

Información General

Curso : Cómputo Evolutivo
Semestre : 2025 - 2
Profesores : Katya Rodríguez Vázquez
: Augusto César Poot Hernández
Entrega : Febrero 23 de 2025
Alumno : Pablo Uriel Benítez Ramírez, 418003561

1. Algoritmo genético simple

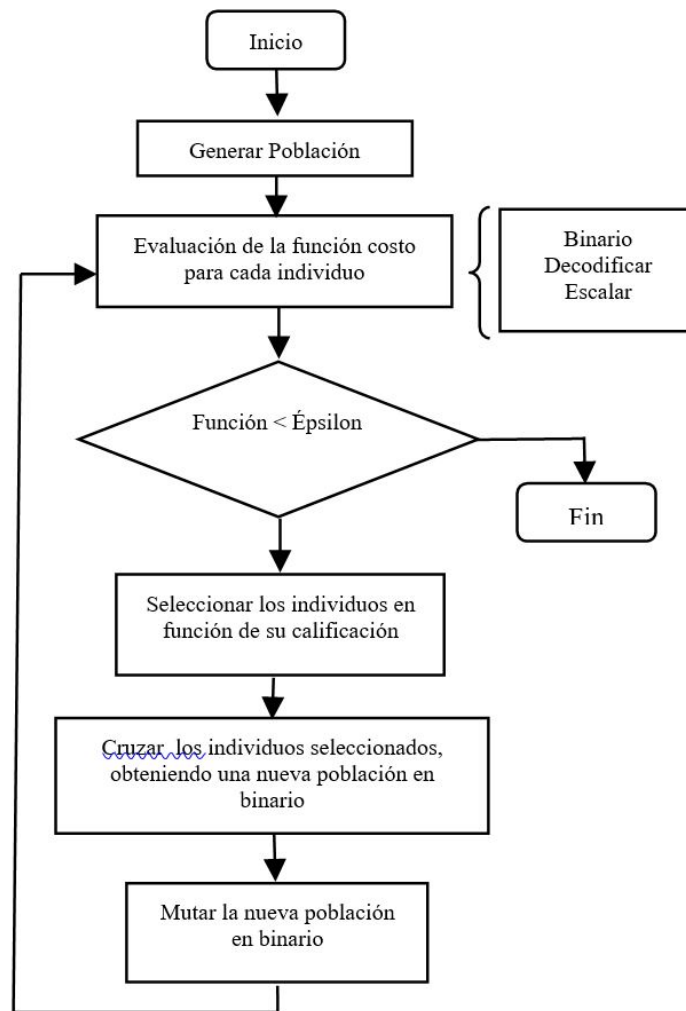


Figura 1: Diagrama de algoritmo para cómputo evolutivo [1]

Implementa el algoritmo genético simple con las siguientes características:

1. Codificación binaria, *representación de los cromosomas mediante una estructura de símbolos. Obtener la longitud entre el rango mínimo y máximo de cada variable.*
2. Selección, hacer una selección proporcional a la aptitud de los individuos. Los mejor evaluados tendrán mayores probabilidades de sobrevivir.

Definición 1.1. [2] La *evaluación* de los individuos se genera a partir de la *decodificación* del genotipo dada la siguiente fórmula:

$$x = a + decimal(g) \left(\frac{b - a}{2^m - 1} \right)$$

donde g es la cadena de 0's y 1's que representa a cada individuo, a y b son los valores que acotan a x ($a \leq x \leq b$), y m es la longitud del genotipo.

Definición 1.2. [2] La *selección* de individuos x 's de una población es de acuerdo al valor de *fitness* (adaptabilidad, aptitud, desempeño) $f(x)$. Hacer una selección proporcional a la aptitud de los individuos. Los mejor evaluados tendrán mayores probabilidades de sobrevivir.

Definición 1.3. La *aptitud* a_n se obtiene a partir de la evaluación de $f(x_1, \dots, x_n)$ para cada individuo

$$a_n = f_n \text{ o bien } a_n = \frac{1}{f_n + \epsilon}, \epsilon > 0 \text{ para otros casos}$$

La aptitud total de la población es la suma de total de aptitudes,

$$S = \sum_{i=1}^n a_n$$

Definición 1.4. La *probabilidad de selección* para cada individuo está dada por

$$p_n = \frac{a_n}{S}$$

Definición 1.5. La *aptitud acumulada* es un vector acumulativo donde cada n posición representa la suma acumulada de las probabilidades hasta el individuo n

$$acc_n = \sum_{j=1}^n p_j$$

Se consideraron dos funciones para la evaluación:

$$a_n = f(x) \quad x = (x_1, \dots, x_n) \quad (1)$$

$$a_n = \frac{1}{f(x) + \epsilon} \quad x = (x_1, \dots, x_n), \epsilon = 1e^{-6} \quad (2)$$

Definición 1.6. [2] El *método de ruleta* (Goldberg 1989), simple pero ineficiente: $O(n^2)$ [3]. cada individuo de la población recibe una probabilidad de ser seleccionado que es proporcional a su aptitud.

Para seleccionar a un individuo,

- a) Genera un número aleatorio $t \in [0, 1]$
- b) Recorre el vector acumulado hasta encontrar el primer índice n donde:

$$acc_n \geq t$$

Definición 1.7. El *método estocástico universal (SUS)* es una variante del método de ruleta (Definición 1.6) destinada a reducir el riesgo de convergencia prematura. Se puede implementar según el siguiente pseudocódigo:

```

SUS(Population, N)
  F := total fitness of Population
  N := number of offspring to keep
  P := distance between the pointers (F/N)
  Start := random number between 0 and P
  Pointers := [Start + i*P | i in [0..(N-1)]]
  return RWS(Population, Pointers)

RWS(Population, Points)
  Keep = []
  for P in Points
    I := 0
    while fitness sum of Population[0..I] < P
      I++
    add Population[I] to Keep
  return Keep

```

Definición 1.8. El *método de torneo binario* implica realizar varios "torneos" entre dos individuos elegidos al azar de la población. El ganador de cada torneo (el que tiene mejor condición física) es seleccionado para la cruce.

```

choose k (the tournament size) individuals from the population at random
choose the best individual from the tournament with probability p
choose the second best individual with probability p*(1-p)
choose the third best individual with probability p*((1-p)^2)
and so on

```

Figura 2: Caption

3. Cruza en un punto

Definición 1.9. [2] El *operador de cruzamiento* permite explotar el espacio de búsqueda al combinar nociones (subcadenas) para formar nuevas ideas (nuevas soluciones).

4. Mutación aleatoria

Definición 1.10. [2] La mutación es un proceso en el cual el alelo de un gen es aleatoriamente reemplazando por otro para producir una nueva estructura. La probabilidad de mutación P_m en cada gen es pequeña. La mutación de cada posición es independiente de la acción en otra posición.

5. La probabilidad de cruce, probabilidad de mutación, tamaño de la población y número de generación quedan a su elección.

6. Precisión de 3 dígitos decimales.

1.1. Esfera

($n = 2$ & $n = 5$) $-10 \leq x \leq 10$

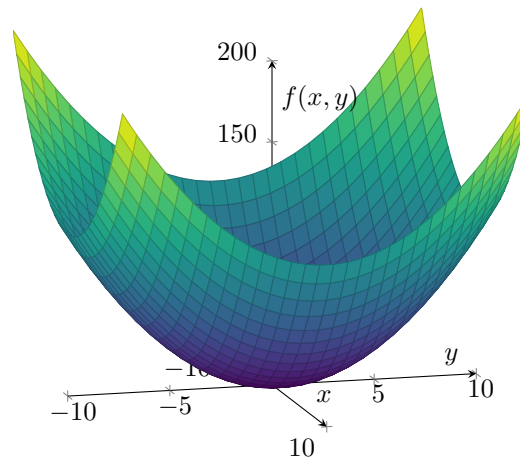
$$f(\mathbf{x}) = \sum_{i=1}^n x_i^2$$

1.1.1. $n=2$

Para $n = 2$ la función es

$$f(\mathbf{x}) = \sum_{i=1}^2 x_i^2 = x_1^2 + x_2^2, \quad f(x, y) = x^2 + y^2$$

la cual es un paraboloide de 3 dimensiones, que se ve así



Con un máximo $f(x, y) = 200$ y un mínimo de $f(x, y) = 0$.

