



## **Inteligencia Artificial**

### **Implementación problemas de búsqueda**

**Eliecer Ureche Torres 2019214060**

**Airton Sampayo 2020114027**

**Keyner Barrios 2019214046**

**Juan Diego Marin 2020114021**

**Universidad Magdalena**

**Ingeniería de sistemas  
Santa Marta, Colombia**

**D.T.C.H**

**2024**

## Esquema de representación

La función EggHolder es un conocido problema de optimización utilizado en matemáticas y ciencia de la computación para evaluar y colocar a prueba algoritmos de optimización. Esta función se utiliza como caso de prueba para la búsqueda de mínimos locales en un espacio multidimensional. Su nombre “EggHolder”, se deriva de la forma de la función, que se asemeja a un soporte para huevos(vea figura 1), con múltiples “cuencas” en la que se buscan los mínimos locales.[2]

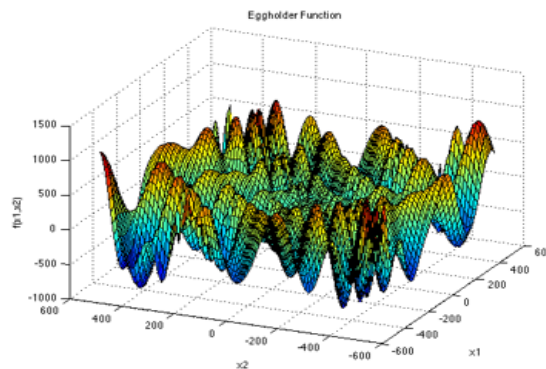


Figura 1. EggHolder

La función EggHolder se define matemáticamente como:

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

Esta función se encuentra en dos dimensiones y debido a su forma característica, con varias depresiones en forma de cuenco, resulta difícil de optimizar debido a la presencia de múltiples mínimos locales.

**Dominio de entrada:** generalmente se evalúa sobre el cuadrado  $X_i \in [-512, 512]$  para todo  $i = 1, 2$ .

**Objetivo:** El objetivo es encontrar el punto  $(x, y)$  que minimice el valor de la función  $f(x, y)$ , el mínimo global que es el que se espera encontrar de esta función es:

$$f(\mathbf{x}^*) = -959.6407, \text{ at } \mathbf{x}^* = (512, 404.2319)$$

## Estrategia para generar nuevas soluciones

### 1 Estrategias de evolución

#### I Definir las Funciones:

Función objetivo:

Para lo cual se usó la función asignada, **EggHolder**, se estableció de la siguiente manera:

```
fit = lambda x, y: -(y + 47) * math.sin(math.sqrt(abs(x/2+(y + 47)))) -
x * math.sin(abs(x - (y +47)))
```

Función de evaluación:

```
def eval_fitness(population):
    return [fit(x, y) for x, y in population]
```

Encargada de **evaluar el fitness** (calidad) de cada individuo en la población. calcula el **valor** de la función **fit** para cada par(x,y) en la población y devuelve un lista con los valores de **fitness**.

Función de Mutación:

```
def mutar(parent, sigma):
    x, y = parent
    if sigma <= epsilon:
        sigma = epsilon
    return (x + sigma * random.gauss(0, 0.5), y + sigma * random.gauss(0, 0.5))
```

Esta función introduce mutación en un individuo padre. Se agrega un pequeño valor aleatorio a cada variable del padre, teniendo en cuenta el parámetro sigma, que controla la magnitud de la mutación.

Función de Cruce o recombinación:

```
def recombinarLocal(parent1, parent2):
    numRandom = random.uniform(0,1)
    return ((parent1[0] * numRandom) + (parent2[0] * (1 - numRandom)),
            (parent1[1] * numRandom) + (parent2[1] * (1 - numRandom)))
```

Esta función se realiza recombinación local entre dos individuos padres. Promedia ponderadamente los valores de cada variable de los padres para generar un nuevo individuo descendiente.

## II Establecer variables:

**epsilon:** Valor mínimo permitido para sigma.

**n:** Dimensión del problema (en este caso, 2 variables).

**t:** Parámetro usado para el cálculo inicial de sigmaAct.

**sigmaIni:** Valor inicial para sigmaAct.

**sigmaAct:** Valor actual de la desviación estándar usada en la mutación. Se ajusta en cada generación.

**miu:** Tamaño de la población de padres.

**lda:** Tamaño total de la población (padres + descendientes).

**gens:** Número máximo de generaciones.

**start y end:** Límites del rango de búsqueda para las variables.

**sesenta:** Número de descendientes creados por mutación.

**cuarenta:** Número de descendientes creados por recombinación local.

**random.seed(1):** Fija la semilla para la generación de números aleatorios, permitiendo

resultados replicables.

**parents:** Lista que contiene los individuos de la población de padres.

**MejorMin:** Mejor valor de fitness encontrado hasta el momento.

**MejorXY:** Coordenadas del mejor individuo encontrado (valores de x e y).

**MejorIteracion:** Generación en la que se encontró el mejor individuo.

**HistorialFitness:** Lista que guarda el mejor valor de fitness en cada generación.

### III Uso de funciones:

```
for i in range(gens):
    # offspringsRecom = [recombinarGlobal(parents) for _ in range(cuarenta)]
    offspringsRecom = [recombinarLocal(random.choice(parents), random.choice(parents)) for _ in range(cuarenta)]
    # offspringsRecom = [recombinarDiscreta(parents) for _ in range(cuarenta)]
    offspringsMutar = [mutar(random.choice(parents), sigmaAct) for _ in range(sesenta)]
    population = parents + offspringsMutar + offspringsRecom
    fitness = eval_fitness(population)
    MejorMinimoGen = min(fitness)
    HistorialFitness.append(MejorMinimoGen)
    if MejorMinimoGen < MejorMin:
        MejorMin = MejorMinimoGen
        MejorXY = population[fitness.index(MejorMinimoGen)]
        MejorIteracion = i
    parents = [population[j] for j in sorted(range(len(population)), key=lambda x: fitness[x])[:miu]]
```

se itera a lo largo del número de generación establecidas, en cada iteración se hace lo siguiente:

1. se crea la población de descendientes con:
  - **offspringsRecom:** Descendientes creados por recombinación local.
  - **offspringsMuar:** Descendientes creados por mutación.
2. Se combina la población de padres con los descendientes para formar la población completa.
3. Se evalúa el fitness de todos los individuos en la población.
4. Se actualiza el mejor fitness encontrado (**mejorMin**), las coordenadas del mejor individuo (**MejorXY**) y la mejor generación(**MejorIteración**) si se encuentra un nuevo mínimo.
5. Se selecciona los **miu** mejores individuos de la población completa para formar la nueva población de padres.

## 2 Enfriamiento simulado

### I Definición de variables:

**límites:** Es un tupla que define los límites del espacio de búsqueda (mínimo y máximo para x y para y)

**iteraciones:** EL número máximo del salto aleatorio para explorar el espacio de búsqueda.

**temp:** La Temperatura inicial.

**enfriamientoRate:** La tasa de enfriamiento de la temperatura en cada iteración.

**tempFinal:** La temperatura final.

## II Definición de objetivo:

```
def Funcion(x, y):  
    return -(y + 47) * math.sin(math.sqrt(abs(x/2 + (y + 47)))) - x * math.sin(math.sqrt(abs(x - (y + 47))))
```

la función requerida para minimizar, en este caso es la función **EggHolder**.

