Algoritmos Evolutivos

TP3

Juan Pablo Schamun

# Ejercicio1

1. Se intenta utilizando PSO con restricciones maximizar la siguiente función:

* f(A, B, C, D) = 375\*A + 275\*B + 475\*C + 325\*D

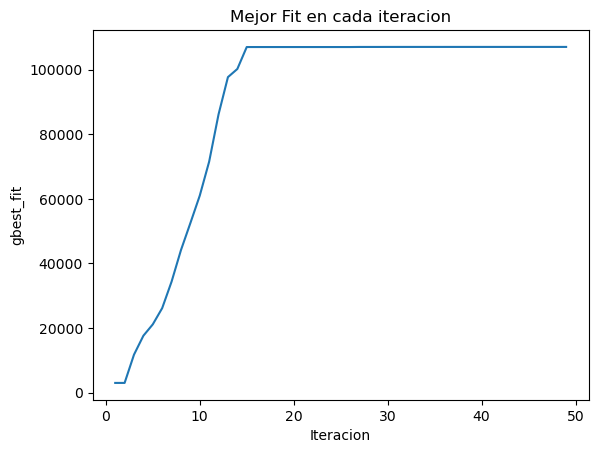
Sujeto a:

* g1(X) = 2.5\*A + 1.5\*B + 2.75\*C + 2\*D <= 640
* g2(X) = 3.5\*A + 3\*B + 3\*C + 2\*D <= 960
* A, B, C, D >= 0.

El algoritmo se encuentra en el git

1. Solución encontrada:
   * Solución óptima: X = [52.6250, 40.8868,120.7945, 57.4612]
   * Valor óptimo: y = 107030.5
2. URL GitHub:

<https://github.com/juanpsch/AEIV/blob/main/TP3/TP3_1.ipynb>

1. Grafico:
2. Si se reduce en una unidad el tiempo de acabado de la parte b, se cambia la restricción dos por:
   * 3.5\*A + **2**\*B + 3\*C + 2\*D <= 960

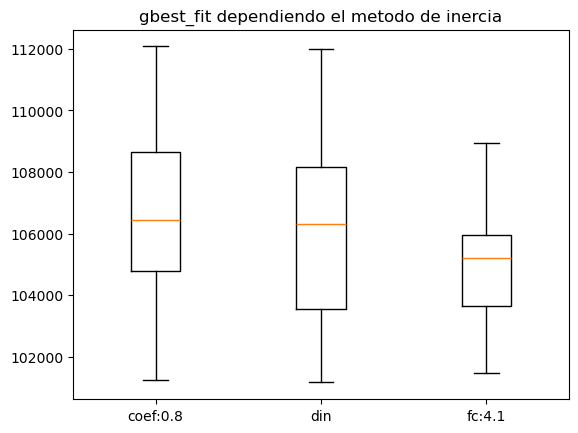
Dando como solución:

* + Solución óptima: X = [84.1278, 71.2885,81.9580, 48.6816]
  + Valor óptimo: y = 105903.8

Se ve que aumentan las cantidades de B de la solución óptima.