1.1.2. n=5

Para $n = 5$ la función es

$$f(\mathbf{x}) = \sum_{i=1}^5 x_i^2 = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2$$

la cual es un hiperparaboloide en 5 dimensiones, que se ve como la anterior si dejamos fijas 3 variables.

Con un máximo $f(x, y) = 500$ y un mínimo de $f(x, y) = 0$.

1.1.3. Codificación binaria

El total de valores a representar se obtiene multiplicando la longitud por el requerimiento de precisión.

$$Longitud = 10 - (-10) = 20$$

$$L_{valores} = 20 \cdot 1000 = 20,000$$

Por lo tanto, la cantidad de bits necesarios para representar el total de valores es

$$2^{14} \leq 20,000 \leq 2^{15} \implies 2^{15} = 32,768$$

1.2. Esfera Ruleta

1.2.1. Evaluación n=2

Método de selección ruleta [1.6](#).

Parámetros

```
m = 15           # longitud del genotipo
pob_size = 50    # tamaño de la población
a = -10          # valor mínimo de x
b = 10           # valor máximo de x
decimales=3      # número de decimales
n = 2            # variables a considerar
t = m * n        # total de bits
epsilon = 1e-6   # cota de selección
proba_cruza = 0.8 # probabilidad de cruza
```

```

proba_muta = 0.02 # probabilidad de mutación
generaciones = 100 # número de generaciones
target = -10000    # objetivo de minimización

```

Función 1. Resultado:

Generaciones: 100

Mejor genotipo: [0 1 1 1 1 0 1 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1]

Mejor fenotipo : 0.168

Mejor evaluación $f(x)$: 0.168

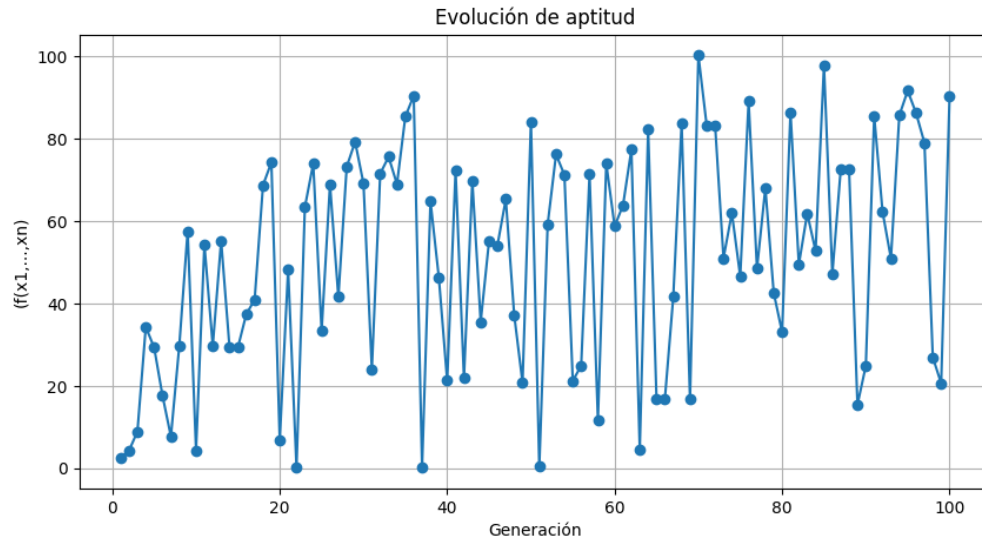


Figura 3: Representación de función 1

Función 2. Resultado:

Generaciones: 6

Mejor genotipo: [1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0]

Mejor fenotipo (x): 0.0

Mejor evaluación $f(x)$: 0.0

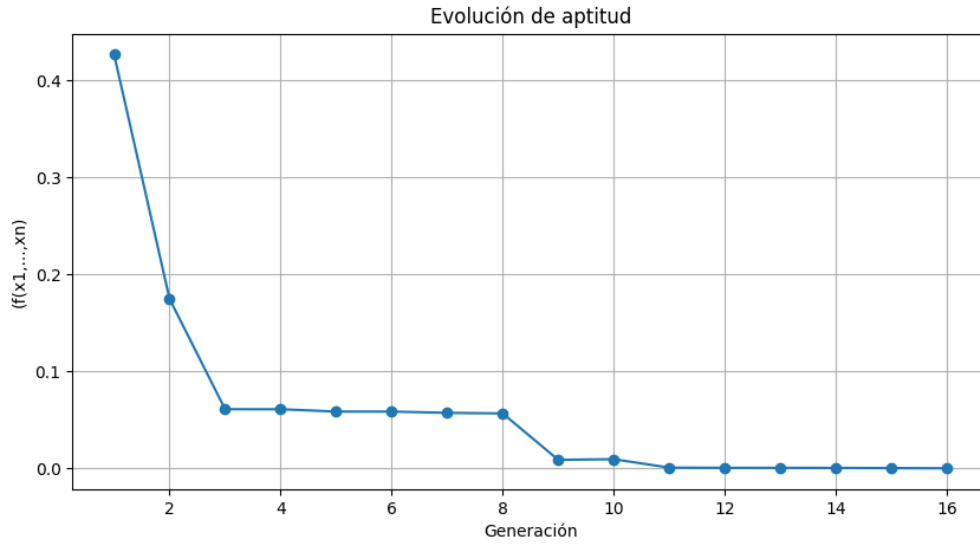


Figura 4: Representación de función 2

1.2.2. Evaluación n=5

Método de selección ruleta 1.6,

Parámetros

```

m = 15          # longitud del genotipo
pob_size = 50   # tamaño de la población
a = -10         # valor mínimo de x
b = 10          # valor máximo de x
decimales=3     # número de decimales
n = 5           # variables a considerar
t = m * n       # total de bits
epsilon = 1e-6  # cota de selección
proba_cruza = 0.8 # probabilidad de cruza
proba_muta = 0.02 # probabilidad de mutación
generaciones = 100 # número de generaciones
target = 0      # objetivo de minimización

```

Función 1. Resultado:

Generaciones: 100

Mejor genotipo: [1 0 0 1 0 0 0 0 1 0 1 1 1 1 0 1 0 1 0 1 0 0 1 1 1 0 0 1 1 0 1 1 0 0 1 0
1 1 1 1 1 1 0 0 1 0 0 0 1 0 1 0 0 0 1 1 0 1 1 0 1 1 1 1 0 0 0 0 0 1 0]

Mejor fenotipo (x): 19.578

Mejor evaluación f(x): 19.578

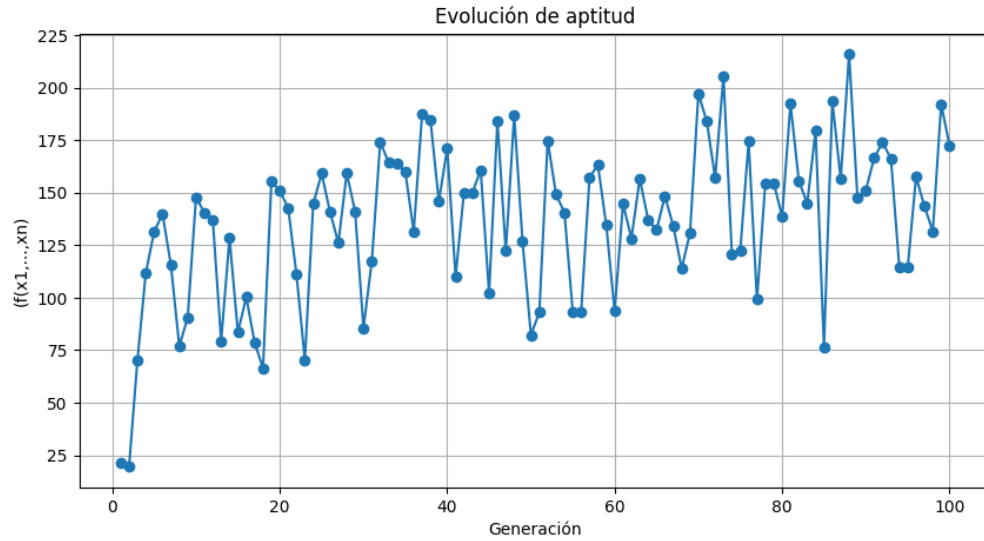


Figura 5: Representación de función 1

Función 2. Resultado:

Generaciones: 42

Mejor genotipo: [0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1
1 1 1 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 0 1 0]

Mejor fenotipo (x): 0.0

Mejor evaluación $f(x)$: 0.0

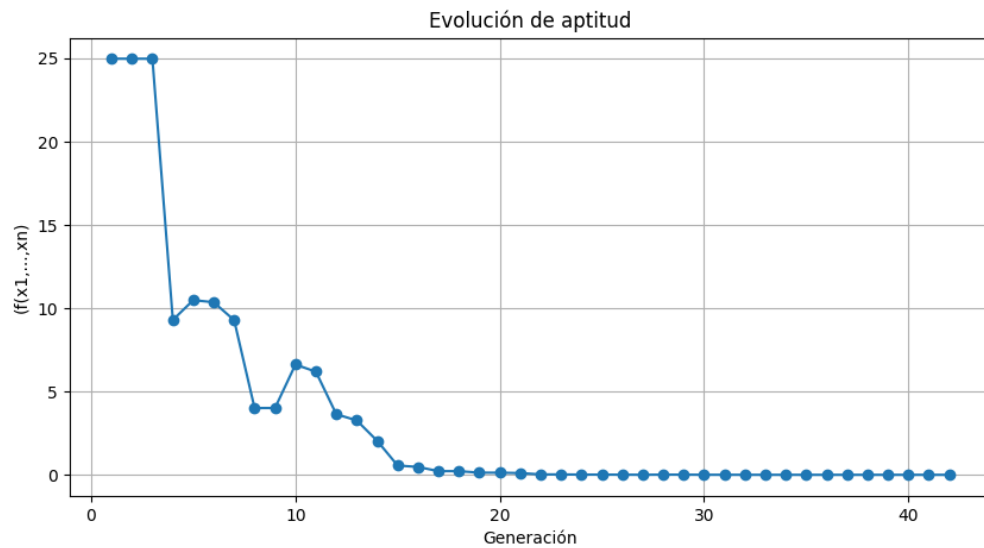


Figura 6: Representación de función 2

1.3. Esfera SUS

1.3.1. Evaluación $n=2$

Método de selección SUS 1.7.

Parámetros

```
m = 15                # longitud del genotipo
pob_size = 50          # tamaño de la población
a = -10                # valor mínimo de x
b = 10                 # valor máximo de x
decimales=3            # número de decimales
n = 2                  # variables a considerar
t = m * n              # total de bits
epsilon = 1e-6         # cota de selección
proba_cruza = 0.8      # probabilidad de cruce
proba_muta = 0.02      # probabilidad de mutación
generaciones = 100     # número de generaciones
target = 0             # objetivo de minimización
```

Función 2. Resultado:

Generaciones: 8

Mejor genotipo: [1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 1 1 1 1 1 1 1 1 0 0 0 0]

Mejor fenotipo (x1,...,xn): [0.006 -0.009]

Mejor evaluación $f(x_1, \dots, x_n)$: 0.0

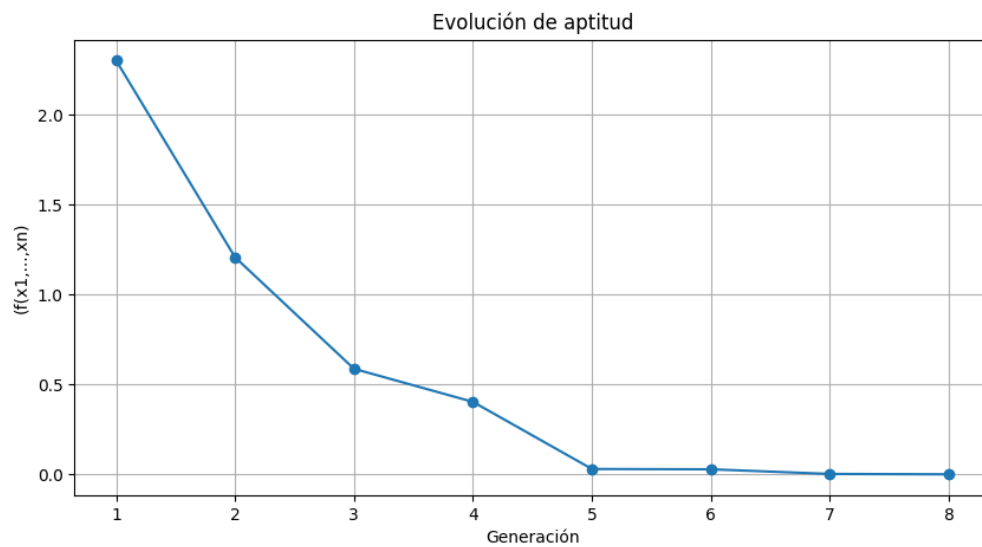


Figura 7: Representación de función 2

1.3.2. Evaluación n=5

1.6

Parámetros

```
m = 15                # longitud del genotipo
pob_size = 50          # tamaño de la población
a = -10                # valor mínimo de x
b = 10                 # valor máximo de x
decimales=3            # número de decimales
n = 5                  # variables a considerar
t = m * n              # total de bits
```



```

epsilon = 1e-6      # cota de selección
proba_cruza = 0.8   # probabilidad de cruza
proba_muta = 0.02   # probabilidad de mutación
generaciones = 100  # número de generaciones
target = 0          # objetivo de minimización

```

Función 2. Resultado:

Generaciones: 77

Mejor genotipo: [1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 1 1 1 1
1 1 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 1 0 1 1]

Mejor fenotipo (x1,...,xn): [0.003 0.003 -0.001 0.006 0.007]

Mejor evaluación $f(x_1, \dots, x_n)$: 0.0

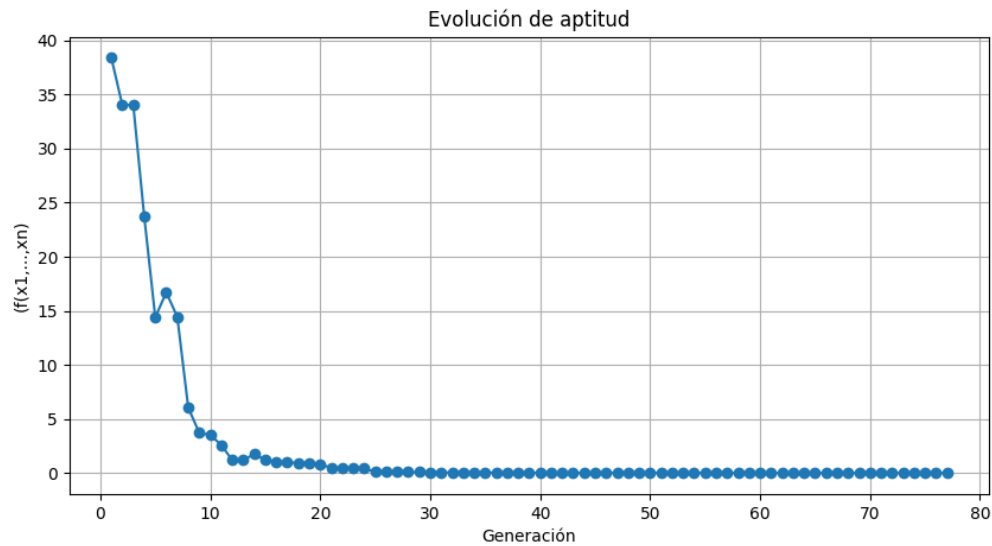


Figura 8: Representación de función 2

1.4. Esfera torneo

1.4.1. Evaluación n=2

Método de selección torneo 1.8.

