

Notebook

August 13, 2024

1 TRABAJO PRÁCTICO 3

1.0.1 Algoritmos Evolutivos

Alumno: Leandro Bello

URL: https://github.com/manco92/Algoritmos_gen

```
[14]: # Importo librerías
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

1.0.2 EJERCICIO 1

Una fábrica produce cuatro tipos de partes automotrices. Cada una de ellas primero se fabrica y luego se le dan los acabados. Las horas de trabajador requeridas y la utilidad para cada parte son las siguientes:

	A	B	C	D
Tiempo de fabricación (hs/100 un)	2.5	1.5	2.75	2
Tiempo de acabados (hs/100 un)	3.5	3	3	2
Utilidad (hs/100 un)	375	275	475	325

Las capacidades de los talleres de fabricación y acabados para el mes siguiente son de 640 y 960 horas, respectivamente. Determinar mediante un algoritmo PSO con restricciones (sin usar bibliotecas para PSO) que cantidad de cada parte debe producirse a fin de maximizar la utilidad y resolver las siguientes consignas: * Transcribir el algoritmo escrito en Python a un archivo .pdf de acuerdo a los siguientes parámetros: número de partículas = 20, máximo número de iteraciones 50, coeficientes de aceleración $c1 = c2 = 1.4944$, factor de inercia $w = 0.6$. * Transcribir al .pdf la solución óptima encontrada (dominio) y el valor objetivo óptimo (imagen). * Indicar en el .pdf la URL del repositorio en donde se encuentra el algoritmo PSO.+ * Realizar un gráfico de línea que muestre gbest (eje de ordenadas) en función de las iteraciones realizadas (eje de abscisas). El gráfico debe contener etiquetas en los ejes, leyenda y un título. El gráfico debe ser pegado en el .pdf. * Explicar (en el .pdf) y demostrar (desde el código fuente) que sucede si se reduce en 1 unidad el tiempo de acabado de la parte B. * Realizar observaciones/comentarios/conclusiones en el .pdf acerca de qué cantidad mínima de partículas es recomendable utilizar para este problema específicamente.

```
[3]: # función objetivo a maximizar
def f(x):
    return 375 * x[0] + 275 * x[1] + 475 * x[2] + 325 * x[3]

# primera restriccion
def g1(x):
    return 2.5 * x[0] + 1.5 * x[1] + 2.75 * x[2] + 2 * x[3] - 640 <= 0

# segunda restriccion
def g2(x):
    return 3.5 * x[0] + 3 * x[1] + 3 * x[2] + 2 * x[3] - 960 <= 0

# tercera restriccion
def g3(x):
    return x[0] >= 0 and x[1] >= 0 and x[2] >= 0 and x[3] >= 0

[4]: # parametros
n_particles = 20 # numero de particulas en el enjambre
n_dimensions = 4 # dimensiones del espacio de busqueda (x1 y x2)
max_iterations = 50 # numero máximo de iteraciones para la optimizacion
c1 = c2 = 1.4944 # coeficientes de aceleracion
w = 0.6 # Factor de inercia

def solve_pso(f, g1, g2, g3, n_particles, n_dimensions, max_iterations, c1, c2, w):

    # inicialización de particulas
    x = np.zeros((n_particles, n_dimensions)) # matriz para las posiciones de
    ↪ las particulas
    v = np.zeros((n_particles, n_dimensions)) # matriz para las velocidades de
    ↪ las particulas
    pbest = np.zeros((n_particles, n_dimensions)) # matriz para los mejores
    ↪ valores personales
    pbest_fit = -np.inf * np.ones(n_particles) # vector para las mejores
    ↪ aptitudes personales (inicialmente -infinito)
    gbest = np.zeros(n_dimensions) # mejor solución global
    gbest_fit = -np.inf # mejor aptitud global (inicialmente -infinito)
    g_best = []

    # inicializacion de particulas factibles
    for i in range(n_particles):
        while True: # bucle para asegurar que la particula sea factible
            x[i] = np.random.uniform(0, 10, n_dimensions) # inicializacion
            ↪ posicion aleatoria en el rango [0, 10]
```

```

        if g1(x[i]) and g2(x[i]) and g3(x[i]): # se comprueba si la
↪posicion cumple las restricciones
            break # Salir del bucle si es factible
        v[i] = np.random.uniform(-1, 1, n_dimensions) # inicializar velocidad
↪aleatoria
        pbest[i] = x[i].copy() # ee establece el mejor valor personal inicial
↪como la posicion actual
        fit = f(x[i]) # calculo la aptitud de la posicion inicial
        if fit > pbest_fit[i]: # si la aptitud es mejor que la mejor conocida
            pbest_fit[i] = fit # se actualiza el mejor valor personal

# Optimizacion
for _ in range(max_iterations): # Repetir hasta el número máximo de
↪iteraciones
    for i in range(n_particles):
        fit = f(x[i]) # Se calcula la aptitud de la posicion actual
        # Se comprueba si la nueva aptitud es mejor y si cumple las
↪restricciones
        if fit > pbest_fit[i] and g1(x[i]) and g2(x[i]) and g3(x[i]):
            pbest_fit[i] = fit # Se actualiza la mejor aptitud personal
            pbest[i] = x[i].copy() # Se actualizar la mejor posicion
↪personal
            if fit > gbest_fit: # Si la nueva aptitud es mejor que la
↪mejor global
                gbest_fit = fit # Se actualizar la mejor aptitud global
                gbest = x[i].copy() # Se actualizar la mejor posicion
↪global

        # actualizacion de la velocidad de la particula
        v[i] = w * v[i] + c1 * np.random.rand() * (pbest[i] - x[i]) + c2 *
↪np.random.rand() * (gbest - x[i])
        x[i] += v[i] # Se actualiza la posicion de la particula

        # se asegura de que la nueva posicion esté dentro de las
↪restricciones
        if not (g1(x[i]) and g2(x[i]) and g3(x[i])):
            # Si la nueva posicion no es válida, revertir a la mejor
↪posicion personal
            x[i] = pbest[i].copy()

    g_bests.append(gbest)

# Se imprime la mejor solucion encontrada y también su valor optimo
mejor_solucion = gbest
valor_optimo = gbest_fit

```

```

    print(f"Mejor solucion: [{gbest[0]:.0f}, {gbest[1]:.0f}, {gbest[2]:.0f}, {gbest[3]:.0f}]")
    print(f"Valor optimo: {gbest_fit}")

    return mejor_solucion, valor_optimo, g_bests

```

```

[5]: solucion_optima, valor_optimo, g_bests = solve_pso(f, g1, g2, g3, n_particles, n_dimensions, max_iterations, c1, c2, w)

```

Mejor solucion: [48, 87, 48, 129]

Valor optimo: 106539.86855501271

```

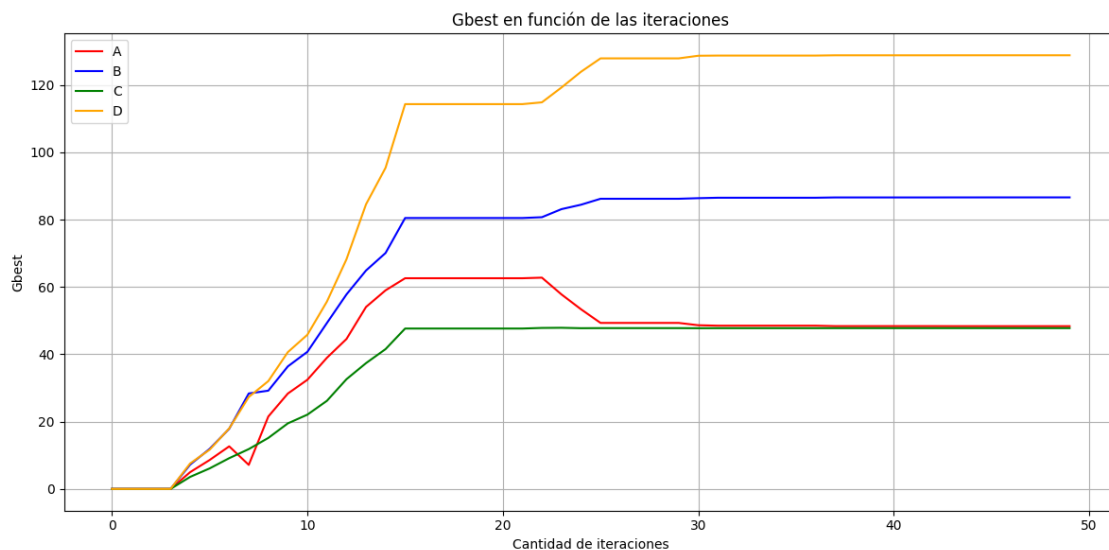
[6]: fig, ax = plt.subplots(figsize=(12,6))

x = np.arange(len(g_bests))
y = g_bests

ax.plot(x, np.array(g_bests)[: , 0], color='red', label='A')
ax.plot(x, np.array(g_bests)[: , 1], color='blue', label='B')
ax.plot(x, np.array(g_bests)[: , 2], color='green', label='C')
ax.plot(x, np.array(g_bests)[: , 3], color='orange', label='D')
ax.set_title('Gbest en función de las iteraciones')
ax.set_xlabel('Cantidad de iteraciones')
ax.set_ylabel('Gbest')

plt.legend()
plt.grid()
plt.tight_layout()
plt.show()

```



Qué ocurre si se reduce en 1 unidad el tiempo de acabado de B

```
[7]: # se modifica la segunda restricción
def g2(x):
    return 3.5 * x[0] + 2 * x[1] + 3 * x[2] + 2 * x[3] - 960 <= 0
```

```
[8]: solucion_optima, valor_optimo, g_bests = solve_pso(f, g1, g2, g3, n_particles,
    ↪ n_dimensions, max_iterations, c1, c2, w)
```

Mejor solucion: [70, 112, 16, 127]

Valor optimo: 105744.98514585469

Al reducir en una unidad el tiempo de acabado de B, la producción A y B baja, mientras que la de C y D crece.

PRODUCCIÓN PREVIA: [81, 59, 83, 61]

PRODUCCIÓN ACTUAL: [51, 47, 104, 78]

El gráfico de gbest en función de la cantidad de iteraciones indica que no son necesarias más de 12 iteraciones, porque ya se halló la solución.

1.0.3 Cálculo de número de partículas mínimo

```
[9]: valores_optimos = []
for n_particles in range(1, 80):
    solucion_optima, valor_optimo, g_bests = solve_pso(f, g1, g2, g3,
    ↪ n_particles, n_dimensions, max_iterations, c1, c2, w)
    valores_optimos.append(valor_optimo)
```

Mejor solucion: [11, 12, 13, 9]

Valor optimo: 16332.50308567811

Mejor solucion: [4, 44, 0, 28]

Valor optimo: 22349.345338534236

Mejor solucion: [9, 55, 23, 73]

Valor optimo: 53108.463804340114

Mejor solucion: [15, 10, 0, 13]

Valor optimo: 12998.444380356079

Mejor solucion: [93, 91, 2, 133]

Valor optimo: 104011.5526971564

Mejor solucion: [86, 78, 48, 88]

Valor optimo: 105115.20526840733

Mejor solucion: [74, 66, 44, 0]

Valor optimo: 66792.18918077924

Mejor solucion: [53, 99, 88, 59]

Valor optimo: 107900.37923755718

Mejor solucion: [66, 101, 77, 57]

Valor optimo: 107249.63761419992

Mejor solucion: [109, 148, 1, 71]

Valor optimo: 105270.01906060203

Mejor solucion: [68, 94, 75, 62]
Valor optimo: 106930.22587301582
Mejor solucion: [0, 16, 165, 80]
Valor optimo: 109149.24038194586
Mejor solucion: [94, 72, 63, 62]
Valor optimo: 105102.14661820144
Mejor solucion: [4, 72, 118, 98]
Valor optimo: 109473.77928066811
Mejor solucion: [66, 72, 73, 85]
Valor optimo: 106229.21617176825
Mejor solucion: [60, 15, 82, 121]
Valor optimo: 104863.97324213383
Mejor solucion: [79, 100, 58, 67]
Valor optimo: 106274.23592706177
Mejor solucion: [92, 95, 57, 56]
Valor optimo: 105689.53653031585
Mejor solucion: [73, 80, 83, 55]
Valor optimo: 106528.67298944578
Mejor solucion: [54, 77, 90, 70]
Valor optimo: 107262.3113028115
Mejor solucion: [7, 91, 111, 91]
Valor optimo: 109733.61972501736
Mejor solucion: [24, 105, 129, 35]
Valor optimo: 110148.93932391133
Mejor solucion: [92, 73, 49, 84]
Valor optimo: 104787.54617093159
Mejor solucion: [58, 67, 91, 73]
Valor optimo: 106838.89399171388
Mejor solucion: [70, 103, 29, 115]
Valor optimo: 105830.22523748886
Mejor solucion: [80, 84, 0, 156]
Valor optimo: 104133.47272380159
Mejor solucion: [98, 72, 68, 50]
Valor optimo: 105106.26763572628
Mejor solucion: [99, 84, 35, 86]
Valor optimo: 104501.31985655363
Mejor solucion: [77, 54, 94, 54]
Valor optimo: 105929.36873357481
Mejor solucion: [128, 46, 90, 2]
Valor optimo: 103995.28540705229
Mejor solucion: [65, 68, 79, 79]
Valor optimo: 106313.9694692454
Mejor solucion: [83, 1, 98, 81]
Valor optimo: 104173.06584578584
Mejor solucion: [47, 55, 89, 97]
Valor optimo: 106785.68028029322
Mejor solucion: [90, 70, 53, 81]
Valor optimo: 104850.6544510638

Mejor solucion: [70, 57, 80, 80]
Valor optimo: 105822.87963134576
Mejor solucion: [100, 91, 84, 12]
Valor optimo: 106087.16304465353
Mejor solucion: [44, 68, 88, 93]
Valor optimo: 107218.01521608428
Mejor solucion: [60, 54, 76, 99]
Valor optimo: 105962.03791981346
Mejor solucion: [82, 60, 87, 53]
Valor optimo: 105774.55046211783
Mejor solucion: [0, 33, 112, 141]
Valor optimo: 108160.94586480042
Mejor solucion: [84, 81, 60, 72]
Valor optimo: 105592.79317897573
Mejor solucion: [75, 73, 75, 68]
Valor optimo: 106075.35142398566
Mejor solucion: [60, 58, 71, 104]
Valor optimo: 105929.27431865936
Mejor solucion: [86, 72, 80, 48]
Valor optimo: 105815.62000136741
Mejor solucion: [101, 37, 51, 96]
Valor optimo: 103436.46584411255
Mejor solucion: [0, 88, 121, 87]
Valor optimo: 110125.03192442696
Mejor solucion: [60, 78, 112, 32]
Valor optimo: 107696.86121865019
Mejor solucion: [98, 96, 86, 7]
Valor optimo: 106346.40145652322
Mejor solucion: [83, 96, 77, 38]
Valor optimo: 106574.19040537947
Mejor solucion: [73, 121, 76, 34]
Valor optimo: 107626.53995169509
Mejor solucion: [36, 88, 78, 102]
Valor optimo: 107846.21070796269
Mejor solucion: [36, 104, 98, 62]
Valor optimo: 108890.13435122861
Mejor solucion: [124, 13, 68, 61]
Valor optimo: 102435.58006623418
Mejor solucion: [24, 14, 124, 109]
Valor optimo: 107202.50454681195
Mejor solucion: [30, 86, 117, 57]
Valor optimo: 109034.74258164642
Mejor solucion: [84, 56, 83, 60]
Valor optimo: 105439.28779813348
Mejor solucion: [0, 139, 60, 133]
Valor optimo: 109998.30476401179
Mejor solucion: [74, 59, 75, 80]
Valor optimo: 105646.86715749982

