Algoritmos Genéticos

Grado en Ingeniería Informática

Organización y Gestión de Empresas

José Manuel Galán
Luis R. Izquierdo

Universidad de Burgos

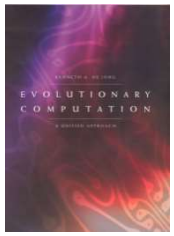
Estructura de la presentación

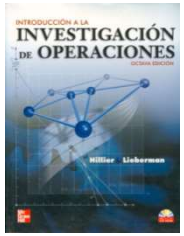
Aproximación evolutiva
Evolución artificial
Algoritmo genético
- Representación
- Inicialización
- Fitness
- Selección
- Cruce
- Mutación
- Reemplazamiento
- Criterio de parada

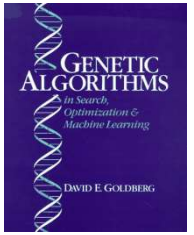
Algunos comentarios adicionales
Resumen

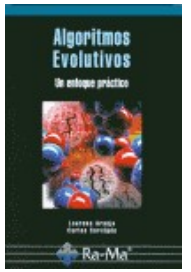
Bibliografía

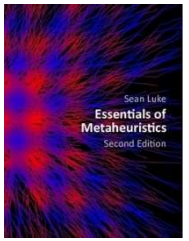
- Kenneth A. De Jong. Evolutionary Computation. A Unified Approach, (2006). MIT Press.
- Hillier, F.S., Lieberman, G.S. (2008) Introducción a la Investigación de Operaciones. McGraw-Hill. México D.F.
- D.E. Goldberg, Algoritmos Genéticos in Search, Optimization and Machine Learning. Addison Wesley, (1989)
- B. Melián, J.A. Moreno Pérez, J.M. Moreno Vega. Metaheurísticas: un visión global. Revista Iberoamericana de Inteligencia Artificial 19 (2003) 7-28
- Araujo, L., Cervigón, C. (2009) Algoritmos Evolutivos. Ra-Ma.
- Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada
- Luke, Sean (2014) Essentials of Metaheuristics. George Mason Univesity https://cs.gmu.edu/~sean/book/metaheuristics/
- Pérez, E. (2010) Guía para recién llegados a los Algorimos Genéticos. Universidad de Vallaoolid.





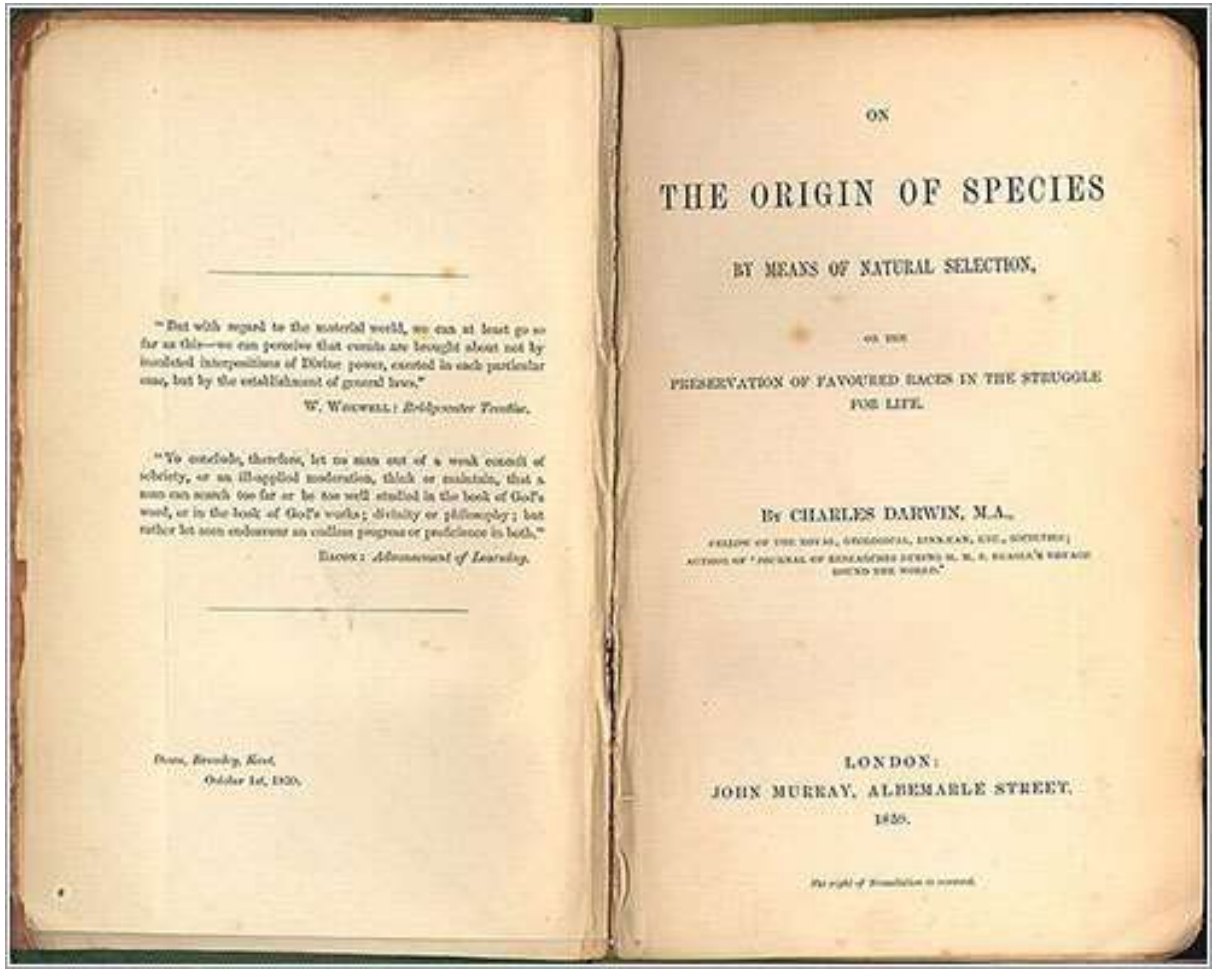






http://sci2s.ugr.es/

Evolución natural



"But with regard to the material world, we can at least go so far as this—we can perceive that events are brought about not by insulated interpositions of Divine power, exerted in each particular case, but by the establishment of general laws."

W. WHEWELL: *Bridgewater Treatise.*

"To conclude, therefore, let no man out of a weak conceit of sobriety, or an ill-applied moderation, think or maintain, that a man can search too far or be too well studied in the book of God's word, or in the book of God's works; divinity or philosophy; but rather let men endeavour an endless progress or proficience in both."

BACON: *Advancement of Learning.*

Down, Bromley, Kent,
October 1st, 1859.

ON

THE ORIGIN OF SPECIES

BY MEANS OF NATURAL SELECTION,

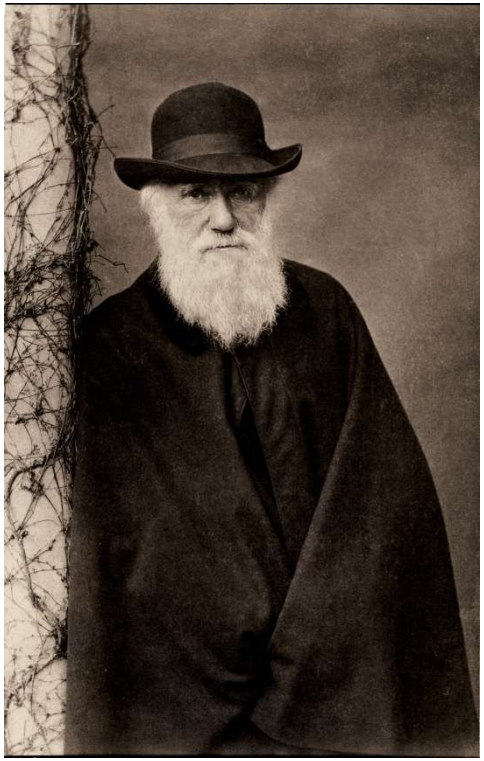
OR THE

PRESERVATION OF FAVOURED RACES IN THE STRUGGLE FOR LIFE.

By CHARLES DARWIN, M.A.,

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

LONDON:
JOHN MURRAY, ALBEMARLE STREET.
1859.



Source: Darwin, C. (1859). On the Origin of Species by Means of Natural Selección or the Preservations of Favoured Races in the Struggle for Life. London: John

Sistema evolutivo

- Diversidad: las entidades del sistema no son todas iguales; muestran diferencias que afectan a la llamada fitness individual.
- Selección: el mecanismo de selección es una fuerza discriminante que favorece a un conjunto específico de entidades en lugar de a otras
- Replicación / Herencia / Conservación: las propiedades de las entidades en el sistema (o las entidades en sí mismas) son conservadas, replicadas o heredadas de una generación a otra al menos en cierto grado

Richard Dawkins. (2006): *El gen egoísta (3ª ed.). Oxford University Press. ISBN 0-1992-9114-4*

Source: Izquierdo, L.R. (2008). Advancing Learning and Evolutionary Game Theory with an Application to Social Dilemmas. PhD Thesis.

Biología evolutiva

- La evolución está centrada en los genes frente a las visiones centradas en los organismos vivos
- La selección natural es el vínculo entre los genes y la actuación (fitness) de sus estructuras decodificadas
- La reproducción es el punto en el que la evolución tiene lugar

Source: Richard Dawkins. (2006): The Selfish Gen (3ª ed.). Oxford University Press.

Biología evolutiva

- Creacionismo es la creencia religiosa de que el universo y la vida se originaron “a partir de actos específicos de creación divina”, en contraposición a la conclusión científica de que son consecuencia de procesos naturales

- El Lamarckismo (o herencia lamarckiana) es la idea de que un organismo puede pasar características adquiridas durante la vida a sus descendientes (también conocida como herencia de los rasgos adquiridos o herencia blanda)

Source: https://en.wikipedia.org/wiki/Lamarckism
https://en.wikipedia.org/wiki/Creationism

Algoritmo genético

- un conjunto prescrito de instrucciones o reglas bien definidas, ordenadas y finitas que permite llevar a cabo una actividad mediante pasos sucesivos
- En los 70, John Henry Holland en su libro *"Adaptation in Natural and Artificial Systems"* desarrolló una de las más prometedoras líneas de la inteligencia artificial: los algoritmos genéticos. Estos algoritmos se basan en la evolución biológica.
- *"Evolution as an engine for adaptation"*

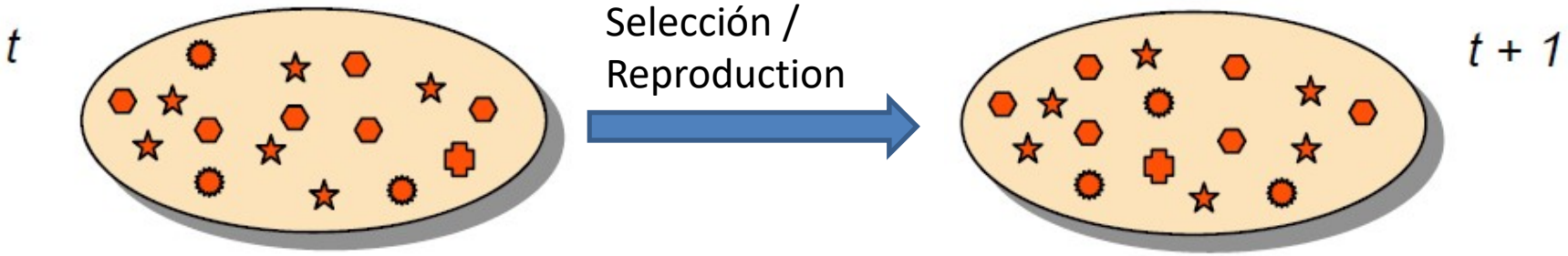
SANTA FE INSTITUTE

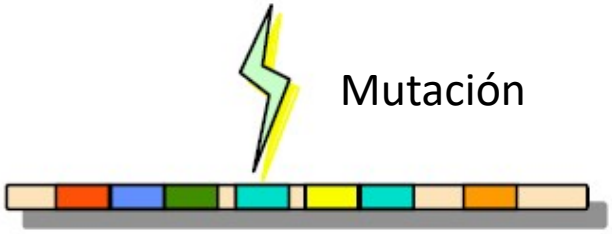
Emergence: From Chaos to Order
John H. Holland. Redwood City, California: Addison-Wesley. 1998
Cloth: ISBN 0-201-14943-5

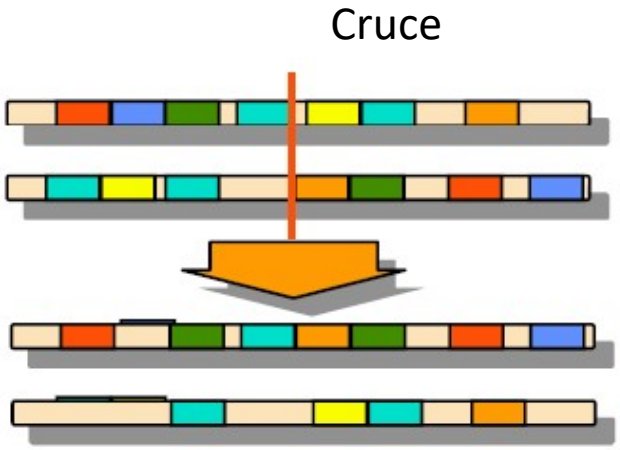
Computación evolutiva

- Métodos basados en poblaciones en los que los individuos representan soluciones
- Algoritmos de optimización aleatoria que simulan un proceso evolutivo en un ordenador
- Con frecuencia proporcionan buenos resultados en problemas difíciles
- Hay diferentes aproximaciones: algoritmos genéticos, algoritmos meméticos, programación genética…

