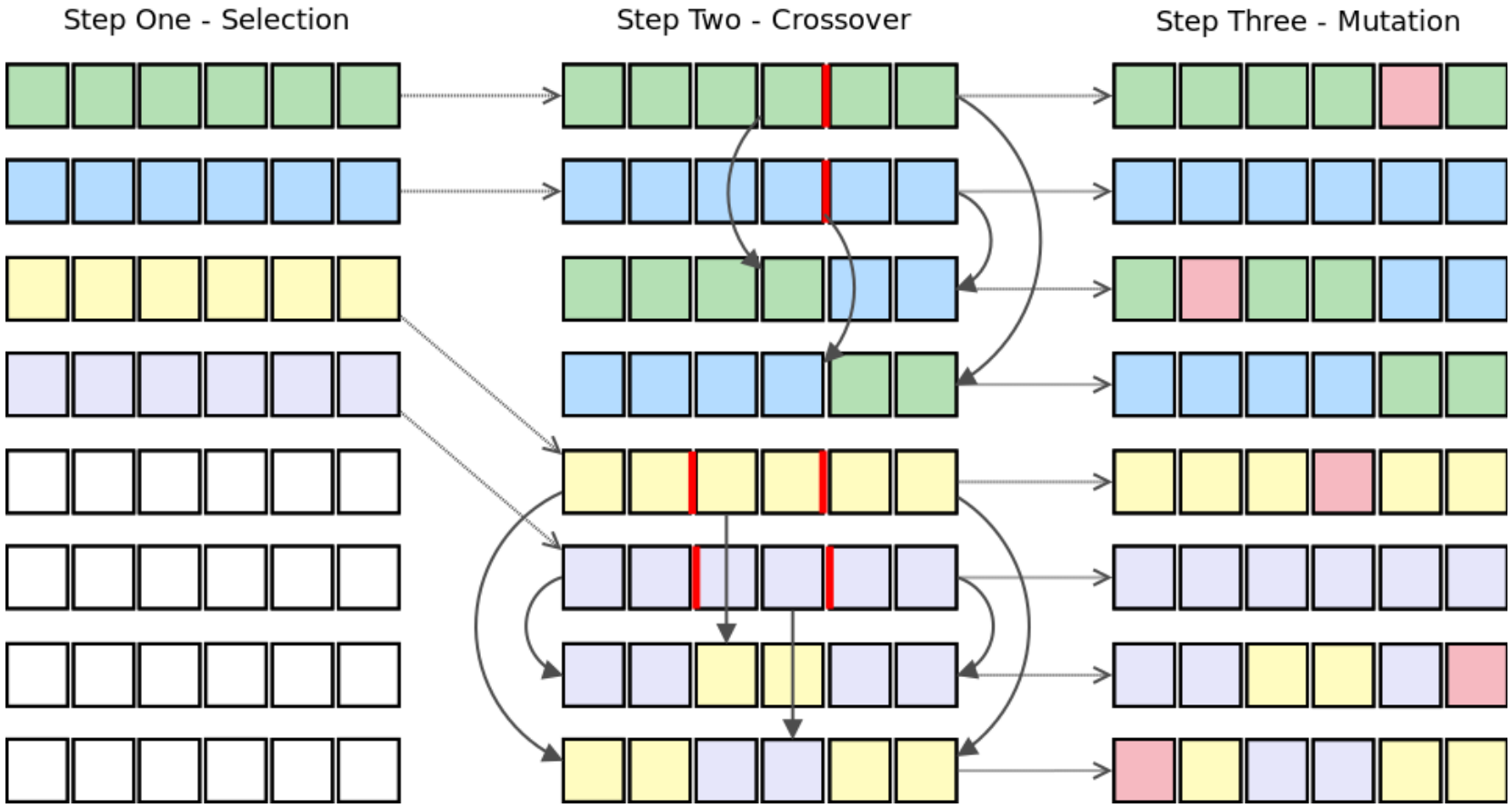


Mutación de orden

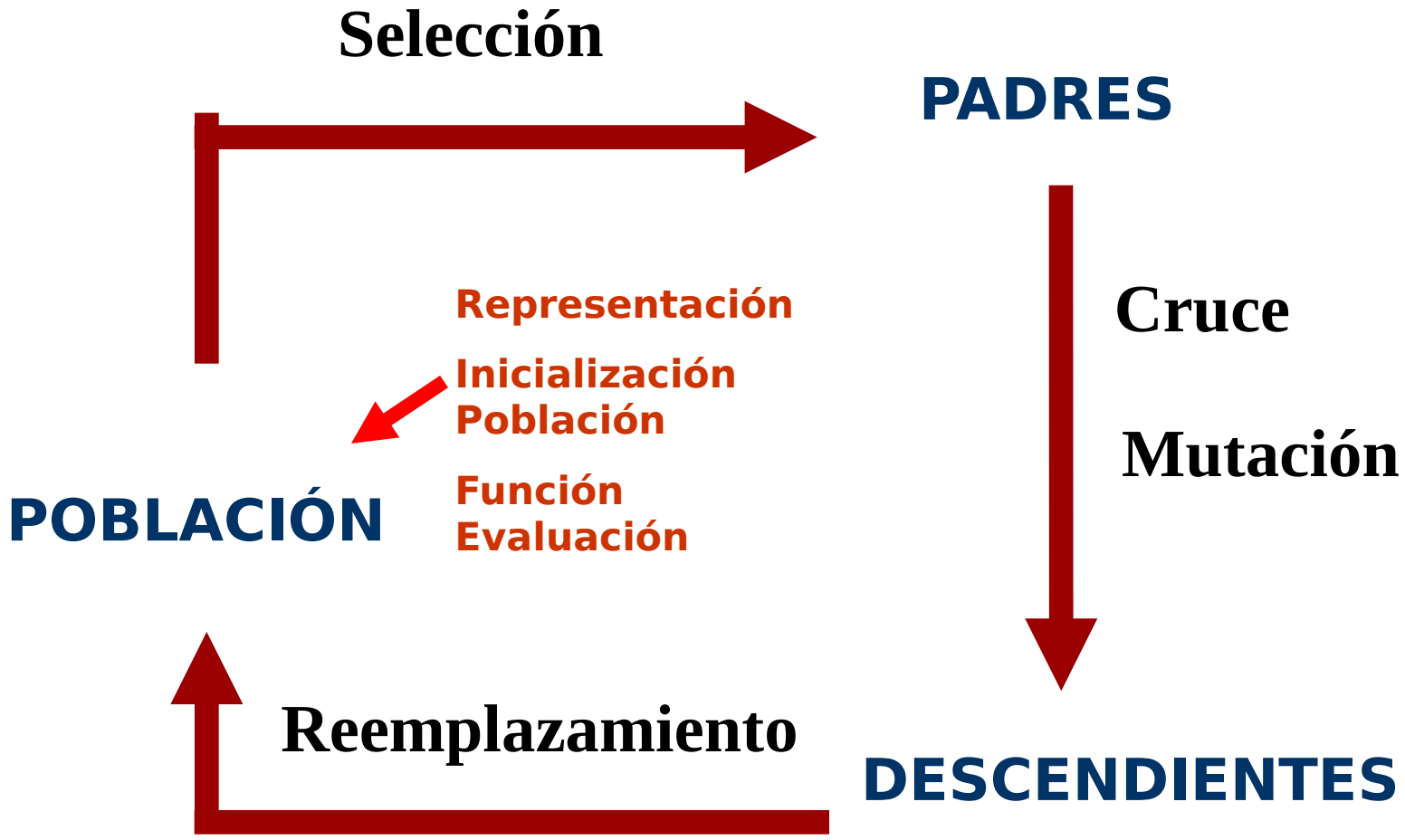
Selección aleatoria de dos genes e intercambio de ambos.



Proceso de AG



Proceso de AG



Reemplazo

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.