## III Función principal y utilización:

```
def simulated_annealing(funcion, limites, iteraciones, salto, temp, enfriamientoRate, tempFinal):  
  
    xIni, yIni = random.uniform(*limites), random.uniform(*limites)  
    evalIni = funcion(xIni, yIni)  
    xMejor, yMejor, evalMejor = xIni, yIni, evalIni  
    iteracionMejor = 0  
    valores_por_iteracion = []  
  
    for i in range(iteraciones):  
        temp *= enfriamientoRate  
        if temp > tempFinal:  
            xNueva, yNueva = xIni + salto * (random.uniform(-1.0, 1.0)), yIni + salto * (random.uniform(-1.0, 1.0))  
            xNueva, yNueva = max(limites[0], min(xNueva, limites[1])), max(limites[0], min(yNueva, limites[1]))  
            evalNueva = funcion(xNueva, yNueva)  
            diff = evalNueva - evalIni  
            valores_por_iteracion.append(evalNueva)  
  
            if diff < 0 or random.random() < math.exp(-diff / temp):  
                xIni, yIni, evalIni = xNueva, yNueva, evalNueva  
                if evalIni < evalMejor:  
                    xMejor, yMejor, evalMejor = xIni, yIni, evalIni  
                    iteracionMejor = i  
            else: break  
  
    return (xMejor, yMejor, evalMejor, iteracionMejor, valores_por_iteracion)
```

Recibe como parámetros, la **función objetivo** (Función) y las variables establecidas, de la siguiente manera:

```
xMejor, yMejor, evalMejor, iteracionMejor, valores_por_iteracion = simulated_annealing(Funcion, Limites, Iteraciones, Salto, TempIni, EnfriamientoRate, TempFinal)
```

se toman lo que se quiere hallar y se igual a la función principal, donde se le pasan los requisitos que pide, luego la función principal hace lo siguiente con lo que recibe:

### Inicialización:

1 Selecciona valores aleatorios iniciales para x e y dentro de los límites.

2 Se calcula el valor de la **función objetivo** para la solución inicial.

3 Se inicializan las variables para almacenar la mejor solución encontrada hasta el momento, es decir calor y coordenadas x e y.

4 Se crea una lista para almacenar el valor de la función objetivo en cada iteración.

### Iteración:

1 Se reduce la temperatura en cada iteración según la tasa de enfriamiento.

2 Se generan nuevos valores aleatorios para x e y dentro de un rango definido para el salto y se limita su rango a los límites establecidos.

3 Se calcula el valor de la **función objetivo** para la nueva solución.

4 Se calcula la diferencia entre el valor de la nueva solución y el valor actual.

5 Se acepta la nueva solución si:

- La diferencia es negativa(indica una mejora)
- Ocurre con una probabilidad que depende de la diferencia y la temperatura actual, permitiendo escapar de mínimos locales.

6 Se actualiza la solución actual y la mejor solución encontrada si corresponde.

7 Se guarda el valor de la función objetivo en la lista de valores por iteración.

### Parada:

El proceso continúa hasta que se alcance el número máximo de iteraciones o la temperatura llegue a un valor mínimo.

## RESULTADOS:

### Estrategia de evolución

Los Primeros parámetros utilizados fueron los siguientes:

| Primera Configuración  |      |
|------------------------|------|
| Número de iteraciones  | 20   |
| Tasa de cruzamiento    | 0,60 |
| Tasa de mutación       | 0,40 |
| Tamaño de la población | 21   |

Los tiempos de ejecución y número de iteraciones necesarios para encontrar la solución para cada ejecución se muestra en la siguiente tabla:

| Ejecución | Tiempo ejecución/segundos | Mejor mínimo | Mejor (x,y) | Mejor Iteración |
|-----------|---------------------------|--------------|-------------|-----------------|
| n         |                           |              |             |                 |

|    |           |           |   |    |
|----|-----------|-----------|---|----|
| 1  | 0,0054911 | -838,1145 | (511.19240131901455,<br>374.7349005728338)    | 19 |
| 2  | 0,0097367 | -711,0176 | (-311.2709106538821,<br>-457.20947261450425)  | 18 |
| 3  | 0,013502  | -614,1676 | (-276.0424246750138,<br>291.1449339937901)    | 19 |
| 4  | 0,0113514 | -733,0923 | (-219.61586832030713,<br>473.2365571490874)   | 19 |
| 5  | 0,0122718 | -743,9348 | (-372.1948290838148,<br>-423.90768158786534)  | 15 |
| 6  | 0,0147241 | -765,3885 | (-424.84679412795276,<br>-404.3160411759114)  | 17 |
| 7  | 0,0138689 | -925,9303 | (420.696299055413,<br>463.2308895724484)      | 18 |
| 8  | 0,013477  | -845,6580 | (-432.40881105550835,<br>367.2812119815703)   | 19 |
| 9  | 0,0180226 | -663,3766 | (-339.84288123016756,<br>-410.34887006852614) | 19 |
| 10 | 0,0094781 | -776,8951 | (-437.6676851696012,<br>-391.9899275848463)   | 18 |
| 11 | 0,0121805 | -793,6652 | (-470.4525288291613,<br>-374.4883999486515)   | 19 |
| 12 | 0,0122932 | -670,7316 | (-313.916333033275,<br>309.816887127442)      | 17 |
| 13 | 0,0148504 | -807,4695 | (-500.68876418670624,<br>-354.4529840700203)  | 19 |
| 14 | 0,0134667 | -912,3355 | (497.0948105291606,<br>448.5742257975461)     | 19 |
| 15 | 0,0169285 | -689,3891 | (-500.37440887113337,<br>142.20525323591443)  | 19 |
| 16 | 0,0144698 | -673,8681 | (250.10764259369301,<br>-470.76479660430135)  | 19 |
| 17 | 0,0129459 | -878,0497 | (-452.04300511536314,<br>379.02469700125516)  | 8  |
| 18 | 0,0131701 | -868,5600 | (311.10967174484256,<br>510.76958339735813)   | 19 |
| 19 | 0,0122661 | -928,1674 | (-485.12666459020045,<br>396.2164473781594)   | 19 |
| 20 | 0,0126508 | -641,4695 | (-299.8056385701829,<br>311.4129672390765)    | 17 |

|    |           |           |  |    |
|----|-----------|-----------|--|----|
| 21 | 0,0132508 | -943,6475 | (517.5782934757564,<br>393.6037899060748)    | 19 |
| 22 | 0,0068369 | -501,0903 | (207.3528193463954,<br>256.2039147744849)    | 18 |
| 23 | 0,0067476 | -726,5786 | (-350.8749545724214,<br>329.3566257860634)   | 19 |
| 24 | 0,0067428 | -669,9024 | (-312.71737774205906,<br>311.03981112095806) | 17 |
| 25 | 0,00724   | -925,9285 | (420.61297456422227,<br>463.14904791925824)  | 19 |
| 26 | 0,0098682 | -921,6188 | (494.83882156405144,<br>402.21966343694226)  | 19 |
| 27 | 0,0060731 | -806,7509 | (-496.8340551140883,<br>-363.1918717150139)  | 19 |
| 28 | 0,0061753 | -648,6855 | (417.5898505108337,<br>-278.1615460889753)   | 18 |
| 29 | 0,0117657 | -808,9528 | (-498.1181682924806,<br>263.8408039575116)   | 19 |
| 30 | 0,0055851 | -845,4724 | (-321.555810887889,<br>515.7465503445153)    | 19 |