Elementos básicos

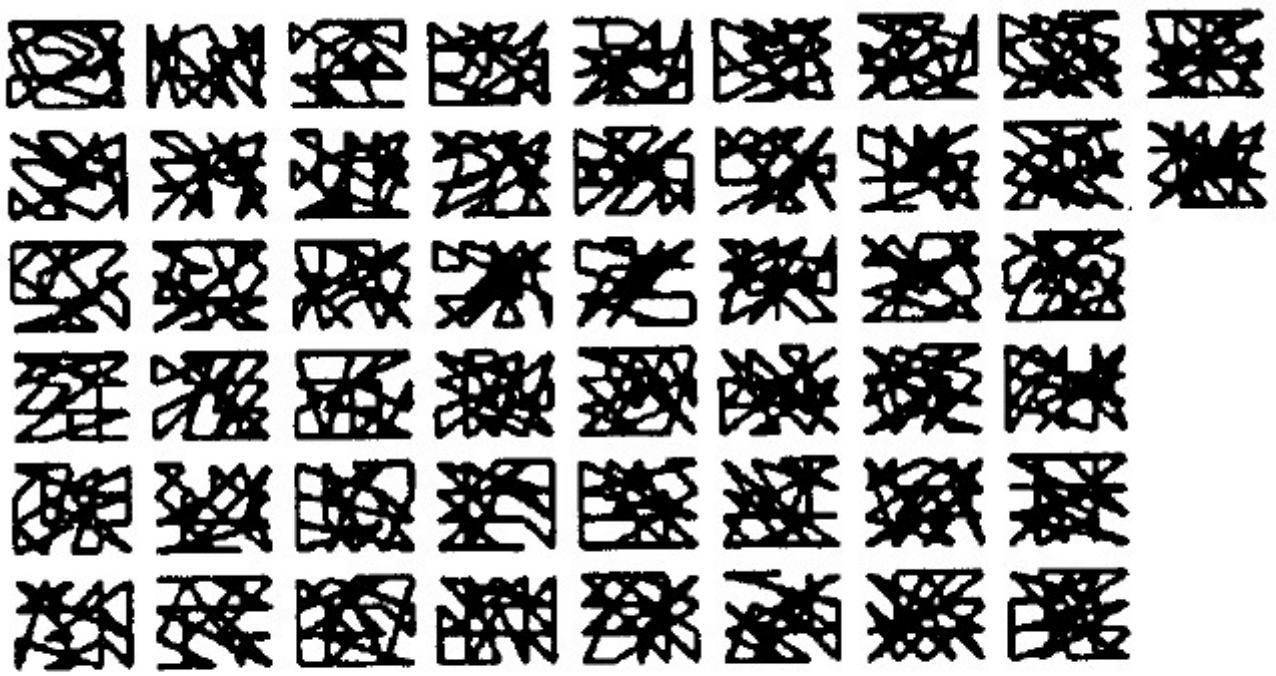






Based on Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

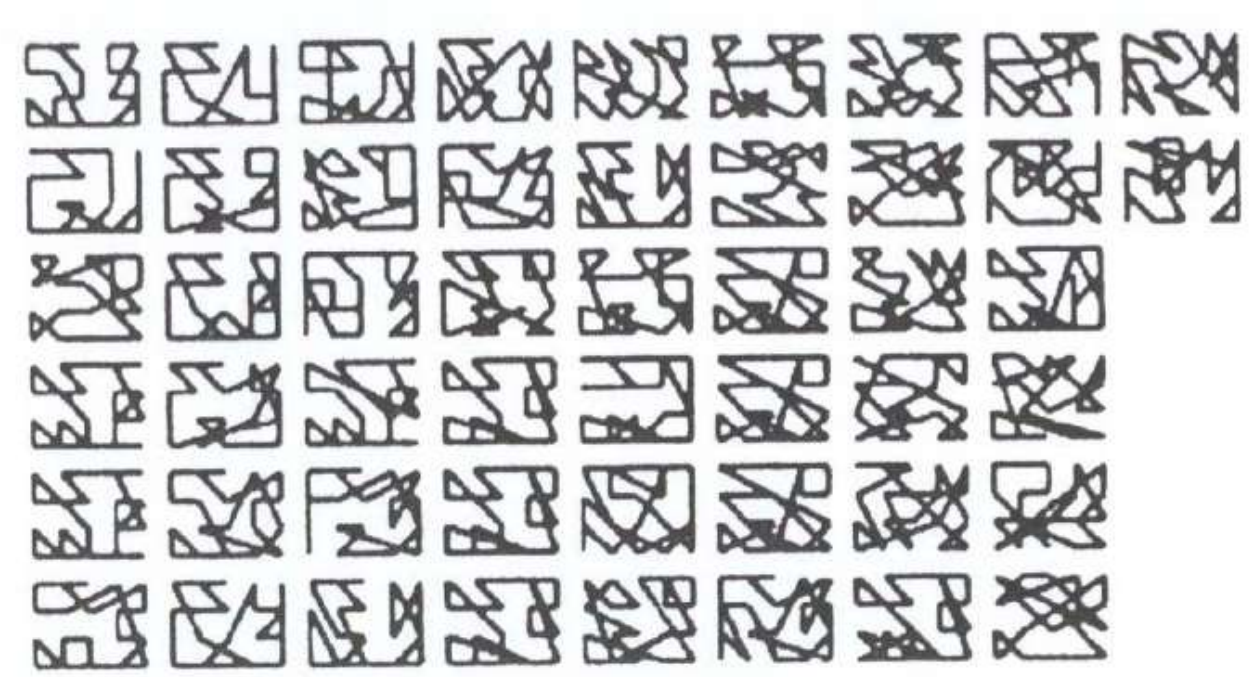
Idea intuitiva: el problema del viajante



(0)

Source: Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

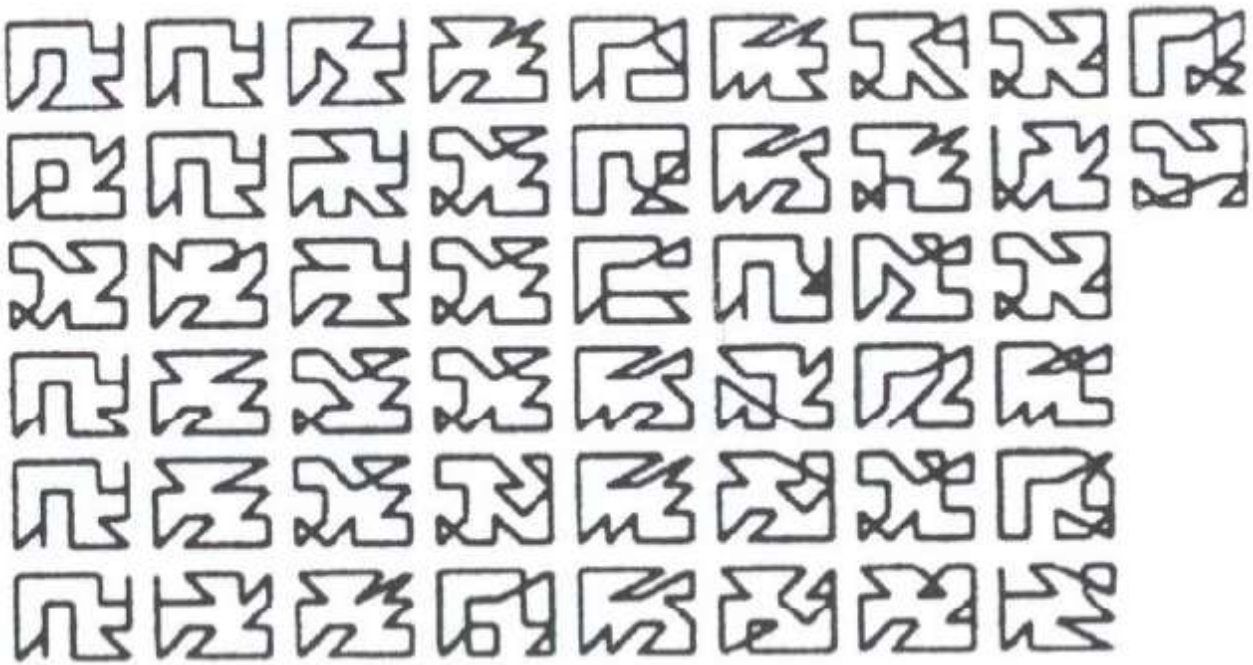
Idea intuitiva: el problema del viajante



(10)

Source: Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

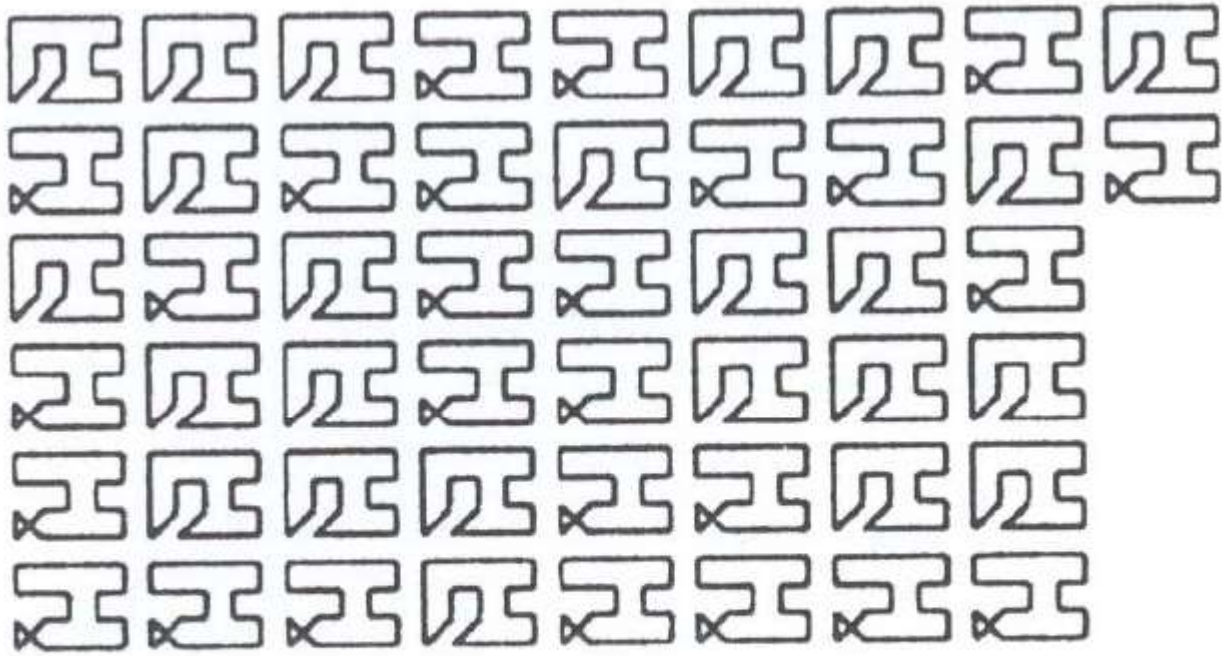
Idea intuitiva: el problema del viajante



(30)

Source: Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

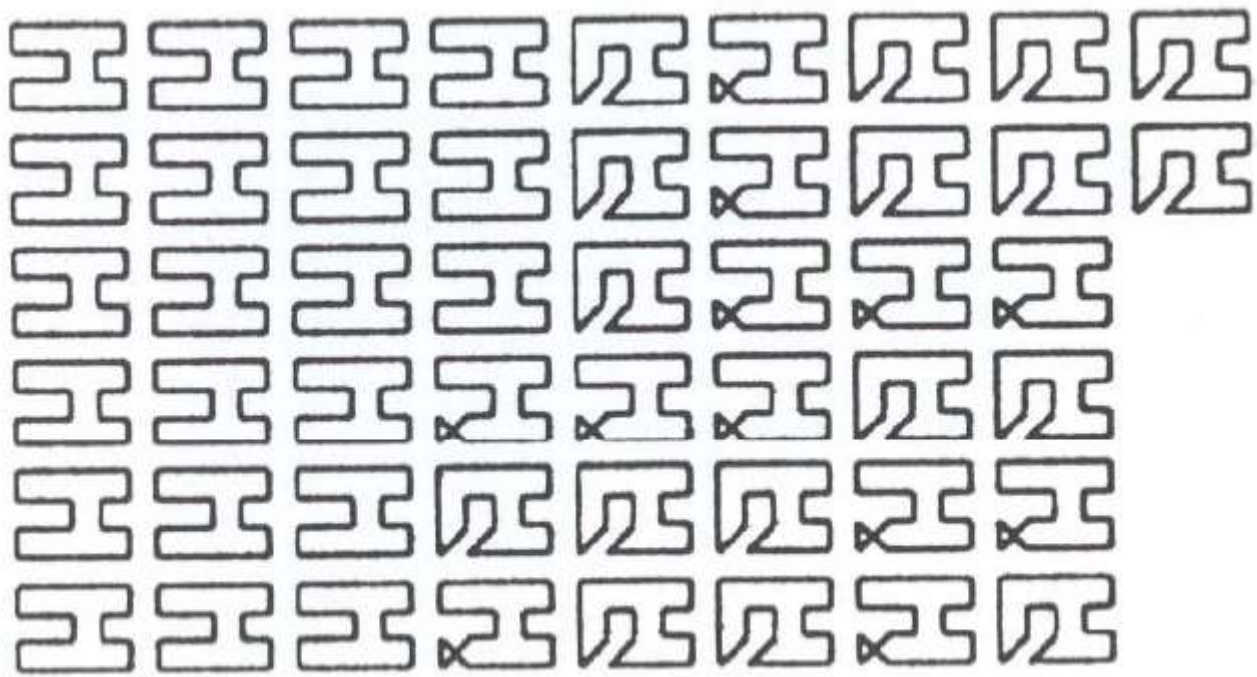
Idea intuitiva: el problema del viajante



