

# COMPUTACIÓN BIOLÓGICA

Pedro Isasi<sup>1</sup>

<sup>1</sup>Departamento de Informática  
Universidad Carlos III de Madrid  
Avda. de la Universidad, 30. 28911 Leganés (Madrid). Spain  
email: isasi@ia.uc3m.es

Presentación

# TEMARIO

- 1 INTRODUCCIÓN
- 2 ALGORITMOS GENÉTICOS
- 3 COMPUTACIÓN EVOLUTIVA**
  - Estrategias Evolutivas
  - Ejemplo de Estrategias Evolutivas
  - Programación Genética
  - Ejemplo de Programación Genética
- 4 COMPUTACIÓN CON INSPIRACIÓN BIOLÓGICA
- 5 BIOLOGÍA Y COMPUTACIÓN

# TEMARIO

- 1 INTRODUCCIÓN
- 2 ALGORITMOS GENÉTICOS
- 3 **COMPUTACIÓN EVOLUTIVA**
  - Estrategias Evolutivas
  - **Ejemplo de Estrategias Evolutivas**
  - Programación Genética
  - Ejemplo de Programación Genética
- 4 COMPUTACIÓN CON INSPIRACIÓN BIOLÓGICA
- 5 BIOLOGÍA Y COMPUTACIÓN

# TEMARIO

- 1 INTRODUCCIÓN
- 2 ALGORITMOS GENÉTICOS
- 3 **COMPUTACIÓN EVOLUTIVA**
  - Estrategias Evolutivas
  - Ejemplo de Estrategias Evolutivas
  - **Programación Genética**
  - Ejemplo de Programación Genética
- 4 COMPUTACIÓN CON INSPIRACIÓN BIOLÓGICA
- 5 BIOLOGÍA Y COMPUTACIÓN






# ORIGENES

- Técnica derivada de los Algoritmos Genéticos ideada en 1989 por John Koza
- Se modifica el sistema de representación de modo que:
  - La representación mantenga su carácter de simbología discreta. Los elementos han de poder ser elegidos entre un conjunto de elementos finito
  - Se aumente el poder de representación de los individuos, pudiendo diseñarse soluciones con mayor contenido semántico
- Esto permite generar soluciones simbólicas de gran nivel de abstracción
  - Aparcar un camión articulado. Se maneja el volante, la caja de cambios, la velocidad...
  - En problemas numéricos pueden generar las fórmulas analíticas:  
 $g(x)$  tal que  $g(x) = \int f(x)$

# REPRESENTACIÓN DE LAS SOLUCIONES

- Los individuos se representan mediante expresiones-s (sintaxis de Lisp)
- Son árboles cuyos nodos son funciones y las hojas operandos
- Hay que definir el conjunto de funciones permitidas y el de terminales
- Sea el problema de resolución de ecuaciones de segundo grado:  
$$a \cdot x^2 + b \cdot x + c = 0$$

## Operadores

	Resta
	Suma
	Division
	Multiplicacion
	Raiz cuadrada

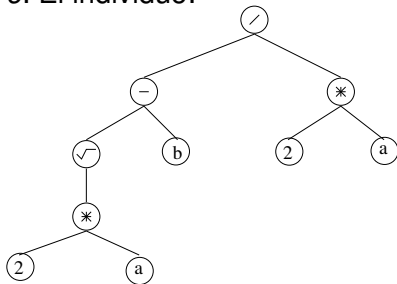
## Terminales






# SOLUCIÓN A ECUACIONES DE 2º GRADO

- Cada individuo representa a una ecuación de tres variables,  $a$ ,  $b$  y  $c$ . El individuo:



se corresponde con la ecuación:

$$x = \frac{\sqrt{2 \cdot a - b}}{2 \cdot a}$$

## EVALUACIÓN DE UN INDIVIDUO

- Se genera un conjunto de ecuaciones de segundo grado, y se calcula cómo resuelve el individuo dicho conjunto
- Cada ecuación del conjunto de ejemplos contendrá un valor para las variables  $a$ ,  $b$  y  $c$
- Se sustituyen dichos valores en la ecuación representada por el individuo a evaluar, y así se obtiene un valor para la  $x$
- Se sustituye dicho valor en la ecuación de segundo grado, y se comprueba si el resultado es cero
- Si es así, el individuo resuelve la ecuación-ejemplo, si no es así, la diferencia se utilizará para el cálculo del fitness del individuo