Resultados 21 ejecución de la configuración

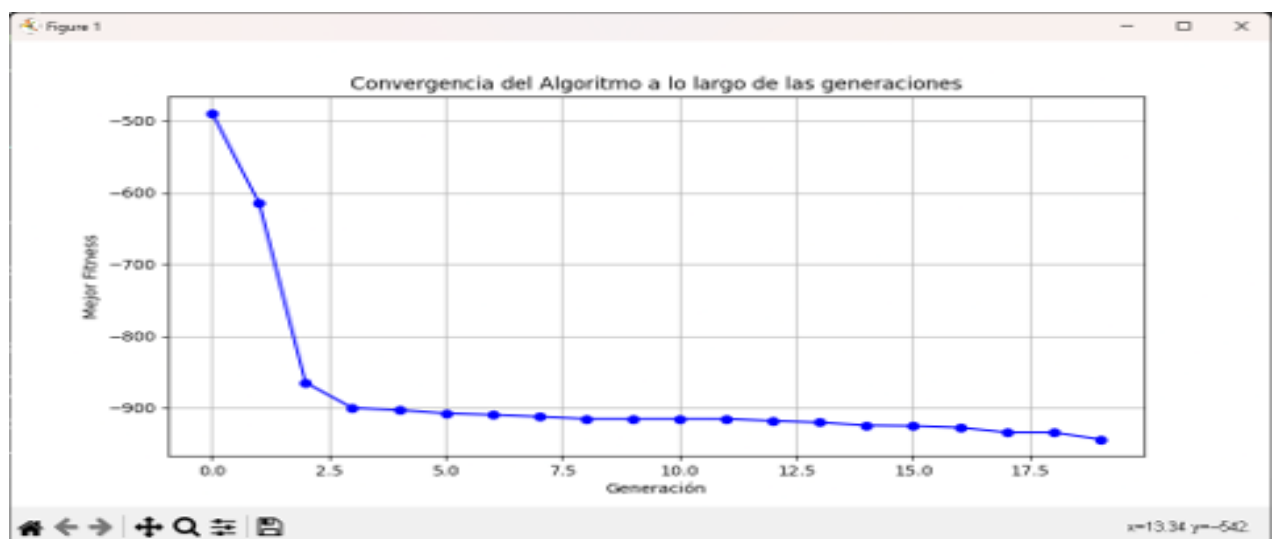
Resultados de la ejecución

El tiempo en encontrar la mejor solución fue de: 0.013250800002424512

La mejor solución encontrada: -943.6475171870759

Valores con los que se obtuvo esa solución: (517.5782934757564, 393.6037899060748)

Mejor solución encontrada en la iteración: 19





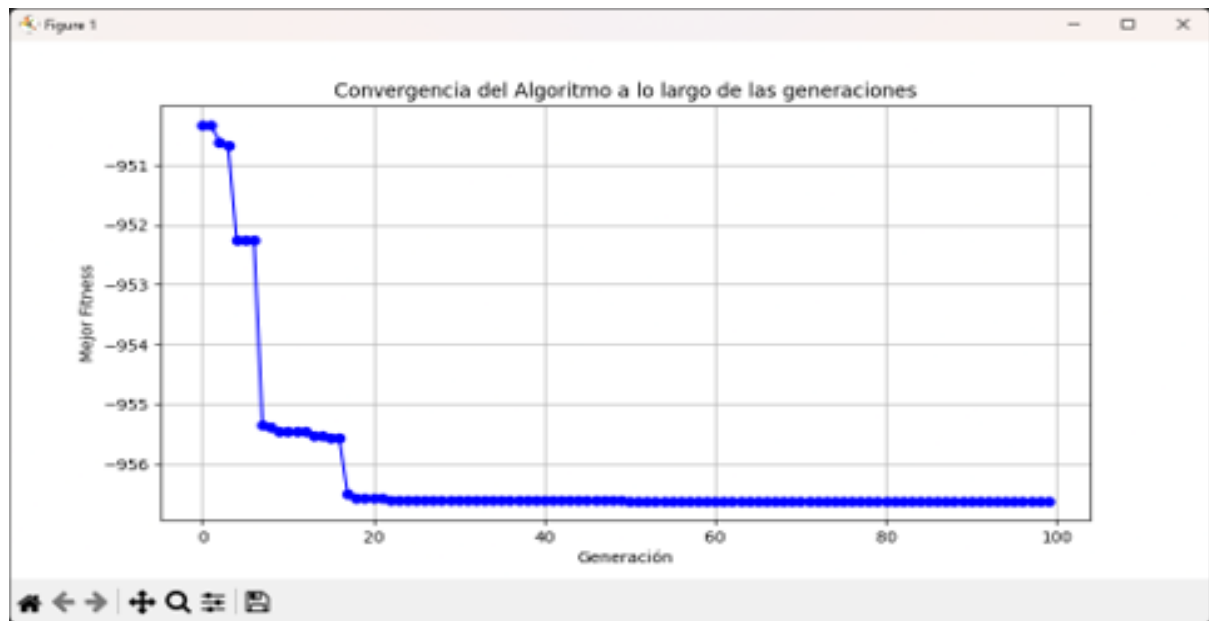
Los segundos parámetros utilizados fueron los siguientes:

| Segunda Configuración  |       |
|------------------------|-------|
| Número de iteraciones  | 100   |
| Tasa de cruzamiento    | 0,23  |
| Tasa de mutación       | 0,77  |
| Tamaño de la población | 2.800 |

Los tiempos de ejecución y número de iteraciones necesarios para encontrar la solución para cada ejecución se muestra en la siguiente tabla:

| Ejecución | Tiempo ejecución/segundos | Mejor mínimo | Mejor (x,y)                                 | Mejor Iteración |
|-----------|---------------------------|--------------|---|-----------------|
| 1         | 4,7876                    | -973,5621    | (515.4491987128564,<br>416.6122655690456)   | 50              |
| 2         | 4,7302                    | -969,3240    | (507.02267478204146,<br>420.7521015126496)  | 66              |
| 3         | 4,6511                    | -969,3240    | (507.0226750801486,<br>420.75210181857847)  | 47              |
| 4         | 5,0008                    | -971,4429    | (511.2358288699028,<br>418.6820757137574)   | 46              |
| 5         | 4,6573                    | -969,3240    | (507.022674795753,<br>420.752101535007)     | 83              |
| 6         | 4,9279                    | -975,6814    | (519.6627866275741,<br>414.54267341710533)  | 55              |
| 7         | 4,7861                    | -972,3239    | (-514.9769177582616,<br>410.34795035573853) | 34              |
| 8         | 4,8030                    | -971,4429    | (511.23582895984123,<br>418.68207581393534) | 33              |
| 9         | 4,8410                    | -965,0868    | (498.59700318366896,<br>424.8927894294429)  | 34              |
| 10        | 5,0887                    | -973,5621    | (515.4491981865344,<br>416.612265047761)    | 53              |
| 11        | 4,9033                    | -967,2053    | (502.80973380930084,<br>422.82234035678766) | 33              |
| 12        | 4,8476                    | -973,5621    | (515.449198319802,<br>416.61226518107077)   | 64              |
| 13        | 4,8863                    | -973,5621    | (515.449198508029,<br>416.6122653768926)    | 50              |
| 14        | 4,9107                    | -969,3240    | (507.0226749767951,<br>420.75210171583075)  | 57              |