```

Mejor solucion: [81, 35, 73, 92]
Valor optimo: 104626.37460041977
Mejor solucion: [77, 70, 68, 77]
Valor optimo: 105703.11753169262
Mejor solucion: [68, 62, 84, 73]
Valor optimo: 106155.7005176834
Mejor solucion: [48, 91, 78, 84]
Valor optimo: 107540.86445086423
Mejor solucion: [75, 50, 93, 61]
Valor optimo: 105811.0509705036
Mejor solucion: [64, 39, 116, 51]
Valor optimo: 106507.59555927126
Mejor solucion: [55, 75, 75, 92]
Valor optimo: 106710.2946261981
Mejor solucion: [61, 53, 92, 77]
Valor optimo: 106342.35737381785
Mejor solucion: [43, 145, 114, 0]
Valor optimo: 110366.18895442277
Mejor solucion: [0, 135, 122, 50]
Valor optimo: 111649.31643661464
Mejor solucion: [96, 82, 46, 75]
Valor optimo: 104894.65436537436
Mejor solucion: [6, 141, 95, 75]
Valor optimo: 110890.28177334259
Mejor solucion: [0, 65, 128, 94]
Valor optimo: 109646.10942728212
Mejor solucion: [76, 67, 73, 74]
Valor optimo: 105765.93575211562
Mejor solucion: [64, 85, 73, 76]
Valor optimo: 106733.3525328993
Mejor solucion: [0, 163, 98, 63]
Valor optimo: 111836.40136682428
Mejor solucion: [84, 72, 74, 60]
Valor optimo: 105691.1203097401
Mejor solucion: [97, 3, 94, 68]
Valor optimo: 103701.57599036105
Mejor solucion: [75, 54, 62, 100]
Valor optimo: 105083.7422772265
Mejor solucion: [87, 96, 65, 50]
Valor optimo: 106093.96657986622
Mejor solucion: [78, 61, 90, 53]
Valor optimo: 106018.38888833081

```

```
[10]: fig, ax = plt.subplots(figsize=(12,6))
```

```

x = np.arange(79)
y = valores_optimos

```

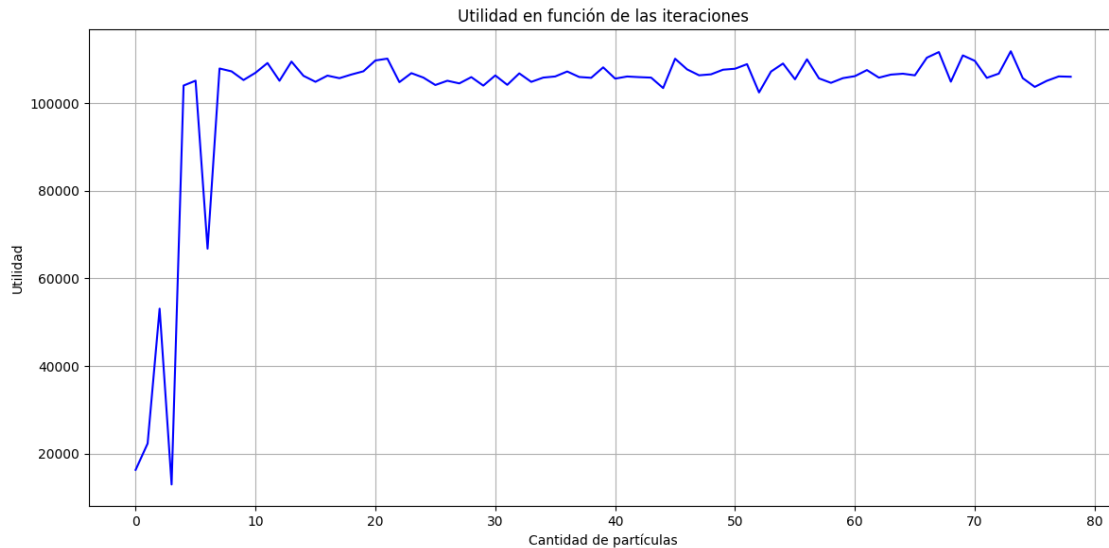


```

ax.plot(x, y, color='blue')
ax.set_title('Utilidad en función de las iteraciones')
ax.set_xlabel('Cantidad de partículas')
ax.set_ylabel('Utilidad')

plt.grid()
plt.tight_layout()
plt.show()

```



Como se puede ver en el gráfico anterior, el algoritmo de optimización es estocástico. Pero se ve claramente que a partir de 4 partículas es suficiente.

1.0.4 Calcular distribuciones de valores de producción

```

[ ]: # parametros
n_particles = 5 # numero de particulas en el enjambre
n_dimensions = 4 # dimensiones del espacio de busqueda (x1 y x2)
max_iterations = 50 # numero máximo de iteraciones para la optimizacion
c1 = c2 = 1.4944 # coeficientes de aceleracion
w = 0.6 # Factor de inercia

soluciones_optimas = []
for _ in range(200):
    solucion_optima, valor_optimo, g_bests = solve_pso(f, g1, g2, g3,
    ↪n_particles, n_dimensions, max_iterations, c1, c2, w)
    soluciones_optimas.append(solucion_optima)

```

```

[13]: soluciones_optimas = np.array(soluciones_optimas)

```

```
[ ]: fig, ax = plt.subplots(1,4, figsize=(20,5))
    sns.kdeplot()
```

soluciones_optimas

1.0.5 EJERCICIO 2

Un fabricante de equipos de cómputo produce dos tipos de impresoras (impresoras de tipo 1 e impresoras de tipo 2). Los recursos necesarios para producirlas así como las utilidades correspondientes son los que siguen:

Equipo	Capital (/un)	Mano de obra (hs/u)	Utilidad (/un)
Impresora 1	300	20	500
Impresora 2	400	10	400

Si cada día se dispone de \$127000 de capital y 4270 horas de mano de obra, ¿qué cantidad de cada equipo debe producirse a diario a fin de maximizar la utilidad? Escriba el algoritmo PSO con restricciones (sin usar bibliotecas para PSO) que permita optimizar la utilidad y resolver cumpliendo con las siguientes consignas: * Transcribir el algoritmo escrito en Python a un archivo .pdf de acuerdo a los siguientes parámetros: número de partículas = 10, máximo número de iteraciones 80, coeficientes de aceleración $c1 = c2 = 2$, factor de inercia $w = 0.5$. * Transcribir al .pdf la solución óptima encontrada (dominio) y el valor objetivo óptimo (imagen). * Indicar en el .pdf la URL del repositorio en donde se encuentra el algoritmo PSO. * Realizar un gráfico de línea que muestre gbest (eje de ordenadas) en función de las iteraciones realizadas (eje de abscisas). El gráfico debe contener etiquetas en los ejes, leyenda y un título. El gráfico debe ser pegado en el .pdf. * Explicar (en el .pdf) y demostrar (desde el código fuente) que sucede si se incrementa en 1 unidad la cantidad horas requeridas para fabricar la Impresora 2. * Realizar observaciones/comentarios/conclusiones en el .pdf acerca de qué cantidad mínima de partículas es recomendable utilizar para este problema específicamente.

```
[467]: # funcion objetivo hiperboloide eliptico
def funcion_objetivo(x):
    return np.sin(x)+np.sin(x**2)
```

```
[468]: def solve_PSO(funcion_objetivo, num_particulas, cantidad_iteraciones, c1, c2,
    ↪w, limite_inf, limite_sup):

    # inicializacion
    particulas = np.random.uniform(limite_inf, limite_sup, num_particulas) #
    ↪posiciones iniciales de las particulas

    velocidades = np.zeros((num_particulas)) # inicializacion de la matriz de
    ↪velocidades en cero

    # inicializacion de pbest y gbest
    pbest = particulas.copy() # mejores posiciones personales iniciales
```

```

    fitness_pbest = np.empty(num_particulas) # mejores fitness personales
    ↪ iniciales
    for i in range(num_particulas):
        fitness_pbest[i] = funcion_objetivo(particulas[i])

    gbest = pbest[np.argmin(fitness_pbest)] # mejor posicion global inicial
    fitness_gbest = np.min(fitness_pbest) # fitness global inicial

    g_bests = []

    # busqueda
    for iteracion in range(cantidad_iteraciones):
        for i in range(num_particulas): # iteracion sobre cada partícula
            r1, r2 = np.random.rand(), np.random.rand() # generacion dos
            ↪ numeros aleatorios

            # actualizacion de la velocidad de la partícula en cada dimension
            velocidades[i] = (w * velocidades[i] + c1 * r1 * (pbest[i] -
            ↪ particulas[i]) + c2 * r2 * (gbest - particulas[i]))

            particulas[i] = particulas[i] + velocidades[i] # cctualizacion de
            ↪ la posicion de la partícula en cada dimension

            # mantenimiento de las partículas dentro de los limites
            particulas[i] = np.clip(particulas[i], limite_inf, limite_sup)

            fitness = funcion_objetivo(particulas[i]) # Evaluacion de la
            ↪ funcion objetivo para la nueva posicion

            # actualizacion el mejor personal
            if fitness > fitness_pbest[i]:
                fitness_pbest[i] = fitness # actualizacion del mejor fitness
            ↪ personal

                pbest[i] = particulas[i].copy() # actualizacion de la mejor
            ↪ posicion personal

            # actualizacion del mejor global
            if fitness > fitness_gbest:
                fitness_gbest = fitness # actualizacion del mejor fitness
            ↪ global

                gbest = particulas[i].copy() # actualizacion de la mejor
            ↪ posicion global

    g_bests.append(gbest)

    # imprimir el mejor global en cada iteracion

```

```

        print(f"Iteración {iteracion + 1}: Mejor posición global {gbest}, Valor_
↪{fitness_gbest}")

    # resultado
    solucion_optima = gbest # mejor posicion global final
    valor_optimo = fitness_gbest # mejor fitness global final

    print("\nSolucion optima (x):", solucion_optima)
    print("Valor optimo:", valor_optimo)

    return solucion_optima, valor_optimo, g_bests

```

Con 2 partículas

```

[469]: # parametros
num_particulas = 2 # numero de particulas
cantidad_iteraciones = 30 # maximo numero de iteraciones
c1 = 1.49 # componente cognitivo
c2 = 1.49 # componente social
w = 0.5 # factor de inercia
limite_inf = 0 # limite inferior de busqueda
limite_sup = 10 # limite superior de busqueda

solucion_optima, valor_optimo, g_bests = solve_PSO(funcion_objetivo,
↪num_particulas, cantidad_iteraciones, c1, c2, w, limite_inf, limite_sup)

```