## EVALUACIÓN DE UN INDIVIDUO

- Por ejemplo la ecuación:

$$2x^2 + 3x - 1 = 0$$

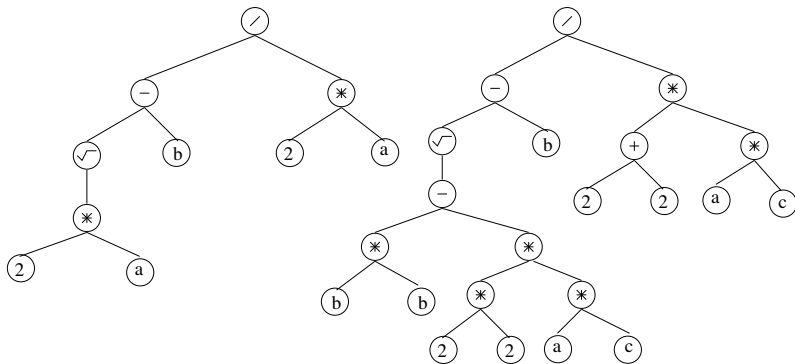
- Sustituyendo estos valores en el individuo ejemplo anterior se obtiene:

$$x = \frac{\sqrt{2 \cdot 2} - 3}{2 \cdot 2} = -4$$

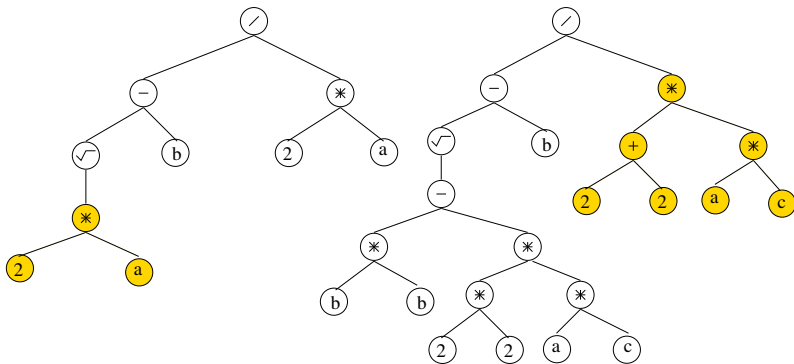
- Sustituyendo  $x$  por su valor  $-4$  en la ecuación de segundo grado se obtiene:

$$2(-4)^2 + 3(-4) - 1 = 19$$

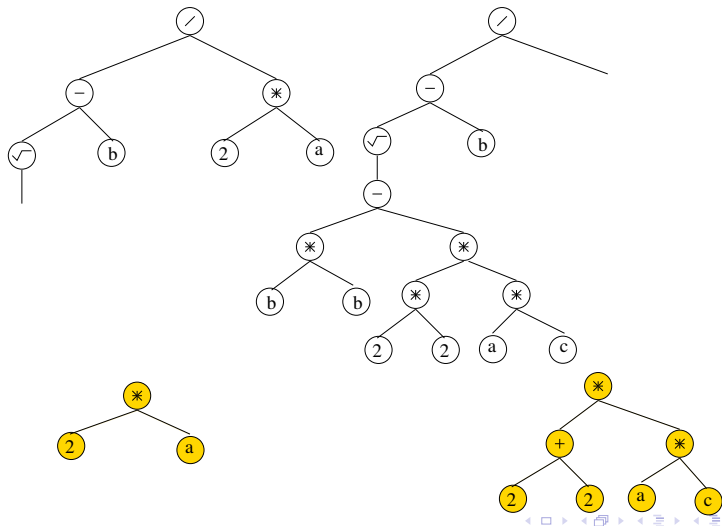
# OPERADOR DE CRUCE



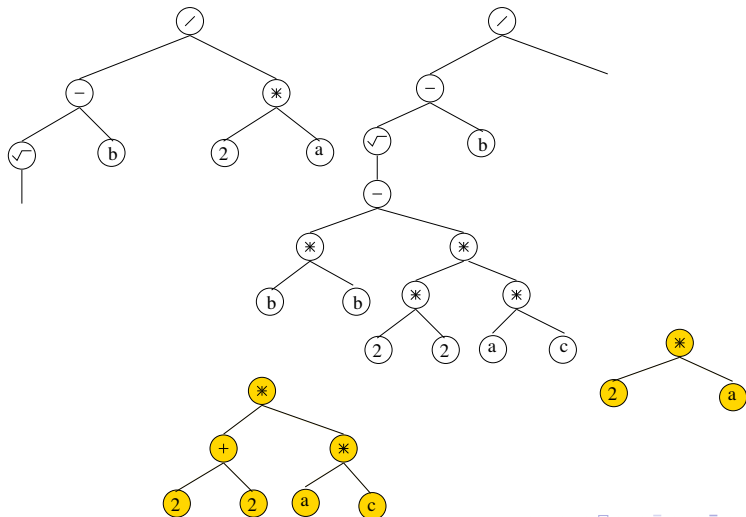
# OPERADOR DE CRUCE



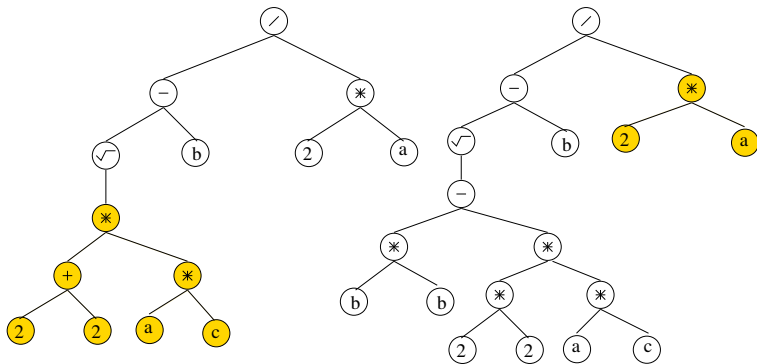
# OPERADOR DE CRUCE



# OPERADOR DE CRUCE

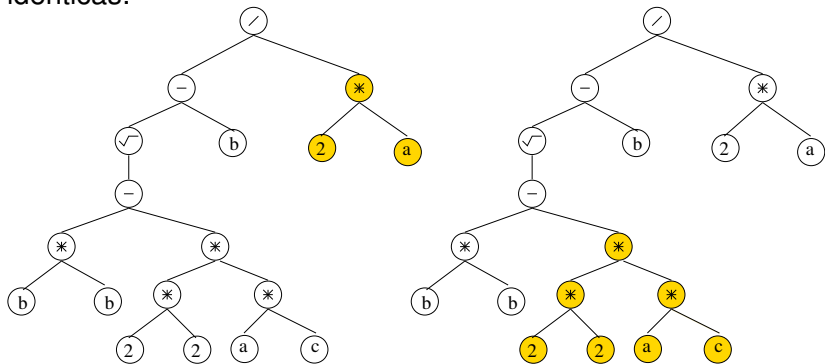


# OPERADOR DE CRUCE



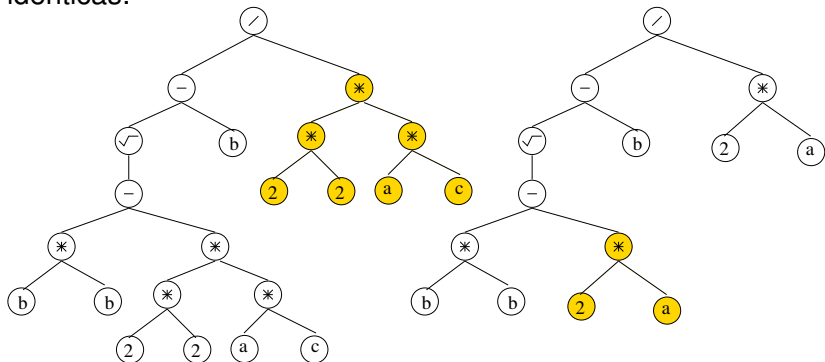
# MEJORA DEL CRUCE

Son capaces de generar soluciones nuevas a partir de soluciones idénticas:



# MEJORA DEL CRUCE

Son capaces de generar soluciones nuevas a partir de soluciones idénticas:



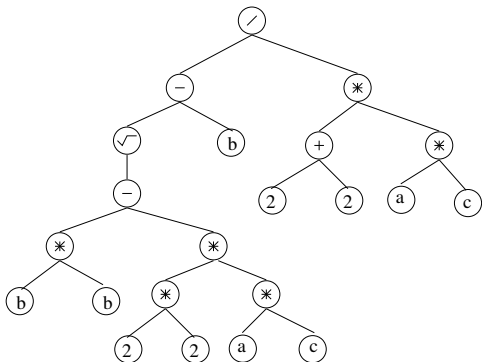


# LA MUTACIÓN

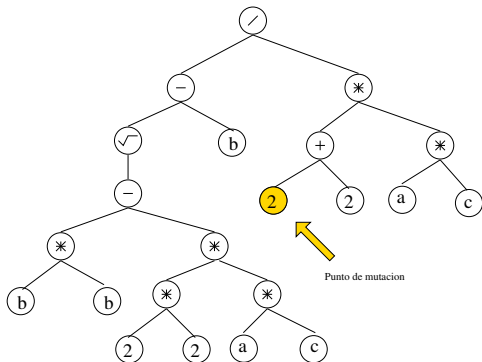
La mutación consiste en generar un nuevo programa a partir de un único progenitor. Existen tres tipos de mutación:

- Mutación terminal simple
- Mutación funcional simple
- Mutación de árbol

# MUTACIÓN TERMINAL SIMPLE

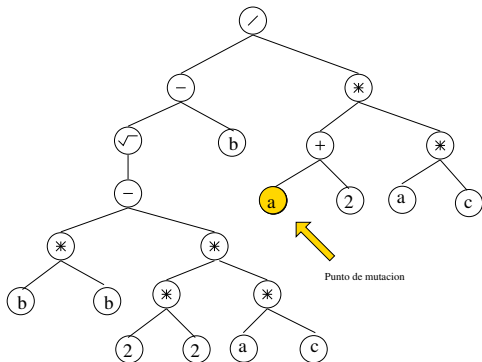


# MUTACIÓN TERMINAL SIMPLE



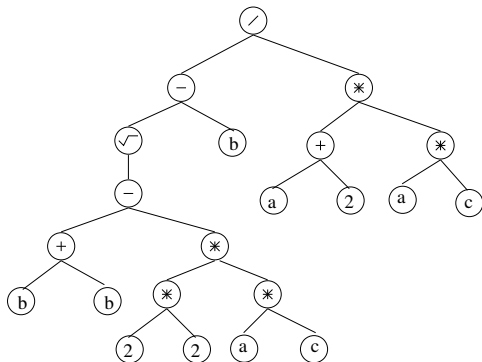
- Se selecciona aleatoriamente un símbolo terminal dentro del individuo

# MUTACIÓN TERMINAL SIMPLE

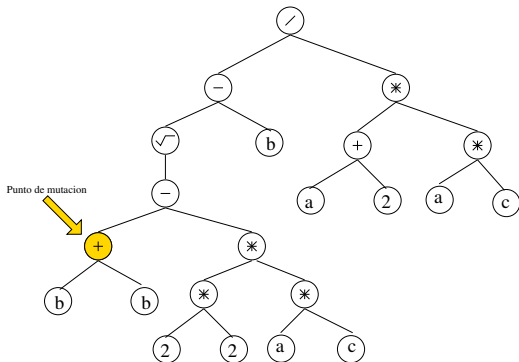


- Se selecciona aleatoriamente un símbolo terminal dentro del individuo
- Se sustituye por otro diferente del conjunto de símbolos terminales posibles

# MUTACIÓN FUNCIONAL SIMPLE

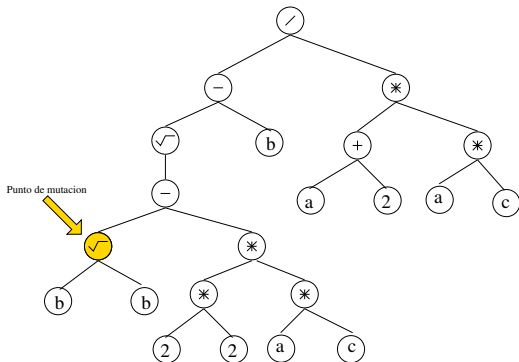


# MUTACIÓN FUNCIONAL SIMPLE



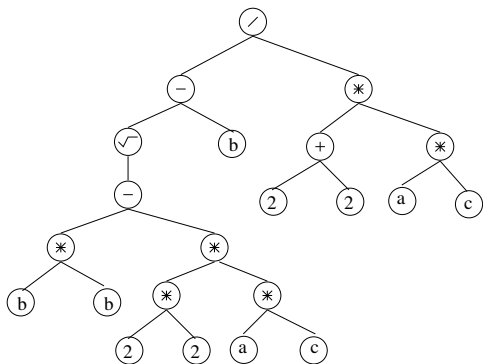
- Se selecciona aleatoriamente una función dentro del individuo