|    |        |           |   |    |
|----|--------|-----------|---|----|
| 15 | 4,8959 | -969,3240 | (507.0226748992859,<br>420.752101638293)    | 85 |
| 16 | 4,7947 | -967,2053 | (502.8097339878312,<br>422.8223405390812)   | 56 |
| 17 | 4,8318 | -972,3239 | (-514.9769176960984,<br>410.3479504148135)  | 34 |
| 18 | 5,7385 | -975,6814 | (519.6627863668411,<br>414.5426731576637)   | 64 |
| 19 | 4,5997 | -969,3240 | (507.02267503347036,<br>420.75210176948315) | 69 |
| 20 | 4,6941 | -973,5621 | (515.4491981927423,<br>416.6122650522111)   | 72 |
| 21 | 4,6178 | -967,2053 | (502.80973394221263,<br>422.8223404793293)  | 32 |
| 22 | 4,6500 | -971,4429 | (511.2358287630508,<br>418.68207560131225)  | 39 |
| 23 | 4,6960 | -973,5621 | (515.4491986096691,<br>416.61226546946483)  | 40 |
| 24 | 4,5845 | -972,3239 | (-514.9769177057466,<br>410.34795041963724) | 32 |
| 25 | 4,6111 | -967,2053 | (502.80973359109146,<br>422.82234013173854) | 27 |
| 26 | 4,6258 | -956,6150 | (481.7481274947845,<br>433.17663159576983)  | 50 |
| 27 | 4,8176 | -967,2053 | (502.80973423871484,<br>422.8223407731365)  | 60 |
| 28 | 4,9537 | -971,4429 | (511.235829142973,<br>418.6820759802935)    | 40 |
| 29 | 4,8017 | -971,4429 | (511.23582917091005,<br>418.6820760113427)  | 24 |
| 30 | 4,8012 | -960,8505 | (490.1721609822649,<br>429.0343063596499)   | 37 |



Resultados 26 ejecución de la configuración

El tiempo en encontrar la mejor solución fue de: 4.6258362999979

La mejor solución encontrada: -956.6150107560228

Valores con los que se obtuvo esa solución: (481.7481274947845, 433.17663159576983)

Mejor solución encontrada en la iteración: 50

Los terceros parámetros utilizados fueron los siguientes:

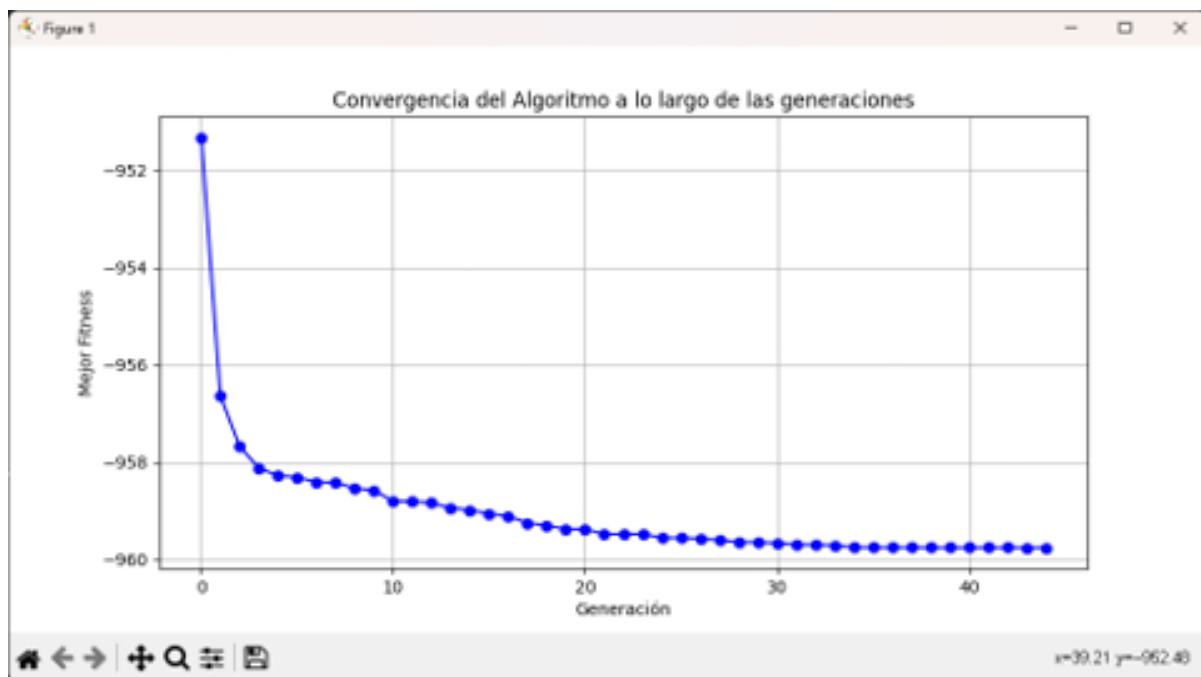
| Mejor Configuración    |      |
|------------------------|------|
| Número de iteraciones  | 45   |
| Tasa de cruzamiento    | 0,55 |
| Tasa de mutación       | 0,45 |
| Tamaño de la población | 500  |

Los tiempos de ejecución y número de iteraciones necesarios para encontrar la solución para cada ejecución se muestra en la siguiente tabla:

| Ejecución | Tiempo ejecución/segundos | Mejor mínimo | Mejor (x,y)                            | Mejor Iteración |
|-----------|---------------------------|--------------|--|-----------------|
| 1         | 0,4022                    | -948,5263    | (465.1320431960618, 438.5477327419028) | 44              |
| 2         | 0,4095                    | -961,2520    | (494.0659157298358, 420.3520494861524) | 44              |
| 3         | 0,3732                    | -958,7327    | (485.9600444102582, 431.1053691949316) | 23              |