Reemplazo Estacionario

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:**

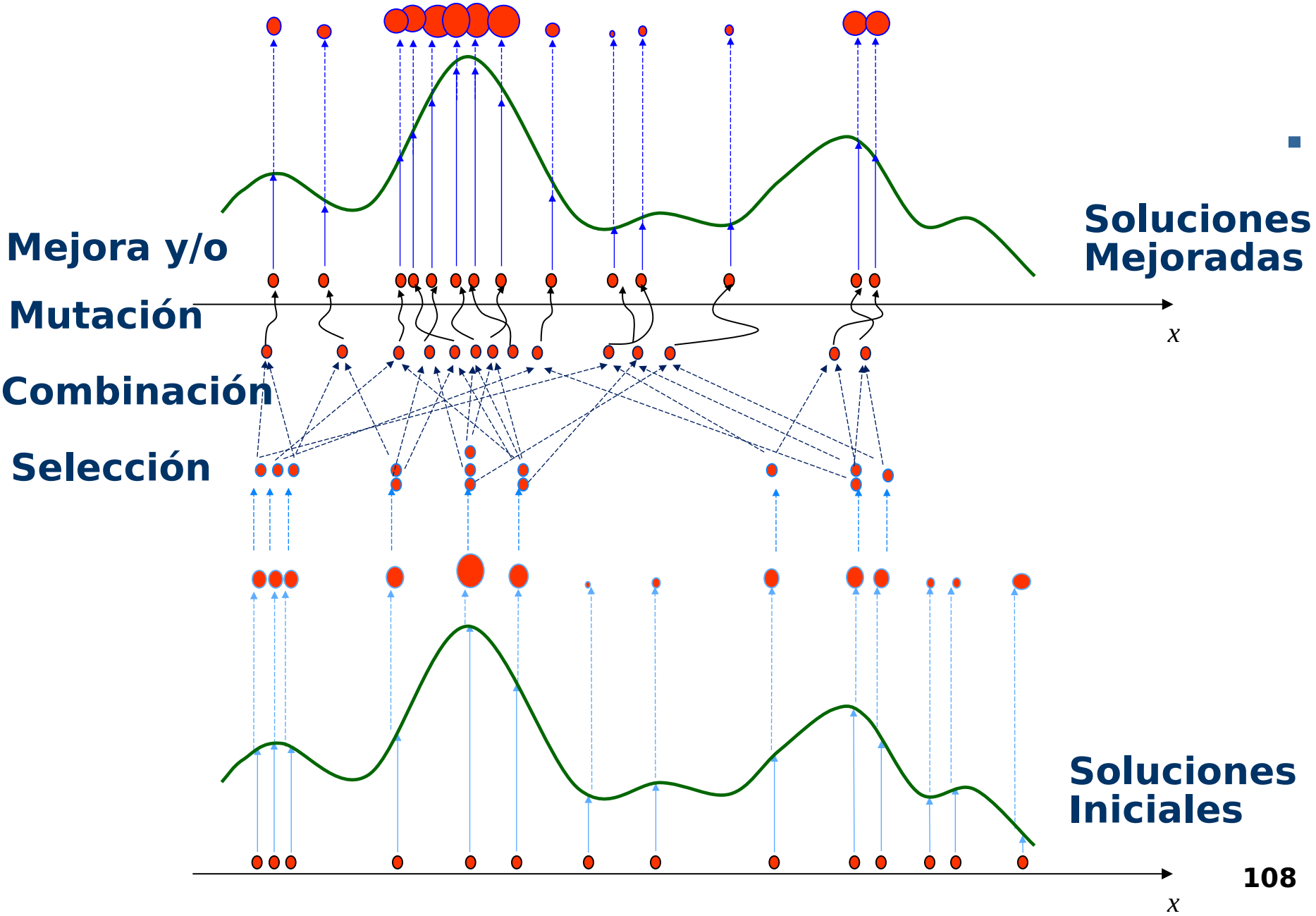
  Fijar el máximo número de evaluaciones
- **Límite sobre la paciencia del usuario:**

  Después de algunas iteraciones sin mejora.

Proceso de AG

PROCESO ITERATIVO + CRITERIO DE PARADA

Efecto de la Evolución



Sobre su uso

- **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/chequear la actuación de un algoritmo sobre ejemplos simples si se desea trabajar con casos reales.**
- **Existe una 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 puede ser extensible a otros casos, es el error en el que incurren algunos autores)

¿Alguna Pregunta?