```

Iteración 1: Mejor posición global 6.51593363527382, Valor -0.7682946103484992
Iteración 2: Mejor posición global 6.51593363527382, Valor -0.7682946103484992
Iteración 3: Mejor posición global 6.51593363527382, Valor -0.7682946103484992
Iteración 4: Mejor posición global 6.51593363527382, Valor -0.7682946103484992
Iteración 5: Mejor posición global 6.51593363527382, Valor -0.7682946103484992
Iteración 6: Mejor posición global 6.51593363527382, Valor -0.7682946103484992
Iteración 7: Mejor posición global 6.51593363527382, Valor -0.7682946103484992
Iteración 8: Mejor posición global 6.51593363527382, Valor -0.7682946103484992
Iteración 9: Mejor posición global 6.51593363527382, Valor -0.7682946103484992
Iteración 10: Mejor posición global 6.51593363527382, Valor -0.7682946103484992
Iteración 11: Mejor posición global 6.51593363527382, Valor -0.7682946103484992
Iteración 12: Mejor posición global 6.51593363527382, Valor -0.7682946103484992
Iteración 13: Mejor posición global 6.51593363527382, Valor -0.7682946103484992
Iteración 14: Mejor posición global 6.51593363527382, Valor -0.7682946103484992
Iteración 15: Mejor posición global 6.51593363527382, Valor -0.7682946103484992
Iteración 16: Mejor posición global 6.51593363527382, Valor -0.7682946103484992
Iteración 17: Mejor posición global 6.51593363527382, Valor -0.7682946103484992
Iteración 18: Mejor posición global 6.51593363527382, Valor -0.7682946103484992
Iteración 19: Mejor posición global 6.51593363527382, Valor -0.7682946103484992
Iteración 20: Mejor posición global 6.51593363527382, Valor -0.7682946103484992
Iteración 21: Mejor posición global 6.51593363527382, Valor -0.7682946103484992
Iteración 22: Mejor posición global 6.51593363527382, Valor -0.7682946103484992

```

Iteración 23: Mejor posición global 6.51593363527382, Valor -0.7682946103484992
 Iteración 24: Mejor posición global 6.51593363527382, Valor -0.7682946103484992
 Iteración 25: Mejor posición global 6.51593363527382, Valor -0.7682946103484992
 Iteración 26: Mejor posición global 6.51593363527382, Valor -0.7682946103484992
 Iteración 27: Mejor posición global 6.51593363527382, Valor -0.7682946103484992
 Iteración 28: Mejor posición global 6.51593363527382, Valor -0.7682946103484992
 Iteración 29: Mejor posición global 6.51593363527382, Valor -0.7682946103484992
 Iteración 30: Mejor posición global 6.51593363527382, Valor -0.7682946103484992

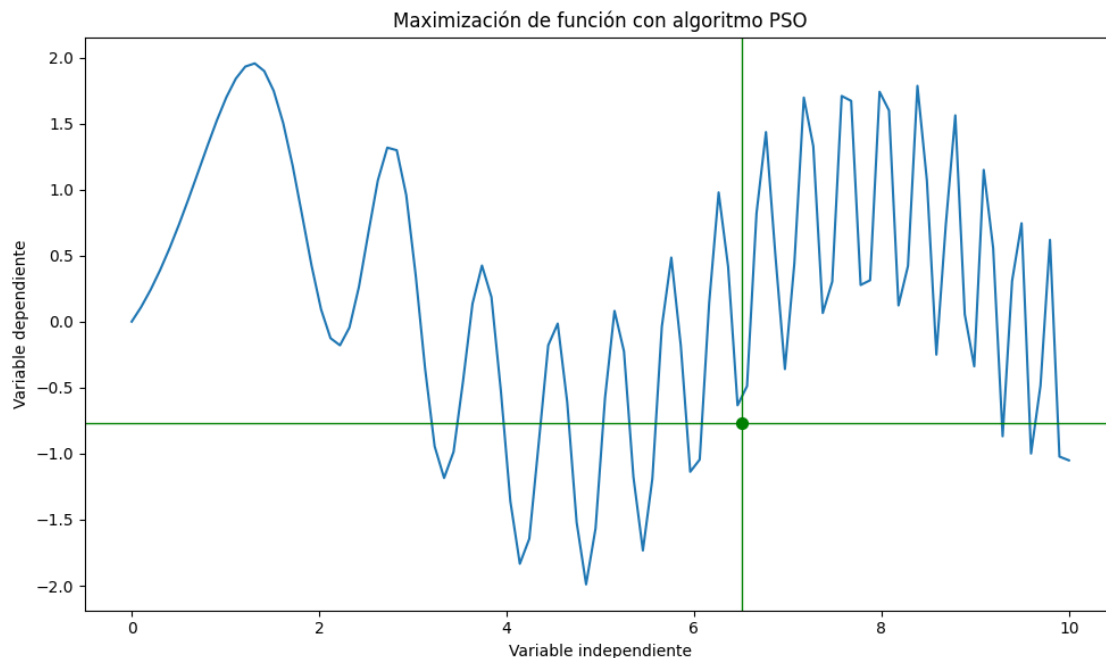
Solucion optima (x): 6.51593363527382

Valor optimo: -0.7682946103484992

```
[470]: fig, ax = plt.subplots(figsize=(10,6))

x = np.linspace(limite_inf, limite_sup, 100)
y = funcion_objetivo(x)

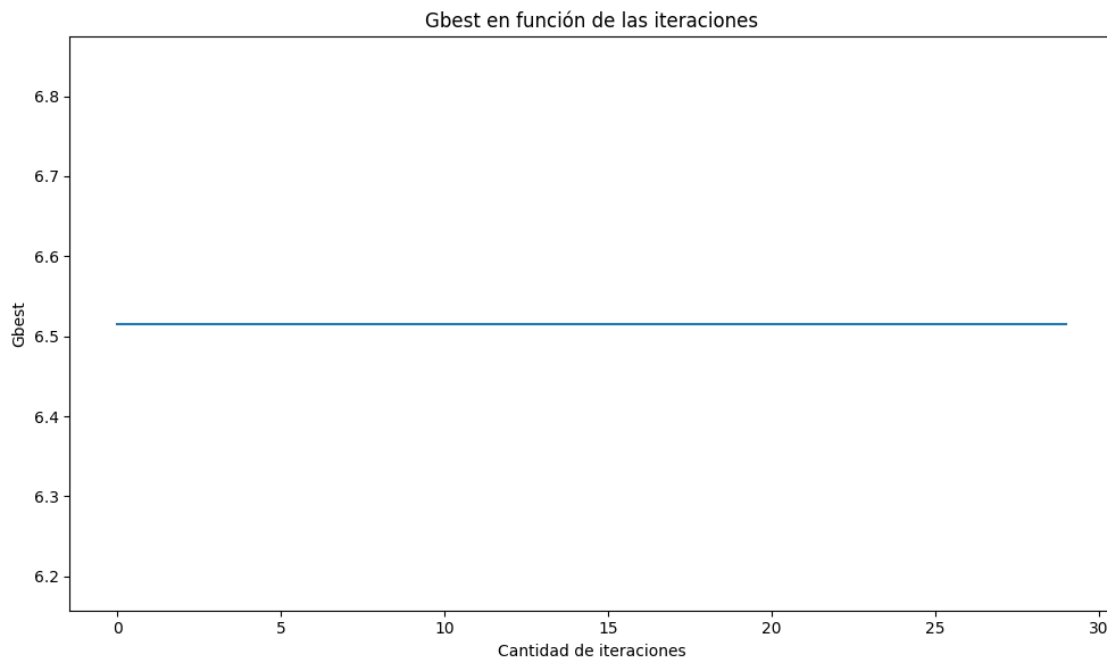
ax.plot(x, y)
ax.scatter(solucion_optima, valor_optimo, s=50, c='green')
ax.axvline(x=solucion_optima, c='green', linewidth=1)
ax.axhline(y=valor_optimo, c='green', linewidth=1)
ax.set_title('Maximización de función con algoritmo PSO')
ax.set_xlabel('Variable independiente')
ax.set_ylabel('Variable dependiente')
plt.tight_layout()
plt.show()
```



```
[471]: fig, ax = plt.subplots(figsize=(10,6))

x = np.arange(cantidad_iteraciones)
y = g_bests

ax.plot(x, y)
ax.set_title('Gbest en función de las iteraciones')
ax.set_xlabel('Cantidad de iteraciones')
ax.set_ylabel('Gbest')
plt.tight_layout()
plt.show()
```



Con 4 partículas

```
[472]: # parametros
num_particulas = 4 # numero de particulas
cantidad_iteraciones = 30 # maximo numero de iteraciones
c1 = 1.49 # componente cognitivo
c2 = 1.49 # componente social
w = 0.5 # factor de inercia
limite_inf = 0 # limite inferior de busqueda
limite_sup = 10 # limite superior de busqueda

solucion_optima, valor_optimo, g_bests = solve_PSO(funcion_objetivo,
↪ num_particulas, cantidad_iteraciones, c1, c2, w, limite_inf, limite_sup)
```

Iteración 1: Mejor posición global 9.094499363899265, Valor 1.180837376271547

Iteración 2: Mejor posición global 9.118373630030703, Valor 1.295863301550468
 Iteración 3: Mejor posición global 7.61678082880788, Valor 1.966584586717199
 Iteración 4: Mejor posición global 7.61678082880788, Valor 1.966584586717199
 Iteración 5: Mejor posición global 7.61678082880788, Valor 1.966584586717199
 Iteración 6: Mejor posición global 7.61678082880788, Valor 1.966584586717199
 Iteración 7: Mejor posición global 8.016835226052132, Valor 1.9779368526285899
 Iteración 8: Mejor posición global 8.016835226052132, Valor 1.9779368526285899
 Iteración 9: Mejor posición global 8.016835226052132, Valor 1.9779368526285899
 Iteración 10: Mejor posición global 8.016835226052132, Valor 1.9779368526285899
 Iteración 11: Mejor posición global 8.016835226052132, Valor 1.9779368526285899
 Iteración 12: Mejor posición global 8.02829464151312, Valor 1.9835525492730806
 Iteración 13: Mejor posición global 8.02829464151312, Valor 1.9835525492730806
 Iteración 14: Mejor posición global 8.02829464151312, Valor 1.9835525492730806
 Iteración 15: Mejor posición global 8.02829464151312, Valor 1.9835525492730806
 Iteración 16: Mejor posición global 8.02829464151312, Valor 1.9835525492730806
 Iteración 17: Mejor posición global 8.02829464151312, Valor 1.9835525492730806
 Iteración 18: Mejor posición global 8.02829464151312, Valor 1.9835525492730806
 Iteración 19: Mejor posición global 8.022193529136144, Valor 1.9847785715953963
 Iteración 20: Mejor posición global 8.025399955284835, Valor 1.9853341625515926
 Iteración 21: Mejor posición global 8.025399955284835, Valor 1.9853341625515926
 Iteración 22: Mejor posición global 8.025399955284835, Valor 1.9853341625515926
 Iteración 23: Mejor posición global 8.025399955284835, Valor 1.9853341625515926
 Iteración 24: Mejor posición global 8.025399955284835, Valor 1.9853341625515926
 Iteración 25: Mejor posición global 8.025399955284835, Valor 1.9853341625515926
 Iteración 26: Mejor posición global 8.024508173247725, Valor 1.9854463664175066
 Iteración 27: Mejor posición global 8.024508173247725, Valor 1.9854463664175066
 Iteración 28: Mejor posición global 8.024508173247725, Valor 1.9854463664175066
 Iteración 29: Mejor posición global 8.024508173247725, Valor 1.9854463664175066
 Iteración 30: Mejor posición global 8.024508173247725, Valor 1.9854463664175066

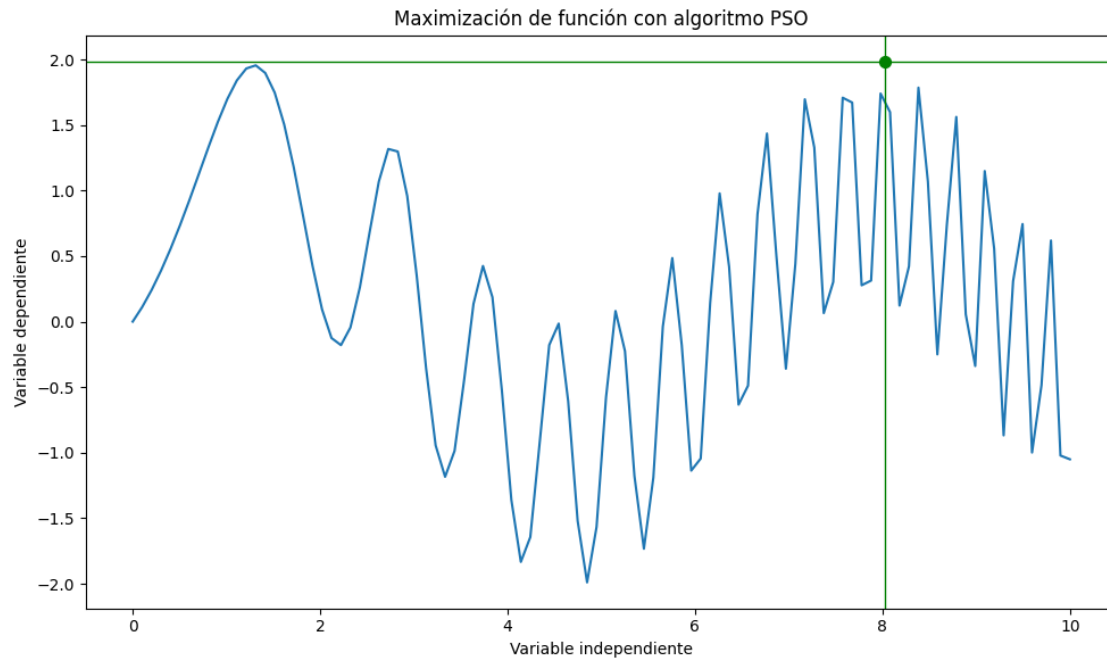
Solucion optima (x): 8.024508173247725
 Valor optimo: 1.9854463664175066

```

[473]: fig, ax = plt.subplots(figsize=(10,6))

x = np.linspace(limite_inf, limite_sup, 100)
y = funcion_objetivo(x)

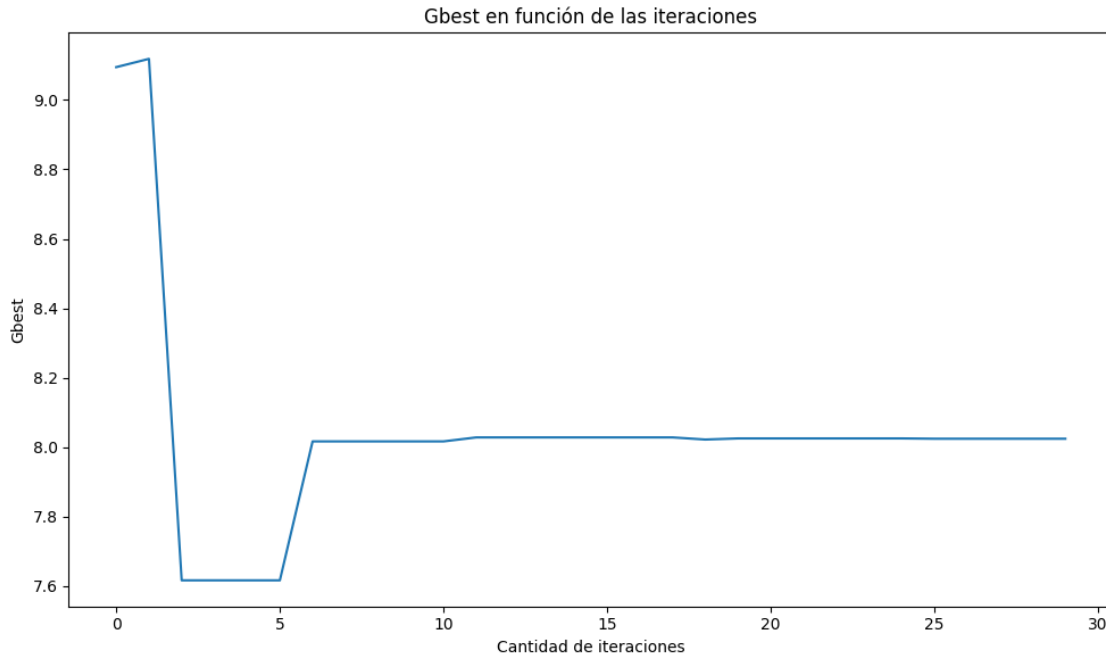
ax.plot(x, y)
ax.scatter(solucion_optima, valor_optimo, s=50, c='green')
ax.axvline(x=solucion_optima, c='green', linewidth=1)
ax.axhline(y=valor_optimo, c='green', linewidth=1)
ax.set_title('Maximización de función con algoritmo PSO')
ax.set_xlabel('Variable independiente')
ax.set_ylabel('Variable dependiente')
plt.tight_layout()
plt.show()
  
```