(50)

Source: Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

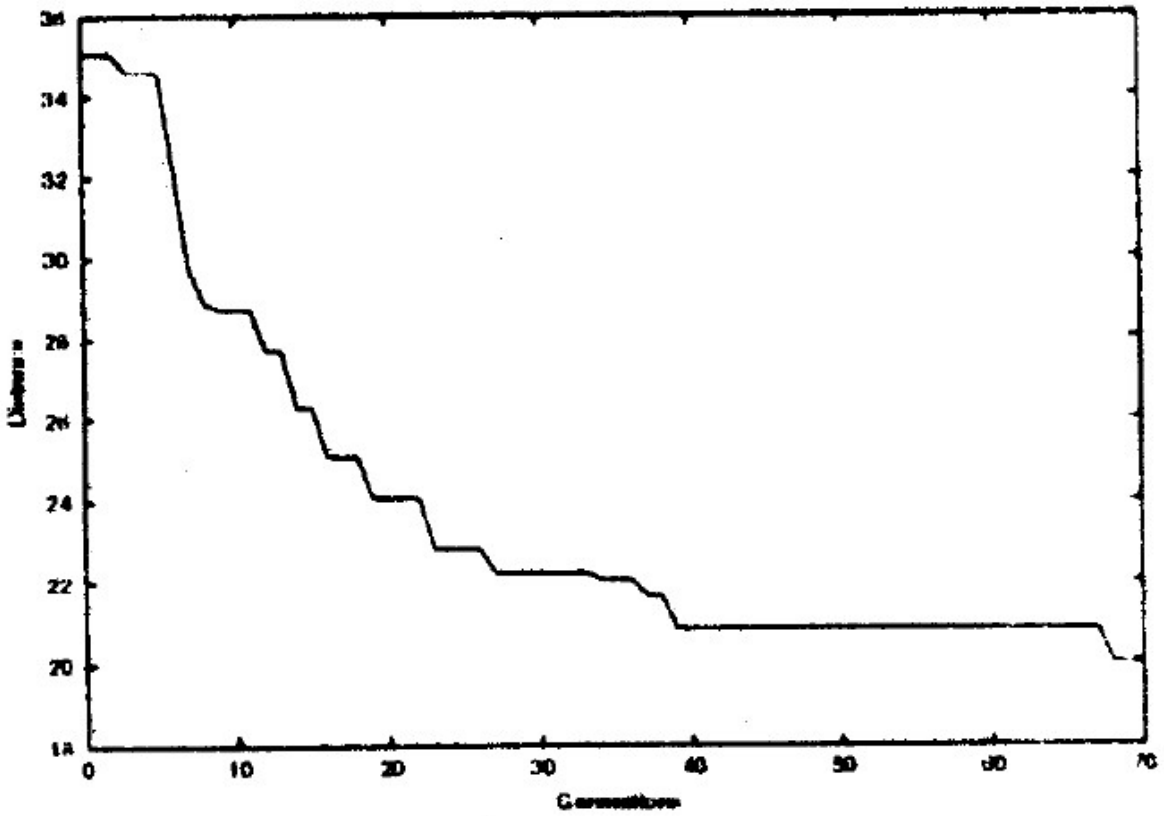
Idea intuitiva: el problema del viajante



(70)

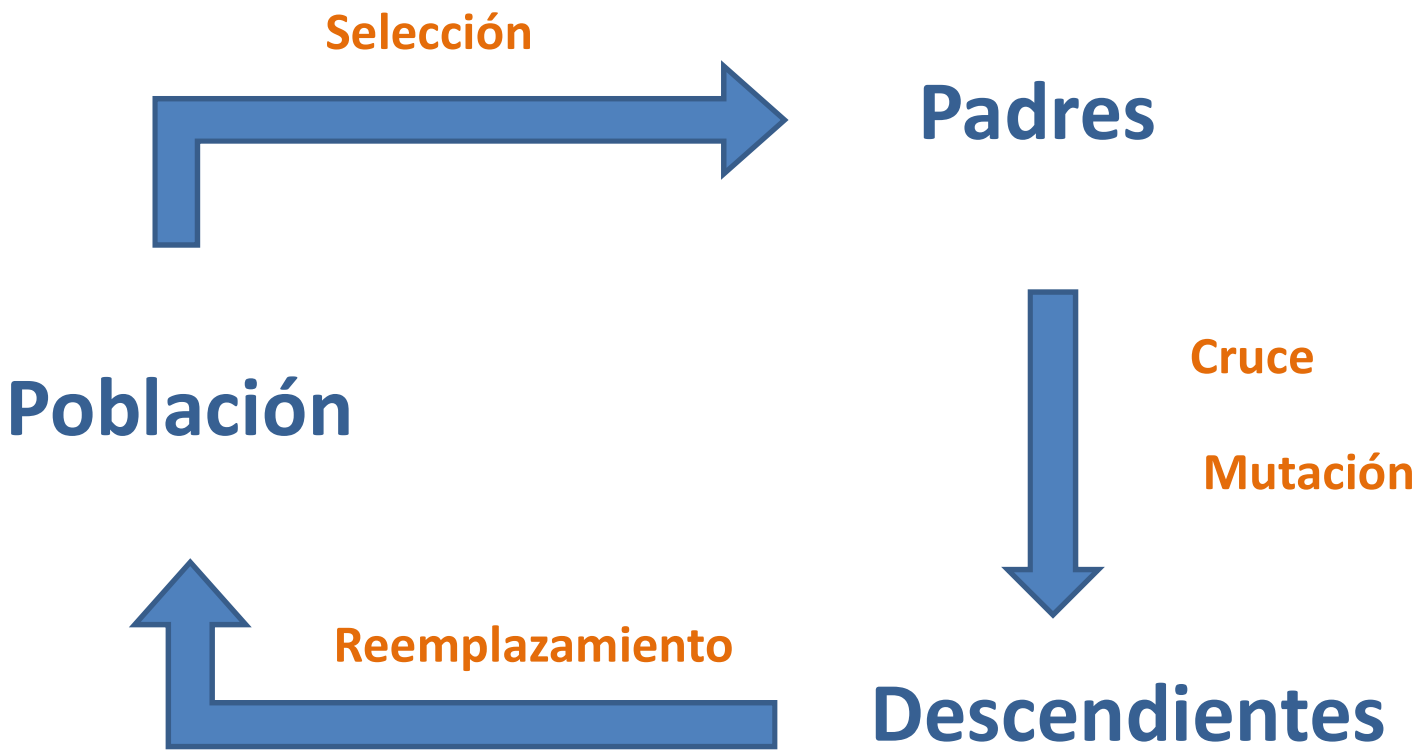
Source: Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

Idea intuitiva: el problema del viajante



Source: Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

Ciclo evolutivo



Based on Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

Estructura del algoritmo

```
Begin (1)
    t=0
    Inicializar P(t)
    Evaluar P(t)
    While (criterio de parada) do
    Begin(2)
        t=t+1
        Selecciona P(t) from P(t-1)
        Recombinar P(t)
        Mutar P(t)
        Evaluar P(t)
    Final (2)
Final (1)
```


Source: Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

Pseudocódigo (algoritmo básico)

Algorithm 20 *The Genetic Algorithm (GA)*

```
1:  popsize ← desired population size                    ▷ This is basically λ. Make it even.
2:  P ← {}
3:  for popsize times do
4:      P ← P ∪ {new random individual}
5:  Best ← □
6:  repeat
7:      for each individual P_i ∈ P do
8:          AssessFitness(P_i)
9:          if Best = □ or Fitness(P_i) > Fitness(Best) then
10:             Best ← P_i
11:     Q ← {}                          ▷ Here's where we begin to deviate from (μ, λ)
12:     for popsize/2 times do
13:         Parent P_a ← SelectWithReplacement(P)
14:         Parent P_b ← SelectWithReplacement(P)
15:         Children C_a, C_b ← Crossover(Copy(P_a), Copy(P_b))
16:         Q ← Q ∪ {Mutate(C_a), Mutate(C_b)}
17:     P ← Q                                                ▷ End of deviation
18: until Best is the ideal solution or we have run out of time
19: return Best
```


Source: Luke, Sean (2014) Essentials of Metaheuristics. George Mason Univesity https://cs.gmu.edu/~sean/book/metaheuristics/

Pasos para diseñar un algoritmo genético

- Diseño de la representación
- Decidir como inicializar la población
- Diseñar la correspondencia entre genotipo y fenotipo
- Diseñar como evaluar la fitness de un individuo
- Diseñar el operador mutación adecuado
- Diseñar el operador de cruce apropiado
- Diseñar el mecanismo de selección
- Diseñar el proceso de reemplazamiento
- Definir la condición de parada

Source: Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

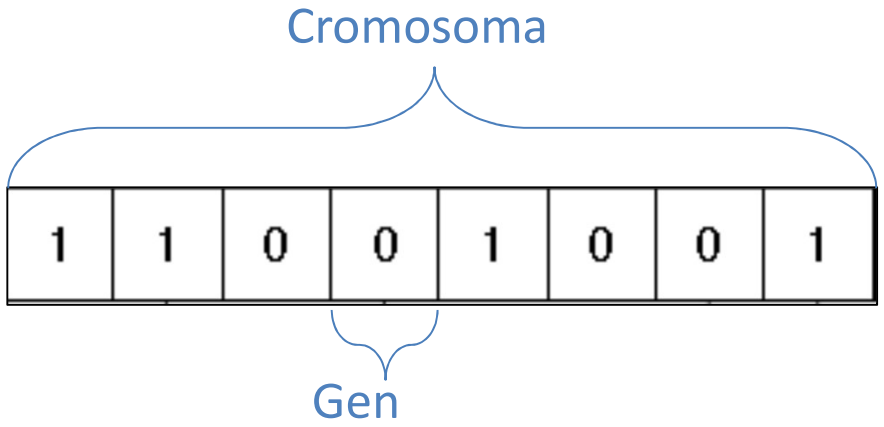
Representación

- Cada individuo (solución) debe representarse de acuerdo al problema a solucionar
- **Genotipo o genoma:** la estructura de datos de un individuo tal y como se utiliza durante el proceso de reproducción (cruce y mutación) (*breeding*)
- **Fenotipo:** como opera el individuo durante la evaluación de la fitness

Source: Luke, Sean (2014) Essentials of Metaheuristics. George Mason Univesity https://cs.gmu.edu/~sean/book/metaheuristics/

Codificación binaria

- **Cromosoma**: un genotipo en forma de un vector de longitud fija
- **Gen**: una posición particular en un cromosoma
- **Alelo**: un valor particular de un gen

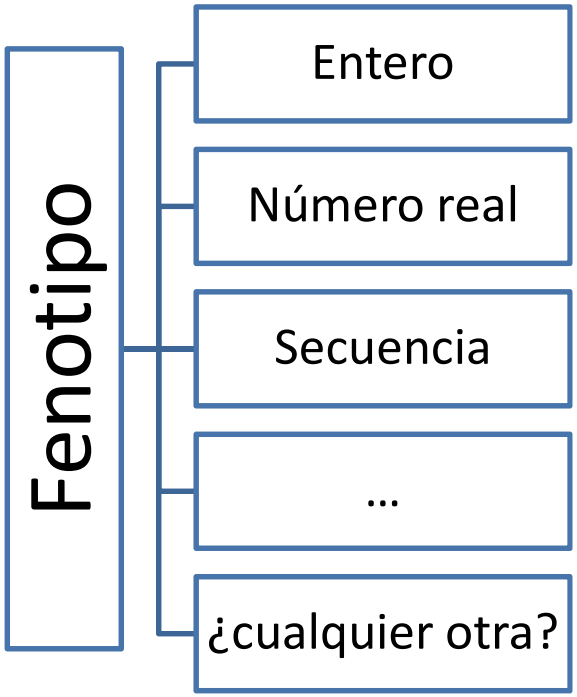


Source: Luke, Sean (2014) Essentials of Metaheuristics. George Mason Univesity https://cs.gmu.edu/~sean/book/metaheuristics/

Codificación binaria

Genotipo:

Representación binaria de 8 bits

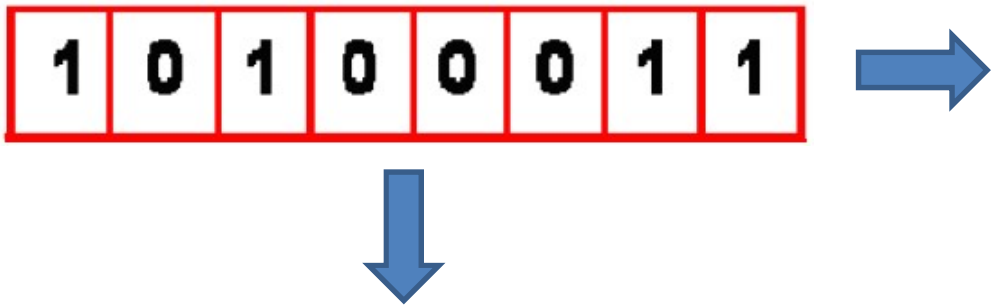


Source: Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

Codificación binaria

Genotipo:
Representación binaria de 8 bits

Fenotipo:
Entero



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

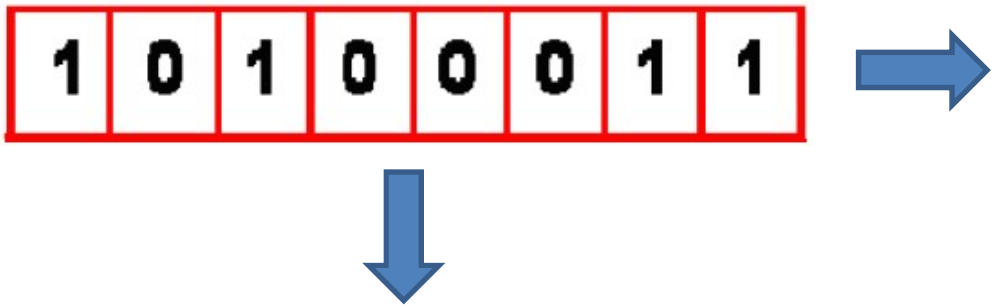
$$128 + 32 + 2 + 1 = 163$$

Source: Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

Codificación binaria

Genotipo:
Representación binaria de 8 bits

Fenotipo:
Número real entre 2.5 y 20.5



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

Source: Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

Codificación real

- Otra forma de representación natural de muchos problemas es utilizar números reales como genes
- El cromosoma de los individuos viene representado como un vector fijo de números reales