Igual el algoritmo tiene cierta variabilidad

1. Gráfico de BoxPlots con 3 métodos de inercia



# Ejercicio2

1. Introduciendo a=12 y b=35 queda la función

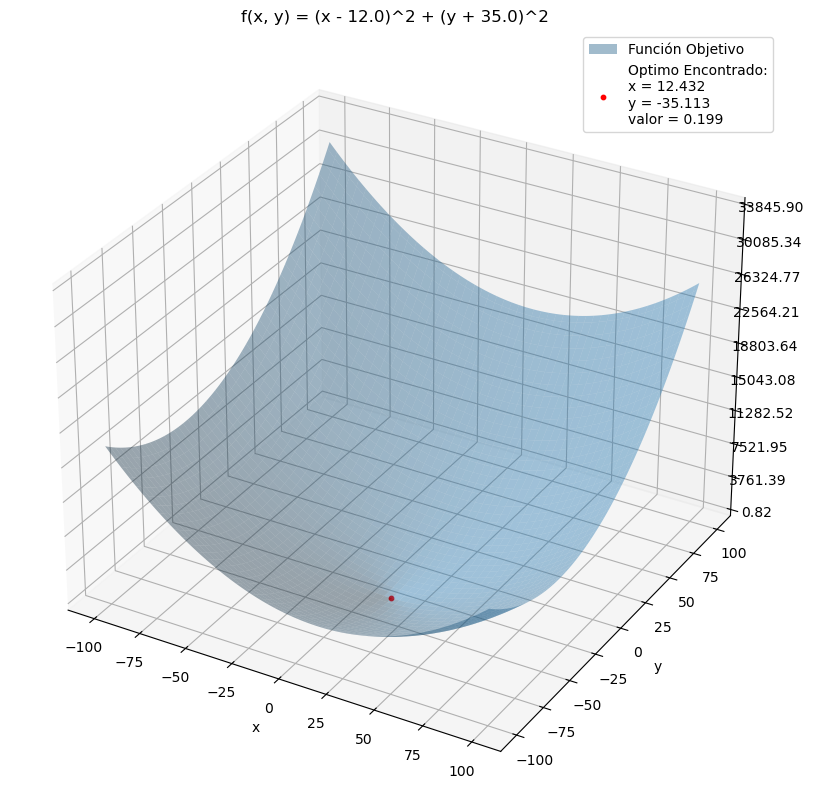
***f(x, y) = (x - 12)^2 + (y + 35)^2*** para minimizar

* 1. Solución óptima: **x = 12.432**; **y = -35.113**
  2. Valor óptimo**: f(x,y) = 0.119**

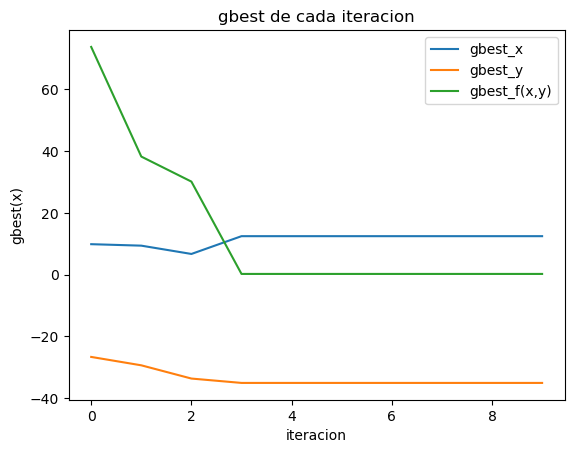
1. URL GitHub:

<https://github.com/juanpsch/AEIV/blob/main/TP2/TP2_2.ipynb>

1. Gráfico de función objetivo



1. Gráfico de gbest:



1. Cambiando w=0 la función

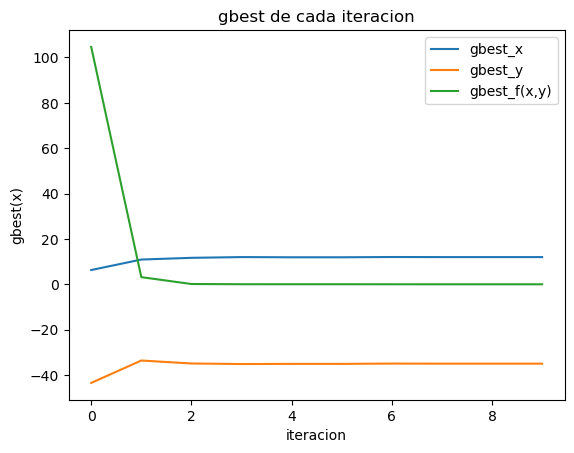
***f(x, y) = (x - 12)^2 + (y + 35)^2*** para minimizar tiene

* 1. Solución óptima: **x = 11.97**; **y = -35.02**

Valor óptimo**: f(x,y) = 0.0008**

Se obtiene un óptimo mejor de manera sistemática. Al parecer al ser una función unimodal la inercia no ayuda a converger más rápidamente, ya que no hay mínimos locales en donde pueda quedare estancada la partícula.