```
[474]: fig, ax = plt.subplots(figsize=(10,6))

x = np.arange(cantidad_iteraciones)
y = g_bests

ax.plot(x, y)
ax.set_title('Gbest en función de las iteraciones')
ax.set_xlabel('Cantidad de iteraciones')
ax.set_ylabel('Gbest')
plt.tight_layout()
plt.show()
```

Con 6 partículas

```
[475]: # parametros
num_particulas = 6 # numero de particulas
cantidad_iteraciones = 30 # maximo numero de iteraciones
c1 = 1.49 # componente cognitivo
c2 = 1.49 # componente social
w = 0.5 # factor de inercia
limite_inf = 0 # limite inferior de busqueda
limite_sup = 10 # limite superior de busqueda

solucion_optima, valor_optimo, g_bests = solve_PSO(funcion_objetivo,
↪ num_particulas, cantidad_iteraciones, c1, c2, w, limite_inf, limite_sup)
```

```
Iteración 1: Mejor posición global 0.5203314898328146, Valor 0.7646170201288318
Iteración 2: Mejor posición global 1.2153353612930071, Valor 1.9330942599905843
Iteración 3: Mejor posición global 1.2153353612930071, Valor 1.9330942599905843
Iteración 4: Mejor posición global 1.2153353612930071, Valor 1.9330942599905843
Iteración 5: Mejor posición global 1.2153353612930071, Valor 1.9330942599905843
Iteración 6: Mejor posición global 1.2153353612930071, Valor 1.9330942599905843
Iteración 7: Mejor posición global 1.2667444503209953, Valor 1.953558545539272
Iteración 8: Mejor posición global 1.3094304553356553, Valor 1.9557148323917897
Iteración 9: Mejor posición global 1.3094304553356553, Valor 1.9557148323917897
Iteración 10: Mejor posición global 1.299341688089008, Valor 1.9564875722191717
Iteración 11: Mejor posición global 1.299341688089008, Valor 1.9564875722191717
Iteración 12: Mejor posición global 1.299341688089008, Valor 1.9564875722191717
Iteración 13: Mejor posición global 1.299341688089008, Valor 1.9564875722191717
```

```

Iteración 14: Mejor posición global 1.299341688089008, Valor 1.9564875722191717
Iteración 15: Mejor posición global 1.299341688089008, Valor 1.9564875722191717
Iteración 16: Mejor posición global 1.2965514703477115, Valor 1.9565586697255495
Iteración 17: Mejor posición global 1.2949923379369894, Valor 1.9565717081604785
Iteración 18: Mejor posición global 1.2949923379369894, Valor 1.9565717081604785
Iteración 19: Mejor posición global 1.2949923379369894, Valor 1.9565717081604785
Iteración 20: Mejor posición global 1.2949923379369894, Valor 1.9565717081604785
Iteración 21: Mejor posición global 1.2949082684012738, Valor 1.9565718689027005
Iteración 22: Mejor posición global 1.2946664322924237, Valor 1.9565720222387801
Iteración 23: Mejor posición global 1.2946664322924237, Valor 1.9565720222387801
Iteración 24: Mejor posición global 1.2946664322924237, Valor 1.9565720222387801
Iteración 25: Mejor posición global 1.2947164635842092, Valor 1.9565720281398362
Iteración 26: Mejor posición global 1.2947164635842092, Valor 1.9565720281398362
Iteración 27: Mejor posición global 1.2947149953543935, Valor 1.9565720282461787
Iteración 28: Mejor posición global 1.2947149953543935, Valor 1.9565720282461787
Iteración 29: Mejor posición global 1.2947149953543935, Valor 1.9565720282461787
Iteración 30: Mejor posición global 1.2947149953543935, Valor 1.9565720282461787

```

Solucion optima (x): 1.2947149953543935

Valor optimo: 1.9565720282461787

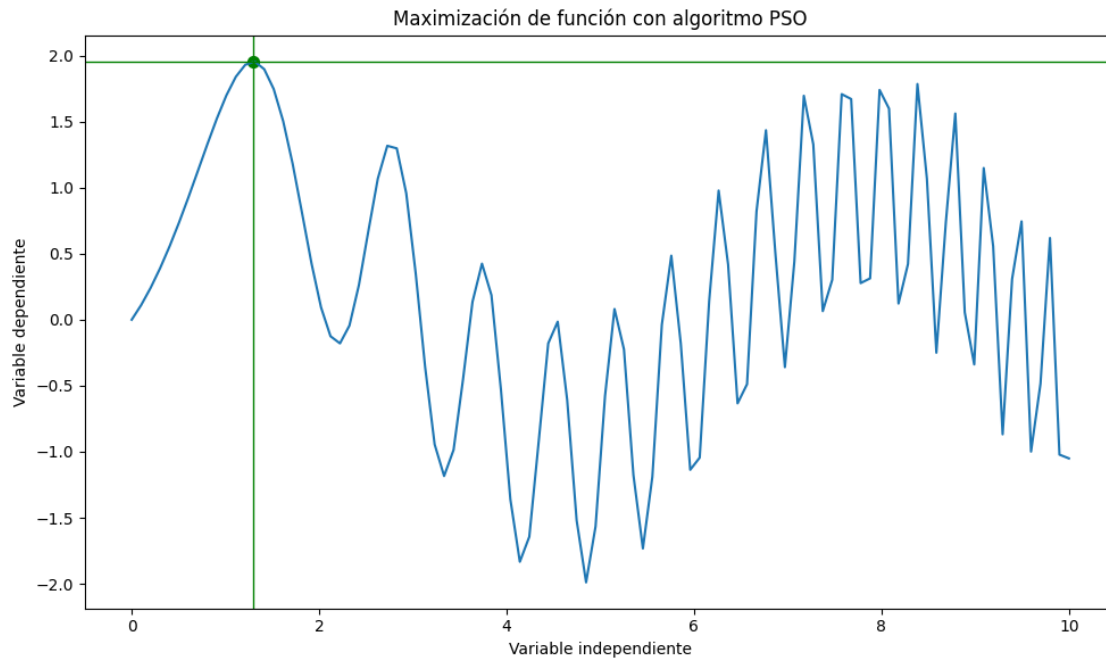
```

[476]: fig, ax = plt.subplots(figsize=(10,6))

x = np.linspace(limite_inf, limite_sup, 100)
y = funcion_objetivo(x)

ax.plot(x, y)
ax.scatter(solucion_optima, valor_optimo, s=50, c='green')
ax.axvline(x=solucion_optima, c='green', linewidth=1)
ax.axhline(y=valor_optimo, c='green', linewidth=1)
ax.set_title('Maximización de función con algoritmo PSO')
ax.set_xlabel('Variable independiente')
ax.set_ylabel('Variable dependiente')
plt.tight_layout()
plt.show()

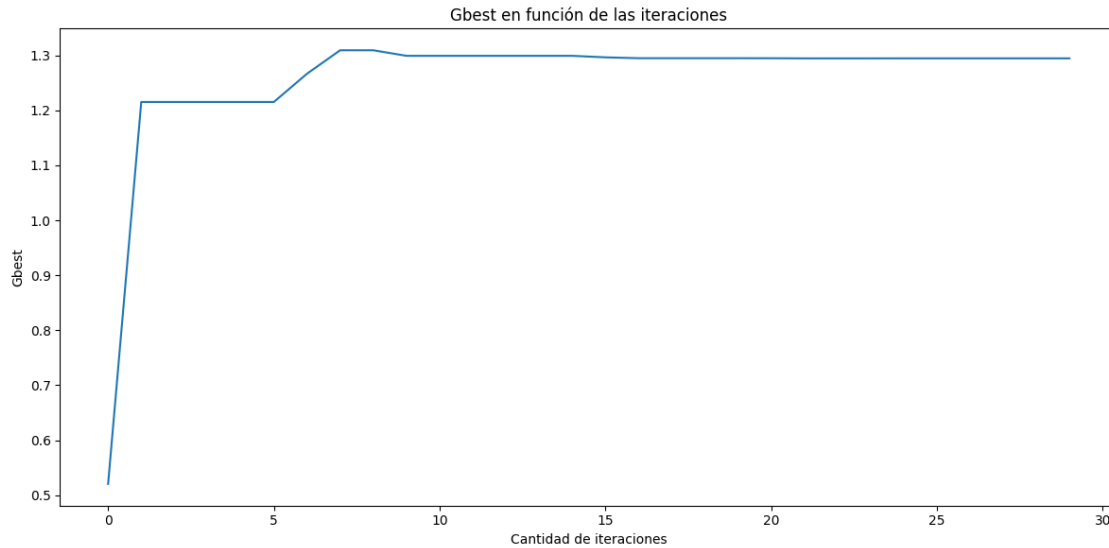
```



```
[477]: fig, ax = plt.subplots(figsize=(12,6))

x = np.arange(cantidad_iteraciones)
y = g_bests

ax.plot(x, y)
ax.set_title('Gbest en función de las iteraciones')
ax.set_xlabel('Cantidad de iteraciones')
ax.set_ylabel('Gbest')
plt.tight_layout()
plt.show()
```



Con 10 partículas

```
[478]: # parametros
num_particulas = 10 # numero de particulas
cantidad_iteraciones = 30 # maximo numero de iteraciones
c1 = 1.49 # componente cognitivo
c2 = 1.49 # componente social
w = 0.5 # factor de inercia
limite_inf = 0 # limite inferior de busqueda
limite_sup = 10 # limite superior de busqueda

solucion_optima, valor_optimo, g_bests = solve_PSO(funcion_objetivo,
↪ num_particulas, cantidad_iteraciones, c1, c2, w, limite_inf, limite_sup)
```

```
Iteración 1: Mejor posición global 1.5184462798819833, Valor 1.7405394050135388
Iteración 2: Mejor posición global 1.5184462798819833, Valor 1.7405394050135388
Iteración 3: Mejor posición global 1.5184462798819833, Valor 1.7405394050135388
Iteración 4: Mejor posición global 1.5184462798819833, Valor 1.7405394050135388
Iteración 5: Mejor posición global 1.2725536765884482, Valor 1.9546737669180514
Iteración 6: Mejor posición global 1.2725536765884482, Valor 1.9546737669180514
Iteración 7: Mejor posición global 1.2738976397503343, Valor 1.9548957176843018
Iteración 8: Mejor posición global 1.2983349910625934, Valor 1.956520304313674
Iteración 9: Mejor posición global 1.2983349910625934, Valor 1.956520304313674
Iteración 10: Mejor posición global 1.2983349910625934, Valor 1.956520304313674
Iteración 11: Mejor posición global 1.2983349910625934, Valor 1.956520304313674
Iteración 12: Mejor posición global 1.2940454201784444, Valor 1.9565703159938164
Iteración 13: Mejor posición global 1.2940454201784444, Valor 1.9565703159938164
Iteración 14: Mejor posición global 1.2945060270525932, Valor 1.956571871009313
Iteración 15: Mejor posición global 1.2945060270525932, Valor 1.956571871009313
Iteración 16: Mejor posición global 1.2945060270525932, Valor 1.956571871009313
```

```

Iteración 17: Mejor posición global 1.2945060270525932, Valor 1.956571871009313
Iteración 18: Mejor posición global 1.2945060270525932, Valor 1.956571871009313
Iteración 19: Mejor posición global 1.2945060270525932, Valor 1.956571871009313
Iteración 20: Mejor posición global 1.2945076105755886, Valor 1.956571873487928
Iteración 21: Mejor posición global 1.2946901076273643, Valor 1.9565720274773293
Iteración 22: Mejor posición global 1.2947216441885303, Valor 1.9565720276295746
Iteración 23: Mejor posición global 1.2947216441885303, Valor 1.9565720276295746
Iteración 24: Mejor posición global 1.2947216441885303, Valor 1.9565720276295746
Iteración 25: Mejor posición global 1.2947216441885303, Valor 1.9565720276295746
Iteración 26: Mejor posición global 1.2947216441885303, Valor 1.9565720276295746
Iteración 27: Mejor posición global 1.2946964417533804, Valor 1.9565720281337124
Iteración 28: Mejor posición global 1.2947116234303158, Valor 1.956572028426422
Iteración 29: Mejor posición global 1.2947116234303158, Valor 1.956572028426422
Iteración 30: Mejor posición global 1.2947116234303158, Valor 1.956572028426422

```

Solucion optima (x): 1.2947116234303158

Valor optimo: 1.956572028426422

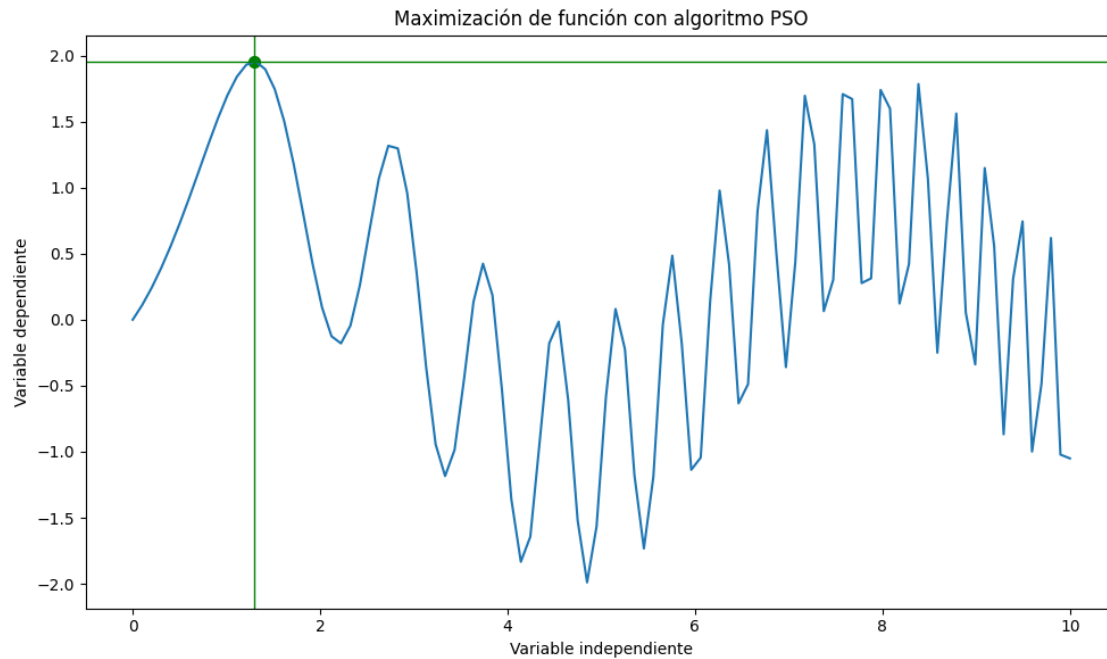
```