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

- Normalmente el fenotipo es un número real obtenido de una función de acuerdo a la siguiente forma:

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

Source: Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

Codificación permutacional o de orden

- Cuando una solución es una secuencia, una posible forma de ordenar algo, los individuos se representan mediante permutaciones

7	3	6	8	2	4	1	5

- Se utiliza esta representación en los problemas de scheduling
- El ejemplo más popular en el que se utiliza es en el problema del viajante, en el que a cada ciudad se asigna un único entero entre 1 y n
- Este tipo de codificación requiere operador especiales para garantizar que el resultado es también una permutación

Source: Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

Inicialización

- Si es posible uniforme a lo largo de todo el espacio de soluciones
  - En el caso de codificación binaria: 0 ó 1 con probabilidad 0.5 para cada gen
  - Con codificación real: uniforme en los diferentes intervalos
- Soluciones optimizadas de heurísticas previas no suele ser una buena idea

Source: Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

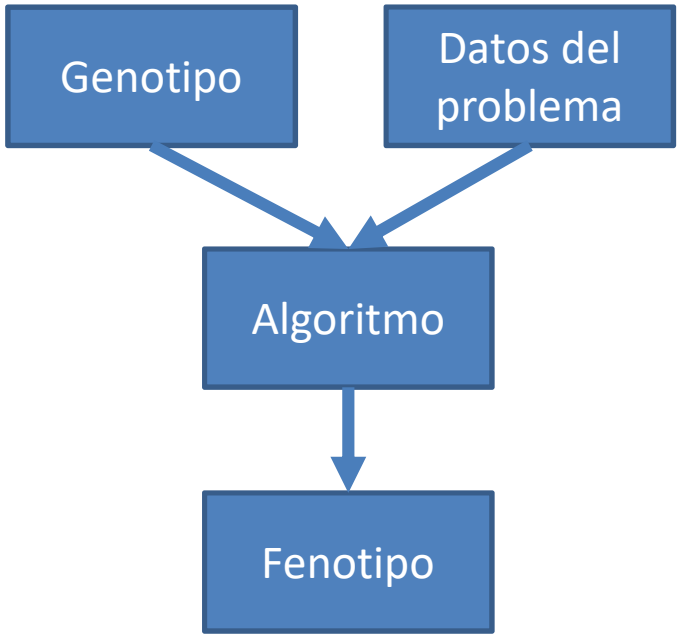
Evaluación de la función de fitness

- **Fitness:** calidad de la solución
- **Fitness landscape:** función objetivo
- Normalmente es el paso más costoso computacionalmente en aplicaciones reales
- Puede ser una subrutina, un simulador o cualquier proceso externo (e.g. experimentos con robots,…)
- En algunas situaciones se utilizan funciones aproximadas para tratar de reducir el coste computacional
- En caso de restricciones, se pueden integrar como coste adicional en la función de fitness
- Se pueden utilizar funciones multiobjetivo

Source: Luke, Sean (2014) Essentials of Metaheuristics. George Mason Univesity
https://cs.gmu.edu/~sean/book/metaheuristics/
Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

Correspondencia entre genotipo y fenotipo

- En algunos caso el proceso es simple
- Pero en otros, el genotipo puede ser un conjunto de parámetros de otro algoritmo o una simulación que debe ser ejecutada para obtener el fenotipo



Source: Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

Ciclo evolutivo. Diseño de un algoritmo genético(I)

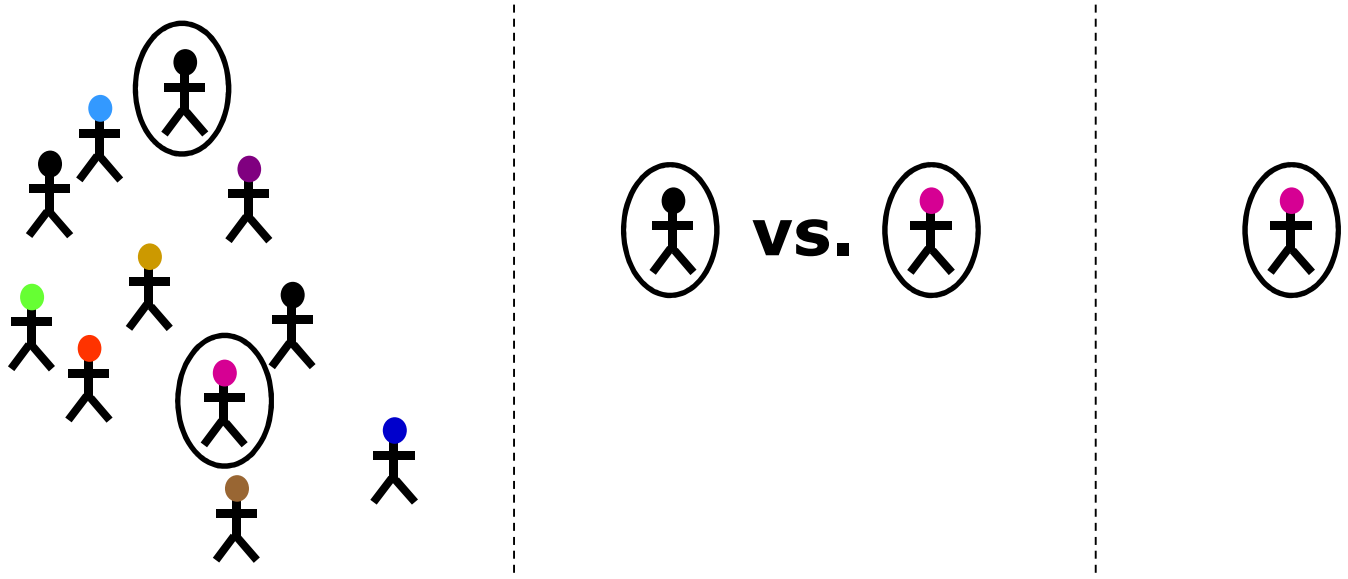
Estrategia de selección

- **Selección:** elección de los individuos basada en la fitness
- Los mejores individuos deben tener una probabilidad mayor de ser elegidos como padres
- Pero los individuos no tan buenos deben tener también opción de reproducirse. Es posible que incluyan material genético útil
- Esta idea define **la presión de selección** del proceso, el grado en el que la reproducción es dirigida por los mejores individuos

Source: Luke, Sean (2014) Essentials of Metaheuristics. George Mason Univesity https://cs.gmu.edu/~sean/book/metaheuristics/
Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

Mecanismos de selección populares

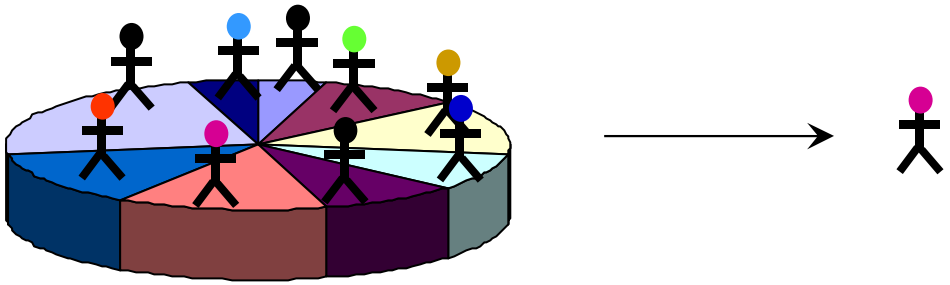
- **Torneo aleatorio (Random tournament)**



Source: Galan, J. M. & Izquierdo, L. R. (2005). Appearances Can Be Deceiving: Lessons Learned Re-Implementing Axelrod's 'Evolutionary Approach to Norms'. Journal of Artificial Societies and Social Simulation 8(3)2. http://jasss.soc.surrey.ac.uk/8/3/2.html

Mecanismos de selección populares

- **Selección por ruleta (Roulette wheel)**

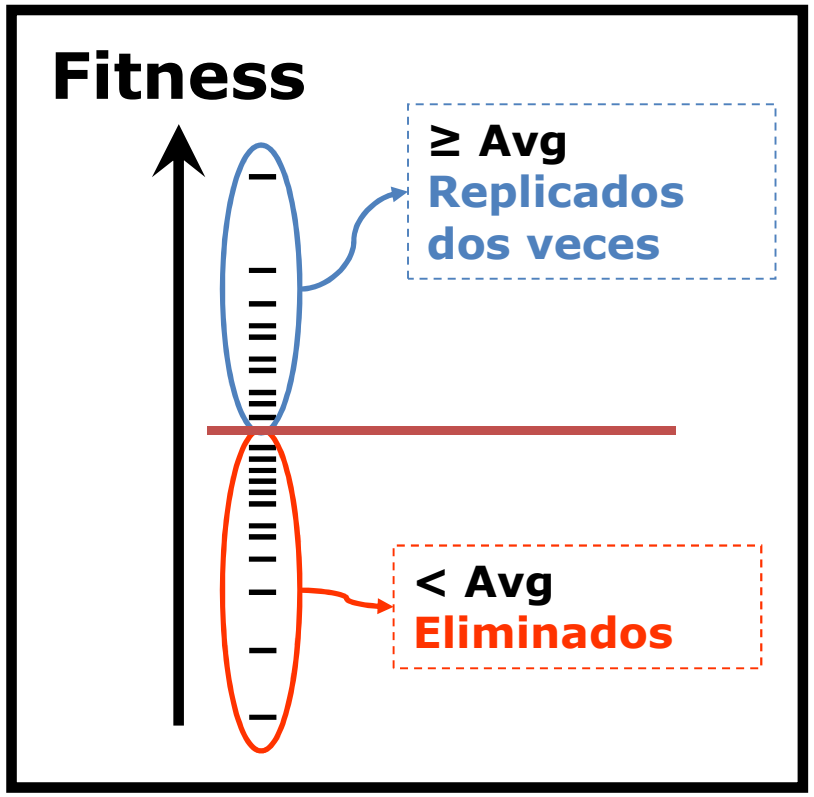


Source: Galan, J. M. & Izquierdo, L. R. (2005). Appearances Can Be Deceiving: Lessons Learned Re-Implementing Axelrod's 'Evolutionary Approach to Norms'. Journal of Artificial Societies and Social Simulation 8(3)2. http://jasss.soc.surrey.ac.uk/8/3/2.html

Mecanismos de selección populares

- **Selección media (Average selection)**

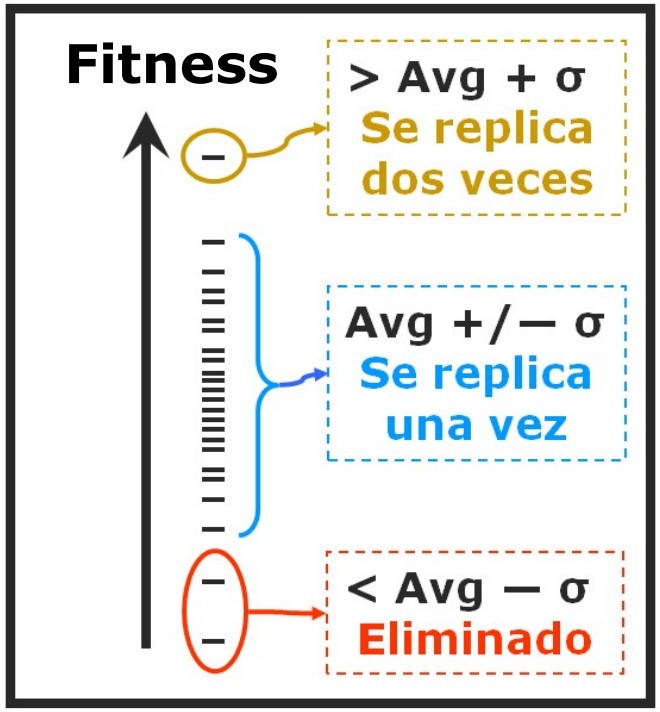
  Para mantener la población constante es mejor utilizar la mediana en lugar de la media



Source: Galan, J. M. & Izquierdo, L. R. (2005). Appearances Can Be Deceiving: Lessons Learned Re-Implementing Axelrod's 'Evolutionary Approach to Norms'. Journal of Artificial Societies and Social Simulation 8(3)2. http://jasss.soc.surrey.ac.uk/8/3/2.html

