



Universidad
Francisco de Vitoria
UFV Madrid

Inteligencia Artificial II

Tema 8: Algoritmos Genéticos



- Ubicación
 - Bloque III: **COMPUTACION EVOLUTIVA**
 - Tema 8: *Algoritmos Genéticos*
- Objetivos
 - Definir un AG como **mecanismo computacional de optimización**
 - Entender la razón del funcionamiento de los AG basados en su **origen biológico**
 - Comprender el significado de los **parámetros de un AG**
 - Entender los **operadores genéticos** que se utilizan en computación evolutiva, derivados de la genética molecular



1. Introducción
2. Fundamentos biológicos
3. Historia
4. ¿Qué es un AG?
5. Elementos de un AG
6. Operadores
 1. Operador Selección
 2. Operador Entrecruzamiento
 3. Operador Mutación
7. Aplicaciones
8. Recursos



1. Introducción
2. Fundamentos biológicos
3. Historia
4. ¿Qué es un AG?
5. Elementos de un AG
6. Operadores
 1. Operador Selección
 2. Operador Entrecruzamiento
 3. Operador Mutación
7. Aplicaciones
8. Recursos

1. Introducción



Un algoritmo genético (AG) es una técnica de optimización basada en la búsqueda e inspirada en la teoría de la evolución de Darwin y Wallace



pequeños cambios heredables en los seres vivos

+

selección natural

=

cambio en la Naturaleza
y generación de nuevas especies
(*supervivencia del más adaptado*)

4. ¿Qué es un AG?

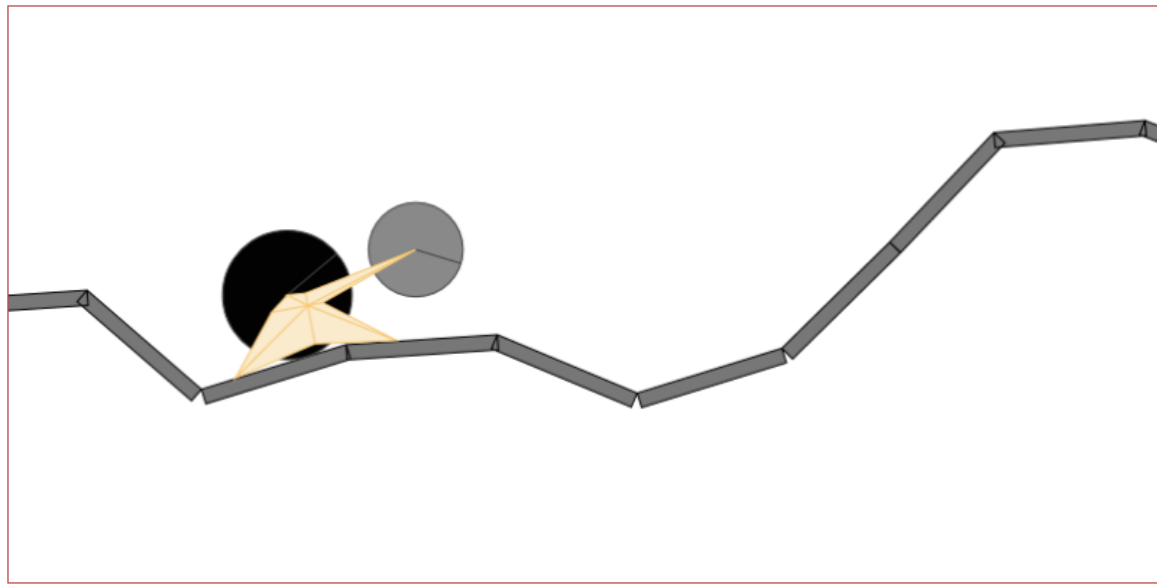


- La resolución de un problema de optimización mediante un AG es una técnica:
 - Basada en poblaciones:
 - En vez de una solución tentativa al problema, que se va alterando hasta obtener la solución óptima...
 - se llevan a cabo mejoras sucesivas sobre un conjunto de candidatos.
 - Que aplica los principios de la **Evolución y Selección Natural**:
 - Nuevas soluciones (hijos) aparecen como resultado de cambios en las soluciones que ya existen (progenitores)
 - Las soluciones que resuelven mejor el problema generan más descendencia con mayor probabilidad que las otras.

4. ¿Qué es un AG?



https://rednuht.org/genetic_cars_2/



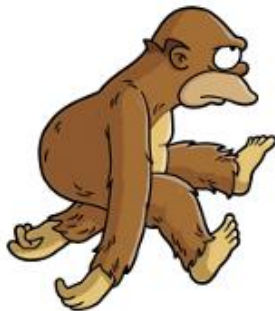
1. Introducción



EVOLUCIÓN = Reproducción errónea + Selección natural



MONKIUS EATALOTIS



CHIMPUS IMBECILUS



APEIS STUPIDIUS



NEANDERSLOB



HOMERSAPIEN

HOMERSAPIEN



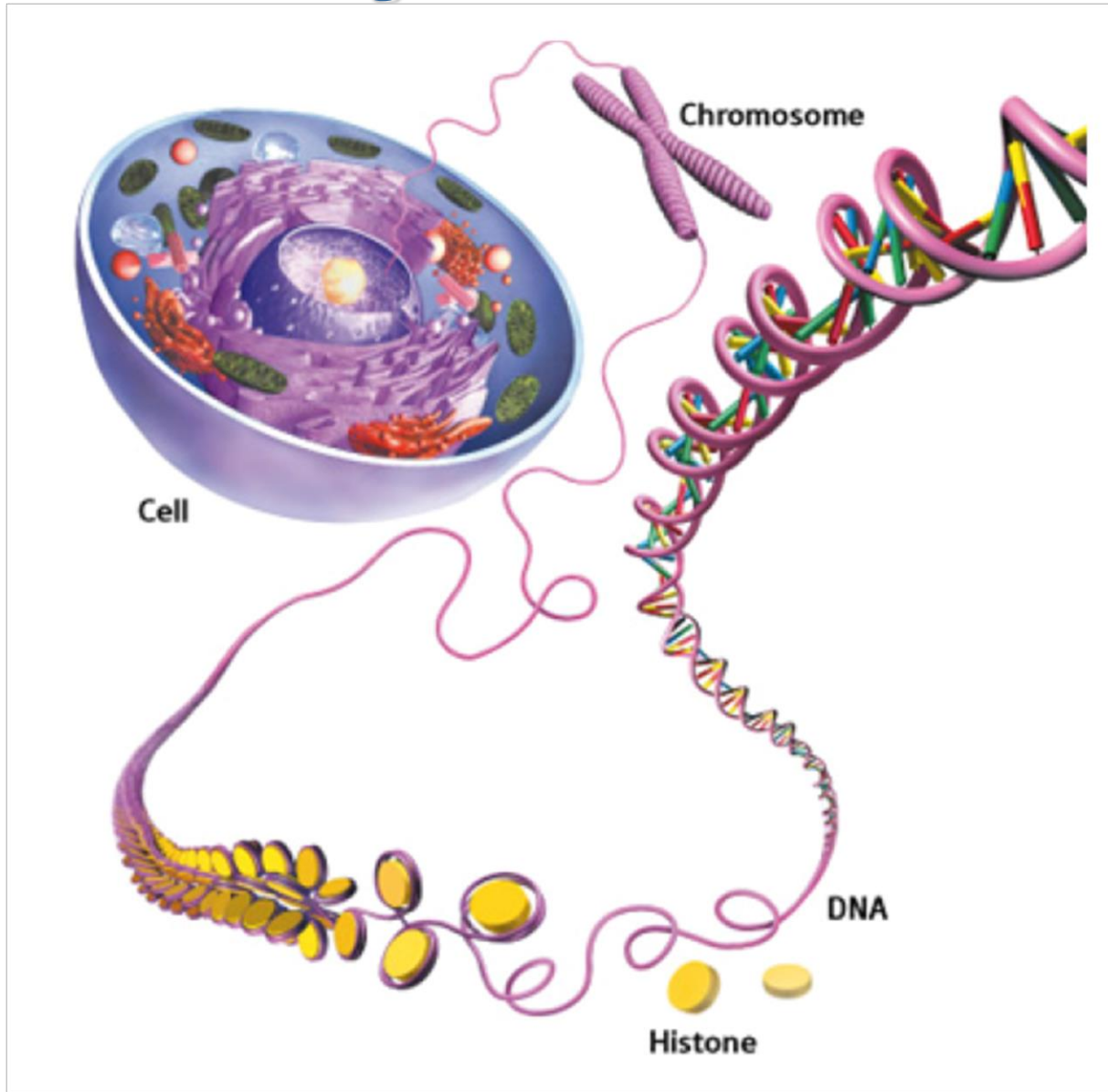
1. Introducción
2. Fundamentos biológicos
3. Historia
4. ¿Qué es un AG?
5. Elementos de un AG
6. Operadores
 1. Operador Selección
 2. Operador Entrecruzamiento
 3. Operador Mutación
7. Aplicaciones
8. Recursos