[479]: fig, ax = plt.subplots(figsize=(10,6))

x = np.linspace(limite_inf, limite_sup, 100)
y = funcion_objetivo(x)

ax.plot(x, y)
ax.scatter(solucion_optima, valor_optimo, s=50, c='green')
ax.axvline(x=solucion_optima, c='green', linewidth=1)
ax.axhline(y=valor_optimo, c='green', linewidth=1)
ax.set_title('Maximización de función con algoritmo PSO')
ax.set_xlabel('Variable independiente')
ax.set_ylabel('Variable dependiente')
plt.tight_layout()
plt.show()

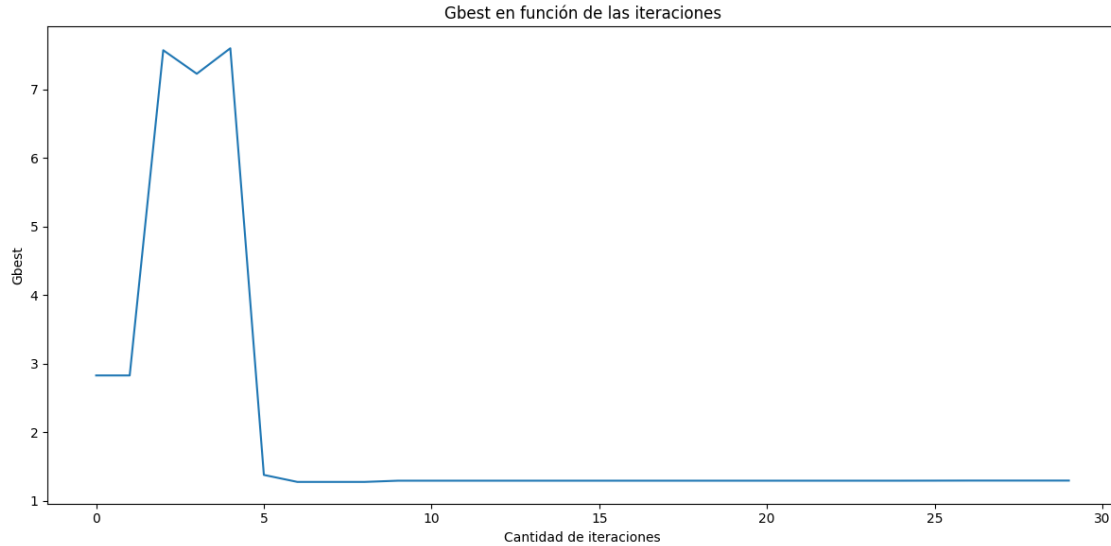
```



```
[444]: fig, ax = plt.subplots(figsize=(12,6))

x = np.arange(cantidad_iteraciones)
y = g_bests

ax.plot(x, y)
ax.set_title('Gbest en función de las iteraciones')
ax.set_xlabel('Cantidad de iteraciones')
ax.set_ylabel('Gbest')
plt.tight_layout()
plt.show()
```



Cuando se varía el número de partículas, se está permitiendo una exploración más amplia de posibles soluciones. Cuando se optó hasta 6 partículas, el algoritmo eligió un máximo local. Sin embargo, cuando se aumentó el número de partículas, el algoritmo halló el máximo global.

1.0.6 EJERCICIO 3

Dada la siguiente función perteneciente a un paraboloide elíptico de la forma:

$$f(x, y) = (x - a)^2 + (y + b)^2$$

donde, las constantes a y b son valores reales ingresados por el usuario a través de la consola, con intervalos de:

$$-100 \leq x \leq 100 - 100 \leq y \leq 100 - 50 \leq a \leq 50 - 50 \leq b \leq 50$$

escribir en Python un algoritmo PSO para la minimización de la función $f(x, y)$ que cumpla con las siguientes consignas: * Transcribir el algoritmo utilizando los siguientes parámetros: número de partículas = 20, máximo número de iteraciones = 10, coeficientes de aceleración $c1 = c2 = 2$, peso de inercia $w = 0.7$. * Indicar la URL del repositorio en donde se encuentra el algoritmo PSO. * Graficar usando matplotlib la función objetivo $f(x, y)$ y agregar un punto rojo en donde el algoritmo haya encontrado el valor mínimo. El gráfico debe contener etiquetas en los ejes, leyenda y un título. * Realizar un gráfico de línea que muestre gbest en función de las iteraciones realizadas. * Transcribir la solución óptima encontrada (dominio) y el valor objetivo óptimo (imagen). * Establecer el coeficiente de inercia w en 0, ejecutar el algoritmo y realizar observaciones/comentarios/conclusiones sobre los resultados observados. * Reescribir el algoritmo PSO para que cumpla nuevamente con los ítems A hasta F pero usando la biblioteca pswarm (`from pswarm import pso`). * Realizar observaciones/comentarios/conclusiones comparando los resultados obtenidos sin pswarm y con pswarm.

```

[480]: # funcion objetivo hiperboloide eliptico
def funcion_objetivo(a, b, x, y):
    return (x-a)**2+(y+b)**2

[481]: def solve_PSO(funcion_objetivo, num_particulas, dim, cantidad_iteraciones, c1,
    ↪c2, w, limite_inf, limite_sup, input_a, input_b):

    # inicializacion
    particulas = np.random.uniform(limite_inf, limite_sup, (num_particulas,
    ↪dim)) # posiciones iniciales de las particulas

    velocidades = np.zeros((num_particulas, dim)) # inicializacion de la
    ↪matriz de velocidades en cero

    # inicializacion de pbest y gbest
    pbest = particulas.copy() # mejores posiciones personales iniciales

    fitness_pbest = np.empty(num_particulas) # mejores fitness personales
    ↪iniciales
    for i in range(num_particulas):
        fitness_pbest[i] = funcion_objetivo(input_a, input_b, particulas[i][0],
    ↪particulas[i][1])

    gbest = pbest[np.argmin(fitness_pbest)] # mejor posicion global inicial
    fitness_gbest = np.min(fitness_pbest) # fitness global inicial

    g_best = []

    # busqueda
    for iteracion in range(cantidad_iteraciones):
        for i in range(num_particulas): # iteracion sobre cada partícula
            r1, r2 = np.random.rand(), np.random.rand() # generacion dos
            ↪numeros aleatorios

            # actualizacion de la velocidad de la partícula en cada dimension
            for d in range(dim):
                velocidades[i][d] = (w * velocidades[i][d] + c1 * r1 *
    ↪(pbest[i][d] - particulas[i][d]) + c2 * r2 * (gbest[d] - particulas[i][d]))

            for d in range(dim):
                particulas[i][d] = particulas[i][d] + velocidades[i][d] #
    ↪actualizacion de la posicion de la partícula en cada dimension

            # mantenimiento de las partículas dentro de los limites
            particulas[i][d] = np.clip(particulas[i][d], limite_inf,
    ↪limite_sup)

```



```

        fitness = funcion_objetivo(input_a, input_b, particulas[i][0],
↪particulas[i][1]) # Evaluacion de la funcion objetivo para la nueva posicion

        # actualizacion el mejor personal
        if fitness < fitness_pbest[i]:
            fitness_pbest[i] = fitness # actualizacion del mejor fitness
↪personal
            pbest[i] = particulas[i].copy() # actualizacion de la mejor
↪posicion personal

        # actualizacion del mejor global
        if fitness < fitness_gbest:
            fitness_gbest = fitness # actualizacion del mejor fitness
↪global
            gbest = particulas[i].copy() # actualizacion de la mejor
↪posicion global

        g_bests.append(gbest)

        # imprimir el mejor global en cada iteracion
        print(f"Iteración {iteracion + 1}: Mejor posición global {gbest}, Valor
↪{fitness_gbest}")

        # resultado
        solucion_optima = gbest # mejor posicion global final
        valor_optimo = fitness_gbest # mejor fitness global final

        print("\nSolucion optima (x, y):", solucion_optima)
        print("Valor optimo:", valor_optimo)

        return solucion_optima, valor_optimo, g_bests

```

```

[482]: from mpl_toolkits.mplot3d import Axes3D

def plot_3d(funcion_objetivo, limite_inf, limite_sup, min_x, min_y):
    # Crear una malla de puntos para x y y
    x = np.linspace(limite_inf, limite_sup, 400)
    y = np.linspace(limite_inf, limite_sup, 400)
    X, Y = np.meshgrid(x, y)
    Z = funcion_objetivo(input_a, input_b, X, Y)

    # Mínimo teórico de la función
    min_z = funcion_objetivo(input_a, input_b, min_x, min_y)

    # Configurar la gráfica 3D
    fig = plt.figure()

```

```

ax = fig.add_subplot(111, projection='3d')

# Graficar la superficie
ax.plot_surface(X, Y, Z, cmap='viridis', alpha=0.7)

# Graficar las líneas que indican el mínimo
ax.plot([min_x, min_x], [min_y, min_y], [-100, min_z], color='r',
↪linestyle='--') # Línea vertical desde el mínimo hasta el plano XY
ax.plot([min_x, min_x], [-100, min_y], [min_z, min_z], color='r',
↪linestyle='--') # Línea en el plano XY desde el eje Y al mínimo
ax.plot([-100, min_x], [min_y, min_y], [min_z, min_z], color='r',
↪linestyle='--') # Línea en el plano XY desde el eje X al mínimo

# Graficar el punto que indica el mínimo
ax.scatter(min_x, min_y, min_z, color='red', s=50, label='Mínimo')

# Etiquetas de los ejes
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')

# Título de la gráfica
ax.set_title('Gráfico 3D con el Mínimo de la Función')

# Mostrar la leyenda
ax.legend()

# Mostrar la gráfica
plt.show()

```

```

[483]: # parametros
num_particulas = 20 # numero de particulas
dim = 2 # dimensiones
cantidad_iteraciones = 10 # maximo numero de iteraciones
c1 = 2.0 # componente cognitivo
c2 = 2.0 # componente social
w = 0.7 # factor de inercia
limite_inf = -100 # limite inferior de busqueda
limite_sup = 100 # limite superior de busqueda

input_a = input("Introduzca el valor del parámetro a")
input_a = max(min(float(input_a), 60), -60)
print("a: ", input_a)

input_b = input("Introduzca el valor del parámetro b")
input_b = max(min(float(input_b), 60), -60)
print("b: ", input_b)

```

```

solucion_optima, valor_optimo, g_bests = solve_PSO(funcion_objetivo,
↳ num_particulas, dim, cantidad_iteraciones, c1, c2, w, limite_inf,
↳ limite_sup, input_a, input_b)

```

a: -30.0

b: 30.0

Iteración 1: Mejor posición global [-36.74289885 -34.00817634], Valor 61.53216249661317

Iteración 2: Mejor posición global [-36.74289885 -34.00817634], Valor 61.53216249661317

Iteración 3: Mejor posición global [-28.70518693 -33.67665592], Valor 15.194339625157106

Iteración 4: Mejor posición global [-31.11005373 -28.21404641], Valor 4.421849526425924

Iteración 5: Mejor posición global [-31.11005373 -28.21404641], Valor 4.421849526425924

Iteración 6: Mejor posición global [-31.11005373 -28.21404641], Valor 4.421849526425924

Iteración 7: Mejor posición global [-31.11005373 -28.21404641], Valor 4.421849526425924

Iteración 8: Mejor posición global [-31.11005373 -28.21404641], Valor 4.421849526425924

Iteración 9: Mejor posición global [-31.11005373 -28.21404641], Valor 4.421849526425924

Iteración 10: Mejor posición global [-31.11005373 -28.21404641], Valor 4.421849526425924

Solucion optima (x, y): [-31.11005373 -28.21404641]

Valor optimo: 4.421849526425924

```

[484]: fig, ax = plt.subplots(1,2,figsize=(12,6))

x1 = np.linspace(limite_inf, limite_sup, 100)
x2 = np.linspace(limite_inf, limite_sup, 100)
y = funcion_objetivo(input_a, input_b, x1, x2)

# Variable X
ax[0].plot(x1, y)
ax[0].scatter(solucion_optima[0], valor_optimo, s=50, c='green')
ax[0].axvline(x=solucion_optima[0], c='green', linewidth=1)
ax[0].axhline(y=valor_optimo, c='green', linewidth=1)
ax[0].set_title('Minimización de función con algoritmo PSO (Variable X)')
ax[0].set_xlabel('Variable independiente X')
ax[0].set_ylabel('Variable dependiente')