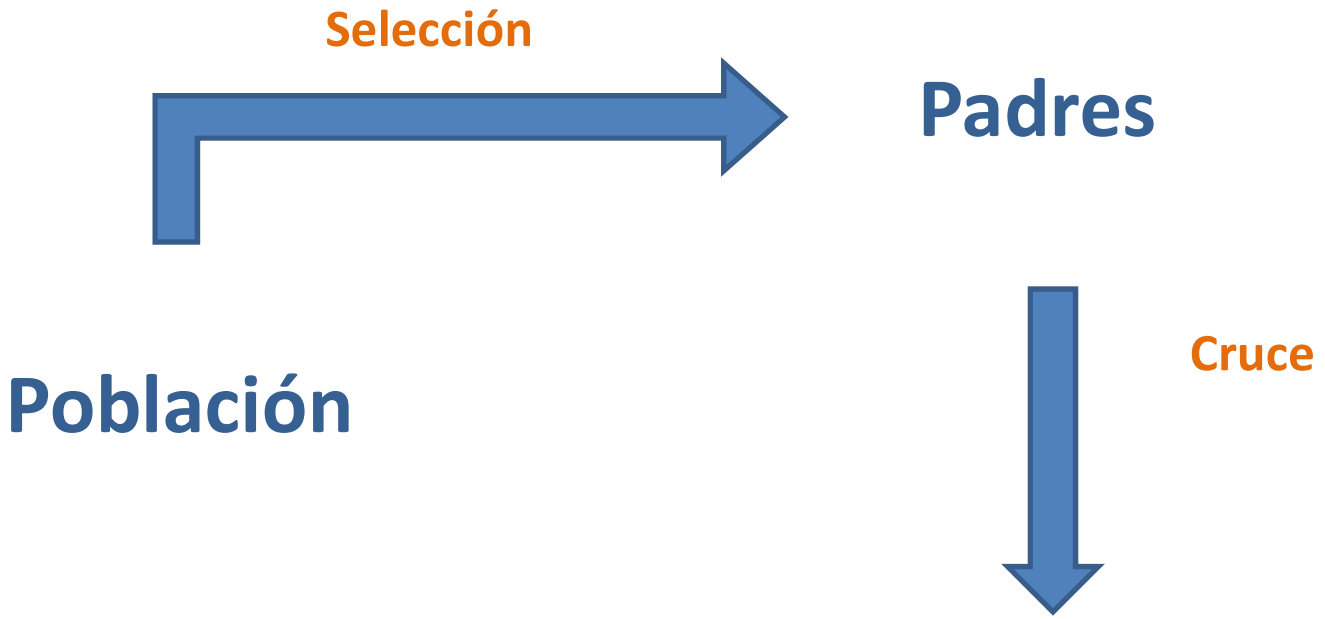
Mecanismos de selección populares

- **Otros métodos de truncamiento**



Source: Galan, J. M. & Izquierdo, L. R. (2005). Appearances Can Be Deceiving: Lessons Learned Re-Implementing Axelrod's 'Evolutionary Approach to Norms'. Journal of Artificial Societies and Social Simulation 8(3)2. http://jasss.soc.surrey.ac.uk/8/3/2.html

Ciclo evolutivo. Diseño de un algoritmo genético(II)



Based on Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

Operador cruce

- **Recombinación o cruce**: un sistema especial de modificación de soluciones que normalmente conlleva elegir dos padres, intercambiar secciones de ambos y (normalmente) producir dos hijos. Normalmente se considera un sistema de reproducción "sexual".
- Se pueden utilizar diferentes operadores cruce en el mismo algoritmo
- Aspectos relevantes:
  - Los hijos deben heredar características de cada padre
  - Es dependiente de la representación del genotipo
  - La recombinación debe producir cromosomas válidos
  - Normalmente se utiliza asociado a una cierta probabilidad (P_c entre 0.6 y 0.9). No todos los padres son recombinados ya que el operador cruce puede ser destructivo

Source: Luke, Sean (2014) Essentials of Metaheuristics. George Mason Univesity https://cs.gmu.edu/~sean/book/metaheuristics/
Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

Cruce para codificación binaria

- **Cruce en un punto**

Algorithm 23 *One-Point Crossover*

1: $\vec{v} \leftarrow$ first vector $\langle v_1, v_2, \ldots v_l \rangle$ to be crossed over
2: $\vec{w} \leftarrow$ second vector $\langle w_1, w_2, \ldots w_l \rangle$ to be crossed over

3: $c \leftarrow$ random integer chosen uniformly from 1 to l inclusive
4: **if** $c \neq 1$ **then**
5: **for** i from 1 to $c-1$ **do**
6: Swap the values of v_i and w_i
7: **return** $\vec{v}$ and $\vec{w}$

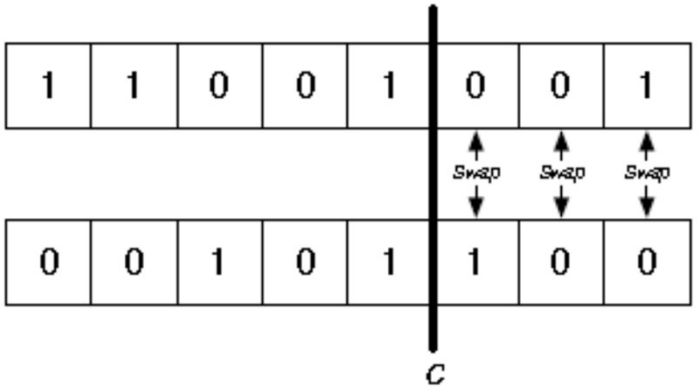


Figure 9 One-Point Crossover.

Source: Luke, Sean (2014) Essentials of Metaheuristics. George Mason Univesity https://cs.gmu.edu/~sean/book/metaheuristics/

Cruce para codificación binaria

- **Cruce en dos puntos**

Algorithm 24 *Two-Point Crossover*

1: $\vec{v} \leftarrow$ first vector $\langle v_1, v_2, ...v_l \rangle$ to be crossed over
2: $\vec{w} \leftarrow$ second vector $\langle w_1, w_2, ...w_l \rangle$ to be crossed over

3: $c \leftarrow$ random integer chosen uniformly from 1 to l inclusive
4: $d \leftarrow$ random integer chosen uniformly from 1 to l inclusive
5: **if** $c > d$ **then**
6: Swap c and d
7: **if** $c \neq d$ **then**
8: **for** i from c to $d - 1$ **do**
9: Swap the values of v_i and w_i
10: **return** $\vec{v}$ and $\vec{w}$

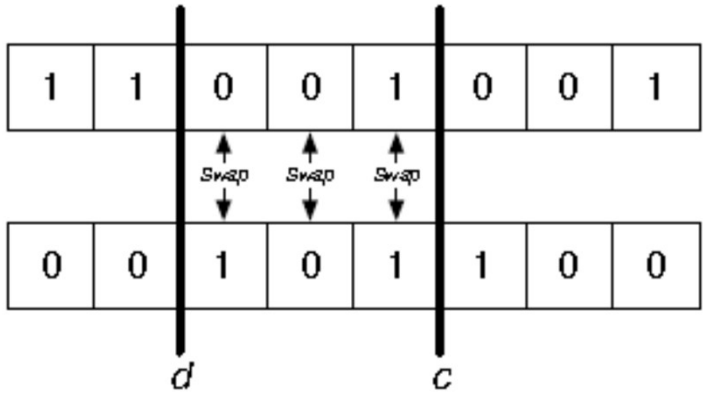


Figure 10 Two-Point Crossover.

[24]We can generalize two-point crossover into a **Multi-Point Crossover**: pick n random points and sort them smallest first: $c_1, c_2, ..., c_n$. Now swap indexes in the region between c_1 and c_2, and between c_3 and c_4, and likewise c_5 and c_6, etc.

Source: Luke, Sean (2014) Essentials of Metaheuristics. George Mason Univesity https://cs.gmu.edu/~sean/book/metaheuristics/

Cruce para codificación binaria

- **Cruce uniforme**

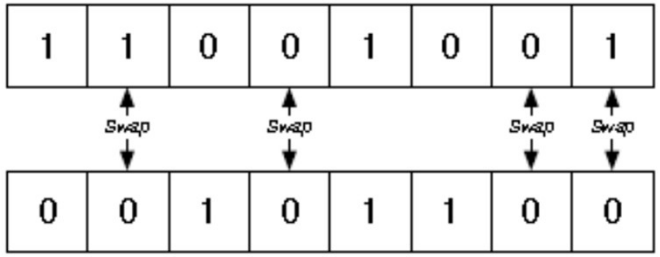


Figure 11 Uniform Crossover.

Algorithm 25 *Uniform Crossover*

1: $p \leftarrow$ probability of swapping an index ▷ Often p is set to $1/l$. At any rate, $p \leq 0.5$
2: $\vec{v} \leftarrow$ first vector $\langle v_1, v_2, ...v_l \rangle$ to be crossed over
3: $\vec{w} \leftarrow$ second vector $\langle w_1, w_2, ...w_l \rangle$ to be crossed over

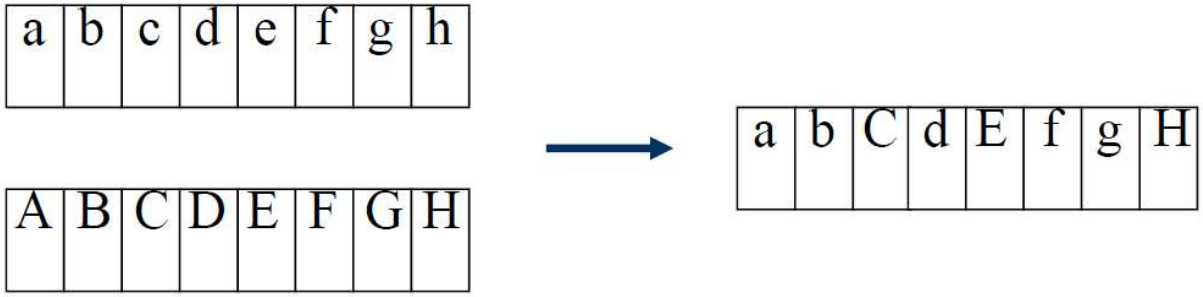
4: **for** i from 1 to l **do**
5: **if** $p \geq$ random number chosen uniformly from 0.0 to 1.0 inclusive **then**
6: Swap the values of v_i and w_i
7: **return** $\vec{v}$ and $\vec{w}$

[25]The original uniform crossover assumed $p = 1/2$, and was first proposed in David Ackley, 1987, *A Connectionist Machine for Genetic Hillclimbing*, Kluwer Academic Publishers. The more general form, for arbitrary p, is sometimes called *parameterized* uniform crossover.

Source: Luke, Sean (2014) Essentials of Metaheuristics. George Mason Univesity https://cs.gmu.edu/~sean/book/metaheuristics/

Cruce para codificación real

- **Cruce uniforme**



Algorithm 25 *Uniform Crossover*

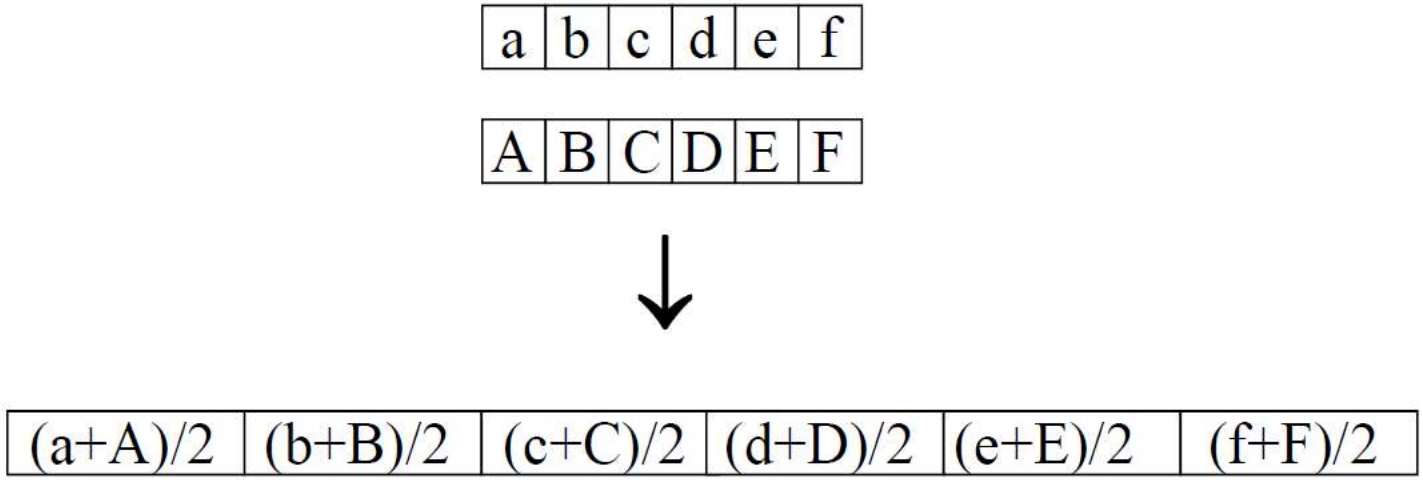
1: $p \leftarrow$ probability of swapping an index ▷ Often p is set to $1/l$. At any rate, $p \leq 0.5$
2: $\vec{v} \leftarrow$ first vector $\langle v_1, v_2, ...v_l \rangle$ to be crossed over
3: $\vec{w} \leftarrow$ second vector $\langle w_1, w_2, ...w_l \rangle$ to be crossed over

4: **for** i from 1 to l **do**
5: **if** $p \geq$ random number chosen uniformly from 0.0 to 1.0 inclusive **then**
6: Swap the values of v_i and w_i
7: **return** $\vec{v}$ and $\vec{w}$

Source: Luke, Sean (2014) Essentials of Metaheuristics. George Mason Univesity
https://cs.gmu.edu/~sean/book/metaheuristics/
Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