2. Fundamentos biológicos



- La Genética posibilita los AG por el conocimiento de los mecanismos de herencia y la estructura molecular del ADN
 - Gregor Mendel (1865)
 - Los caracteres (información genética) se heredan de forma discreta, del padre/madre, dominante/recesivo
 - La información genética está en los genes
 - Los valores que puedan tomar los genes se llaman alelos
 - Walther Flemming (1882)
 - La información genética está estructurada en forma de cromosomas (1 cromosoma = varios genes).
 - El conjunto de cromosomas constituye el Genoma
 - 24 cromosomas
 - 25/30K genes
 - Watson y Crick (1950)
 - El cromosoma está compuesto de DNA superenrollado "sujeto" por unas proteínas (histonas)
 - Distinción genotipo/fenotipo → expresión genética

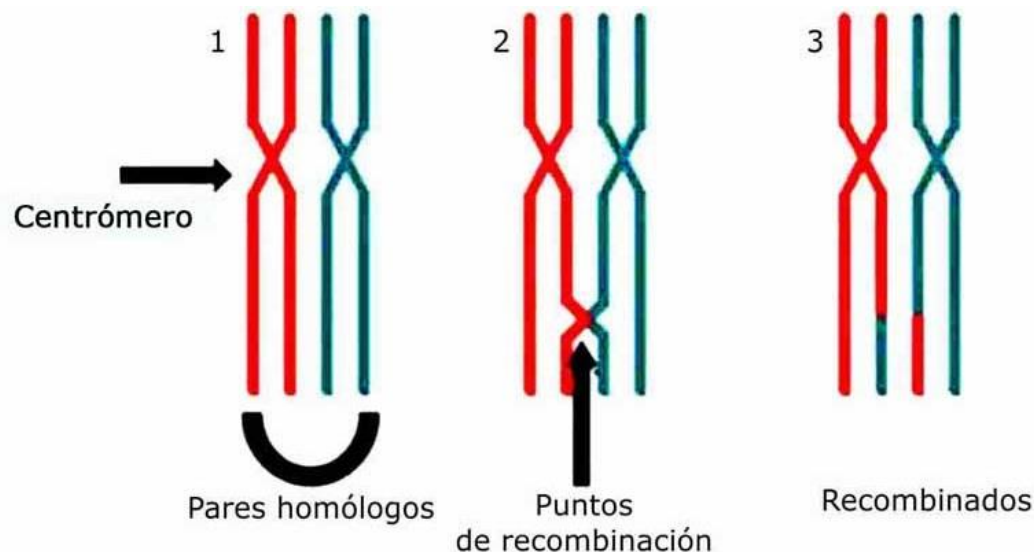
2. Fundamentos biológicos



2. Fundamentos biológicos



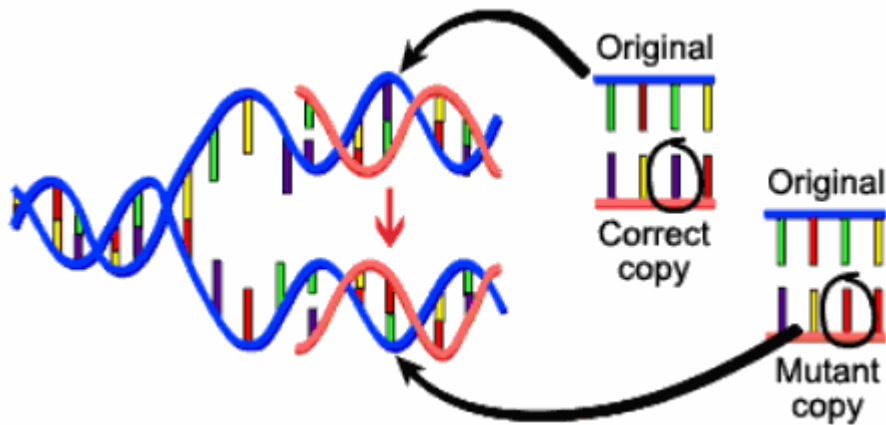
- Durante el proceso de duplicación del material genético pueden ocurrir fenómenos que alteran la secuencia de ADN, de forma que el hijo **no es copia exacta** del padre
- Proceso de replicación sexual (meiosis): RECOMBINACIÓN → mecanismo programado
 - Durante la generación de células sexuales se combinan los cromosomas homólogos



2. Fundamentos biológicos



- Proceso de replicación asexual (mitosis): MUTACIONES → accidente
 - Alteración del código genético
 - Existen sustancias mutagénicas



- Estos fenómenos garantizan variación genética en la población (actúan sobre el *genotipo*)
- En cambio, la selección natural y la deriva genética reducen la diversidad (actúan sobre el *fenotipo*)

2. Fundamentos biológicos





1. Introducción
2. Fundamentos biológicos
3. Historia
4. ¿Qué es un AG?
5. Elementos de un AG
6. Operadores
 1. Operador Selección
 2. Operador Entrecruzamiento
 3. Operador Mutación
7. Aplicaciones
8. Recursos

3. Historia



- Los primeros intentos de aplicar de manera formal la teoría de la evolución, aparecieron en las áreas de control de procesos estadísticos, aprendizaje de máquina y optimización de funciones.
- [1950] **Alan Turing** reconoció que hay una conexión entre el aprendizaje máquina y la evolución
 - Posibilidad de desarrollar programas de ajedrez usando esta técnica.
- [1950's] **Barricelli** lleva a cabo la primera simulación que usaba principios evolutivos
 - Utilizó los mismos procedimientos generales que se usan hoy en día en la disciplina conocida como vida artificial.
 - Bremermann, Friedberg y Box

3. Historia



- [1970] **John Holland**
 - Métodos adaptativos
 - Búsqueda y optimización
 - Evolución de la población
 - Acciones aleatorias semejantes
 - Inspirados en la evolución biológica
 - Basada en la probabilidad