Parámetros

```

m = 15                # longitud del genotipo
pob_size = 50         # tamaño de la población
a = -10               # valor mínimo de x
b = 10                # valor máximo de x
decimales=3           # número de decimales
n = 2                 # variables a considerar
t = m * n              # total de bits
epsilon = 1e-6        # cota de selección
proba_cruza = 0.8     # probabilidad de cruza
proba_muta = 0.02     # probabilidad de mutación
generaciones = 100    # número de generaciones
target = 0            # objetivo de minimización

```

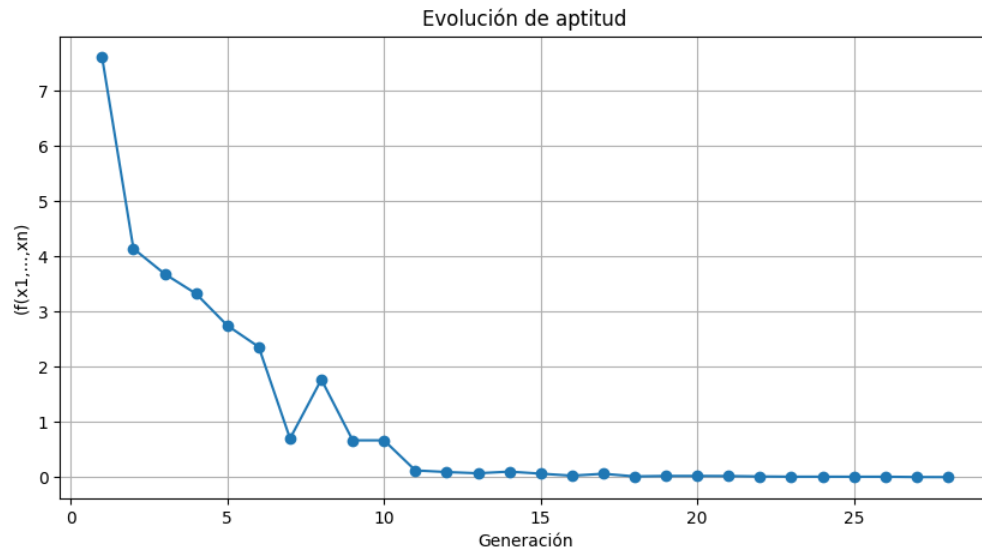
Resultado:

Generaciones: 28

Mejor genotipo: [1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0 1 0 1]

Mejor fenotipo (x1,...,xn): 0.000205 3

Mejor evaluación $f(x_1, \dots, x_n)$: 0.0



1.4.2. Evaluación n=5

Método de selección torneo [1.8](#).

Parámetros

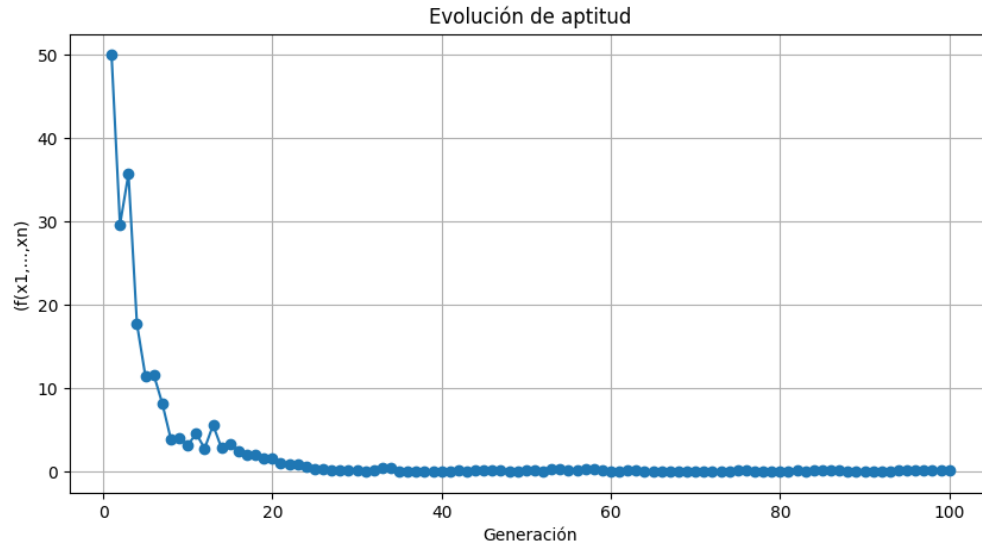
```
m = 15                # longitud del genotipo
pob_size = 50          # tamaño de la población
a = -10                # valor mínimo de x
b = 10                 # valor máximo de x
decimales=3           # número de decimales
n = 5                  # variables a considerar
t = m * n              # total de bits
epsilon = 1e-6         # cota de selección
proba_cruza = 0.8      # probabilidad de cruce
proba_muta = 0.02      # probabilidad de mutación
generaciones = 100    # número de generaciones
target = 0             # objetivo de minimización
```

Generaciones: 100

Mejor genotipo: [1 0 0 0 0 0 0 0 1 0 1 0 1 1 1 0 1 1 1 1 1 1 0 0 0 0 1 0 0 0 1 1 1 1 1 1
1 1 1 1 0 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 0 0 0 0 1 1 0 1 1 0
0]

Mejor fenotipo (x1,...,xn): 0.01284 3

Mejor evaluación $f(x_1, \dots, x_n)$: 0.013



1.5. Esfera Vasconcelos

1.5.1. Evaluación n=2

Método de selección Vasconcelos [1.8](#).

Parámetros

```

m = 15                # longitud del genotipo
pob_size = 50         # tamaño de la población
a = -10               # valor mínimo de x
b = 10                # valor máximo de x
decimales=3           # número de decimales
n = 2                 # variables a considerar
t = m * n             # total de bits
epsilon = 1e-6        # cota de selección
proba_cruza = 0.8     # probabilidad de cruza
proba_muta = 0.02     # probabilidad de mutación
generaciones = 100    # número de generaciones
target = 0            # objetivo de minimización

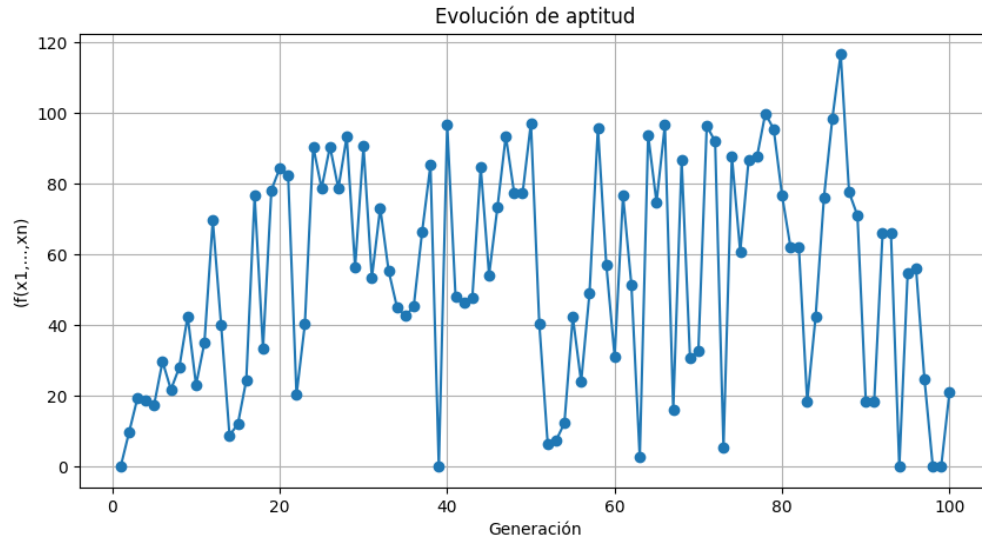
```

Resultado:

```

Generaciones: 100
Mejor genotipo: [1 0 0 0 0 0 0 1 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1]
Mejor fenotipo (x1,...,xn): 0.020241999999999996 3
Mejor evaluación f(x1,...,xn): 0.02

```



1.5.2. Evaluación $n=5$

Método de selección Vasconcelos [1.8](#).