# MUTACIÓN FUNCIONAL SIMPLE



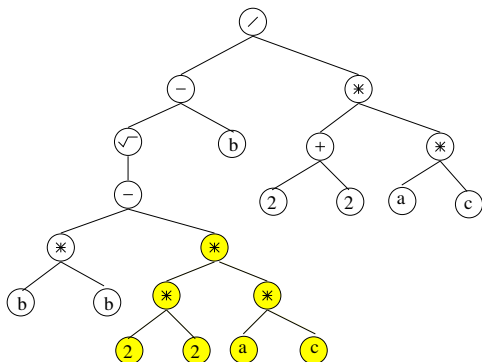
- Se selecciona aleatoriamente una función dentro del individuo
- Se sustituye por otra diferente del conjunto de funciones posibles

# MUTACIÓN DE ÁRBOL





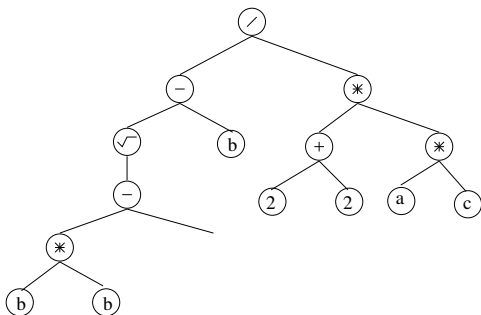
# MUTACIÓN DE ÁRBOL



Zona de mutacion

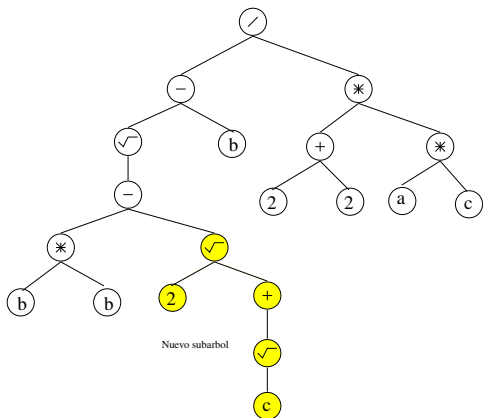
- Se selecciona un subárbol del individuo, igual que en el operador de sobrecruzamiento

# MUTACIÓN DE ÁRBOL



- Se selecciona un subárbol del individuo, igual que en el operador de sobrecruzamiento
- Se elimina totalmente el subárbol seleccionado

# MUTACIÓN DE ÁRBOL



- Se selecciona un subárbol del individuo, igual que en el operador de sobrecruzamiento
- Se elimina totalmente el subárbol seleccionado
- En su lugar se incorpora un nuevo subárbol generado aleatoriamente

# PROCEDIMIENTO

- 1 Generar una población inicial de programas aleatoriamente:

## PROCEDIMIENTO POBLACIÓN INICIAL

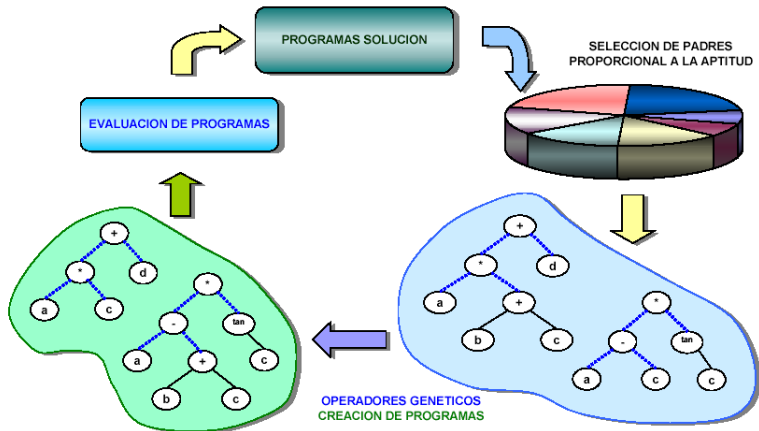
- 1 Se asignan probabilidades a la elección de terminal y función respectivamente
- 2 Se selecciona entre terminal y función mediante las probabilidades anteriores
- 3 Si se ha seleccionado función se elige entre las del conjunto de funciones, y se crea un nodo del árbol
- 4 Si se ha seleccionado terminal se elige entre los símbolos del conjunto de terminales, y se crea una hoja del árbol. Si el símbolo es una constante, se genera un valor para esa constante de forma aleatoria en el rango especificado