Cruce para codificación real

- **Cruce aritmético**

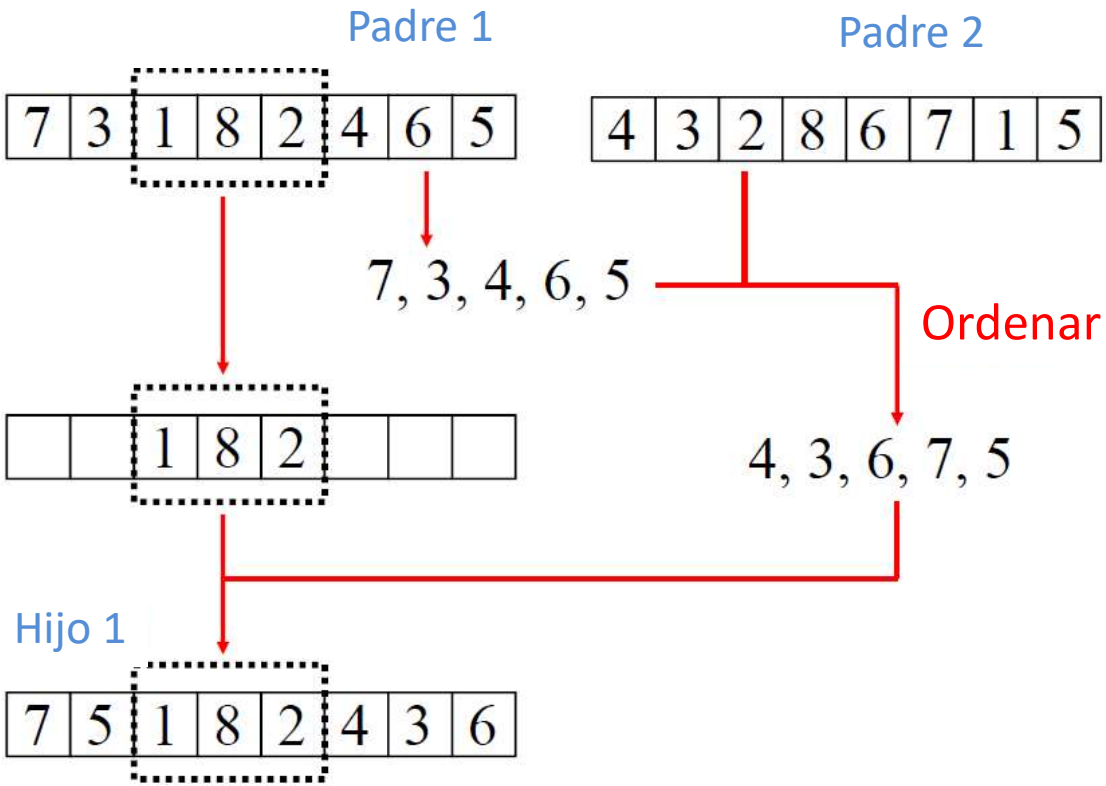


Hay muchísimas opciones para cruce real

Source: Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

Cruce para representación permutacional

- **Cruce OX (order)**



Source: Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

Cruce para representación permutacional

- **Cruce PMX (partially matched)**
  - Selecciona aleatoriamente un segmento central y establece una correspondencia entre los genes de cada padre
  - Cada hijo contiene el elemento central de uno de los padres y el máximos número de posiciones correspondientes al otro padre. Si no es posible la asignación, se realiza a partir de la correspondencia

Padre 1 = (1 2 3 | 4 5 6 7 | 8 9)
Padre 2 = (4 5 3 | 1 8 7 6 | 9 2)
Hijo' 1 = (* * * | 1 8 7 6 | * *)
Hijo' 2 = (* * * | 4 5 6 7 | * *)
Correspondencia : (1-4, 8-5, 7-6, 6-7)
Hijo 1 = (1-4 2 3 | 1 8 7 6 | 8-5 9) = (4 2 3 | 1 8 7 6 | 5 9)
Hijo 2 = (4-1 5-8 3 | 4 5 6 7 | 9 2) = (1 8 3 | 4 5 6 7 | 9 2)

Source: Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

Cruce para representación permutacional

- **Cruce CX (cycle)**
  - Elige aleatoriamente una posición y sigue las restricciones creadas hasta que encuentres un ciclo
  - Repite el proceso hasta generar un hijo

Padre 1 = (1 2 3 4 5 6 7 8)

Padre 2 = (2 4 6 8 7 5 3 1)

Elija aleatoriamente entre 1 y 2 (la primera posición de ambos vectores). Imagine que sale 1

$$(1\ *\ *\ *\ *\ *\ *\ *) \rightarrow (1\ *\ *\ *\ *\ *\ *\ 8) \rightarrow (1\ *\ *\ 4\ *\ *\ *\ 8) \rightarrow (1\ 2\ *\ 4\ *\ *\ *\ 8)$$

Se ha encontrado un ciclo.

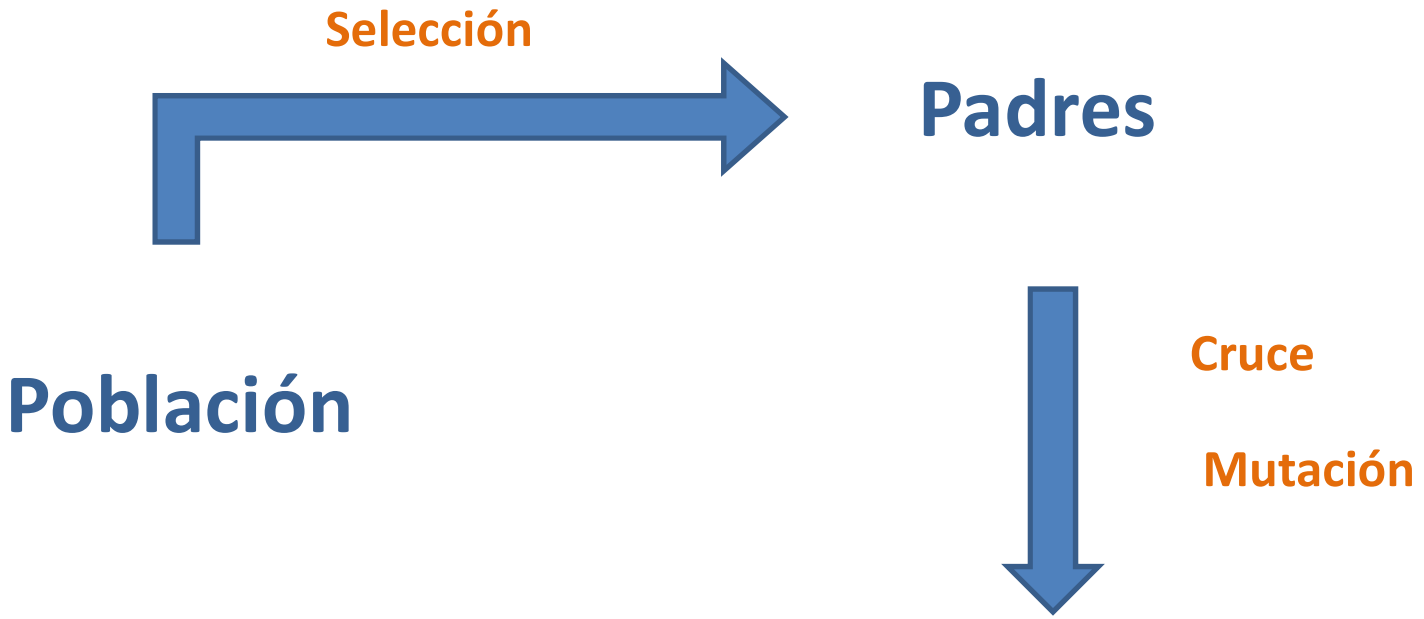
Elija aleatoriamente entre 3 y 6 (la primera posición libre de ambos vectores). Imagine un 6

Hijo = (1 2 6 4 * * * 8)

Hijo = (1 2 6 4 7 5 3 8)

Source: Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

Ciclo evolutivo. Diseño de un algoritmo genético(III)



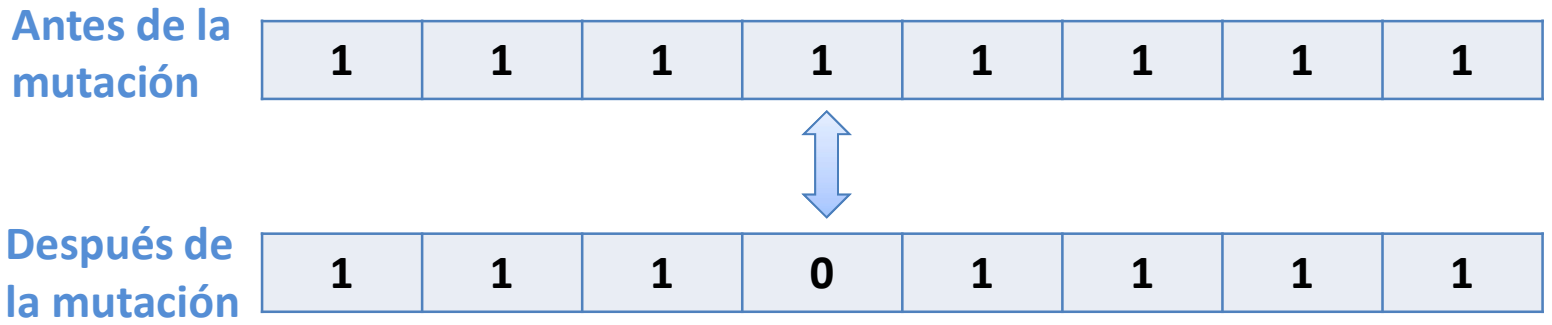
Operador mutación

- **Mutación:** modificación. A veces se considera reproducción "asexual"
- Se pueden utilizar diferentes operadores en el mismo algoritmo
- Aspectos relevantes
  - Cualquier punto del espacio de búsqueda debería poder ser alcanzado
  - La cantidad de mutación se debe controlar
  - La mutación debe producir cromosomas válidos
  - Se debe utilizar como una pequeña probabilidad para cada nuevo hijo tras el operador cruce (incluyendo también los padres no recombinados)

Source: Luke, Sean (2014) Essentials of Metaheuristics. George Mason Univesity https://cs.gmu.edu/~sean/book/metaheuristics/
Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

Mutación para codificación binaria

- Probabilidad P_m para cada gen

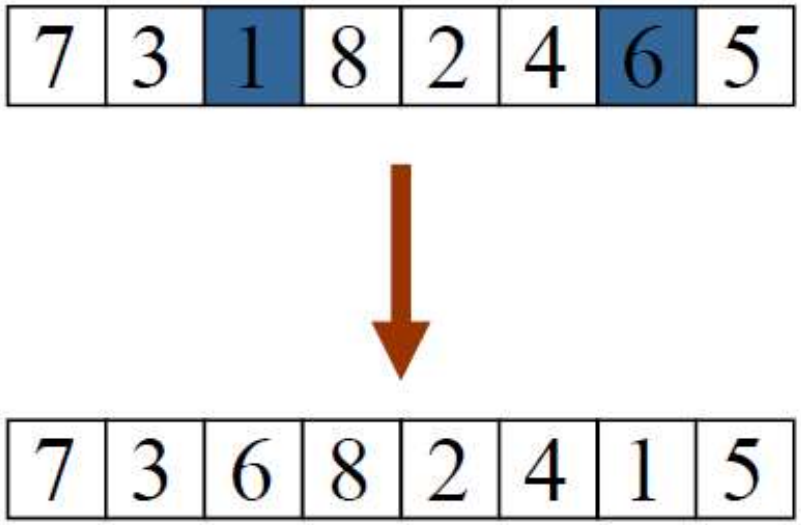


Mutación para codificación real

- Perturbación aleatoria
- Normalmente mediante ruido blanco $N(0,\sigma)$
  - 0 es la media
  - σ es la desviación estándar
  - $x'_i = x_i + N(0,\sigma_i)$ para cada parámetro

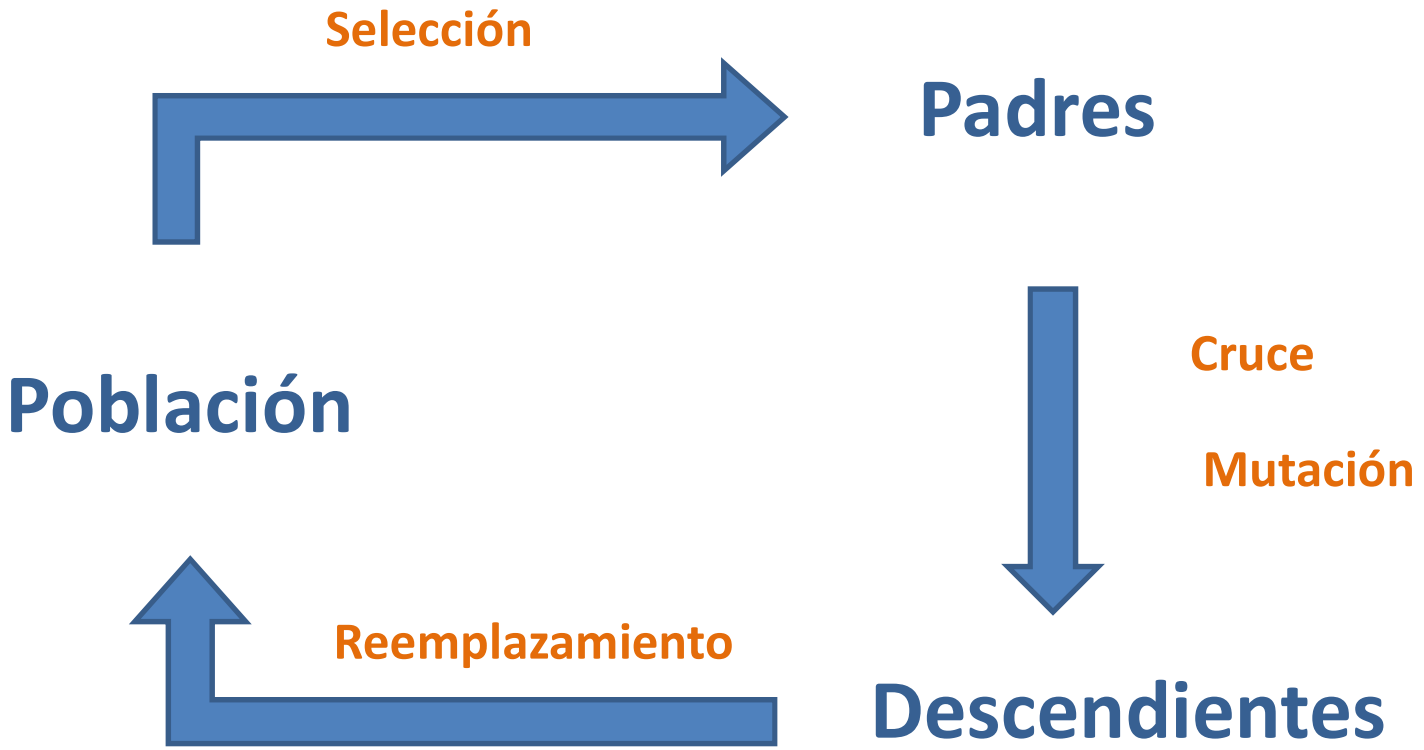
Mutación para codificación permutacional