Se observa en la siguiente figura, como converge mucho más rápidamente



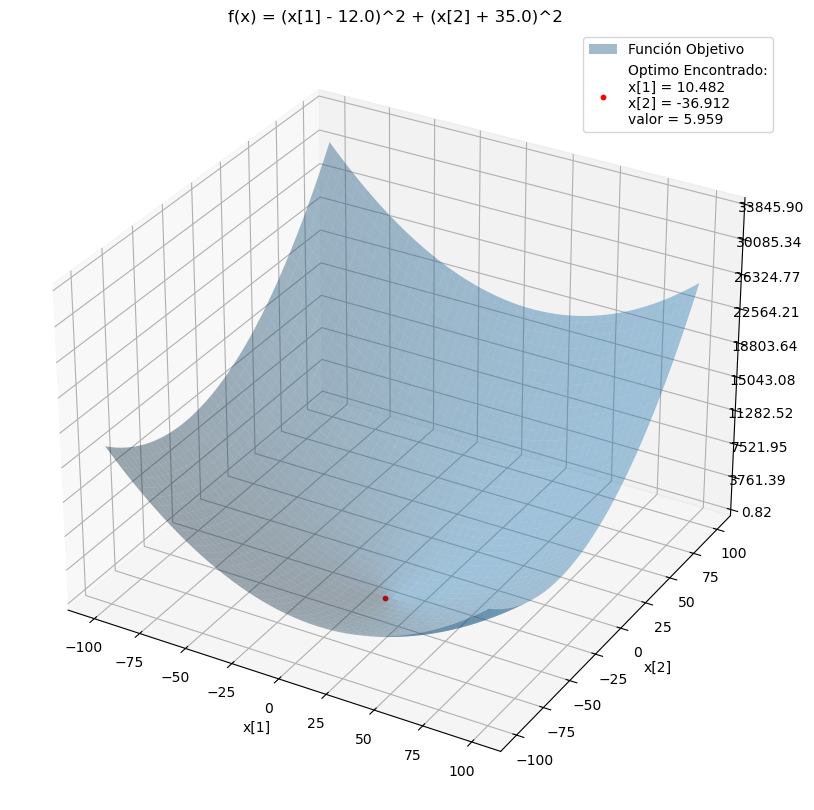
1. Repetir con pyswarm:
   1. Introduciendo a=12 y b=35 queda la función

***f(x, y) = (x1 - 12)^2 + (x2 + 35)^2*** para minimizar

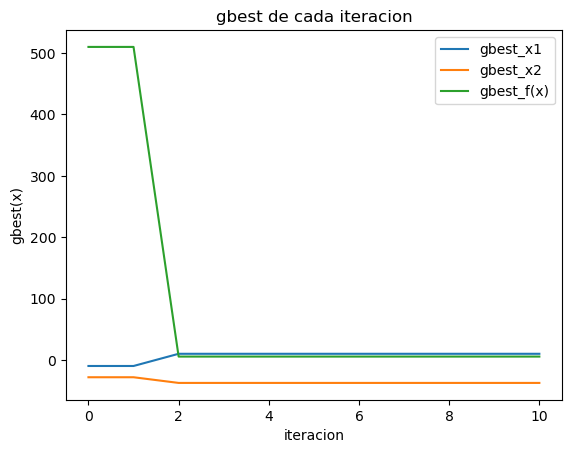
* + - Solución óptima: **x1 = 10.492**; **x2 = -36.912**
    - Valor óptimo**: f(x,y) = 5.959**
  1. URL GitHub:

<https://github.com/juanpsch/AEIV/blob/main/TP2/TP2_2.ipynb>

* 1. Gráfico de función objetivo:



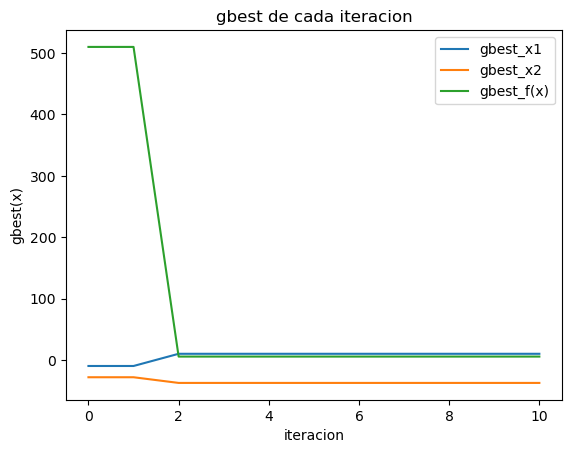
* 1. Gráfico de gbest:



* 1. Cambiando w=0 la función

***f(xy) = (x1 - 12)^2 + (x2 + 35)^2*** para minimizar tiene

Solución óptima: **x1 = 11.981**; **x2 = -34.969**



1. Utilizando pyswarm, se tarda más iteraciones en llegar a un óptimo de valores similares al algoritmo casero. Al poner la inercia en cero, esta diferencia ya no es significativa.