# Variable Y

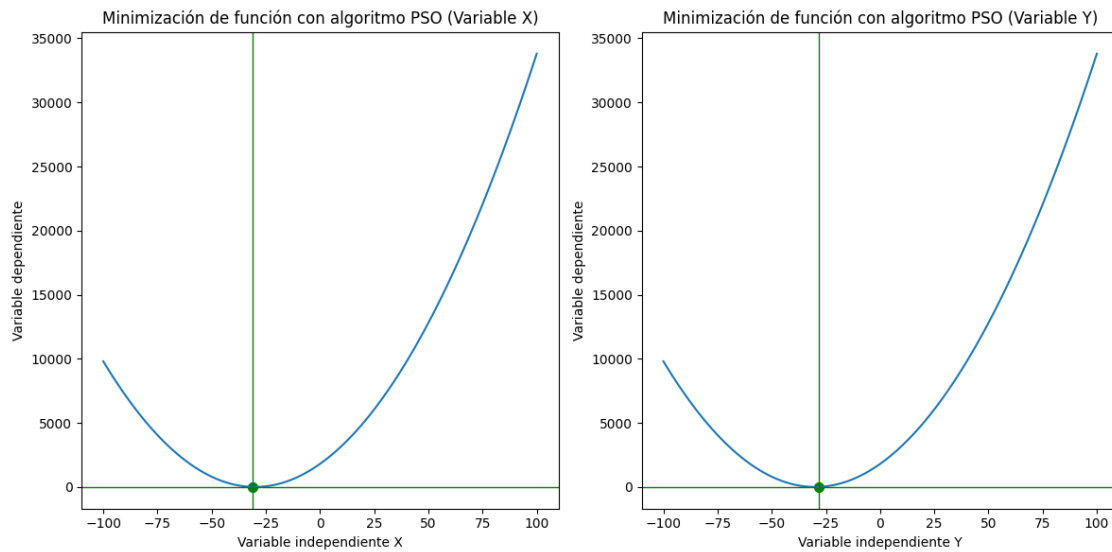
```

```

ax[1].plot(x2, y)
ax[1].scatter(solucion_optima[1], valor_optimo, s=50, c='green')
ax[1].axvline(x=solucion_optima[1], c='green', linewidth=1)
ax[1].axhline(y=valor_optimo, c='green', linewidth=1)
ax[1].set_title('Minimización de función con algoritmo PSO (Variable Y)')
ax[1].set_xlabel('Variable independiente Y')
ax[1].set_ylabel('Variable dependiente')

plt.tight_layout()
plt.show()

```

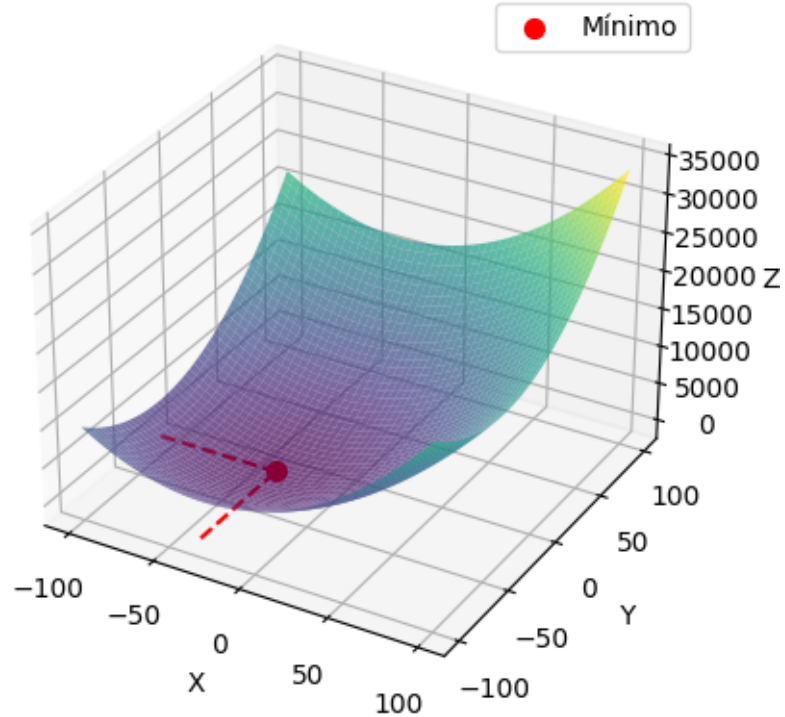


```

[485]: plot_3d(funcion_objetivo, limite_inf, limite_sup, solucion_optima[0],
↪solucion_optima[1])

```

Gráfico 3D con el Mínimo de la Función



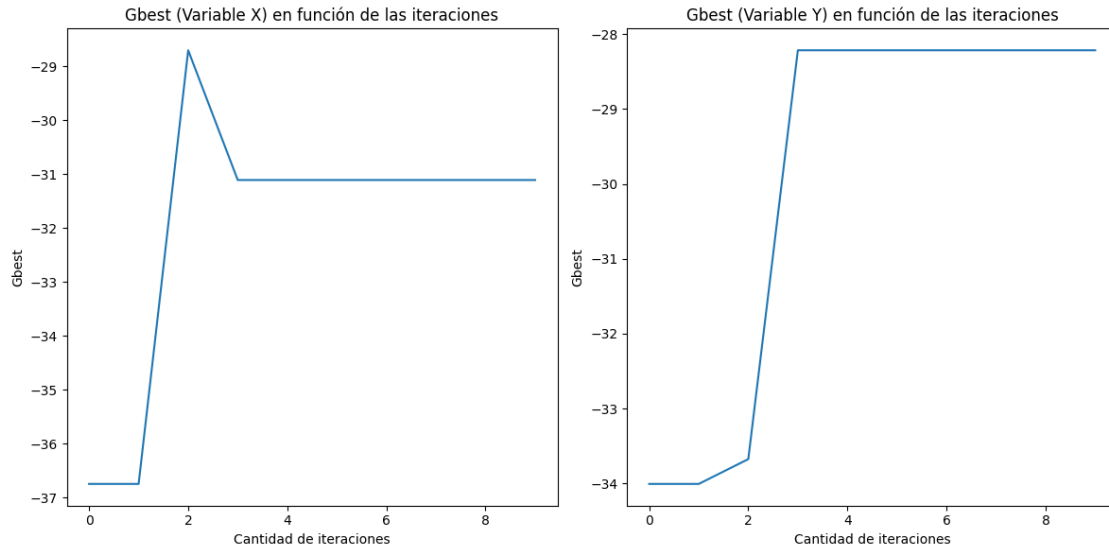
```
[486]: fig, ax = plt.subplots(1,2,figsize=(12,6))

x = np.arange(cantidad_iteraciones)
y1 = np.array(g_bests)[: , 0]
y2 = np.array(g_bests)[: , 1]

# Variable X
ax[0].plot(x, y1)
ax[0].set_title('Gbest (Variable X) en función de las iteraciones')
ax[0].set_xlabel('Cantidad de iteraciones')
ax[0].set_ylabel('Gbest')

# Variable Y
ax[1].plot(x, y2)
ax[1].set_title('Gbest (Variable Y) en función de las iteraciones')
ax[1].set_xlabel('Cantidad de iteraciones')
ax[1].set_ylabel('Gbest')

plt.tight_layout()
plt.show()
```



Con coeficiente de inercia $w = 0$

```
[487]: # parametros
num_particulas = 20 # numero de particulas
dim = 2 # dimensiones
cantidad_iteraciones = 10 # maximo numero de iteraciones
c1 = 2.0 # componente cognitivo
c2 = 2.0 # componente social
w = 0 # factor de inercia
limite_inf = -100 # limite inferior de busqueda
limite_sup = 100 # limite superior de busqueda

solucion_optima, valor_optimo, g_bests = solve_PSO(funcion_objetivo,
↪ num_particulas, dim, cantidad_iteraciones, c1, c2, w, limite_inf,
↪ limite_sup, input_a, input_b)
```

Iteración 1: Mejor posición global [-28.84073814 -31.50777128], Valor 3.617262272494938
 Iteración 2: Mejor posición global [-31.48036611 -29.10389015], Valor 2.994496702309124
 Iteración 3: Mejor posición global [-30.43929614 -29.85877656], Valor 0.21292515864086517
 Iteración 4: Mejor posición global [-30.38941036 -30.10924178], Valor 0.16357419431261852
 Iteración 5: Mejor posición global [-30.22536493 -30.10582213], Valor 0.061987674193137415
 Iteración 6: Mejor posición global [-29.97138855 -30.0066988], Valor 0.0008634891083914363
 Iteración 7: Mejor posición global [-29.99001655 -30.01390895], Valor 0.0002931282459302524

Iteración 8: Mejor posición global [-30.00863388 -30.00257329], Valor
8.116573912683705e-05
Iteración 9: Mejor posición global [-29.99621434 -29.99833131], Valor
1.7115724352406908e-05
Iteración 10: Mejor posición global [-29.99864761 -30.00022896], Valor
1.8813898597703166e-06

Solucion optima (x, y): [-29.99864761 -30.00022896]
Valor optimo: 1.8813898597703166e-06

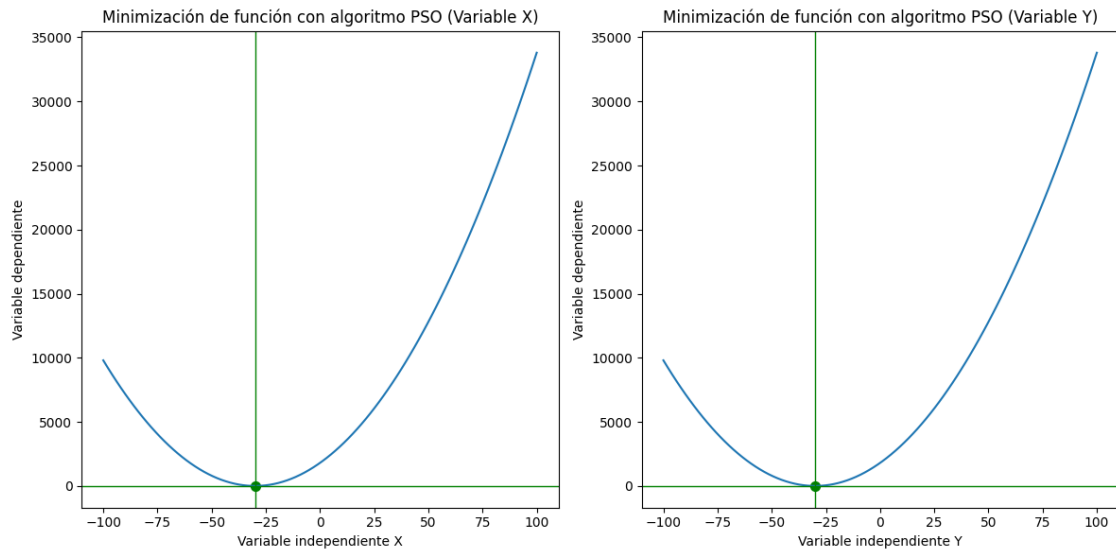
```
[488]: fig, ax = plt.subplots(1,2,figsize=(12,6))

x1 = np.linspace(limite_inf, limite_sup, 100)
x2 = np.linspace(limite_inf, limite_sup, 100)
y = funcion_objetivo(x1, x2, input_a, input_b)

# Variable X
ax[0].plot(x1, y)
ax[0].scatter(solucion_optima[0], valor_optimo, s=50, c='green')
ax[0].axvline(x=solucion_optima[0], c='green', linewidth=1)
ax[0].axhline(y=valor_optimo, c='green', linewidth=1)
ax[0].set_title('Minimización de función con algoritmo PSO (Variable X)')
ax[0].set_xlabel('Variable independiente X')
ax[0].set_ylabel('Variable dependiente')

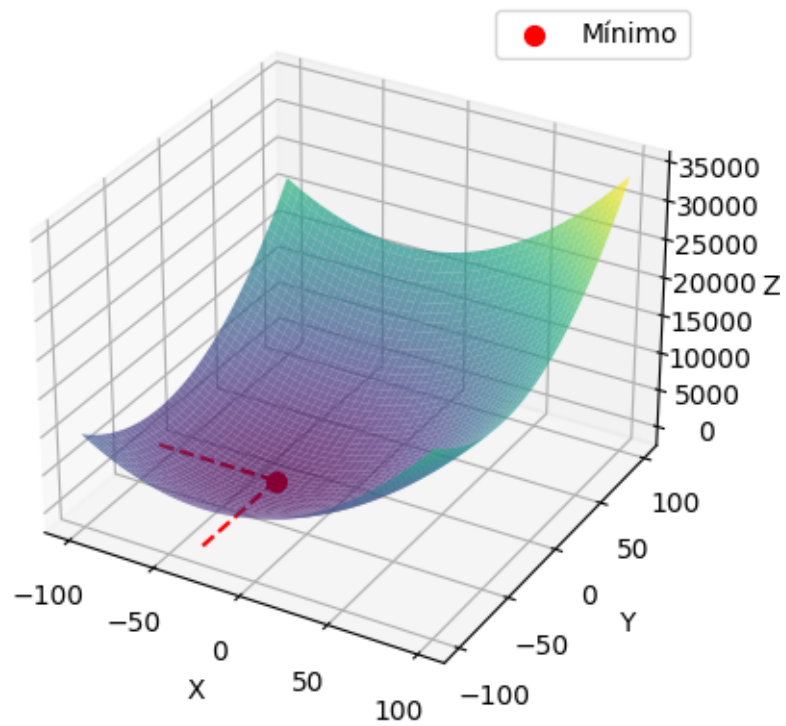
# Variable Y
ax[1].plot(x2, y)
ax[1].scatter(solucion_optima[1], valor_optimo, s=50, c='green')
ax[1].axvline(x=solucion_optima[1], c='green', linewidth=1)
ax[1].axhline(y=valor_optimo, c='green', linewidth=1)
ax[1].set_title('Minimización de función con algoritmo PSO (Variable Y)')
ax[1].set_xlabel('Variable independiente Y')
ax[1].set_ylabel('Variable dependiente')