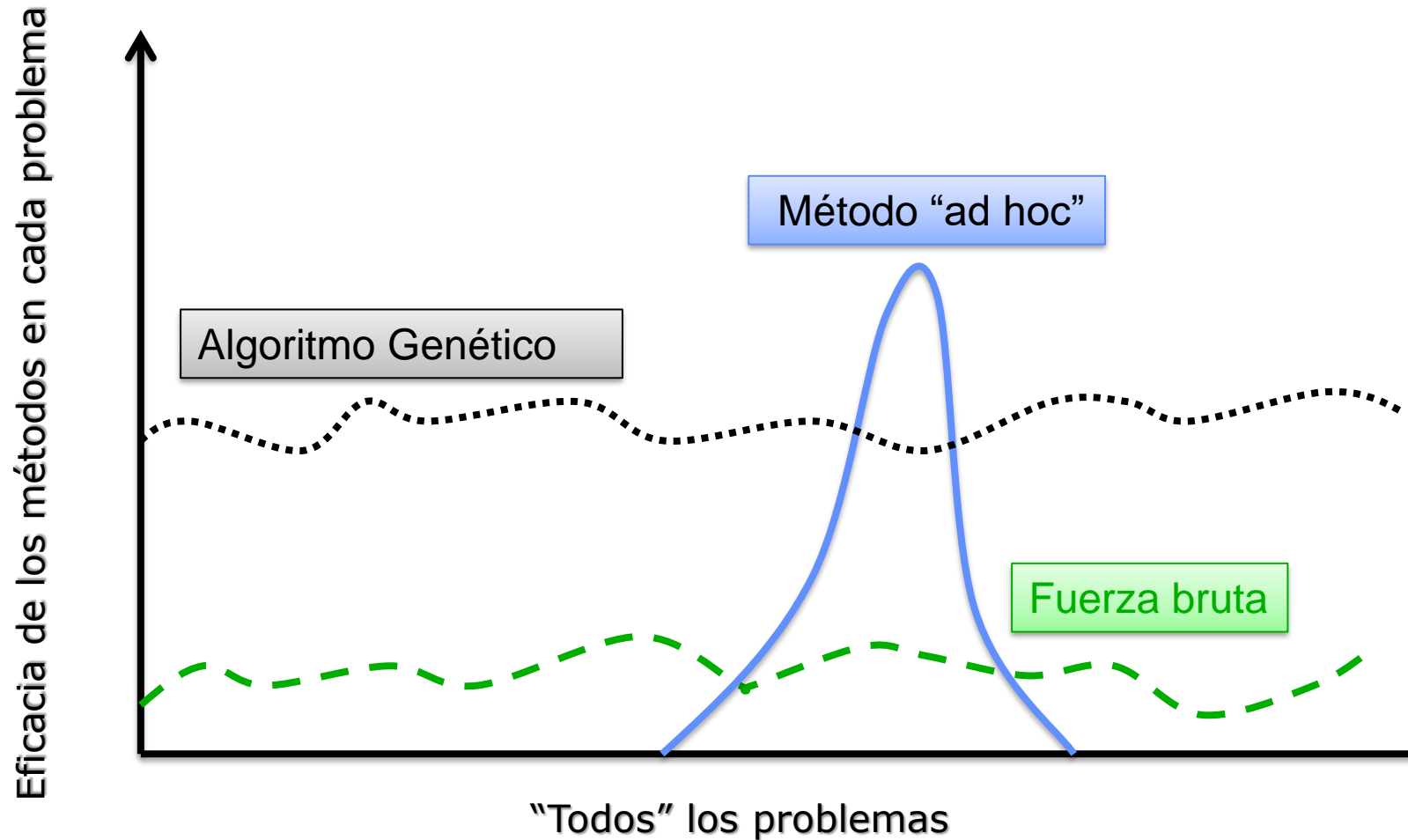
- [1985] **David Goldberg**, alumno de Holland
 - Primera conferencia sobre Algoritmos Genéticos, ICGA '85.

- [1986] **Hicklin** y **Fujiki** uso de expresiones-S en LISP para representar programas cuyo objetivo era resolver problemas de teoría de juegos.

3. Historia



Planteamiento de Goldberg (1989)



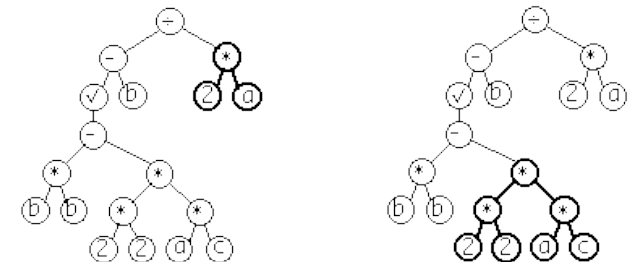
3. Historia



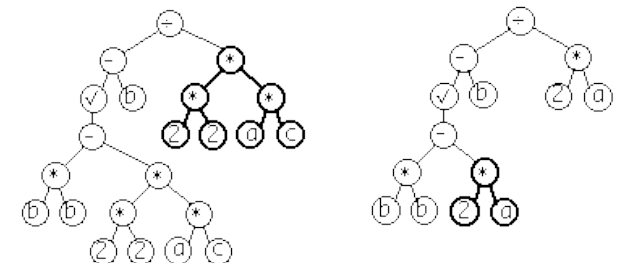
- [1985] **Nichal Cramer** y [1989] **John Koza** proponen (de forma independiente) representar los programas de ordenador como árboles en la que se implementa un operador de cruce para intercambiar sub-árboles entre los diferentes programas de una población generada al azar.
 - Cramer: función de aptitud manual (cuan bueno es un programa dado)
 - Koza: función de aptitud automatizada. Se impuso a la larga, y más tarde se denominó "Programación Genética".

Crossover Operation with Identical Parents

Parents



Children



3. Historia



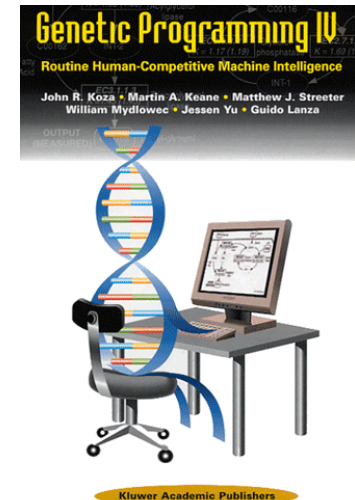
- [principio 90's] **Thomas Ray** desarrolla Tierra
 - Simulador en el que evolucionan programas en lenguaje ensamblador que compiten por ciclos de CPU de un ordenador a la vez que intentan reproducirse (copiarse a sí mismos) en la RAM.
 - Programa único con capacidad de auto-replicarse (ancestro). Con base en este programa se generan "criaturas" nuevas (segmentos de código), las cuales a su vez se podían subdividir para dar origen a nuevas criaturas

```
E:\Programme\TierraWin\Tierra\Tierra.exe
InstExec = 57,695631 Cells = 594 Genotypes = 297 Sizes = 78
Extracted = 0071aaz @ 10 normal
- 36abh 70 : *****
  69abp 31 : *****
  36abz 17 : *****
  40acd 17 : *****
  37abz 14 : *****
    acm 9 : *****
    ach 11 : *****
  38aem 7 : *****
  45adc 9 : *****
  74aax 9 : *****
  36acu 8 : *****
  71aaz 8 : *****
  36aci 5 : ***
  44abu 6 : ***
  39adv 5 : ***
  36acs 7 : *****
  70abn 5 : ***
  36acp 4 : ***
  40acm 4 : ***
  44abo 4 : ***
  46acl 2 : **
  68abg 4 : ***
Press Any Key for Menu ...
```

3. Historia



- [1994] **Michalewicz** propone los Programas Evolutivos
 - Incorporan conocimiento específico del problema a resolver en las estructuras de datos.
 - Los PE's son métodos que incorporan directamente conocimiento específico a los AG al permitir la utilización de estructuras de datos naturales.
 - Esto permite la utilización de operadores genéticos sensibles al contexto, con el fin de mejorar la eficiencia del algoritmo de búsqueda a costa de perder una parte de la propiedad de generalización.





1. Introducción
2. Fundamentos biológicos
3. Historia
4. ¿Qué es un AG?
5. Elementos de un AG
6. Operadores
 1. Operador Selección
 2. Operador Entrecruzamiento
 3. Operador Mutación
7. Aplicaciones
8. Recursos

4. ¿Qué es un AG?