# PROCEDIMIENTO

Mientras no se cumpla el criterio de convergencia:

- 1 Ejecutar cada programa y calcular fitness.
- 2 Crear una población intermedia vacía
- 3 Copiar en ella los mejores individuos
- 4 Mientras la población intermedia no esté completa:
  - 1 Seleccionar un operador entre mutación y sobrecruzamiento
  - 2 Si se ha elegido mutación:
    - 1 Seleccionar un individuo aleatoriamente de la población original
    - 2 Crear un nuevo individuo por mutación del anterior
    - 3 Incluir el nuevo individuo en la población intermedia
  - 3 Si se ha elegido sobrecruzamiento:
    - 1 Seleccionar dos individuos aleatoriamente de la población original
    - 2 Crear dos nuevos individuos por sobrecruzamiento de los anteriores
    - 3 Incluir los nuevos individuos en la población intermedia
- 5 Sustituir a los individuos de la población por los individuos de la población intermedia

# ALGORITMO



# TEMARIO

- 1 INTRODUCCIÓN
- 2 ALGORITMOS GENÉTICOS
- 3 COMPUTACIÓN EVOLUTIVA**
  - Estrategias Evolutivas
  - Ejemplo de Estrategias Evolutivas
  - Programación Genética
  - Ejemplo de Programación Genética**
- 4 COMPUTACIÓN CON INSPIRACIÓN BIOLÓGICA
- 5 BIOLOGÍA Y COMPUTACIÓN

# REGRESIÓN NUMÉRICA

- Se dispone de un conjunto de observaciones de cierto proceso físico en el que se toman medidas de presión, de temperatura y del rendimiento del sistema
- Se desea saber si existe una ecuación matemática ( $f(x_1, x_2)$ ), y de cuál se trata, que permita obtener el rendimiento del sistema a partir de la presión y de la temperatura
- Dicha ecuación permitirá predecir el rendimiento, e incluso manejando las variables, poder asignar el rendimiento más eficaz



# REGRESIÓN NUMÉRICA

- **Terminales.** Las variables consideradas (temperatura y presión) y una constante numérica
- **Funciones.** Aquellos operadores que puedan formar parte de la solución. suma, resta, multiplicación, división y elevado
- Se generan los individuos aleatoriamente:

$$2x_1^{4-x_2} + x_1 \quad x_1^2 + 3,2$$

$$x_2^3 + 1,1 + x_1^2 \quad \frac{x_1^3 - 2x_1}{x_1^{x_2}}$$

- Se evalúan los individuos, en este caso su evaluación será la distancia, para cada punto de la muestra, entre el valor de la función que representa el individuo y el valor de la variable a predecir, la presión

# REGRESIÓN NUMÉRICA. EVALUACIÓN

Datos

$y^*$	$x_1$	$x_2$
0	3	2
0,5	2	1
1	1,5	0,6
2	3	0,5
2,5	8	0,1
5	8,5	-1

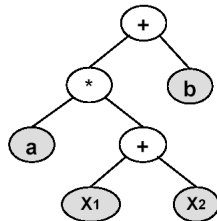


Modelo  
supuesto

$$y = a \cdot (x_1 + x_2) + b$$



representación  
en árbol



# REGRESIÓN NUMÉRICA. EVALUACIÓN

- La evaluación de los individuos se puede representar mediante la diferencia entre las curvas que representan a cada uno de ellos y la solución

a	0,5
b	-1

y	x1	x2	error
1,5	3	2	1,5
0,5	2	1	3E-11
0,05	1,5	0,6	0,95
0,75	3	0,5	1,25
3,05	8	0,1	0,55
2,75	8,5	-1	2,25

suma de errores	6,5
<b>aptitud</b>	0,1515

- 1 INTRODUCCIÓN
- 2 ALGORITMOS GENÉTICOS
- 3 COMPUTACIÓN EVOLUTIVA
  - Estrategias Evolutivas
  - Ejemplo de Estrategias Evolutivas
  - Programación Genética
  - Ejemplo de Programación Genética
- 4 COMPUTACIÓN CON INSPIRACIÓN BIOLÓGICA
- 5 **BIOLÓGÍA Y COMPUTACIÓN**