plt.tight_layout()
plt.show()
```



```
[489]: plot_3d(funcion_objetivo, limite_inf, limite_sup, solucion_optima[0],
↪solucion_optima[1])
```

Gráfico 3D con el Mínimo de la Función



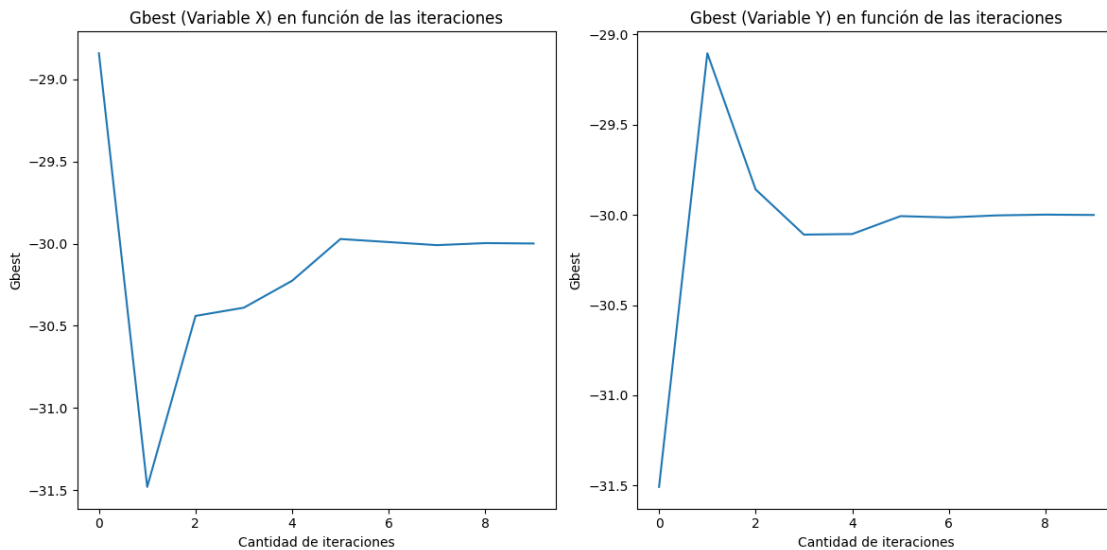

```
[490]: fig, ax = plt.subplots(1,2,figsize=(12,6))

x = np.arange(cantidad_iteraciones)
y1 = np.array(g_bests)[: , 0]
y2 = np.array(g_bests)[: , 1]

# Variable X
ax[0].plot(x, y1)
ax[0].set_title('Gbest (Variable X) en función de las iteraciones')
ax[0].set_xlabel('Cantidad de iteraciones')
ax[0].set_ylabel('Gbest')

# Variable Y
ax[1].plot(x, y2)
ax[1].set_title('Gbest (Variable Y) en función de las iteraciones')
ax[1].set_xlabel('Cantidad de iteraciones')
ax[1].set_ylabel('Gbest')

plt.tight_layout()
plt.show()
```



Los valores halados fueron muy similares, tanto con $w = 0.7$ como con $w = 0$. Teóricamente, el valor de coeficiente de inercia favorece una exploración más local cuanto menor sea su valor, mientras que favorecerá una exploración más global cuanto mayor sea.

Con PySwarm

```
[491]: from pyswarm import pso
```

```
# funcion objetivo hiperboloide eliptico
def funcion_objetivo(x):
    return (x[0] - input_a) ** 2 + (x[1] + input_b) ** 2
```

```
[492]: # parametros
num_particulas = 20 # numero de particulas
dim = 2 # dimensiones
cantidad_iteraciones = 10 # maximo numero de iteraciones
c1 = 2.0 # componente cognitivo
c2 = 2.0 # componente social
w = 0.7 # factor de inercia

lb = [-100, -100] # limite inf
ub = [100, 100] # limite sup

# Llamada a la función pso
solucion_optima, valor_optimo = pso(
    funcion_objetivo,
    lb,
    ub,
    swarmsize=num_particulas,
    maxiter=cantidad_iteraciones,
    omega=w,
    phip=c1,
    phig=c2,
    debug=False)

# Resultados
print("\nSolución óptima (x, y):", solucion_optima)
print("Valor óptimo:", valor_optimo)
```

Stopping search: maximum iterations reached --> 10

Solución óptima (x, y): [-30.77303178 -26.50588453]

Valor óptimo: 12.806421087006138

```
[493]: fig, ax = plt.subplots(1,2,figsize=(12,6))

x1 = np.linspace(limite_inf, limite_sup, 100)
x2 = np.linspace(limite_inf, limite_sup, 100)
y = [funcion_objetivo(x) for x in np.hstack((x1.reshape(-1,1), x2.
    ↪reshape(-1,1)))]

# Variable X
ax[0].plot(x1, y)
ax[0].scatter(solucion_optima[0], valor_optimo, s=50, c='green')
ax[0].axvline(x=solucion_optima[0], c='green', linewidth=1)
```

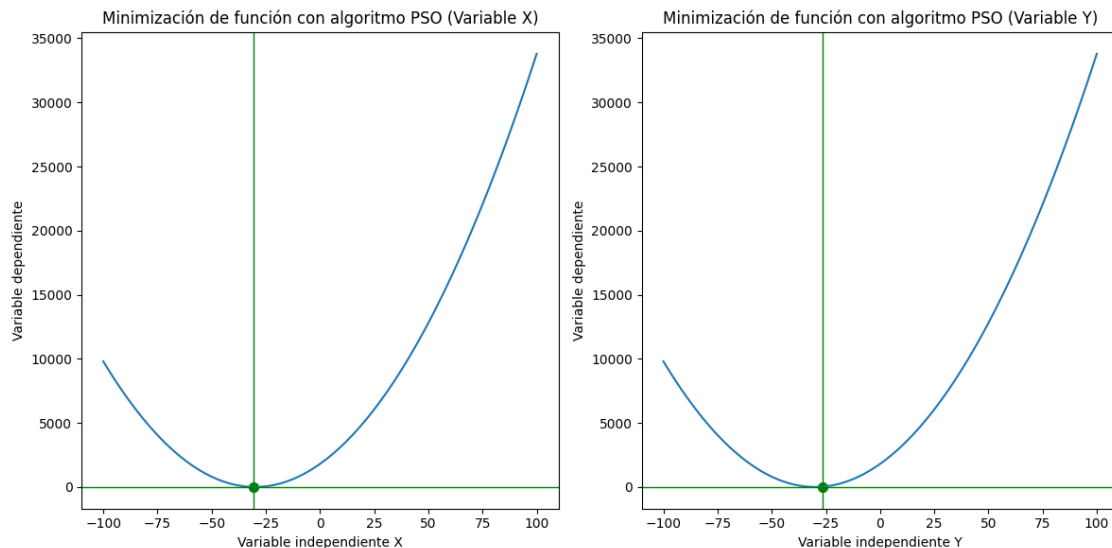
```

ax[0].axhline(y=valor_optimo, c='green', linewidth=1)
ax[0].set_title('Minimización de función con algoritmo PSO (Variable X)')
ax[0].set_xlabel('Variable independiente X')
ax[0].set_ylabel('Variable dependiente')

# Variable Y
ax[1].plot(x2, y)
ax[1].scatter(solucion_optima[1], valor_optimo, s=50, c='green')
ax[1].axvline(x=solucion_optima[1], c='green', linewidth=1)
ax[1].axhline(y=valor_optimo, c='green', linewidth=1)
ax[1].set_title('Minimización de función con algoritmo PSO (Variable Y)')
ax[1].set_xlabel('Variable independiente Y')
ax[1].set_ylabel('Variable dependiente')

plt.tight_layout()
plt.show()

```



El algoritmo PSO de PySwarm es mucho más eficiente. La velocidad de cómputo fue notoria. Además, tiene más hiperparámetros y configuraciones para ajustar. En términos de soluciones halladas, ambos dieron similar. De todas maneras, el ejercicio de optimización fue muy sencillo para que PSO de PySwarm se luzca más.

1.0.7 EJERCICIO 4

Mediante PSO es posible resolver en forma aproximada un sistema de n ecuaciones con n incógnitas clásico del tipo:

$$\begin{cases} f_1(x_1, x_2, \dots, x_n) = 0 \\ f_2(x_1, x_2, \dots, x_n) = 0 \\ \vdots \\ f_n(x_1, x_2, \dots, x_n) = 0 \end{cases}$$

Por ejemplo, el siguiente es un sistema de 2 ecuaciones con 2 incógnitas x_1 y x_2 que puede ser resuelto con PSO:

$$\begin{cases} 3x_1 + 2x_2 = 9 \\ x_1 - 5x_2 = 4 \end{cases}$$

Utilizando la biblioteca pyswarm: * Escribir un algoritmo PSO con parámetros a elección (c_1 , c_2 , w , número de partículas, máximo número de iteraciones) que encuentre x_1 y x_2 para el sistema de ecuaciones anterior (3). Transcribir el código fuente. * Transcribir los valores de x_1 y x_2 encontrados por el algoritmo. * Indicar la URL del repositorio en donde se encuentra el algoritmo PSO. * Realizar observaciones/comentarios/conclusiones sobre: * (i) ¿Cómo eligió los límites superior e inferior de x_1 y x_2 ? * (ii) ¿PSO puede resolver un sistema de n ecuaciones con n incógnitas no lineal?. Demostrar. * (iii) ¿Cómo logró resolver el ejercicio?. * (iv) ¿Los resultados obtenidos guardan relación directa con los valores de los parámetros elegidos?. Demostrar.

Lo que se va a hacer es utilizar la librería Optuna para realizar una búsqueda de hiperparámetros bayesiana. El objetivo será minimizar la función de suma de errores cuadráticos medios y quedarme con esos hiperparámetros.

```
[494]: import optuna
import contextlib # Esto es para evitar los prints excesivos
import os
```

```
[495]: # Función objetivo
def funcion_objetivo(x):
    x1, x2 = x
    eq1 = (3 * x1 + 2 * x2 - 9) ** 2 # error cuadrático de la primera ecuación
    eq2 = (x1 - 5 * x2 - 4) ** 2 # error cuadrático de la segunda ecuación
    return eq1 + eq2
```

```
[496]: # Definir la función que será optimizada por Optuna
def optimizacion(trial):
    # Sugerir valores para los hiperparámetros
    num_particulas = trial.suggest_int('num_particulas', 10, 100)
    cant_iteraciones = trial.suggest_int('cant_iteraciones', 50, 300)
    w = trial.suggest_uniform('w', 0.1, 0.9)
    c1 = trial.suggest_uniform('c1', 0.5, 3.0)
    c2 = trial.suggest_uniform('c2', 0.5, 3.0)
    lb = trial.suggest_uniform('lb', -100, 0)
    ub = trial.suggest_uniform('ub', 0, 100)
```

```

# Límites para x1 y x2
lb_ = [lb, lb]
ub_ = [ub, ub]

# Ejecutar PSO
with open(os.devnull, 'w') as fnull:
    with contextlib.redirect_stdout(fnull):
        best_position, best_value = pso(
            funcion_objetivo,
            lb=lb_,
            ub=ub_,
            swarmsize=num_particulas,
            maxiter=cant_iteraciones,
            omega=w,
            phip=c1,
            phig=c2
        )

# El objetivo de la optimización es minimizar el valor de la función
↪objetivo
return best_value

# Crear un estudio de Optuna y optimizar
study = optuna.create_study(direction='minimize')
study.optimize(optimizacion, n_trials=200)

# Imprimir los resultados
print("Mejor valor objetivo:", study.best_value)
print("Mejores hiperparámetros:", study.best_params)

```

```

[I 2024-08-10 12:41:21,133] A new study created in memory with name:
no-name-d252b2dd-16a2-4e86-845a-79acadd584a2
C:\Users\Leandro\AppData\Local\Temp\ipykernel_12904\1884802977.py:6:
FutureWarning: suggest_uniform has been deprecated in v3.0.0. This feature will
be removed in v6.0.0. See https://github.com/optuna/optuna/releases/tag/v3.0.0.
Use :func:`~optuna.trial.Trial.suggest_float` instead.
    w = trial.suggest_uniform('w', 0.1, 0.9)
C:\Users\Leandro\AppData\Local\Temp\ipykernel_12904\1884802977.py:7:
FutureWarning: suggest_uniform has been deprecated in v3.0.0. This feature will
be removed in v6.0.0. See https://github.com/optuna/optuna/releases/tag/v3.0.0.
Use :func:`~optuna.trial.Trial.suggest_float` instead.
    c1 = trial.suggest_uniform('c1', 0.5, 3.0)
C:\Users\Leandro\AppData\Local\Temp\ipykernel_12904\1884802977.py:8:
FutureWarning: suggest_uniform has been deprecated in v3.0.0. This feature will
be removed in v6.0.0. See https://github.com/optuna/optuna/releases/tag/v3.0.0.
Use :func:`~optuna.trial.Trial.suggest_float` instead.
    c2 = trial.suggest_uniform('c2', 0.5, 3.0)
C:\Users\Leandro\AppData\Local\Temp\ipykernel_12904\1884802977.py:9:

```

FutureWarning: suggest_uniform has been deprecated in v3.0.0. This feature will be removed in v6.0.0. See <https://github.com/optuna/optuna/releases/tag/v3.0.0>. Use :func:`~optuna.trial.Trial.suggest_float` instead.

```
lb = trial.suggest_uniform('lb', -100, 0)
```

C:\Users\Leandro\AppData\Local\Temp\ipykernel_12904\1884802977.py:10:

FutureWarning: suggest_uniform has been deprecated in v3.0.0. This feature will be removed in v6.0.0. See <https://github.com/optuna/optuna/releases/tag/v3.0.0>. Use :func:`~optuna.trial.Trial.suggest_float` instead.