|    |        |           |   |    |
|----|--------|-----------|---|----|
| 4  | 0,3441 | -944,9714 | (458.5734812575012,<br>444.5592510810889)   | 43 |
| 5  | 0,3803 | -953,4389 | (475.430621585375,<br>436.2838944533444)    | 22 |
| 6  | 0,3954 | -945,0142 | (462.3267590405245,<br>435.7518815497114)   | 44 |
| 7  | 0,3826 | -898,7735 | (373.2583305716102,<br>478.644953524588)    | 44 |
| 8  | 0,3797 | -960,8271 | (489.83701417399504,<br>428.6961365892105)  | 44 |
| 9  | 0,3568 | -956,6150 | (481.74771725888513,<br>433.17615098341395) | 44 |
| 10 | 0,3944 | -965,6420 | (499.9596198916727,<br>419.9573656749756)   | 44 |
| 11 | 0,4017 | -953,5961 | (483.93765157738216,<br>422.7999993799547)  | 44 |
| 12 | 0,4258 | -950,2054 | (479.2260878254569,<br>424.386456999709)    | 44 |
| 13 | 0,4052 | -952,6707 | (473.4225563274669,<br>434.27290358265935)  | 43 |
| 14 | 0,4409 | -959,7575 | (-506.60011732320936,<br>406.1584641933936) | 43 |
| 15 | 0,3777 | -954,6128 | (478.54192962632777,<br>429.95551020745546) | 44 |
| 16 | 0,3251 | -962,9686 | (494.3844796314845,<br>426.9634454930237)   | 40 |
| 17 | 0,3448 | -956,6138 | (481.68016539708634,<br>433.10990456961395) | 44 |
| 18 | 0,3618 | -895,8289 | (382.6148490909812,<br>469.1236994480518)   | 44 |
| 19 | 0,3415 | -962,9686 | (494.38447958159895,<br>426.9634454416676)  | 26 |
| 20 | 0,3477 | -955,5562 | (479.65936146106844,<br>434.22943959240627) | 44 |
| 21 | 0,3533 | -938,6228 | (445.9545511143252,<br>450.79007474775494)  | 34 |
| 22 | 0,3643 | -965,0827 | (498.7155794511831,<br>425.0137878231542)   | 44 |
| 23 | 0,3310 | -958,6935 | (485.5258511088994,<br>430.6674545728216)   | 43 |

|    |        |           |   |    |
|----|--------|-----------|---|----|
| 24 | 0,3590 | -917,4720 | (403.8580427554781,<br>471.52533257231414)  | 44 |
| 25 | 0,3515 | -951,3217 | (471.2207242498076,<br>438.3571947594529)   | 43 |
| 26 | 0,3618 | -958,7327 | (485.96004441515856,<br>431.10536920027874) | 24 |
| 27 | 0,3225 | -951,1852 | (470.43088414718085,<br>437.55894600435835) | 44 |
| 28 | 0,3381 | -953,4389 | (475.4306210472579,<br>436.2838939122753)   | 22 |
| 29 | 0,3668 | -964,3422 | (-514.4537279619932,<br>417.16645914048297) | 43 |
| 30 | 0,3719 | -964,4876 | (496.86659603390035,<br>423.1497864914091)  | 44 |



Resultados 14 ejecución de la configuración

El tiempo en encontrar la mejor solución fue de: 0.44088209999972605

La mejor solución encontrada: -959.757541628662

Valores con los que se obtuvo esa solución: (-506.60011732320936, 406.1584641933936)

Mejor solución encontrada en la iteración: 43

### Enfriamiento simulado

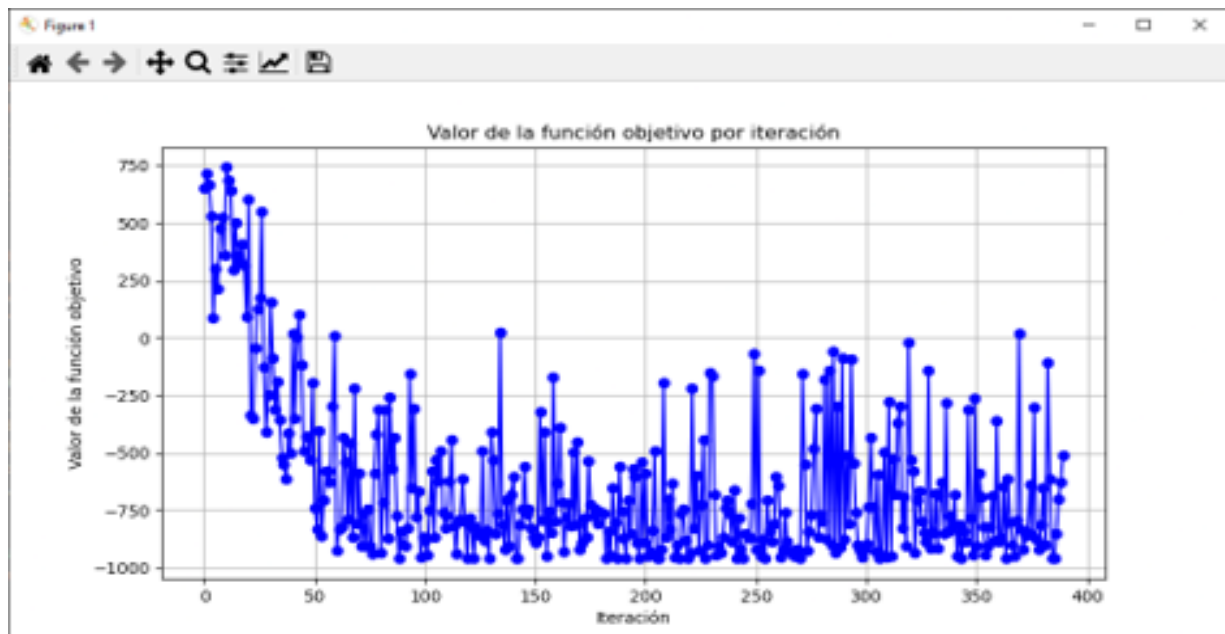
Los primeros parámetros utilizados fueron los siguientes:

| Mejor configuración   |      |
|-----------------------|------|
| temperatura inicial   | 5000 |
| temperatura final     | 0.01 |
| tasa de enfriamiento  | 0.95 |
| número de iteraciones | 5000 |

Los tiempos de ejecución y número de iteraciones necesarios para encontrar la solución para cada ejecución se muestra en la siguiente tabla:

| Ejecución | Tiempo ejecución/segundos   | Mejor mínimo | Mejor (x,y)                                  | Mejor Iteración |
|-----------|-----------------------------|--------------|--|-----------------|
| 1         | 0,000000700005330<br>145359 | -426,3420    | (89.41714325536765;-390.846644<br>5243039)   | 190             |
| 2         | 0,000000700005330<br>145359 | -426,3420    | (418.99496085543376;171.279455<br>1508545)   | 380             |
| 3         | 0,000000700005330<br>145359 | -559,7476    | (-243.55567178686647;274.32435<br>58238662)  | 119             |
| 4         | 0,000000700005330<br>145359 | -469,7186    | (-512.0;-272.2938739820207)                  | 166             |
| 5         | 0,000000700005330<br>145359 | -786,4901    | (-457.37699399697397;-383.05015<br>02454224) | 137             |
| 6         | 0,000000700005330<br>145359 | -493,8775    | (149.37258821226294;301.089915<br>5367147)   | 82              |
| 7         | 0,000000700005330<br>145359 | -206,6040    | (62.3611735347237;-199.1516810<br>616224)    | 207             |
| 8         | 0,000000700005330<br>145359 | -126,1145    | (-44.88981543189848;37.2071277<br>7338341)   | 239             |
| 9         | 0,000000700005330<br>145359 | -306,6765    | (-75.65547036136411;190.638233<br>00802105)  | 270             |
| 10        | 0,000000700005330<br>145359 | -894,4461    | (-466.69940907876287;385.73542<br>12737615)  | 298             |
| 11        | 0,000000700005330<br>145359 | -374,4025    | (-179.77868902815695;-247.91378<br>69098388) | 43              |
| 12        | 0,000000700005330<br>145359 | -959,6400    | (512.0;404.2551492340125)                    | 141             |
| 13        | 0,000000700005330<br>145359 | -559,7774    | (-243.28700650927246;274.46201<br>594167)    | 96              |