- Selecciona aleatoriamente dos posiciones e intercámbialas (también inserción o subinversión – [ver tema recocido simulado])



Source: Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

Ciclo evolutivo. Diseño de un algoritmo genético(IV)



Estrategia de reemplazamiento

- La estrategia de reemplazamiento determina los individuos que son introducidos o eliminados de la población
- Generacional
  - En cada iteración una nueva población completa reemplaza a la anterior
- Estacionaria (steady-state)
  - Una alternativa a la aproximación generacional (la más frecuente) es actualizar la población individuo a individuo (o en pares) en lugar de a todos de una vez

Source: Luke, Sean (2014) Essentials of Metaheuristics. George Mason Univesity https://cs.gmu.edu/~sean/book/metaheuristics/
Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

Estrategia de reemplazamiento

- Una buena idea en el caso de la aproximación generacional es incluir una estrategia de “elitismo” para no perder la mejor solución encontrada
- **Elitismo**: incluir directamente en la siguiente población el mejor individuo (o un truncamiento de n mejores) de la anterior sin cruzar ni mutar. Estos individuos se llaman élites

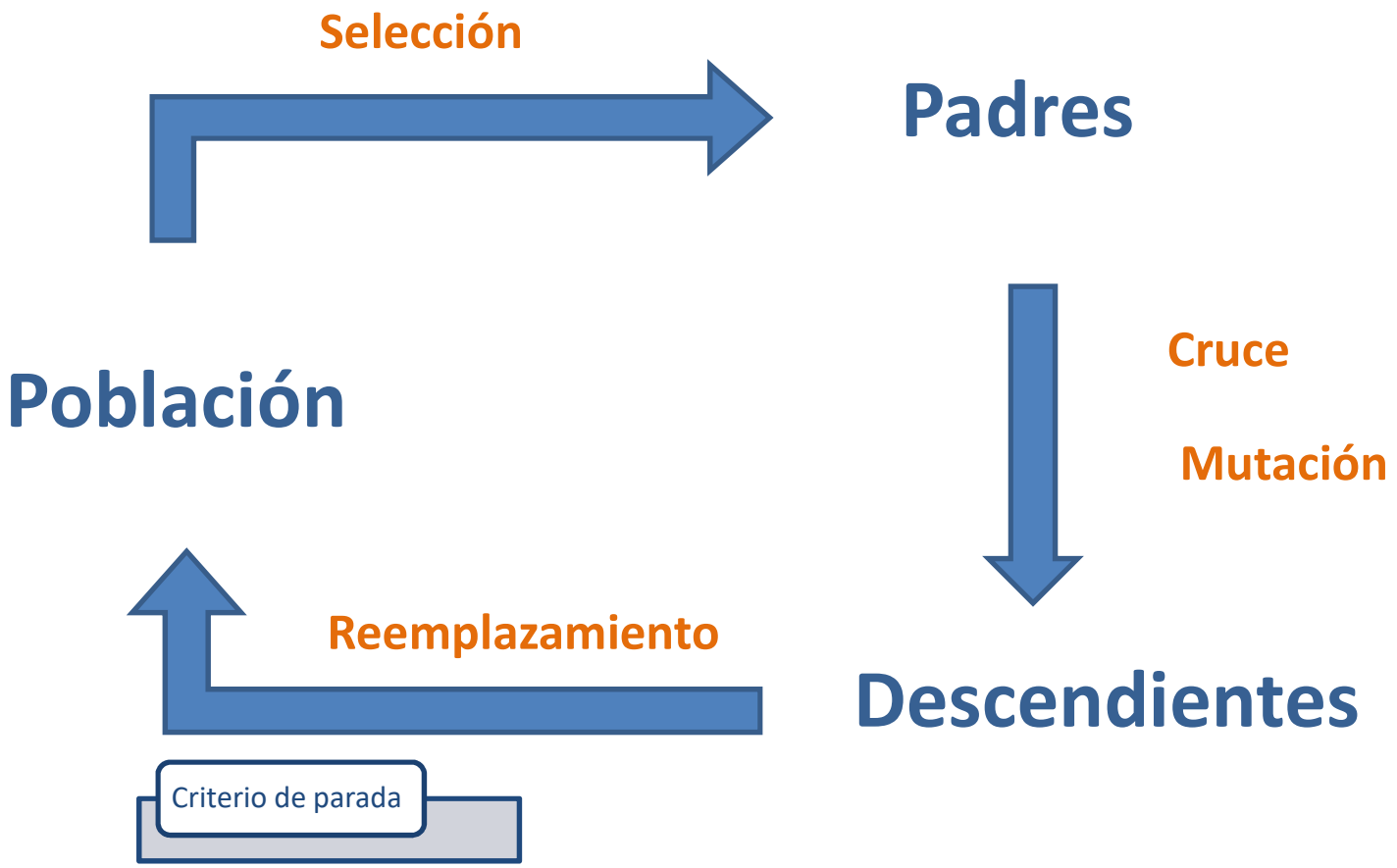
Source: Luke, Sean (2014) Essentials of Metaheuristics. George Mason Univesity https://cs.gmu.edu/~sean/book/metaheuristics/
Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

Criterio de parada

- Si se encuentra el óptimo (requiere conocer cuál es el óptimo a priori, que no suele ser el caso)
- Recursos computacionales limitados. Limitar el número máximo de iteraciones o el tiempo de ejecución
- Después de un tiempo o iteraciones sin mejora

Source: Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

Ciclo evolutivo. Diseño de un algoritmo genético(V)



Based on Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

¡Precaución!

- Como con cualquier algoritmo de optimización aleatoria si es posible utiliza más de una ejecución para sacar conclusiones
  - Utiliza medidas estadísticas (media, mediana…)
  - Haz suficientes replicaciones como para obtener valores significativos
- No ajustes los parámetros del algoritmo en contextos diferentes a los de su uso. No ajustes con casos sencillos si lo utilizarás para resolver casos complejos
- Ten claro si el objetivo es estratégico, táctico u operativo
  - Encontrar una solución buena al menos una vez de varias ejecuciones
  - Encontrar una solución buena en cada ejecución
  - Tiempo de cada ejecución…

Source: Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

Sobre la diversidad genética

- La falta de diversidad implica convergencia a algo similar a un mejor vecino
- Una alta presión selectiva reduce diversidad rápidamente
- Soluciones:
  - Introducir mecanismos de diversidad
  - Reducir la presión selectiva
  - Ejecutar el algoritmo varias veces

Source: Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

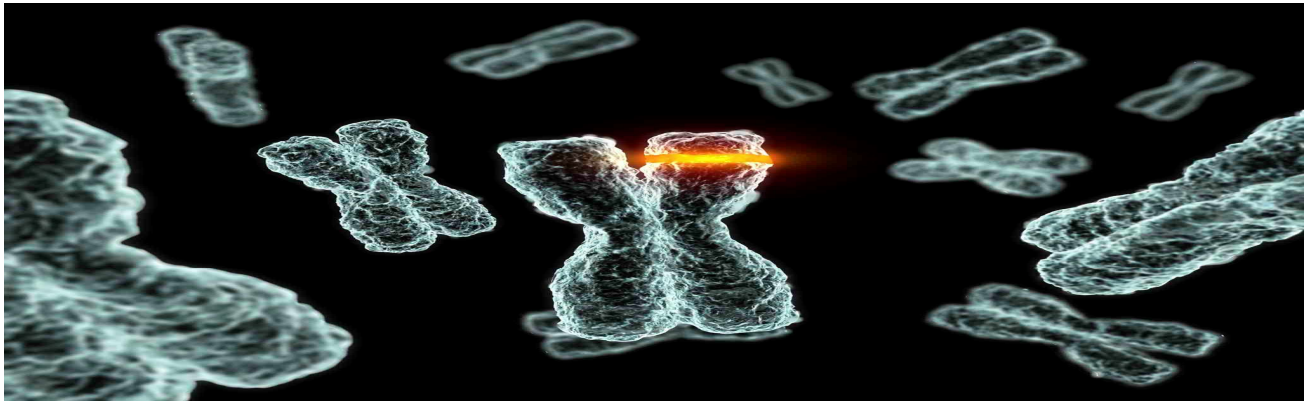
Exploración vs Explotación

- Exploración: muestrear en regiones desconocidas
  - Si exploras demasiado te acercas a la búsqueda aleatoria -> no hay convergencia, malas soluciones
- Explotación: mejorar el mejor individuo
  - Si intensificas demasiado, te acercas a los algoritmos de búsqueda local que se atascan en óptimos locales

Source: Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

Algoritmo genéticos

- Basados en una metáfora biológica: la evolución
- Muchas aplicaciones
- Muy populares
- Muy potentes



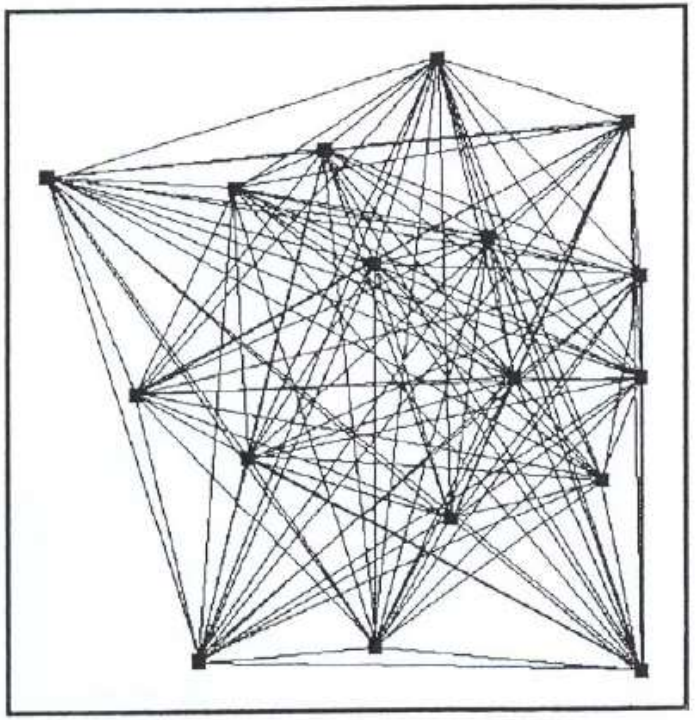
El problema del viajante

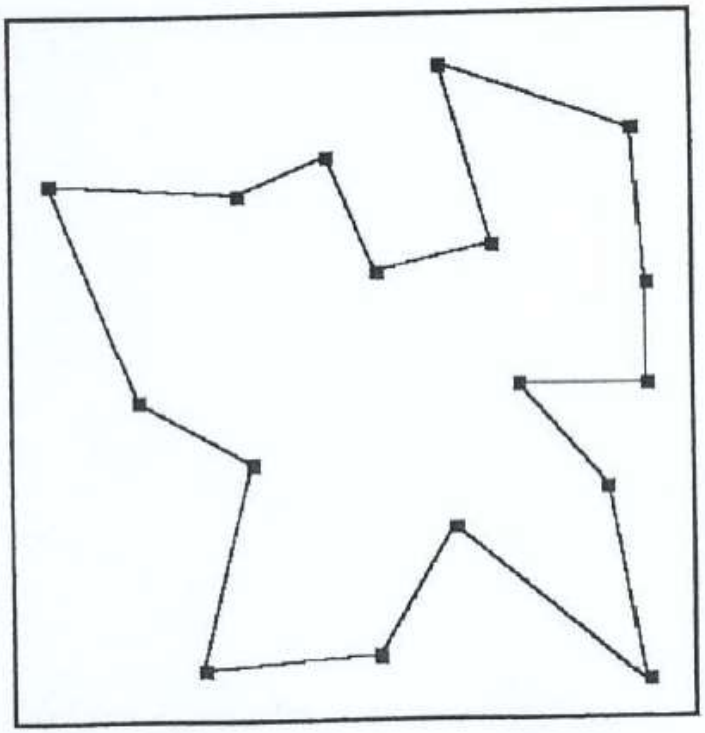
- 17 ciudades
- Representación: Codificación permutacional
- Población: 61 individuos
- Inicialización: aleatoria
- Selección: torneo aleatorios
- Reemplazamiento: generacional con elitismo (1 élite)
- Cruce: operador OX (P_c=0.6)
- Mutación: intercambio de dos genes por cromosoma P_m=0.01
- Fitness:

$$C(S) = \sum_{i=1}^{n-1} \left(D[S[i], S[i+1]] \right) + D[S[n], S[1]]$$

Source: Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

El problema del viajante



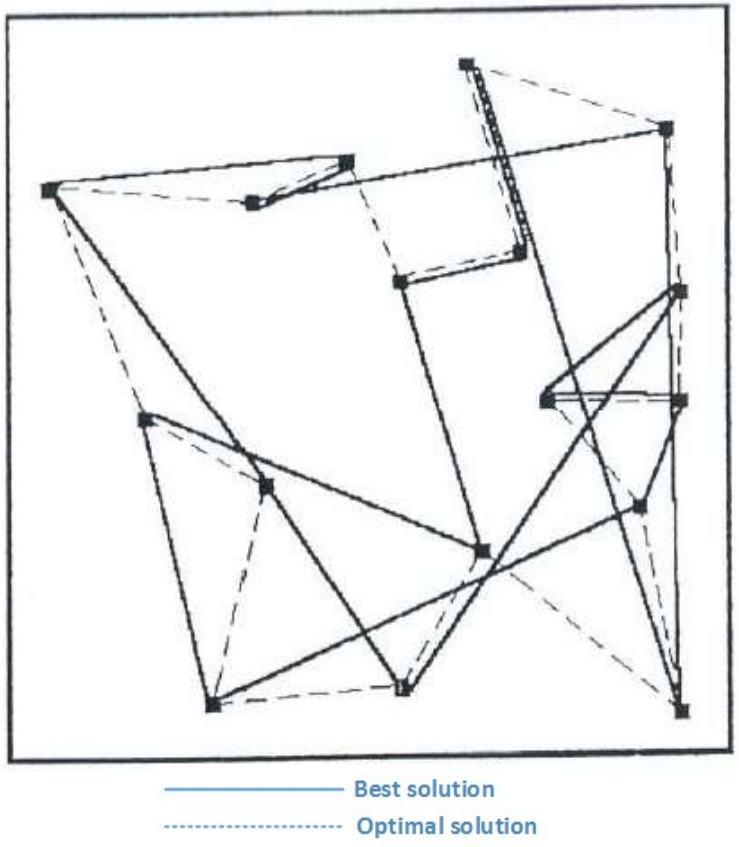


17! ≈ $3.6*10^{14}$ posibles soluciones

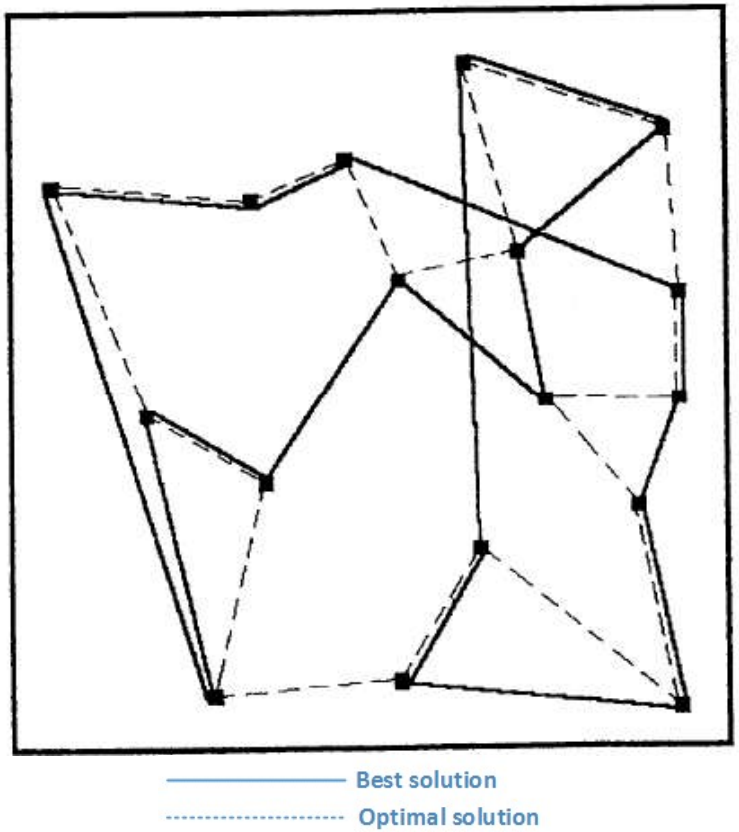
Solución óptima: 226.64

Source: Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

El problema del viajante



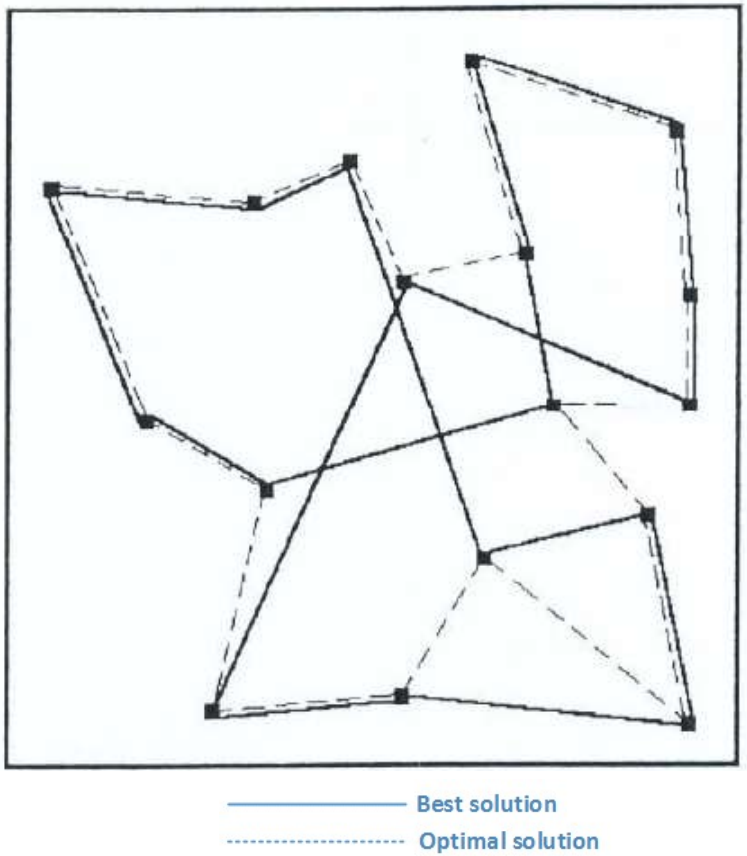
Iteración: 0 Fitness: 403.7



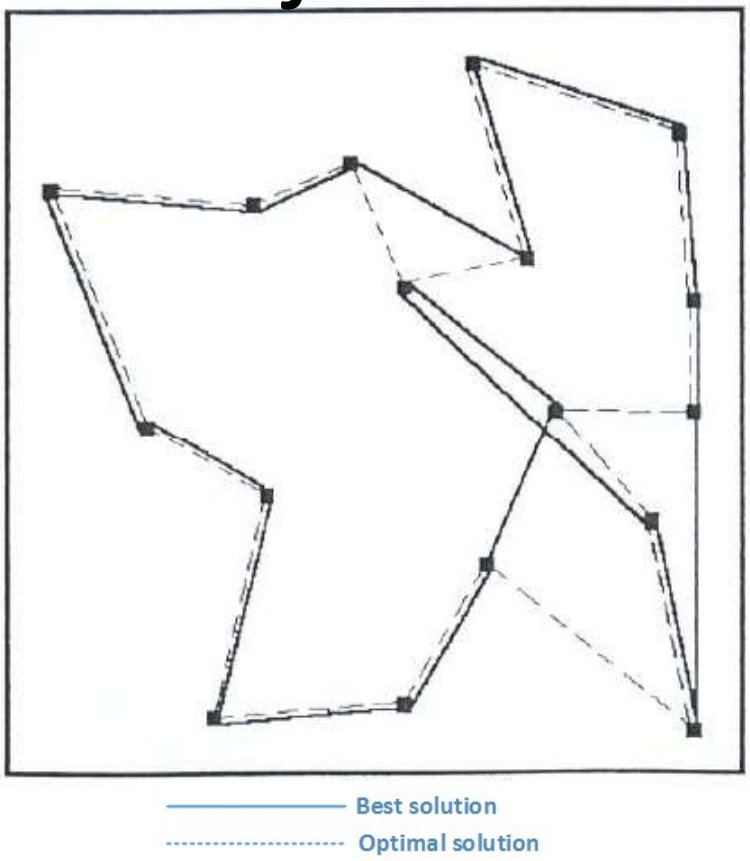
Iteración: 25 Fitness: 303.86

Source: Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

El problema del viajante



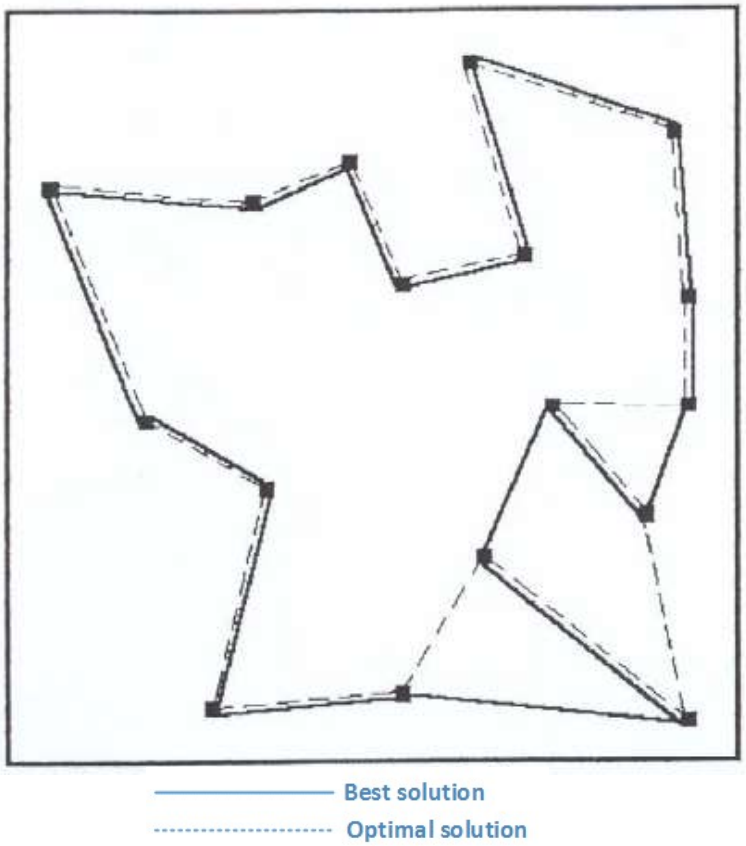
Iteración: 50 Fitness: 293.6



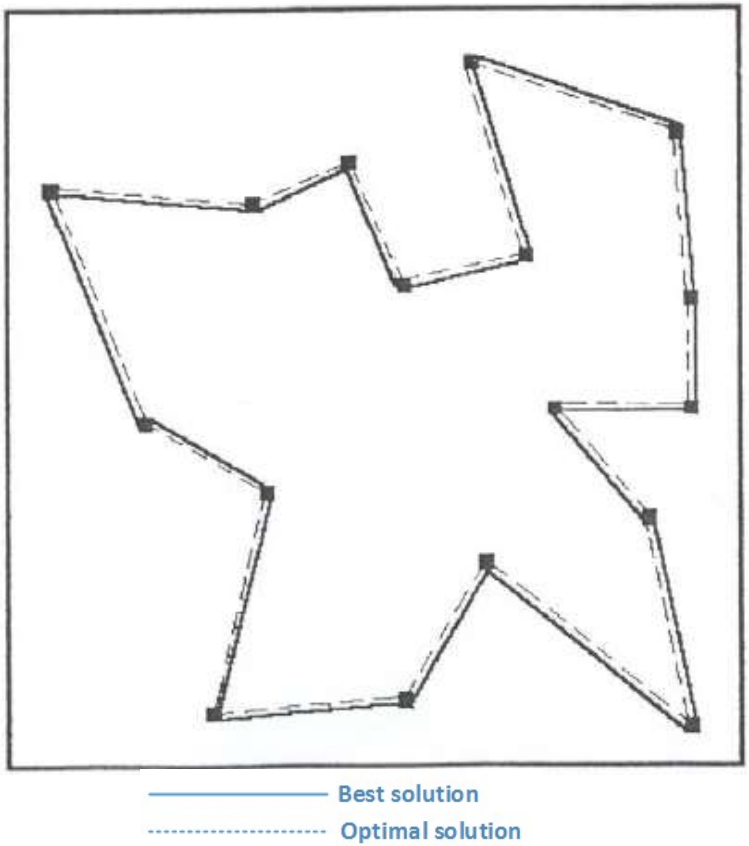
Iteración: 100 Fitness: 256.55

Source: Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada

El problema del viajante

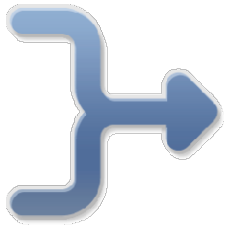


Iteración: 200 Fitness: 231.4



Iteración: 250 Fitness: 226.64

Source: Apuntes Algorítmica. Softcomputing and Soft Computing and Intelligent Information Systems. Universidad de Granada



Resumen

- Aproximación evolutiva (Diversidad, selección y replicación)
- Evolución artificial
- Algoritmo genético
  - Representación
  - Inicialización
  - Fitness
  - Selección
  - Cruce
  - Mutación
  - Reemplazamiento
  - Criterio de parada
- Algunos comentarios adicionales