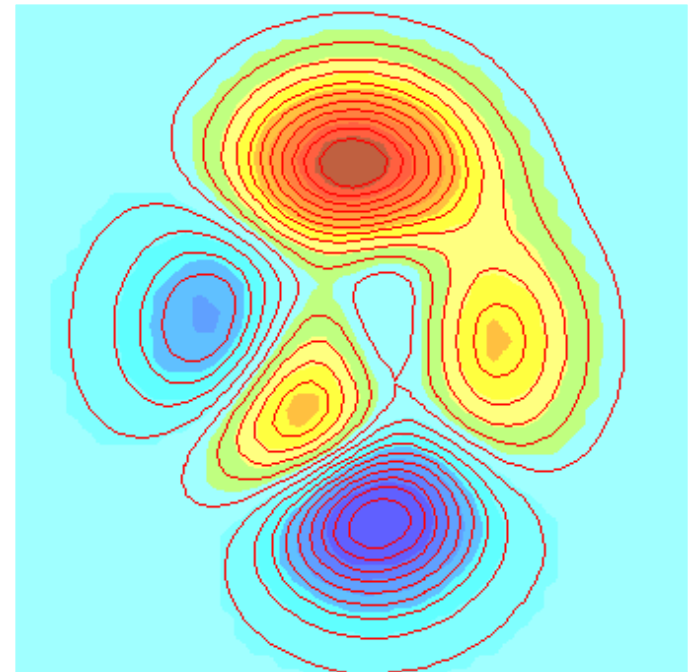
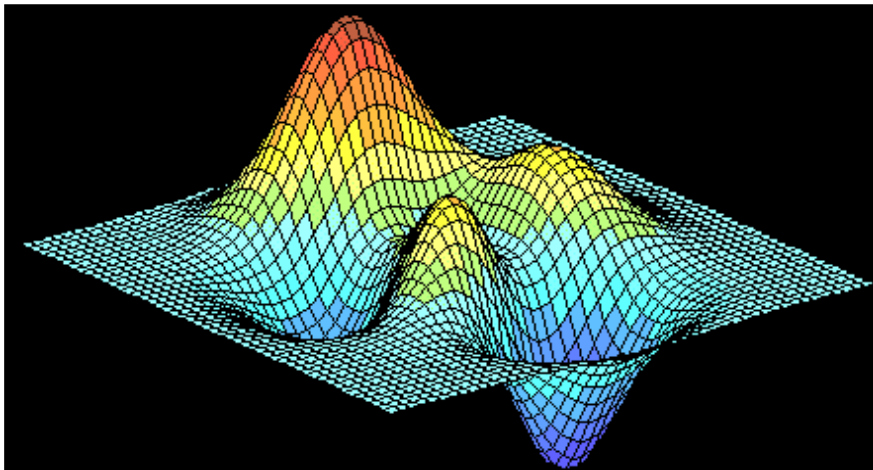
- El proceso de optimización de un problema mediante un AG consiste en:
 - Identificar de qué parámetros depende la solución al problema
 - Generar aleatoriamente un conjunto de soluciones potenciales a ese problema usando esos parámetros y codificadas de alguna manera
 - Definir una métrica que permita evaluar cuantitativamente a cada solución candidata
 - Ejercer una presión selectiva sobre cada solución de forma que se produzca una selección (evolución Darwiniana):
 - Basada en que el tamaño de la población es fijo
 - Sólo aquellas soluciones que resuelvan mejor el problema sobreviven y legan su material genético a las siguientes generaciones.
 - Crear variaciones de las soluciones usando los principios de la Genética Molecular

4. ¿Qué es un AG?



- Ejemplo: encontrar los máximos de la función “picos”

$$z = f(x, y) = 3*(1-x)^2*\exp(-(x^2) - (y+1)^2) - 10*(x/5 - x^3 - y^5)*\exp(-x^2-y^2) - 1/3*\exp(-(x+1)^2 - y^2).$$



4. ¿Qué es un AG?



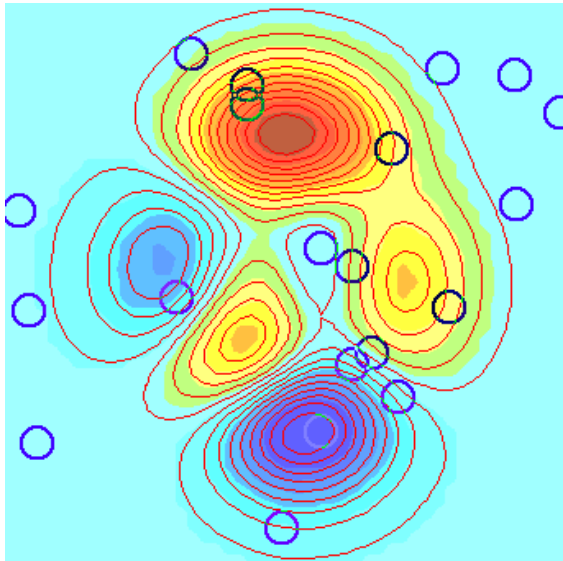
■ Derivadas de la función “picos”

- $$\frac{dz}{dx} = -6*(1-x)*\exp(-x^2-(y+1)^2) - 6*(1-x)^2*x*\exp(-x^2-(y+1)^2) - 10*(1/5-3*x^2)*\exp(-x^2-y^2) + 20*(1/5*x-x^3-y^5)*x*\exp(-x^2-y^2) - 1/3*(-2*x-2)*\exp(-(x+1)^2-y^2)$$
- $$\frac{dz}{dy} = 3*(1-x)^2*(-2*y-2)*\exp(-x^2-(y+1)^2) + 50*y^4*\exp(-x^2-y^2) + 20*(1/5*x-x^3-y^5)*y*\exp(-x^2-y^2) + 2/3*y*\exp(-(x+1)^2-y^2)$$
- $$\frac{d(dz/dx)}{dx} = 36*x*\exp(-x^2-(y+1)^2) - 18*x^2*\exp(-x^2-(y+1)^2) - 24*x^3*\exp(-x^2-(y+1)^2) + 12*x^4*\exp(-x^2-(y+1)^2) + 72*x*\exp(-x^2-y^2) - 148*x^3*\exp(-x^2-y^2) - 20*y^5*\exp(-x^2-y^2) + 40*x^5*\exp(-x^2-y^2) + 40*x^2*\exp(-x^2-y^2)*y^5 - 2/3*\exp(-(x+1)^2-y^2) - 4/3*\exp(-(x+1)^2-y^2)*x^2 - 8/3*\exp(-(x+1)^2-y^2)*x$$
- $$\frac{d(dz/dy)}{dy} = -6*(1-x)^2*\exp(-x^2-(y+1)^2) + 3*(1-x)^2*(-2*y-2)^2*\exp(-x^2-(y+1)^2) + 200*y^3*\exp(-x^2-y^2) - 200*y^5*\exp(-x^2-y^2) + 20*(1/5*x-x^3-y^5)*\exp(-x^2-y^2) - 40*(1/5*x-x^3-y^5)*y^2*\exp(-x^2-y^2) + 2/3*\exp(-(x+1)^2-y^2) - 4/3*y^2*\exp(-(x+1)^2-y^2)$$

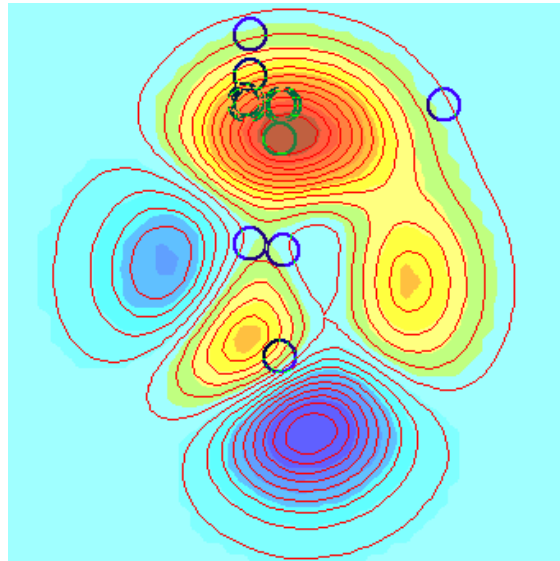
4. ¿Qué es un AG?



