

# Programación evolutiva

## Facultad de Informática U.C.M. Curso 2018/2019

### Práctica 1

**Parte 1:** El objetivo de esta práctica es implementar un algoritmo genético clásico para hallar el máximo o mínimo de diferentes funciones.

#### Diseño del algoritmo:

- **Representación de los individuos:** se representan mediante cadenas binarias que se corresponden con los puntos del espacio de búsqueda.
- **Función de evaluación:** el fitness es el resultado de evaluar la función considerada en el punto que resulta de la decodificación del individuo.
- **Selección:** por ruleta, torneos, estocástico universal y otros métodos vistos en teoría. Se incluirá la opción para seleccionar elitismo.
- **Operadores:** cruce (monopunto, uniforme...) y **mutación** básica.

Consideramos la optimización de las siguientes funciones:

#### ▪ **Función 1:**

$$f(x_1, x_2) = 21.5 + x_1 \cdot \sin(4\pi x_1) + x_2 \cdot \sin(20\pi x_2) :$$

que presenta un máximo de 38.809 en 11.625 y 5.726  $x_1 \in [-3.0, 12.1]$   $x_2 \in [4.1, 5.8]$

#### ▪ **Función 2:**

$$f(\mathbf{x}) = -(x_2 + 47) \sin \left( \sqrt{\left| x_2 + \frac{x_1}{2} + 47 \right|} \right) - x_1 \sin \left( \sqrt{|x_1 - (x_2 + 47)|} \right)$$

que presenta un mínimo de -959.6407 en (512, 404.2319)  $x_i \in [-512, 512]$  para  $i=1..2$

#### ▪ **Función 3:**

$$f(x_i, i = 1..2) = \left( \sum_{i=1}^5 i \cdot \cos((i+1)x_1 + i) \right) \left( \sum_{i=1}^5 i \cdot \cos((i+1)x_2 + i) \right)$$

$x_i \in [-10, 10]$  que presenta 18 mínimos de -186.7309

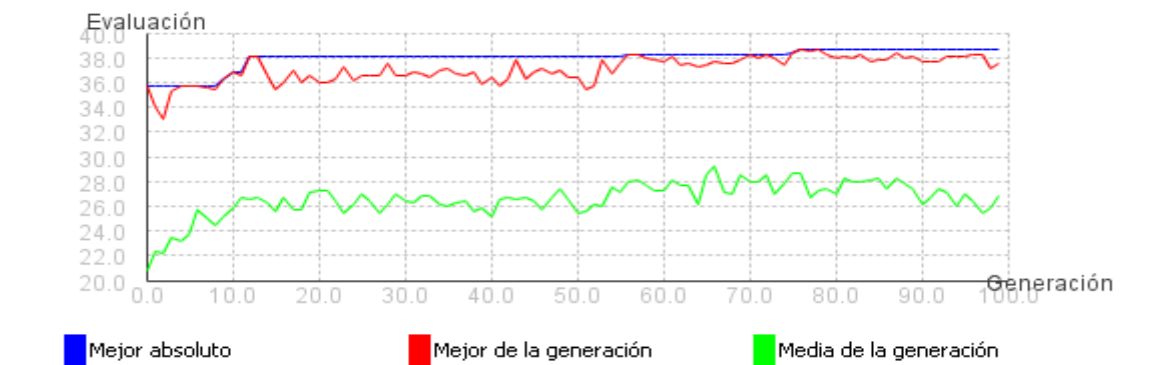
#### ▪ **Función 4:**

$$f(x_i | i = 1..n) = - \sum_{i=1}^n \sin(x_i) \sin^{20} \left( \frac{(i+1)x_i^2}{\pi} \right) : x_i \in [0, \pi]$$

que presenta los siguientes mínimos en función de  $n$  :

N	1	2	3	4	5	6	7
Mínimo	-1	-1.959091	-2.897553	-3.886358	-4.886358	-5.879585	-6.862457

- ❑ Parámetros del algoritmo: La aplicación debe permitir variar los parámetros interactivamente. Los parámetros son: tamaño de la población (100), número de generaciones (100), porcentaje de cruces (60%), porcentaje de mutaciones (5%), precisión o valor de error para la discretización del intervalo (0.001) y posibilidad de seleccionar con o sin elitismo.
- ❑ Representación gráfica de evolución: Representación gráfica en la que se muestre el máximo o mínimo encontrado por el algoritmo y los valores medio y máximo de la aptitud a lo largo de las generaciones para estudiar su evolución. (herramientas *jmathtools*, *jfreechart*...). Además de las gráficas se mostraran los valores obtenidos.



Se pueden incluir en la práctica cualquiera de las mejoras vistas en clase.

**Parte 2:** Se quiere ampliar la práctica para que la función 4 también soporte cromosomas con representación real. Ahora el cromosoma puede estar formado por números reales. Por ejemplo, para  $n=6$  variables, el cromosoma podría ser:

3.1241	2.7112	2.3454	0.3425	1.6832	2.9342
X1	X2	X3	X4	X5	X6

Como operadores de cruce se propone utilizar cruce de un punto, discreto uniforme, aritmético y SBX y Mutación a elegir.

## Entrega

- ❑ **Plazo de entrega: 27 de febrero 16:00.** Debes entregar por el campus un archivo comprimido con el código java de la aplicación (**proyecto en Eclipse o NetBeans**) que incluya una breve memoria que contenga el estudio de las gráficas y resultados obtenidos con cada función. Aquí se valorarán las conclusiones y observaciones que se consideren interesantes respecto al resultado obtenido. Nombre del proyecto-archivo: **G01P1** (por ejemplo, para el grupo 01)
- ❑ La **corrección** será en la sesión de Laboratorio del **1 de Marzo** y deberán estar presentes los dos miembros del grupo, a los que se evaluará por igual mediante una serie de preguntas. Es importante conocer bien la práctica y los aspectos teóricos en los que se basa, pues es lo que determina la calificación. El orden de corrección de cada grupo será el mismo que el orden de entrega por el campus.

Más funciones de prueba: <https://www.sfu.ca/~ssurjano/optimization.html>