|    |                             |           |  |     |
|----|-----------------------------|-----------|--|-----|
| 14 | 0,000000700005330<br>145359 | -373,8434 | (57.77064537790998;-378.589987<br>183309)    | 16  |
| 15 | 0,000000700005330<br>145359 | -715,4010 | (399.3771330803312;-365.869194<br>36870136)  | 27  |
| 16 | 0,000000700005330<br>145359 | -629,6157 | (418.68911005032163;170.911244<br>44445433)  | 279 |
| 17 | 0,000000700005330<br>145359 | -559,6065 | (-242.74932138802166;275.17007<br>003359)    | 187 |
| 18 | 0,000000700005330<br>145359 | -785,6868 | (-459.2714261904075;-384.368730<br>1179436)  | 230 |
| 19 | 0,000000700005330<br>145359 | -629,2529 | (419.69856964488025;171.000102<br>70621425)  | 117 |
| 20 | 0,000000700005330<br>145359 | -206,5715 | (13.142601281518491;151.391944<br>24120672)  | 340 |
| 21 | 0,000000700005330<br>145359 | -342,7200 | (276.69182037549996;27.4130162<br>6239847)   | 174 |
| 22 | 0,000000700005330<br>145359 | -426,4469 | (91.14025661476029;-390.828638<br>5684248)   | 217 |
| 23 | 0,000000700005330<br>145359 | -565,6663 | (-108.25161983242116;423.33270<br>04896553)  | 302 |
| 24 | 0,000000700005330<br>145359 | -539,8684 | (241.62703622563643;257.018276<br>92281813)  | 368 |
| 25 | 0,000000700005330<br>145359 | -231,4725 | (-234.3921633314637;-2.43170550<br>14453324) | 41  |
| 26 | 0,000000700005330<br>145359 | -362,4008 | (-123.96174733537035;-290.55742<br>15476889) | 245 |
| 27 | 0,000000700005330<br>145359 | -306,5022 | (-77.95806525392814;191.614208<br>30178224)  | 179 |
| 28 | 0,000000700005330<br>145359 | -578,3953 | (366.35542983165186;-363.21393<br>10317143)  | 22  |
| 29 | 0,000000700005330<br>145359 | -959,6393 | (512.0;404.1969461821113)                    | 326 |
| 30 | 0,000000700005330<br>145359 | -424,8712 | (16.52924806001851;368.2494005<br>115668)    | 264 |



Resultados 12 ejecución de la configuración

El tiempo en encontrar la mejor solución fue de: 0.00000070000533015

La mejor solución encontrada: -959.640043111564

Valores con los que se obtuvo esa solución:  $x=512.000000$ ,  $y=404.255149$ )

Mejor solución encontrada en la iteración: 141

Los segundos parámetros utilizados fueron los siguientes:

| Segunda configuración |       |
|-----------------------|-------|
| temperatura inicial   | 2200  |
| temperatura final     | 0.001 |
| tasa de enfriamiento  | 0.90  |
| número de iteraciones | 1000  |

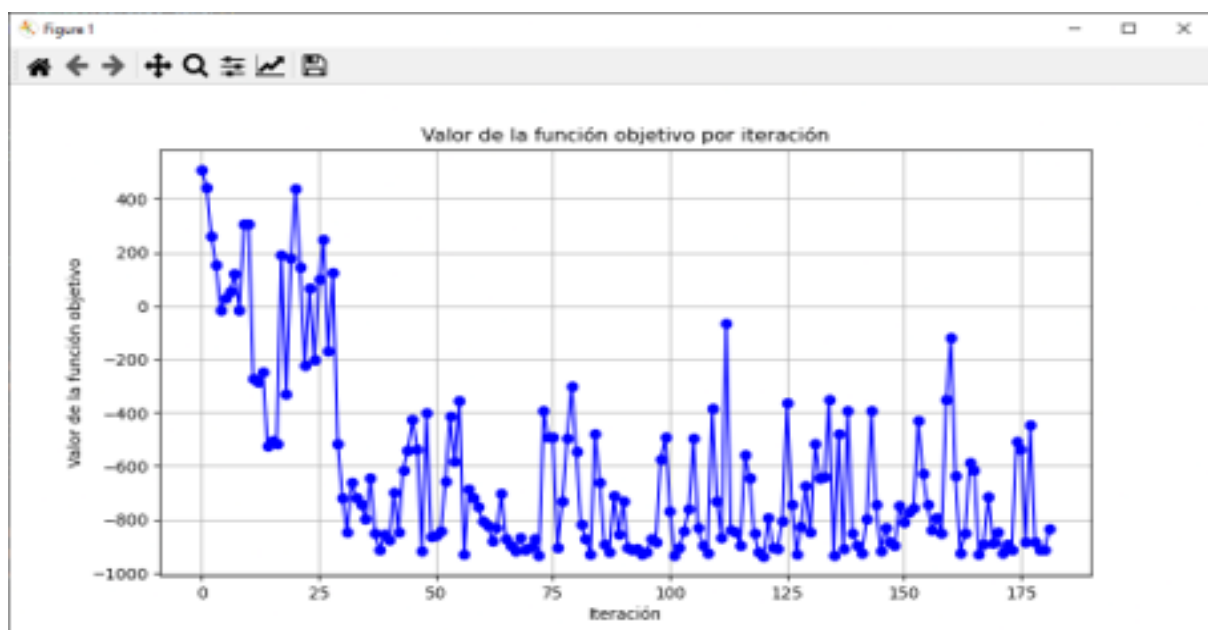
Los tiempos de ejecución y número de iteraciones necesarios para encontrar la solución para cada ejecución se muestra en la siguiente tabla:

| Ejecución<br>n | Tiempo<br>ejecución/segundos | Mejor mínimo | Mejor (x,y)                                   | Mejor<br>Iteración |
|----------------|------------------------------|--------------|---|--------------------|
| 1              | 0,00000013000098988<br>4138  | -557,088     | (-415.2128989728991;100.16620<br>981844576)   | 63                 |
| 2              | 0,00000013000098988<br>4138  | -393,315     | (-187.57149811266453;-258.125<br>93183795553) | 150                |
| 3              | 0,00000013000098988<br>4138  | -126,212     | (-45.074747313281776;37.45533<br>666711907)   | 71                 |