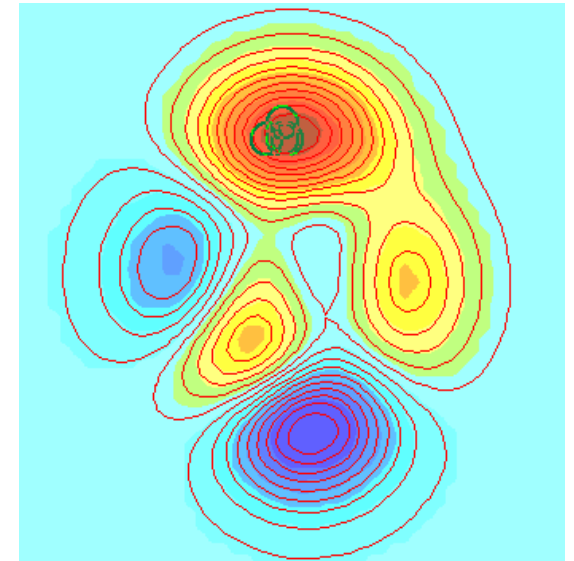
- Proceso del AG



Población inicial



5ª generación



10ª generación

4. ¿Qué es un AG?



Definición de Algoritmo Genético

- Algoritmo matemático sistemático de resolución/optimización de problemas mediante búsqueda
- Altamente paralelo
- Donde un conjunto (población) de soluciones (individuos) a un problema, codificadas en forma de secuencia (cromosoma), asociadas con una cierta función de coste (fitness) y que pueden ir cambiando mediante operadores (evolución), compiten para ver cuál satisface mejor el problema
- *Aunque no necesariamente se obtiene la mejor de todas las soluciones posibles*

4. ¿Qué es un AG?



Definición (John Kosa, 1995)

Es un algoritmo matemático altamente paralelo que transforma con respecto al tiempo un conjunto de objetos matemáticos individuales usando operaciones genéticas modeladas de acuerdo al principio Darwiniano de reproducción y supervivencia del más apto.

Cada uno de estos objetos matemáticos es una cadena de caracteres (letras o números) de longitud fija que se ajusta al modelo de las cadenas de cromosomas, y se les asocia con una cierta función matemática que refleja su aptitud.

4. ¿Qué es un AG?



- Bucle de un AG
 1. Se genera una *población* inicial de soluciones (*individuos*)
 2. Se evalúa la adaptación (*fitness*) de cada individuo
 3. En una generación dada,
 - A. Se toman los individuos mejor adaptados (*mejores soluciones para esa generación*).
 - B. Se cruzan (*recombinan*) generando nuevos individuos que contendrán parte del código genético (información) de sus antecesores.
 - C. Se mejoran (*mutan*) generando nuevos individuos con mejoras sobre sus padres.
 - D. Se evalúa el *fitness* de cada nuevo individuo.
 - E. Se insertan los nuevos individuos en la población.
 - F. Si se ha alcanzado la condición de terminación, se detiene. **SI NO**, se vuelve a 3.

4. ¿Qué es un AG?



BEGIN /* Algoritmo Genético Simple */

1. Generar una población inicial

2. Calcular la función de evaluación de cada individuo (fitness)

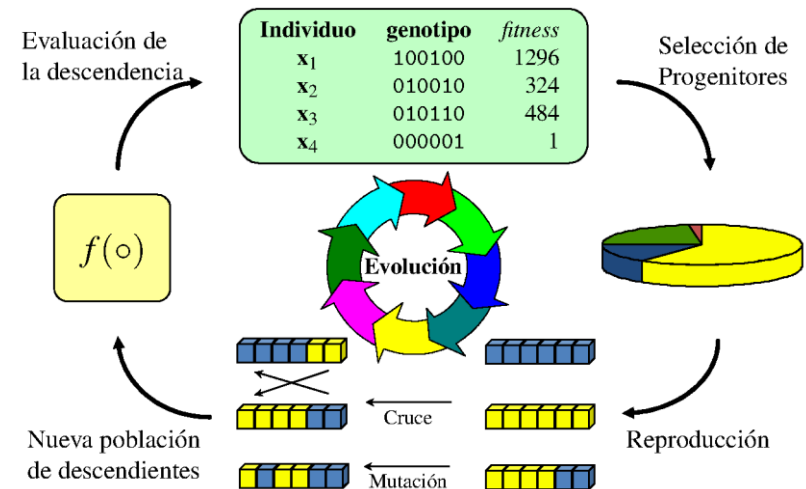
3. WHILE NOT Terminado DO
 /* Producir generación */
 FOR Tamaño_Población DO
 1. Seleccionar progenitores
 2. Cruzar
 3. Mutar
 4. Calcular nuevo fitness
 5. Insertar

END

IF la población ha convergido THEN
 Terminado = TRUE

END

END





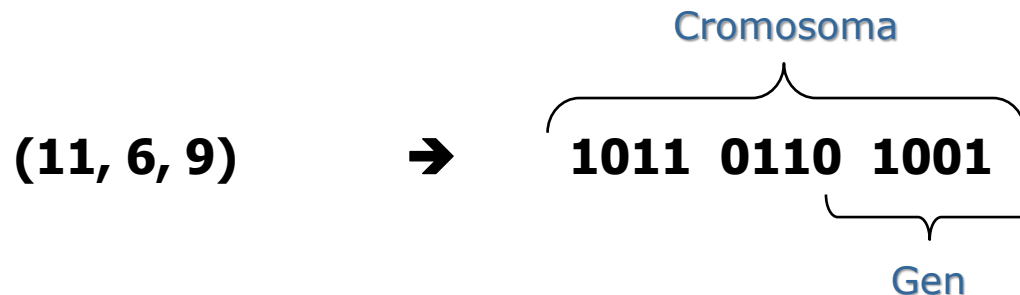
1. Introducción
2. Fundamentos biológicos
3. Historia
4. ¿Qué es un AG?
5. Elementos de un AG
6. Operadores
 1. Operador Selección
 2. Operador Entrecruzamiento
 3. Operador Mutación
7. Aplicaciones
8. Recursos

5. Elementos de un AG



Codificación

- La solución al problema tiene que representarse (codificarse) en un cromosoma (secuencia de bits/caracteres/dígitos...)
- Depende del problema a resolver: puede que una codificación binaria complique innecesariamente el problema.
- Decodificación → obtención del *fenotipo*
- *Codificación binaria (alfabeto cardinalidad = 2)*
 - Los bits pueden codificar números enteros, reales, conjuntos, etc
 - Ventaja: los operadores de cruce y mutación resultan simples
 - Inconveniente: no siempre resulta natural
 - Ejemplo: problema de la mochila (si/no → 0/1)



5. Elementos de un AG



- *Codificación numérica (alfabetos cardinalidad $\neq 2$)*
 - Cadenas de números que representan un número en una secuencia. Puede ser necesario hacer correcciones después del entrecruzamiento/mutación.
 - Ejemplo: problema del viajante de comercio
- *Codificación por valor directo (alfabetos cardinalidad $\neq 2$)*
 - Cada cromosoma es una cadena de valores relacionados con el problema a estudiar.
 - Para problemas que requieren valores de codificación compleja, como números reales, cuya codificación en binario es compleja.
 - Ejemplo: búsqueda de matriz de pesos para una red neuronal.
- El *teorema de los esquemas*, afirma que la codificación sobre la que los algoritmos genéticos funcionan mejor es con un alfabeto de cardinalidad 2.
- Con cardinalidad $\neq 2$ puede ser necesario redefinir los operadores de entrecruzamiento y mutación.