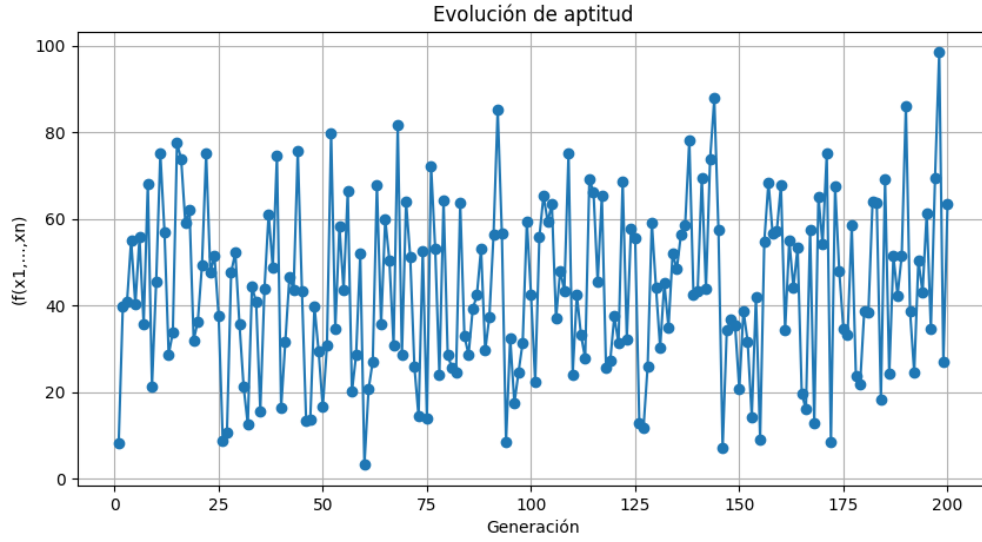
Parámetros

```
m = 15           # longitud del genotipo
pob_size = 50     # tamaño de la población
a = -10          # valor mínimo de x
b = 10           # valor máximo de x
decimales=3      # número de decimales
n = 5            # variables a considerar
t = m * n        # total de bits
epsilon = 1e-6   # cota de selección
proba_cruza = 0.8 # probabilidad de cruce
proba_muta = 0.2 # probabilidad de mutación
generaciones = 200 # número de generaciones
target = 0       # objetivo de minimización
```

Generaciones: 200

Mejor genotipo: [0 1 1 1 1 0 1 0 0 1 1 1 0 1 0 1 1 1 0 0 1 0 1 1 0 0 0 0 1 0 1 1 1 0 0
0 1 1 1 1 1 0 1 0 1 1 1 0 0 0 1 0 1 0 1 0 1 0 0 1 1 1 0 1 1 0 0 0 1 1 0 0 1]

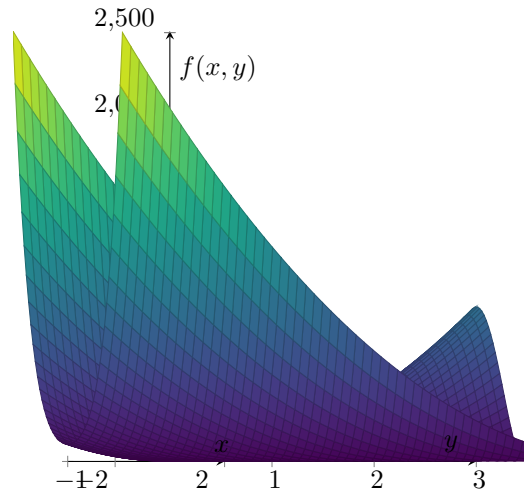
Mejor evaluación $f(x_1, \dots, x_n)$: 3.316



1.6. Rosenbrock

$(n = 2 \text{ \& } n = 5) \quad -10 \leq x \leq 10$

$$f(\mathbf{x}) = \sum_{i=1}^{n-1} \left[100 (x_{i+1} - x_i^2)^2 + (1 - x_i)^2 \right]$$



1.6.1. Codificación binaria

El total de valores a representar se obtiene multiplicando la longitud por el requerimiento de precisión.

$$Longitud = 10 - (-10) = 20$$

$$L_{valores} = 20 \cdot 1000 = 20,000$$

Por lo tanto, la cantidad de bits necesarios para representar el total de valores es

$$2^{14} \leq 20,000 \leq 2^{15} \implies 2^{15} = 32,768$$

A ojo se puede ver que el mínimo es cuando $x=0$.

1.6.2. Evaluación n=2

Parámetros

```
m = 15          # longitud del genotipo
pob_size = 50    # tamaño de la población
a = -10          # valor mínimo de x
b = 10           # valor máximo de x
decimales=3      # número de decimales
n = 2            # variables a considerar
t = m * n        # total de bits
epsilon = 1e-6   # cota de selección
proba_cruza = 0.8 # probabilidad de cruza
proba_muta = 0.02 # probabilidad de mutación
generaciones = 100 # número de generaciones
target = 0       # objetivo de minimización
```

Función 1. Resultado:

Generaciones: 100

Mejor genotipo: [1 0 0 0 0 1 0 0 0 0 1 1 1 0 0 1 0 0 0 0 1 1 0 0 1 0 1 1 0]

Mejor fenotipo (x): 2.384

Mejor evaluación f(x): 2.384

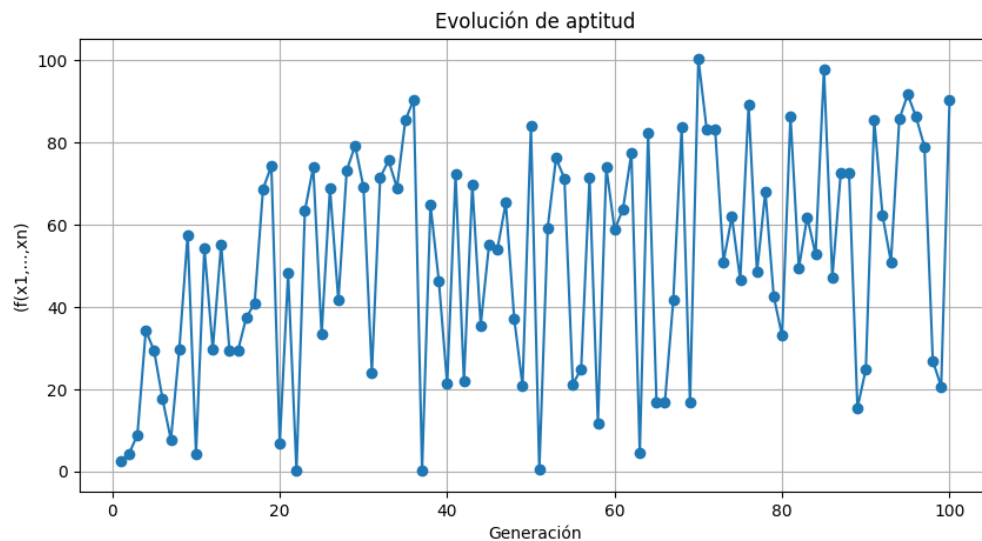


Figura 9: Representación de función 1

Función 2. Resultado:

Generaciones: 100

Mejor genotipo: [1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 1 1 1 1 1 1 1 0]