|    |                             |          |  |     |
|----|-----------------------------|----------|--|-----|
| 4  | 0,00000013000098988<br>4138 | -539,974 | (240.5783440806223;254.871694<br>22117242)   | 54  |
| 5  | 0,00000013000098988<br>4138 | -203,032 | (123.53318965407539;-131.2221<br>2135571237) | 81  |
| 6  | 0,00000013000098988<br>4138 | -557,163 | (-416.1481232286396;97.873207<br>33830651)   | 148 |
| 7  | 0,00000013000098988<br>4138 | -361,938 | (-125.10662727360976;-291.804<br>8482430472) | 132 |
| 8  | 0,00000013000098988<br>4138 | -493,433 | (148.77914858623686;302.20952<br>758618404)  | 169 |
| 9  | 0,00000013000098988<br>4138 | -714,502 | (494.7872857683955;-273.11895<br>33822895)   | 153 |
| 10 | 0,00000013000098988<br>4138 | -206,346 | (13.708782077306672;152.45171<br>736046828)  | 113 |
| 11 | 0,00000013000098988<br>4138 | -716,262 | (401.327940939326;-368.151946<br>069597)     | 97  |
| 12 | 0,00000013000098988<br>4138 | -785,075 | (-453.7326386542278;-380.2116<br>3519755555) | 165 |
| 13 | 0,00000013000098988<br>4138 | -702,893 | (-292.9528305612328;-459.0026<br>351092364)  | 80  |
| 14 | 0,00000013000098988<br>4138 | -306,078 | (-79.24677597593556;191.81293<br>499950613)  | 79  |
| 15 | 0,00000013000098988<br>4138 | -934,969 | (438.0699332801527;452.643890<br>8298319)    | 120 |
| 16 | 0,00000013000098988<br>4138 | -420,418 | (200.5280243703239;-269.59152<br>683361873)  | 87  |
| 17 | 0,00000013000098988<br>4138 | -717,627 | (285.165962946064;-486.709451<br>57247155)   | 144 |
| 18 | 0,00000013000098988<br>4138 | -443,963 | (-285.6650032638704;-209.7896<br>912907725)  | 70  |
| 19 | 0,00000013000098988<br>4138 | -301,781 | (-161.03998819053157;96.25980<br>078792723)  | 96  |
| 20 | 0,00000013000098988<br>4138 | -557,675 | (-332.05338892200336;182.0386<br>57919342)   | 175 |
| 21 | 0,00000013000098988<br>4138 | -554,909 | (-372.78504287301905;141.3096<br>9682370312) | 147 |

|    |                             |          |  |     |
|----|-----------------------------|----------|--|-----|
| 22 | 0,00000013000098988<br>4138 | -306,378 | (-76.95810313332053;190.04375<br>2608718)    | 171 |
| 23 | 0,00000013000098988<br>4138 | -303,037 | (-303.2089610590906;-49.55957<br>424321973)  | 22  |
| 24 | 0,00000013000098988<br>4138 | -299,995 | (-201.73668548289982;55.81853<br>191759427)  | 76  |
| 25 | 0,00000013000098988<br>4138 | -300,794 | (-242.8568497210599;14.095966<br>675240128)  | 120 |
| 26 | 0,00000013000098988<br>4138 | -717,403 | (285.6931798845661;-486.80344<br>551808764)  | 120 |
| 27 | 0,00000013000098988<br>4138 | -210,802 | (-172.79905774030092;-96.4267<br>2390739006) | 73  |
| 28 | 0,00000013000098988<br>4138 | -717,095 | (281.68434467874874;-488.8398<br>017739749)  | 29  |
| 29 | 0,00000013000098988<br>4138 | -565,636 | (-108.40016214838354;423.4258<br>331456735)  | 142 |
| 30 | 0,00000013000098988<br>4138 | -305,849 | (-75.73107711251347;193.26903<br>465499842)  | 122 |



Resultados 15 ejecución de la configuración

El tiempo en encontrar la mejor solución fue de: 0.00000130000989884

La mejor solución encontrada: -934.9694642140739

Valores con los que se obtuvo esa solución:  $x=438.069933$ ,  $y=452.643891$ )

Mejor solución encontrada en la iteración: 120

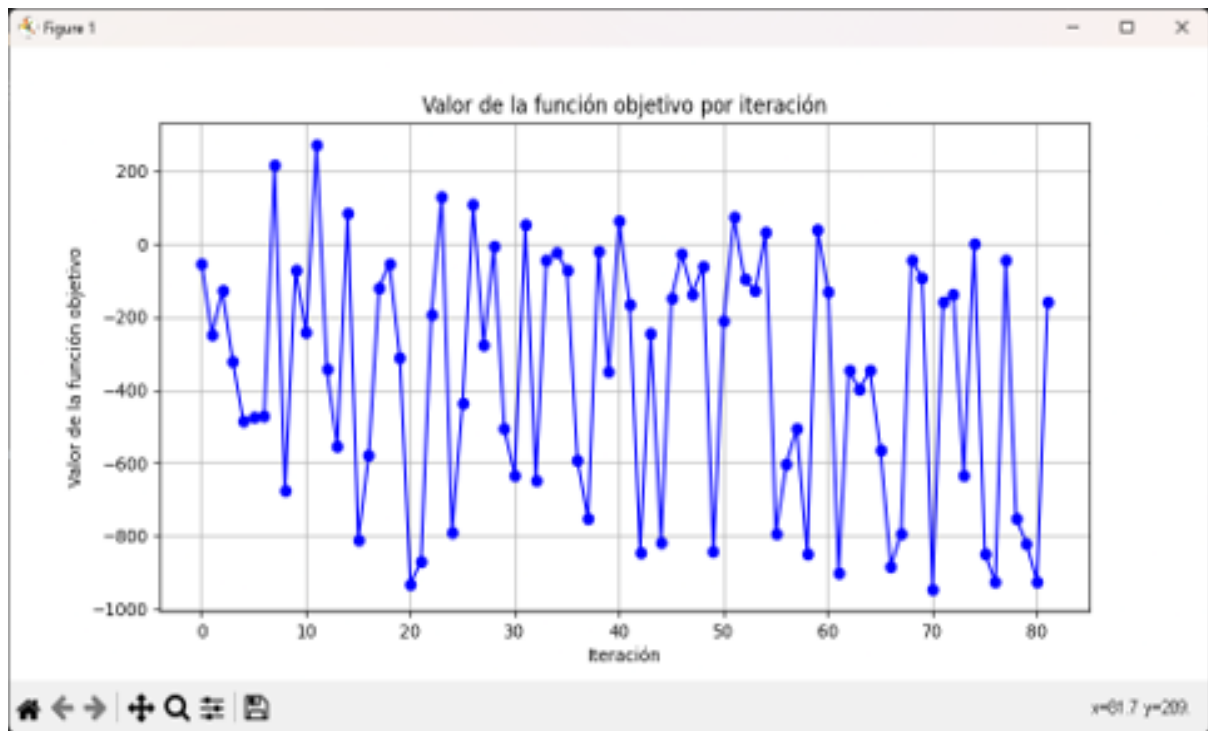
Los terceros parámetros utilizados fueron los siguientes:

| Tercera configuración |        |
|-----------------------|--------|
| temperatura inicial   | 78000  |
| temperatura final     | 0.0001 |
| tasa de enfriamiento  | 0.78   |
| número de iteraciones | 1200   |