5. Elementos de un AG



Individuo:

- Cromosoma (secuencia de caracteres/dígitos) que constituye el código de información sobre el cual opera el AG.

1 0 0 1 1 1 1 0

genotipo

Población:

- Un conjunto de individuos (Cromosomas)

1 0 1 1 0 1 1 0

1 0 0 1 0 1 1 1

1 1 0 1 1 1 1 0

...

1 1 0 1 1 1 1 1

0 0 1 0 1 1 1 0

1 0 0 1 1 0 0 0

1 0 0 0 1 1 0 1

5. Elementos de un AG



Función Fitness/Objetivo/Coste:

- Función que informa de cuan bueno es un individuo dado en la solución de un problema (evalúa el *fenotipo*).
 - Diseñada para cada problema de manera específica
 - Debe reflejar el valor de un individuo de manera real
- Durante la evaluación,
 - Se decodifica el cromosoma convirtiéndose en una serie de parámetros del problema.
 - Se halla la solución del problema a partir de esos parámetros
 - Se le da una puntuación a esa solución mediante la *función de fitness*.
 - Ejemplo TSP
 $F_c(x)$ = Distancia del viaje representado por la secuencia de ciudades en el cromosoma

5. Elementos de un AG



Parámetros

- *Tamaño de la Población*
 - Número de cromosomas en nuestra población para una generación determinada.
 - Si es insuficiente, el AG tiene pocas posibilidades de realizar reproducciones con lo que se realizaría una búsqueda de soluciones escasa y poco óptima.
 - Si es excesiva, el algoritmo genético será excesivamente lento. Hay un límite a partir del cual es ineficiente elevar el tamaño de la población puesto que no se consigue una mayor velocidad en la resolución del problema.
 - Tiene que crecer más o menos con el número de bits del cromosoma.
 - Depende también del ordenador en el que se esté ejecutando.
- *Condición de terminación*
 - Convergencia del algoritmo genético (fitness no varía después de varias generaciones).
 - Número prefijado de generaciones.

5. Elementos de un AG



- **Probabilidad de Selección** (no es un parámetro del sistema)
 - Indica la probabilidad de elegir un cromosoma para ser replicado.
 - Este valor es único calculado para cada individuo.
 - $P_s=0$: el cromosoma no se va a replicar.
 - $P_s \neq 0$: el cromosoma tiene una cierta probabilidad de replicarse.
 - Cálculo:
 - Cociente entre f_i y $\sum_i f_i$
 - Cociente entre f_i y $\max(f_i)$
 - Minimax para f_i
- **Probabilidad de Entrecruzamiento**
 - Indica la frecuencia con la que se producen cruces entre los cromosomas padre.
 - Este valor es igual para todas las operaciones de cruce.
 - $P_c=0$: Los padres no se entrecruzan. Los hijos serán copias exactas de los padres.
 - $P_c \neq 0$: Los padres tienen una cierta probabilidad de entrecruzarse y los hijos tendrán partes de los cromosomas de cada padre.
 - Se elige al azar la posición de cruce

5. Elementos de un AG



- *Probabilidad de Mutación*

- Indica la frecuencia con la que cada gen de un cromosoma es mutados. Se aplica gen a gen.
- Este valor es igual para todos los genes
 - $P_m=0$: Los hijos no mutan. El cromosoma descendiente es idéntico al de origen.
 - $P_m \neq 0$: Los hijos tienen una cierta probabilidad de mutar. El cromosoma descendiente es modificado.

- De todos los elementos de un AG, **solo dos dependen del problema a resolver:**

- Codificación
- Función de Coste

5. Elementos de un AG



Ejercicio:

- Dado una serie de números de longitud arbitraria pero menor de 255 elementos, calcular la posición intermedia de la serie.
 - Si posición máxima – posición mínima
 - Es par, la serie tiene un número impar de términos y hay una posición central: (1...7) → [6, 9, 14, 8, 5, 3, 1]
 - Es impar, la serie tiene un número par de posiciones y hay dos posiciones centrales: (1...6) → [6, 9, 14, 8, 5, 3]
 - Fitness: El óptimo se da cuando
$$|(max_int - k) + (min_int - k)| = 0$$
 - Codificación: Número codificado en base 2

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



1. Introducción
2. Fundamentos biológicos
3. Historia
4. ¿Qué es un AG?
5. Elementos de un AG
6. Operadores
 1. Operador Selección
 2. Operador Entrecruzamiento
 3. Operador Mutación
7. Aplicaciones
8. Recursos

6. Operadores



- Una vez que se ha evaluado cada individuo mediante la función de fitness, se aplican los operadores genéticos:
 - Son las diferentes operaciones que se ejercen sobre una población y que nos permite obtener poblaciones nuevas.

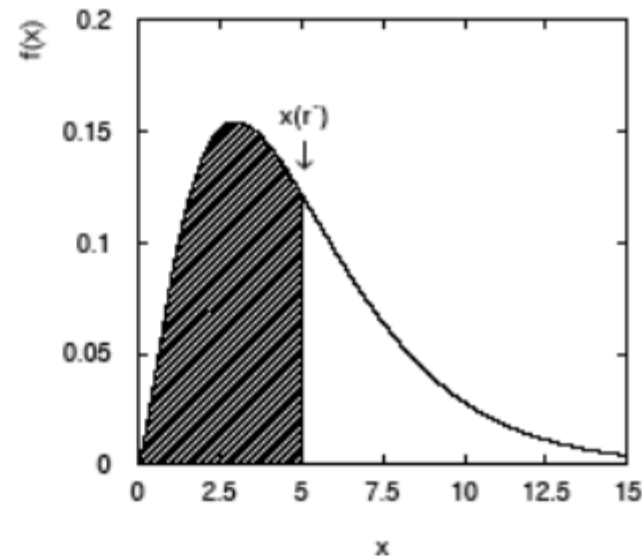
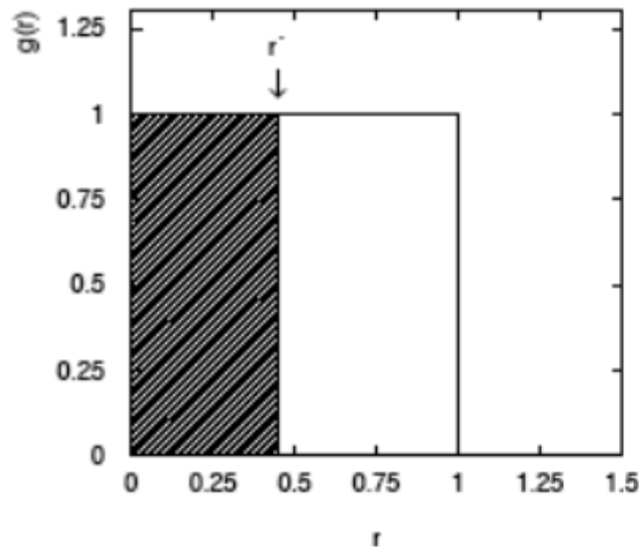
- En todos ellos se emplea el Método de Montecarlo para pasar de probabilidades a acciones concretas sobre cada uno de los individuos o genes
 - Método numérico para resolver problemas matemáticos mediante la simulación de variables aleatorias usando un generador de números aleatorios
 - Bautizado así por su analogía con los juegos de ruleta de los casinos (primer generador de números aleatorios)
 - Desarrollado en el año 1946 por Stanislaw Ulam y John von Neumann en LANL