Mejor fenotipo (x): 0.063

Mejor evaluación f(x): 0.063

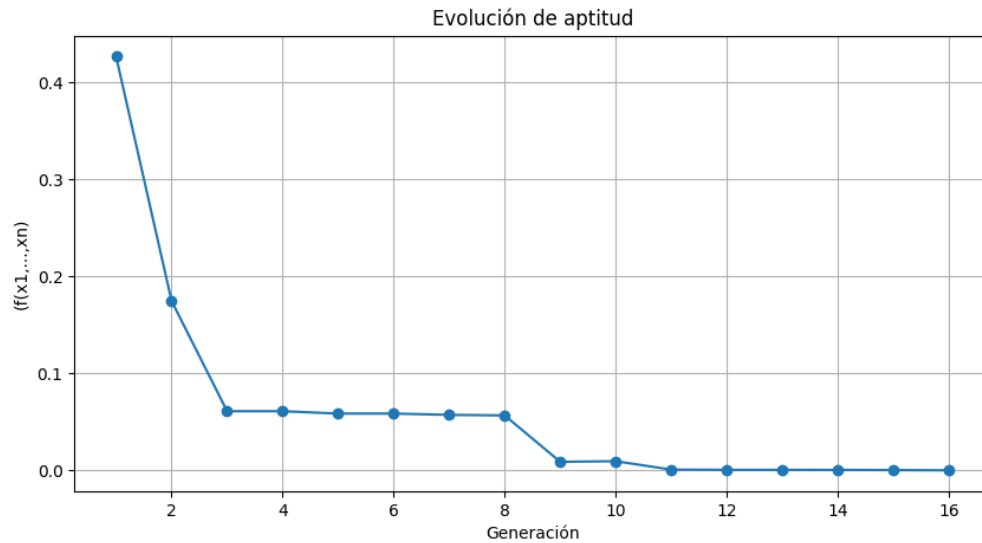


Figura 10: Representación de función 2

1.6.3. Evaluación n=5

Parámetros

```

m = 15           # longitud del genotipo
pob_size = 50    # tamaño de la población
a = -10          # valor mínimo de x
b = 10           # valor máximo de x
decimales=3      # número de decimales
n = 5            # variables a considerar
t = m * n        # total de bits
epsilon = 1e-6   # cota de selección
proba_cruza = 0.8 # probabilidad de cruce
proba_muta = 0.02 # probabilidad de mutación
generaciones = 100 # número de generaciones
target = 0        # objetivo de minimización

```

Función 1. Resultado:

Generaciones: 100

Mejor genotipo: [0 1 0 0 1 1 1 1 0 0 1 0 0 1 1 1 0 0 0 0 0 1 1 0 1 0 1 0 1 0 0 1 0 1 0 1 1
0 1 0 1 0 0 1 1 0 1 1 0 0 1 1 1 0 0 1 1 0 0 1 0 1 1 1 0 0 1 0 0 1 0 1 1 1 1]

Mejor fenotipo (x): 39219.962

Mejor evaluación f(x): 39219.962

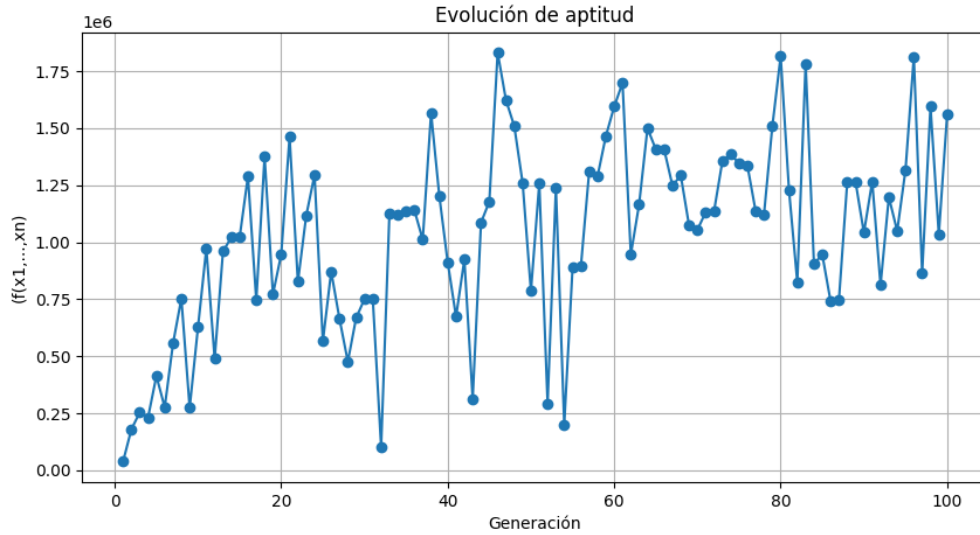


Figura 11: Representación de función 1

Función 2. Resultado:

Generaciones: 100

Mejor genotipo: [1 0 0 0 1 1 1 0 0 1 0 1 0 0 0 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 1 0 0 1
1 1 1 1 1 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 1 0 1 1 1 0 1 0 0 0 0 0 0 1 0 1 0 0]

Mejor fenotipo (x): 3.175

Mejor evaluación f(x): 3.175

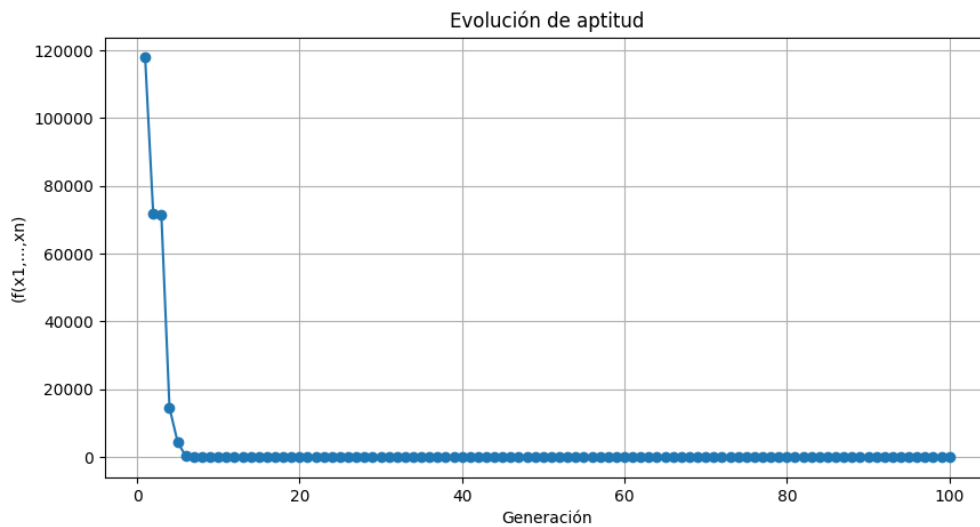
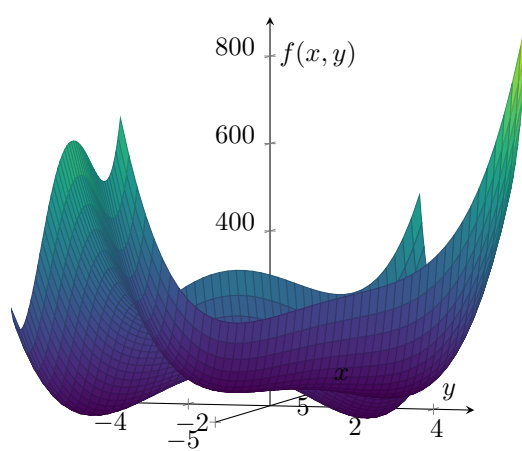


Figura 12: Representación de función 2

1.7. Himmenblau

$$-5 \leq x \leq 5$$

$$f(x, y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2.$$



1.7.1. Codificación binaria

El total de valores a representar se obtiene multiplicando la longitud por el requerimiento de precisión.

$$Longitud = 5 - (-5) = 10$$

$$L_{valores} = 10 \cdot 1000 = 10,000$$

Por lo tanto, la cantidad de bits necesarios para representar el total de valores es

$$2^{13} \leq 10,000 \leq 2^{14} \implies 2^{14} = 16,384$$

Parámetros

```

m = 14           # longitud del genotipo
pob_size = 50    # tamaño de la población
a = -5           # valor mínimo de x
b = 5            # valor máximo de x
decimales=3      # número de decimales
n = 2            # variables a considerar
t = m * n        # total de bits
epsilon = 1e-6   # cota de selección
proba_cruza = 0.8 # probabilidad de cruza
proba_muta = 0.02 # probabilidad de mutación
generaciones = 100 # número de generaciones
target = 0        # objetivo de minimización

```

Función 1. Resultado:

```

Generaciones: 100
Mejor genotipo: [1 1 0 0 1 1 1 0 1 1 1 1 0 0 1 0 1 0 1 0 1 1 0 1 1 1 1 0]
Mejor fenotipo (x1, x2): 1.11
Mejor evaluación f(x1,x2): 1.11