Los tiempos de ejecución y número de iteraciones necesarios para encontrar la solución para cada ejecución se muestra en la siguiente tabla:

| Ejecución | Tiempo ejecución/segundos | Mejor mínimo | Mejor (x,y)                                | Mejor Iteración |
|-----------|---------------------------|--------------|--|-----------------|
| 1         | 0,00000079999881564       | -892,3866    | (-463.94512907635226, 387.69163081045355)  | 80              |
| 2         | 0,00000079999881564       | -515,3221    | (-411.2281701078279, -344.4205716782106)   | 2               |
| 3         | 0,00000079999881564       | -507,3871    | (-406.2658393858681, -154.47432071659003)  | 55              |
| 4         | 0,00000079999881564       | -357,5045    | (-117.55716910268843, -289.1714334417355)  | 66              |
| 5         | 0,00000079999881564       | -342,8570    | (278.72893559878264, 29.97511119840573)    | 66              |
| 6         | 0,00000079999881564       | -342,9293    | (277.99418295119574, 30.391334105580118)   | 50              |
| 7         | 0,00000079999881564       | -546,7666    | (445.46350386699146, 193.5618968327112)    | 1               |
| 8         | 0,00000079999881564       | -405,0645    | (-224.98867058125603, -249.80821091556805) | 49              |
| 9         | 0,00000079999881564       | -781,0173    | (-460.59369573307174, -388.09123893612195) | 73              |
| 10        | 0,00000079999881564       | -202,0293    | (121.9502084661822, -129.00113915704404)   | 47              |
| 11        | 0,00000079999881564       | -668,1657    | (-447.31674199563474, -360.6163368716525)  | 0               |
| 12        | 0,00000079999881564       | -422,0606    | (512, -205.49902928797795)                 | 59              |
| 13        | 0,00000079999881564       | -641,9613    | (-303.8501822722038, -479.22276958437743)  | 15              |

|    |                         |           |   |    |
|----|-------------------------|-----------|---|----|
| 14 | 0,000000799998815<br>64 | -421,5302 | (512, -208.11223965196652)                    | 70 |
| 15 | 0,000000799998815<br>64 | -633,8419 | (-174.85244986764107, -512)                   | 59 |
| 16 | 0,000000799998815<br>64 | -944,7308 | (479.81091793989685,<br>430.9749604225462)    | 70 |
| 17 | 0,000000799998815<br>64 | -748,6633 | (-389.3084909375569,<br>-415.42106692262394)  | 50 |
| 18 | 0,000000799998815<br>64 | -893,3907 | (-468.0384613520712,<br>387.4728559712602)    | 51 |
| 19 | 0,000000799998815<br>64 | -126,5352 | (-12.984222051969887,<br>-167.5296337537705)  | 48 |
| 20 | 0,000000799998815<br>64 | -555,5248 | (-411.0218496019507,<br>98.45924316298456)    | 70 |
| 21 | 0,000000799998815<br>64 | -786,4523 | (-457.53685383517495,<br>-382.9912726261106)  | 71 |
| 22 | 0,000000799998815<br>64 | -558,7724 | (-244.49044068864634,<br>272.88998297337787)  | 69 |
| 23 | 0,000000799998815<br>64 | -506,1140 | (-404.82817981564637,<br>-151.91111702569177) | 71 |
| 24 | 0,000000799998815<br>64 | -210,9471 | (-172.6233186910583,<br>-96.659628841598)     | 78 |
| 25 | 0,000000799998815<br>64 | -736,8069 | (-358.81417488630757,<br>-427.4845170581326)  | 65 |
| 26 | 0,000000799998815<br>64 | -786,4621 | (-457.5400025091415,<br>-383.19699602900573)  | 80 |
| 27 | 0,000000799998815<br>64 | -412,4881 | (261.0442723054652,<br>-201.65171154583788)   | 45 |
| 28 | 0,000000799998815<br>64 | -574,0640 | (317.74870870704837,<br>210.13549695293128)   | 81 |
| 29 | 0,000000799998815<br>64 | -492,8890 | (150.58883018354925,<br>303.69581896250577)   | 59 |
| 30 | 0,000000799998815<br>64 | -425,9793 | (89.12281768302665,<br>-389.4406491515981)    | 80 |



Resultados 16 ejecución de la configuración

El tiempo en encontrar la mejor solución fue de: 0.00000079999881564

La mejor solución encontrada: -944.7307804233793

Valores con los que se obtuvo esa solución:  $x=479.810918$ ,  $y=430.974960$ )

Mejor solución encontrada en la iteración: 70

## Análisis

Para el análisis se tendrán en cuenta las mejores configuraciones de cada método para, la varianza de cada uno en cuanto a la solución encontrada, el tiempo y el número de iteraciones requeridas para converger.

|                               | Enfriamiento simulado   | Estrategias de evolución |
|-------------------------------|-------------------------|--------------------------|
| <b>Varianza</b>               | 48270,55246             | 302,0774816              |
| <b>Tiempo</b>                 | 0,000021000159904360800 | 11,1107                  |
| <b>Tiempo promedio</b>        | 0,000000700005330145359 | 0,370357923333479000000  |
| <b>Número de ejecuciones</b>  | 30                      | 30                       |
| <b>Iteraciones requeridas</b> | <b>141</b>              | <b>43</b>                |
| <b>Solución encontrada</b>    | <b>-959,6400</b>        | <b>-959,7575</b>         |

Se puede ver que los **métodos** a pesar de tener **resultados** son bastante similares, los otros parámetros encontrados a partir de los resultados como la varianza, tiempo (este es el tiempo

total de las 30 ejecuciones), el tiempo promedio de esas 30 ejecuciones y las iteraciones requeridas para poder hallar los resultados en cada **algoritmo**.

### **Comparación entre métodos**

Para hacer las siguientes comparaciones se tomó en cuenta solo la mejor configuración de cada método.

Para el método de **Estrategia de evolución** tenemos que el tiempo promedio para encontrar cada resultado es de 0,370357923333479000000, lo que es un tiempo bastante grande comparado con el método de **Enfriamiento simulado**, el cual necesita un tiempo promedio para alcanzar una solución de 0,000000700005330145359. Lo que nos muestra que el enfriamiento simulado en términos de tiempo es más eficiente que el otro método, esto también puede deberse a la configuración que tiene cada uno, ya que se pueden realizar muchas configuraciones, donde quizás el método de enfriamiento de resultados mayores que el otro método.

Con el método de **Enfriamiento simulado** la cantidad de iteraciones es mayor que en el de **Estrategia de evolución**. El hecho de que las iteraciones sean mayores en el primer método puede mostrar que el método es más ineficiente para mostrar resultado con una menor cantidad de iteraciones, mostrando que la configuración y patrón con la que la Estrategia de evolución a pesar de tener un tiempo de ejecución más alta que el Enfriamiento, no se requieren muchas iteraciones para encontrar un resultado óptimo o lo más acertado posible.

Teniendo en cuenta la comparación de los tiempos de ejecución y del número de iteraciones observamos que al optimizar la función con los algoritmos de **Enfriamiento simulado** y **Estrategias de evolución** logramos valores muy similares sin embargo por tiempos de ejecución podemos optar por decir que el de **Enfriamiento simulado** es más eficiente a pesar de que sus tasas de iteraciones son más altas, sin embargo, el algoritmo de Estrategias de evolución es más eficiente para encontrar la solución ya que requiere de menor cantidad de iteraciones. Se deben tener en cuenta más factores además de los ya mencionados para encontrar una solución verdaderamente óptima para la función objetivo.