6. Operadores



■ Proceso

- Generar una serie de números aleatorios, r_1, r_2, \dots, r_n , distribuidos uniformemente en $[0,1]$
- Usar esta secuencia para producir otra secuencia, x_1, x_2, \dots, x_n , distribuidos de acuerdo a la probabilidad de distribución en la que estamos interesados, llamémosla $f(x)$



6.1 Operador Selección



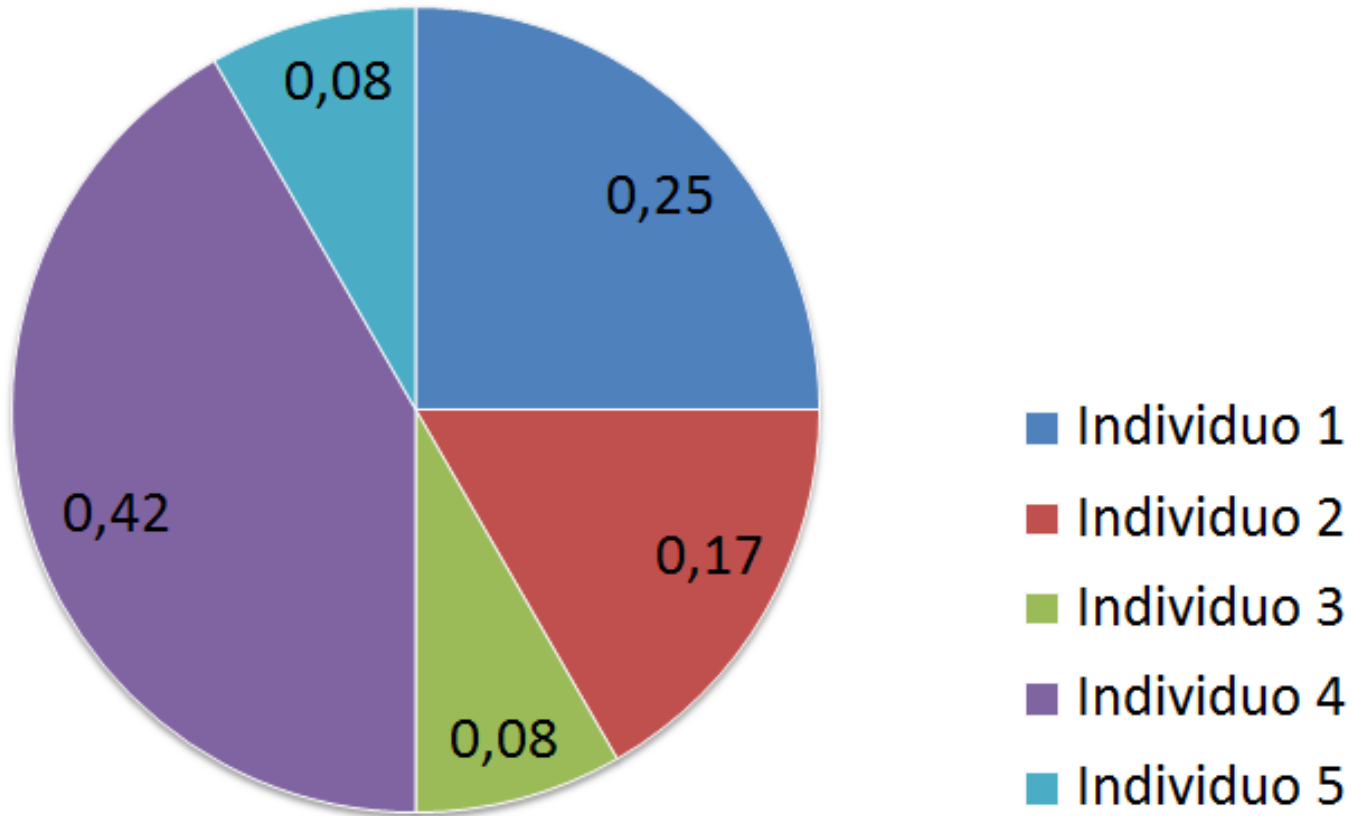
- Mecanismo que favorece la replicación de los individuos con mejor valoración, pero que no impide la *diversidad*
- Se compone de dos partes
 - Determinación de que individuos (cromosomas) de la población serán utilizados para la reproducción (padres).
 - Determinación de los individuos de la población que serán sustituidos por los recién generados

6.1 Operador Selección



- Selección de los padres
 - *Selección por Ruleta (Probabilística)*
 - Se crea una ruleta con los cromosomas presentes en una generación usando como valor
 - Los valores de fitness
 - Los fitness normalizados respecto a la suma de todos ellos
 - Cada cromosoma ocupará un área de esa ruleta mayor o menor en función del valor que tenga cada uno.
 - Se hace girar la ruleta y se selecciona el cromosoma en el que se para la ruleta.
 - Se detendrá con mayor probabilidad sobre el cromosoma con mayor valor (más área), pero los demás también tienen una cierta probabilidad de ser elegidos.
 - Si las probabilidades difieren mucho, este método de selección dará problemas por reducción de la diversidad genética.
 - *Selección por Rango*
 - Los n mejores (mayor valor de P_s) siempre son escogidos para reproducción
 - La diversidad se garantiza introduciendo individuos seleccionados aleatoriamente

6.1 Operador Selección



6.1 Operador Selección

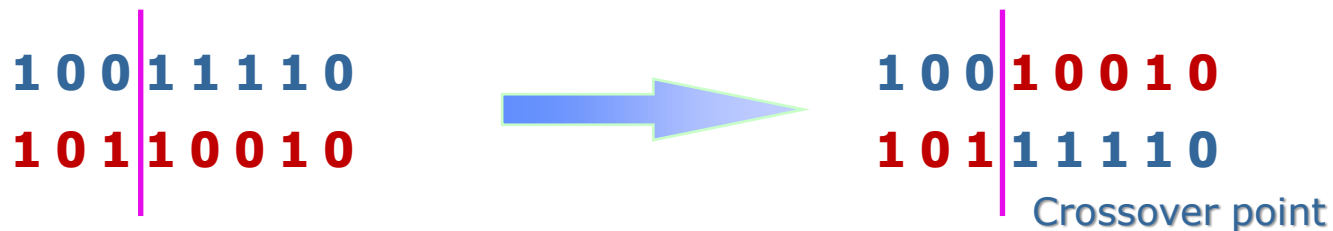


- Eliminación de individuos
 - *Selección elitista*
 - Se copian sin modificar los mejores cromosomas de la generación anterior en la nueva población. El resto se obtienen por selección por ruleta.
 - Puede mejorar el funcionamiento de los AG al evitar que se pierda la mejor solución.
 - *Selección pura*
 - Se sustituye toda la población por reproducción de padres seleccionados
 - Similar a la reproducción de los insectos: una generación pone huevos, muere y es substituida por una nueva
 - *Selección por Estado Estacionario*
 - La descendencia de los individuos seleccionados vuelve a la población preexistente reemplazando a algunos de los miembros menos aptos de la anterior generación
 - Similar al esquema generacional de los mamíferos donde coexisten los padres y sus descendientes

6.2 Operador Entrecruzamiento




- Consiste en unir en alguna forma los cromosomas de los padres que han sido previamente seleccionados de la generación anterior para formar dos descendientes.
- Según Holland, el *crossover* es el principal operador genético. No puede haber un AG si no tiene *crossover*, y sin embargo puede serlo sin mutación.
 - El *teorema de los esquemas* confía en él para hallar la mejor solución a un problema.
 - P_c decide si se produce un cruce o no para un par de cromosomas dados
 - Al azar se elige una posición entre el gen 0 y el gen n