```

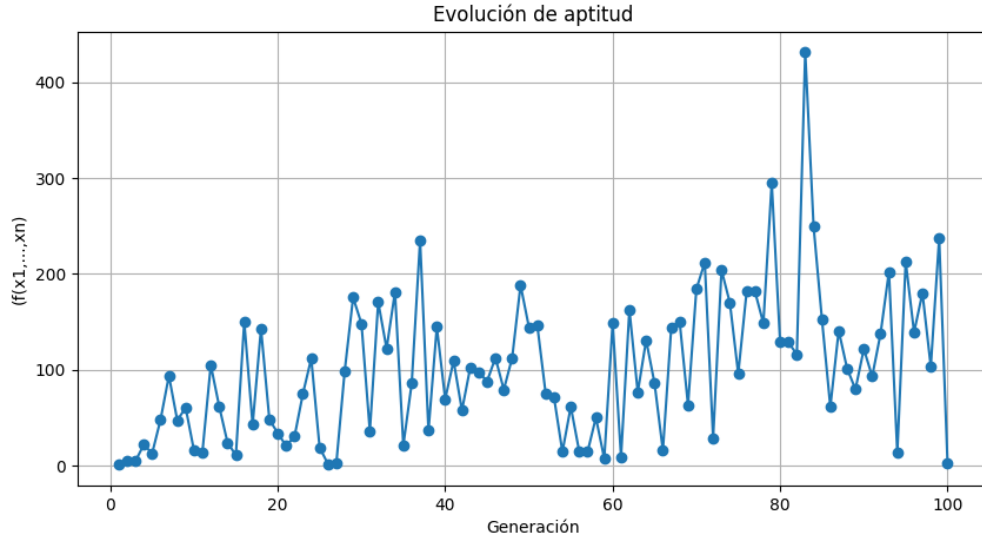


Figura 13: Representación de función 1

Función 2. Resultado:

Generaciones: 100

Mejor genotipo: [1 1 0 0 1 0 1 1 1 1 1 1 1 1 0 1 1 0 0 1 1 1 0 1 0 1 0]

Mejor fenotipo (x1, x2): 0.03

Mejor evaluación $f(x_1, x_2)$: 0.03

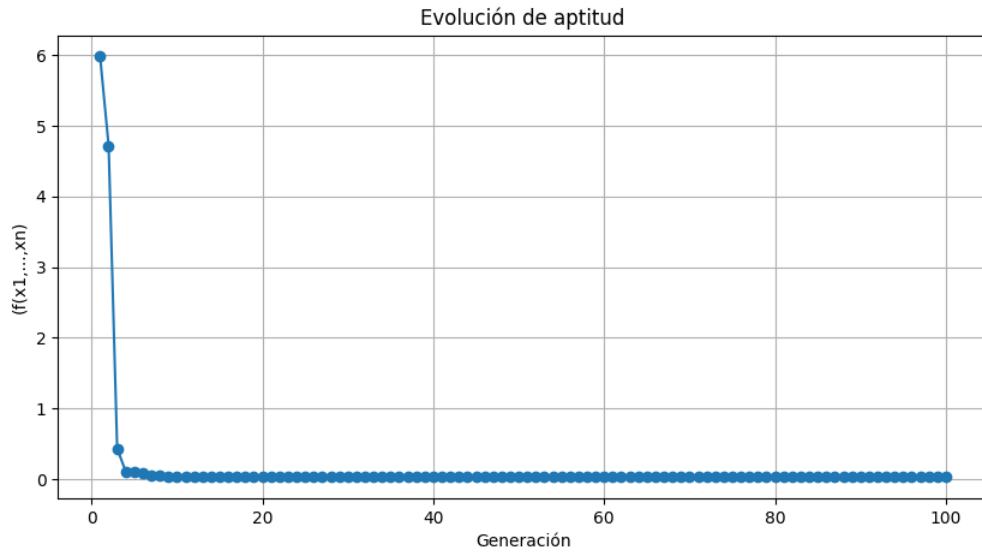


Figura 14: Representación de función 2

1.8. Eggholder

$$-512 \leq x \leq 512$$

$$f(x, y) = -(y + 47) \sin \sqrt{\left| \frac{x}{2} + (y + 47) \right|} - x \sin \sqrt{|x - (y + 47)|}$$

1.8.1. Codificación binaria

El total de valores a representar se obtiene multiplicando la longitud por el requerimiento de precisión.

$$\begin{aligned} \text{Longitud} &= 512 - (-512) = 1024 \\ L_{\text{valores}} &= 1024 \cdot 1000 = 1,024,000 \end{aligned}$$

Por lo tanto, la cantidad de bits necesarios para representar el total de valores es

$$2^{19} \leq 1,024,000 \leq 2^{20} \implies 2^{20} = 1,048,576$$

Parámetros Ruleta

```
m = 20           # longitud del genotipo
pob_size = 50     # tamaño de la población
a = -512          # valor mínimo de x
b = 512           # valor máximo de x
decimales=3       # número de decimales
n = 2             # variables a considerar
t = m * n         # total de bits
epsilon = 1e-6    # cota de selección
proba_cruza = 0.8  # probabilidad de cruza
proba_muta = 0.02  # probabilidad de mutación
generaciones = 100 # número de generaciones
target = -1000     # objetivo de minimización
```

Función 1. Resultado:

Generaciones: 100

Mejor genotipo: [1 1 1 1 1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 1 1 1 0 1 1 0 0 1 1 1 0 0 1 1 0 0 0 0 0]

Mejor fenotipo (x1, x2): -954.663

Mejor evaluación f(x1,x2): -954.663

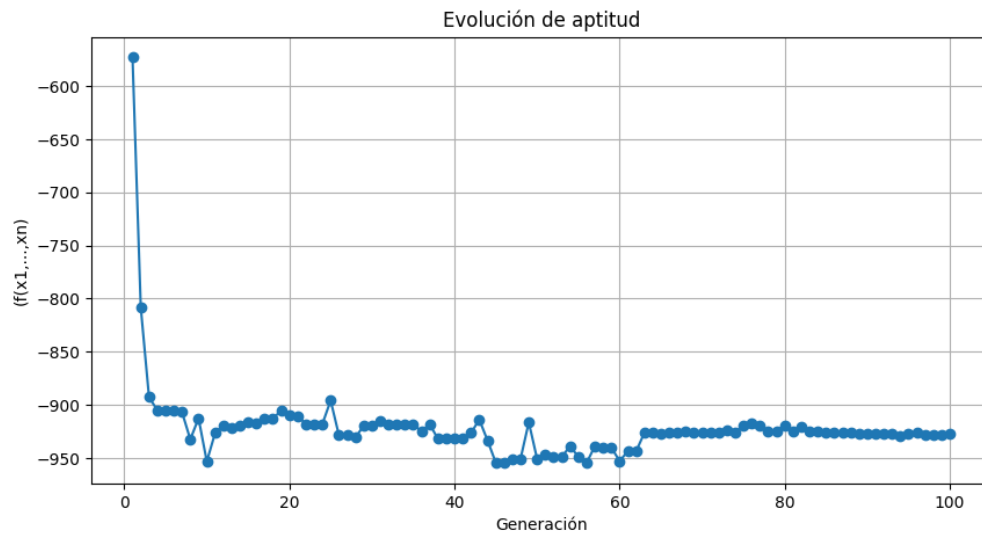


Figura 15: Representación de función 1

Parámetros SUS

```
m = 20                # longitud del genotipo
pob_size = 50         # tamaño de la población
a = -512              # valor mínimo de x
b = 512               # valor máximo de x
decimales=3           # número de decimales
n = 2                 # variables a considerar
t = m * n             # total de bits
epsilon = 1e-6        # cota de selección
proba_cruza = 0.8     # probabilidad de cruce
proba_muta = 0.02     # probabilidad de mutación
generaciones = 200    # número de generaciones
target = -1000        # objetivo de minimización
```

Función 1. Resultado:

Generaciones: 200

Mejor genotipo: [1 1 1 1 0 1 0 1 1 0 1 1 1 0 0 0 1 0 1 1 1 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0]

Mejor fenotipo (x1,...,xn): [470.943 426.003]

Mejor evaluación f(x1,...,xn): -937.865

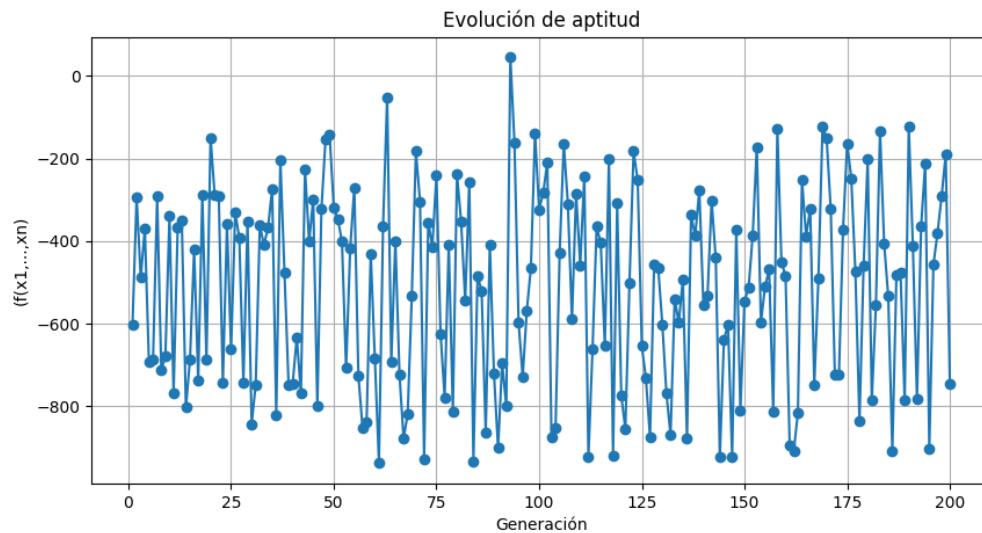


Figura 16: Representación de función 1

Parámetros torneo

```
m = 20                # longitud del genotipo
pob_size = 50         # tamaño de la población
a = -512              # valor mínimo de x
b = 512               # valor máximo de x
decimales=3           # número de decimales
n = 2                 # variables a considerar
t = m * n             # total de bits
epsilon = 1e-6        # cota de selección
proba_cruza = 0.8     # probabilidad de cruce
proba_muta = 0.02     # probabilidad de mutación
```

```

generaciones = 200 # número de generaciones
target = -1000      # objetivo de minimización

```

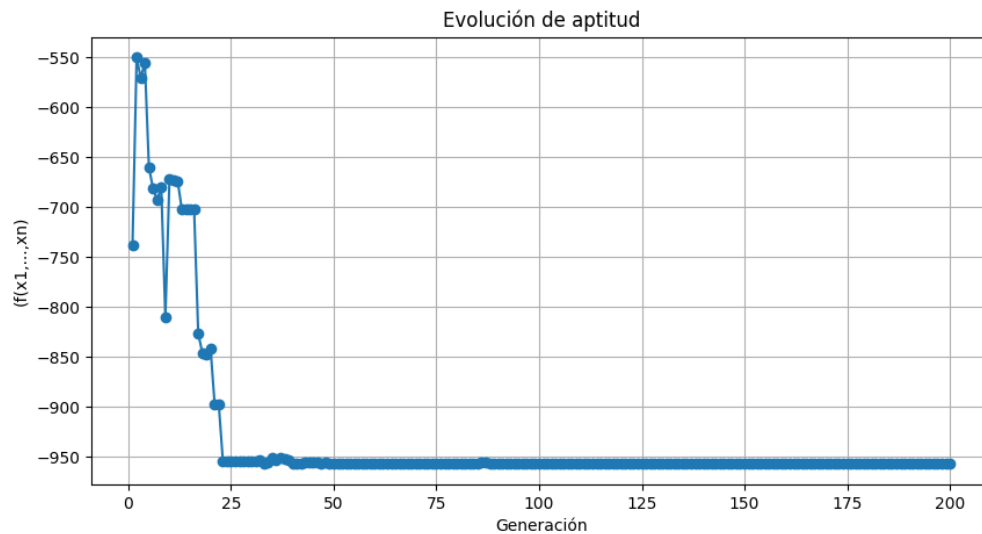
Resultado:

Generaciones: 200

Mejor genotipo: [1 1 1 1 1 0 0 0 1 0 0 1 0 1 1 1 0 0 1 1 1 0 1 1 0 0 0 0 1 1 0 1 1 1 1 0 0 0]

Mejor fenotipo (x1,...,xn): -956.9181651517163 3

Mejor evaluación f(x1,...,xn): -956.918



Parámetros torneo

```

m = 20                # longitud del genotipo
pob_size = 50         # tamaño de la población
a = -512              # valor mínimo de x
b = 512               # valor máximo de x
decimales=3          # número de decimales
n = 2                 # variables a considerar
t = m * n             # total de bits
epsilon = 1e-6        # cota de selección
proba_cruza = 0.8     # probabilidad de cruce
proba_muta = 0.02     # probabilidad de mutación
generaciones = 200    # número de generaciones
target = -1000        # objetivo de minimización

```

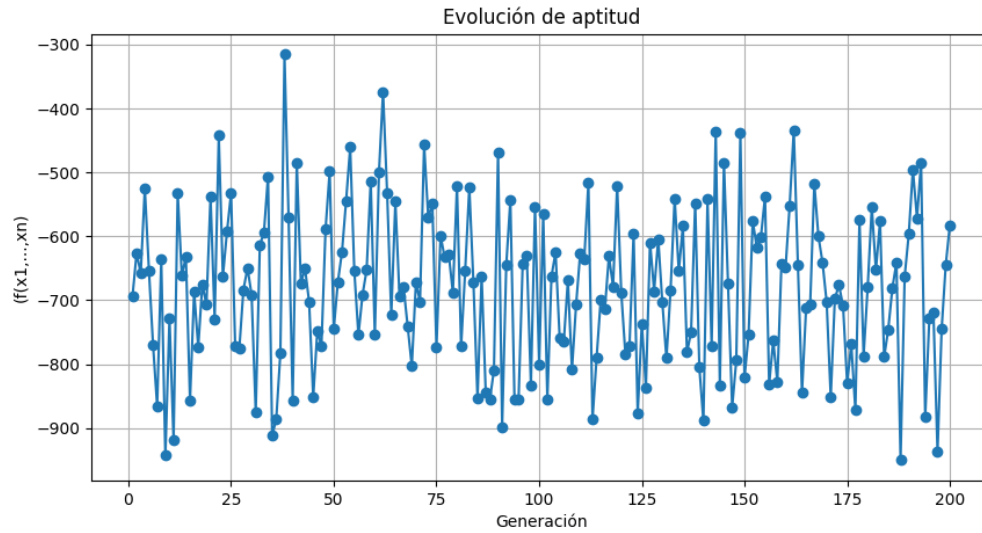
Resultado:

Generaciones: 200

Mejor genotipo: [1 1 1 1 1 0 0 0 0 0 1 0 1 1 1 1 0 0 1 1 1 1 0 1 1 0 1 0 0 1 0 1 0 0 1 0 1 1 1]

Mejor fenotipo (x1,...,xn): -949.9755275204257 3

Mejor evaluación f(x1,...,xn): -949.976



Referencias

- [1] Othon Colorado Arellano. Algoritmo genético aplicado a la sintonización de un controlador pid para un sistema acoplado de tanques. <https://portal.amelica.org/ameli/journal/595/5952866012/html/>. [Acceso el 23/02/2025].
- [2] Katya Rodríguez Vázquez Augusto C. Poot. Computación evolutiva. Curso Cómputo Evolutivo.
- [3] Carlos A. Coello Coello. Introducción a la computación evolutiva. <http://delta.cs.cinvestav.mx/~ccoello/compevol/clase5-2013.pdf>. [Acceso el 23/02/2025].