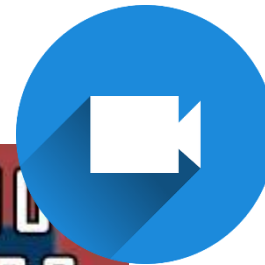
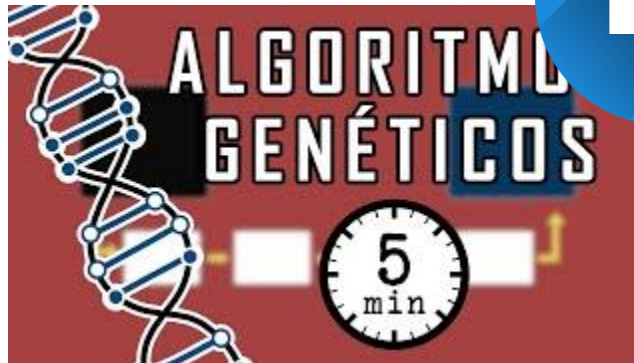


6.3 Operador Mutación



- **Operador de Mutación:** El operador de mutación consiste en la alteración aleatoria de cada uno de los genes del individuo con una probabilidad de mutación P_M .

1 0 0 1 1 1 1 0  1 0 0 1 1 0 1 0
Bit mutado





1. Introducción
2. Fundamentos biológicos
3. Historia
4. ¿Qué es un AG?
5. Elementos de un AG
6. Operadores
 1. Operador Selección
 2. Operador Entrecruzamiento
 3. Operador Mutación
- 7. Aplicaciones**
8. Recursos

7. Aplicaciones



- En la Naturaleza lo que hay que optimizar es la supervivencia → maximizar unos factores y minimizar otros simultáneamente.
 - Un AG, sin embargo, se usa para optimizar sólo una función, no diversas funciones relacionadas entre sí simultáneamente.
 - La optimización multimodal se aborda con un algoritmo genético especializado → **MOGA** (*Multi Objective Genetic Algorithm*).

- Un AG sirve para problemas NP completos (difíciles o casi irresolubles):
 - Alta dimensionalidad
 - Multimodalidad
 - No linealidad
 - Discontinuidades
 - Dependencia temporal

7. Aplicaciones



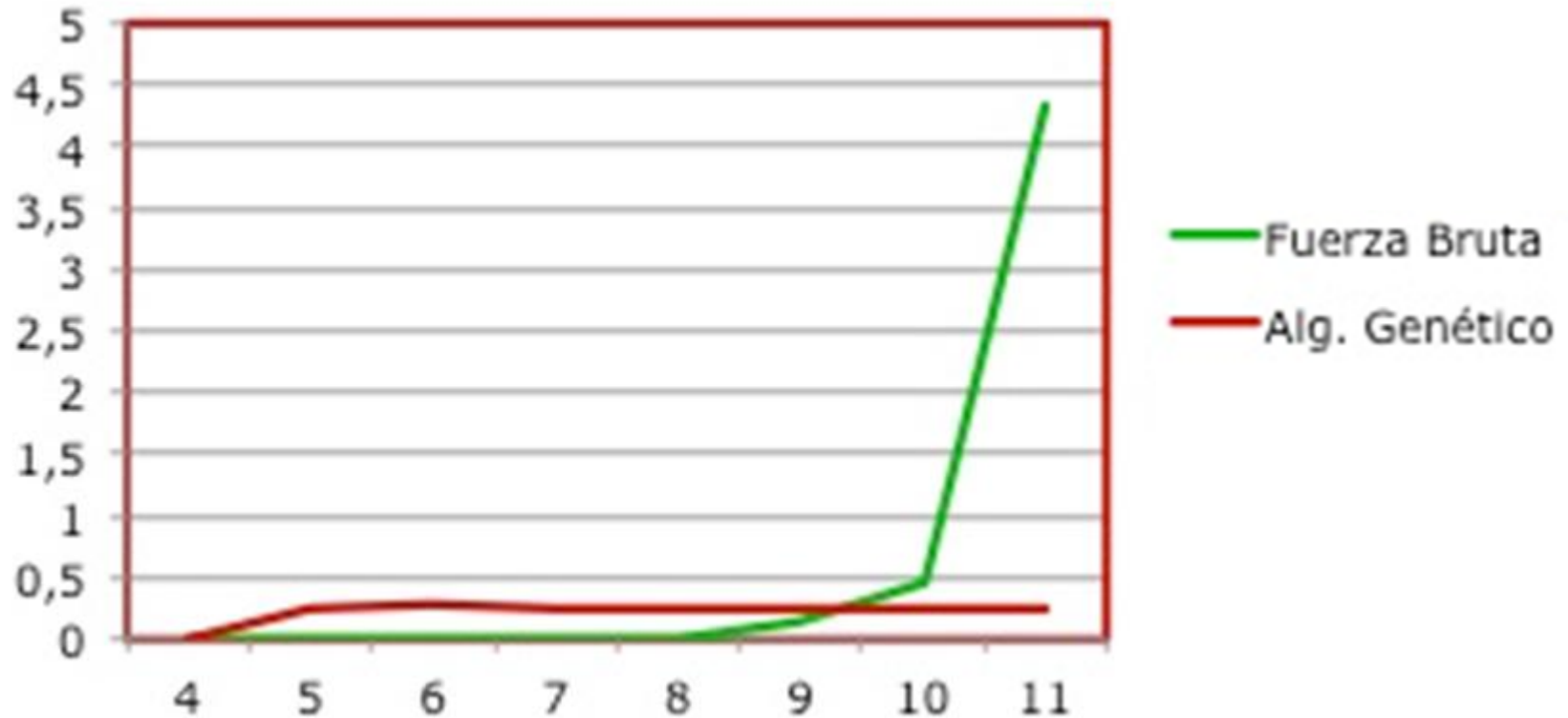
- Comparación AG vs. fuerza bruta para el TSP

<i>Núm Ciudades</i>	Algoritmo genético		Fuerza Bruta	
	<i>Tiempo (seg.)</i>	<i>Distancia (Km)</i>	<i>Tiempo (seg.)</i>	<i>Distancia (Km)</i>
4	0	4637	0,016	4637
5	0,234	5044	0	5044
6	0,297	6410	0,016	6410
7	0,25	7909	0,016	7909
8	0,25	9743	0,015	9743
9	0,25	9753	0,14	9753
10	0,25	9997	0,469	9997
11	0,25	10018	4,343	10018
12	0,25	10280	48,719	10280
13	0,265	10323	617,5	10323
14	0,266	10360	8474,953	10360
15	0,297	10494	??	??

7. Aplicaciones



- Comparación AG vs. fuerza bruta para el TSP



7. Aplicaciones



- Optimización (estructural, de topologías, numérica, combinatoria, etc.)
- Aprendizaje de maquina (sistemas clasificadores)
- Bases de datos (optimización de consultas)
- Reconocimiento de patrones (por ejemplo, imágenes)
- Generación de gramáticas (regulares, libres de contexto, etc.)
- Planeación de movimientos de robots
- Predicción.



1. Introducción
2. Fundamentos biológicos
3. Historia
4. ¿Qué es un AG?
5. Elementos de un AG
6. Operadores
 1. Operador Selección
 2. Operador Entrecruzamiento
 3. Operador Mutación
7. Aplicaciones
8. Recursos

8. Recursos



- Gga4r

- Sencillo, fácil de instalar y en español



- Pyevolve

- Core, mutators, crossovers, selectors, graphics



- JGAP

- Java Genetic Algorithms Package, Netbeans



- C++ UNGenético

- Codificación Híbrida, Lógica difusa, Español

- BoxCar 2D