```
ub = trial.suggest_uniform('ub', 0, 100)
```

[I 2024-08-10 12:41:21,371] Trial 0 finished with value:

2.9446718318952044e-09 and parameters: {'num_particulas': 40, 'cant_iteraciones': 257, 'w': 0.7677895706998955, 'c1': 1.9714555628486794, 'c2': 1.7912683595085797, 'lb': -62.286963711775854, 'ub': 48.56349240777909}.

Best is trial 0 with value: 2.9446718318952044e-09.

[I 2024-08-10 12:41:21,521] Trial 1 finished with value:

7.128234558451984e-09 and parameters: {'num_particulas': 75, 'cant_iteraciones': 138, 'w': 0.6723353958693874, 'c1': 1.9288632469136286, 'c2': 1.768739870484501, 'lb': -29.979119435366513, 'ub': 83.922162715186}. Best is trial 0 with value:

2.9446718318952044e-09.

[I 2024-08-10 12:41:21,591] Trial 2 finished with value:

5.6676419148740185e-06 and parameters: {'num_particulas': 52, 'cant_iteraciones': 102, 'w': 0.5356595282683859, 'c1': 2.032628387087232, 'c2': 2.123355216658025, 'lb': -57.88175700236244, 'ub': 44.81628294474939}. Best is trial 0 with value: 2.9446718318952044e-09.

[I 2024-08-10 12:41:21,668] Trial 3 finished with value:

9.509844194518956e-09 and parameters: {'num_particulas': 79, 'cant_iteraciones': 284, 'w': 0.46120489634168516, 'c1': 1.9730671698541369, 'c2': 1.2486611228849913, 'lb': -49.44851866929979, 'ub': 54.91994113283651}. Best is trial 0 with value: 2.9446718318952044e-09.

[I 2024-08-10 12:41:21,681] Trial 4 finished with value:

475.9080231252671 and parameters: {'num_particulas': 13, 'cant_iteraciones': 159, 'w': 0.10056387366512505, 'c1': 0.7787439971344765, 'c2': 0.6422336944139163, 'lb': -16.27173682322467, 'ub': 97.44107829082702}. Best is trial 0 with value: 2.9446718318952044e-09.

[I 2024-08-10 12:41:21,709] Trial 5 finished with value:

5.514483928500593 and parameters: {'num_particulas': 14, 'cant_iteraciones': 89, 'w': 0.8663097942693797, 'c1': 1.9260959719534538, 'c2': 2.7949944947395524, 'lb': -75.46851574136412, 'ub': 49.15493533944938}. Best is trial 0 with value: 2.9446718318952044e-09.

[I 2024-08-10 12:41:21,753] Trial 6 finished with value:

2.5540556608048445e-09 and parameters: {'num_particulas': 38, 'cant_iteraciones': 179, 'w': 0.2838014414845744, 'c1': 2.6105129468858257, 'c2': 2.173858452893798, 'lb': -81.11807961805582, 'ub': 18.963728400256596}.

Best is trial 6 with value: 2.5540556608048445e-09.

[I 2024-08-10 12:41:21,771] Trial 7 finished with value:

7.775065570807086e-06 and parameters: {'num_particulas': 10, 'cant_iteraciones': 113, 'w': 0.5377887797652231, 'c1': 1.559398551601324, 'c2': 2.2406979315201645, 'lb': -84.3318519381152, 'ub': 28.782206358006402}. Best is trial 6 with value:

2.5540556608048445e-09.

[I 2024-08-10 12:41:21,801] Trial 8 finished with value:
24.353583699135996 and parameters: {'num_particulas': 86, 'cant_iteraciones':
201, 'w': 0.2155772585322321, 'c1': 0.6818678883268896, 'c2':
1.1975725877925756, 'lb': -3.251333436196063, 'ub': 1.554385975364081}. Best is
trial 6 with value: 2.5540556608048445e-09.

[I 2024-08-10 12:41:21,816] Trial 9 finished with value:
5.107938695566155e-08 and parameters: {'num_particulas': 10, 'cant_iteraciones':
200, 'w': 0.449243509273096, 'c1': 1.2249039498416876, 'c2': 1.813316609753231,
'lb': -17.744138400461395, 'ub': 33.90492530415057}. Best is trial 6 with value:
2.5540556608048445e-09.

[I 2024-08-10 12:41:22,040] Trial 10 finished with value:
9.585853134861363e-06 and parameters: {'num_particulas': 38, 'cant_iteraciones':
230, 'w': 0.3043719955961133, 'c1': 2.8865948334605003, 'c2':
2.9652876035173605, 'lb': -99.38758216676284, 'ub': 6.1888984255049095}. Best is
trial 6 with value: 2.5540556608048445e-09.

[I 2024-08-10 12:41:22,321] Trial 11 finished with value:
0.17922704780991666 and parameters: {'num_particulas': 34, 'cant_iteraciones':
288, 'w': 0.8481185421709312, 'c1': 2.595605544634095, 'c2': 2.412622818403037,
'lb': -67.84403838961568, 'ub': 16.803447943832712}. Best is trial 6 with value:
2.5540556608048445e-09.

[I 2024-08-10 12:41:22,409] Trial 12 finished with value:
2.9918388825504465 and parameters: {'num_particulas': 39, 'cant_iteraciones':
52, 'w': 0.7240439241169064, 'c1': 2.408922058529891, 'c2': 2.5274277298745043,
'lb': -53.89196772941899, 'ub': 62.662013848265204}. Best is trial 6 with value:
2.5540556608048445e-09.

[I 2024-08-10 12:41:22,533] Trial 13 finished with value:
3.558559611731984e-09 and parameters: {'num_particulas': 59, 'cant_iteraciones':
238, 'w': 0.3409245251947941, 'c1': 2.413500441566259, 'c2': 2.0407430791808117,
'lb': -88.77397882666915, 'ub': 24.56665766561571}. Best is trial 6 with value:
2.5540556608048445e-09.

[I 2024-08-10 12:41:22,647] Trial 14 finished with value:
3.148263652114581e-08 and parameters: {'num_particulas': 27, 'cant_iteraciones':
251, 'w': 0.6286146431130314, 'c1': 2.9028220949871217, 'c2':
1.5167503872428143, 'lb': -70.3816563180023, 'ub': 15.657850941850395}. Best is
trial 6 with value: 2.5540556608048445e-09.

[I 2024-08-10 12:41:22,859] Trial 15 finished with value:
0.07090541153690938 and parameters: {'num_particulas': 55, 'cant_iteraciones':
183, 'w': 0.7616119461123292, 'c1': 2.274070845653765, 'c2': 2.575434124988038,
'lb': -39.25220491001488, 'ub': 36.25779427299237}. Best is trial 6 with value:
2.5540556608048445e-09.

[I 2024-08-10 12:41:23,046] Trial 16 finished with value:
6.900743390942774e-09 and parameters: {'num_particulas': 99, 'cant_iteraciones':
269, 'w': 0.6277261632964831, 'c1': 1.5627273726887858, 'c2':
1.9907760701099972, 'lb': -61.99023631941267, 'ub': 64.53737405358336}. Best is
trial 6 with value: 2.5540556608048445e-09.

[I 2024-08-10 12:41:23,286] Trial 17 finished with value:
0.00026501717817696454 and parameters: {'num_particulas': 46,

'cant_iteraciones': 226, 'w': 0.7813472762910416, 'c1': 2.650342647591332, 'c2': 2.307966114680278, 'lb': -80.29299416033916, 'ub': 16.664032324762122}. Best is trial 6 with value: 2.5540556608048445e-09.

[I 2024-08-10 12:41:23,396] Trial 18 finished with value: 2.7202936842630803 and parameters: {'num_particulas': 26, 'cant_iteraciones': 153, 'w': 0.8885529264186413, 'c1': 2.933646259752944, 'c2': 1.6058911897324002, 'lb': -71.27709543795088, 'ub': 41.41946123455746}. Best is trial 6 with value: 2.5540556608048445e-09.

[I 2024-08-10 12:41:23,490] Trial 19 finished with value: 3.309780603795406e-08 and parameters: {'num_particulas': 65, 'cant_iteraciones': 201, 'w': 0.36985502875802834, 'c1': 2.2419814413237464, 'c2': 1.9473482404849325, 'lb': -92.4235214885247, 'ub': 27.53491286842183}. Best is trial 6 with value: 2.5540556608048445e-09.

[I 2024-08-10 12:41:23,543] Trial 20 finished with value: 2.196315580570938e-06 and parameters: {'num_particulas': 28, 'cant_iteraciones': 262, 'w': 0.24565572222862475, 'c1': 2.6655330439894307, 'c2': 2.2357105610191828, 'lb': -77.50923507544731, 'ub': 40.63052294278974}. Best is trial 6 with value: 2.5540556608048445e-09.

[I 2024-08-10 12:41:23,642] Trial 21 finished with value: 1.2714586612688567e-09 and parameters: {'num_particulas': 62, 'cant_iteraciones': 233, 'w': 0.3785756497144365, 'c1': 2.3510465924681276, 'c2': 2.049706915983468, 'lb': -89.14281204870836, 'ub': 26.973988903733712}. Best is trial 21 with value: 1.2714586612688567e-09.

[I 2024-08-10 12:41:23,735] Trial 22 finished with value: 9.14400345637698e-09 and parameters: {'num_particulas': 66, 'cant_iteraciones': 216, 'w': 0.36802733378760066, 'c1': 2.179211545036525, 'c2': 1.9010752274142124, 'lb': -88.54904579612717, 'ub': 22.083223042791563}. Best is trial 21 with value: 1.2714586612688567e-09.

[I 2024-08-10 12:41:23,827] Trial 23 finished with value: 1.0768771464496828e-08 and parameters: {'num_particulas': 48, 'cant_iteraciones': 171, 'w': 0.41591931085598266, 'c1': 2.508406283813536, 'c2': 2.0951211929279694, 'lb': -98.22907897056933, 'ub': 9.795411552554537}. Best is trial 21 with value: 1.2714586612688567e-09.

[I 2024-08-10 12:41:23,915] Trial 24 finished with value: 1.1919665990288524e-05 and parameters: {'num_particulas': 45, 'cant_iteraciones': 251, 'w': 0.49240862623921994, 'c1': 2.175701346171879, 'c2': 2.4324167244637147, 'lb': -82.10213461236823, 'ub': 33.72527283340071}. Best is trial 21 with value: 1.2714586612688567e-09.

[I 2024-08-10 12:41:23,999] Trial 25 finished with value: 1.7470392039642104e-08 and parameters: {'num_particulas': 63, 'cant_iteraciones': 293, 'w': 0.2789518166435263, 'c1': 1.7572603000623888, 'c2': 2.6910990291858914, 'lb': -64.61877606295384, 'ub': 8.800864298198768}. Best is trial 21 with value: 1.2714586612688567e-09.

[I 2024-08-10 12:41:24,188] Trial 26 finished with value: 5.7189284567933124e-08 and parameters: {'num_particulas': 73, 'cant_iteraciones': 186, 'w': 0.525778625771774, 'c1': 2.724419509931015, 'c2': 2.234157696462918, 'lb': -75.33052503525097, 'ub': 28.483763357922555}. Best is trial 21 with value: 1.2714586612688567e-09.

[I 2024-08-10 12:41:24,248] Trial 27 finished with value: 2.06145132401094e-09 and parameters: {'num_particulas': 35, 'cant_iteraciones': 271, 'w': 0.21021501539644952, 'c1': 2.388856770086352, 'c2': 1.7335512517165412, 'lb': -91.81738243463138, 'ub': 18.606096709699205}. Best is trial 21 with value: 1.2714586612688567e-09.

[I 2024-08-10 12:41:24,308] Trial 28 finished with value: 7.107929178125922e-08 and parameters: {'num_particulas': 21, 'cant_iteraciones': 137, 'w': 0.2100433158350377, 'c1': 2.489062918172708, 'c2': 1.6391318264604542, 'lb': -92.0115325758835, 'ub': 19.43563560719488}. Best is trial 21 with value: 1.2714586612688567e-09.

[I 2024-08-10 12:41:24,366] Trial 29 finished with value: 6.64792406193698e-10 and parameters: {'num_particulas': 33, 'cant_iteraciones': 271, 'w': 0.14760133192407063, 'c1': 2.784318366337371, 'c2': 1.9082020623485838, 'lb': -84.66500795817052, 'ub': 12.731898291891298}. Best is trial 29 with value: 6.64792406193698e-10.

[I 2024-08-10 12:41:24,418] Trial 30 finished with value: 66.21245225058225 and parameters: {'num_particulas': 32, 'cant_iteraciones': 271, 'w': 0.1604102190266738, 'c1': 2.9677638540330595, 'c2': 1.836302118970465, 'lb': -95.55876118882284, 'ub': 0.5400209982420421}. Best is trial 29 with value: 6.64792406193698e-10.

[I 2024-08-10 12:41:24,476] Trial 31 finished with value: 2.302273676510042e-11 and parameters: {'num_particulas': 20, 'cant_iteraciones': 246, 'w': 0.16587757519176125, 'c1': 2.750358620585019, 'c2': 2.107959716939472, 'lb': -84.25770853426242, 'ub': 13.886667325606695}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:24,528] Trial 32 finished with value: 2.0636160528905436e-08 and parameters: {'num_particulas': 20, 'cant_iteraciones': 246, 'w': 0.14927127960773565, 'c1': 2.7661445839501084, 'c2': 1.995927796901934, 'lb': -86.99039177366129, 'ub': 10.969435725300357}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:24,576] Trial 33 finished with value: 1.9988388998406038e-08 and parameters: {'num_particulas': 23, 'cant_iteraciones': 279, 'w': 0.1005654146298927, 'c1': 2.7889883298581384, 'c2': 1.7376065022725962, 'lb': -91.0664932859917, 'ub': 11.673516863690631}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:24,640] Trial 34 finished with value: 30.727276670216735 and parameters: {'num_particulas': 51, 'cant_iteraciones': 297, 'w': 0.18655868496266959, 'c1': 2.3531765549172667, 'c2': 2.092035059409713, 'lb': -99.87931812583398, 'ub': 4.873596927718173}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:24,689] Trial 35 finished with value: 2.2268869187682024e-07 and parameters: {'num_particulas': 17, 'cant_iteraciones': 218, 'w': 0.22972978882963804, 'c1': 2.518035923708051, 'c2': 1.7535425875550479, 'lb': -83.14495188986473, 'ub': 23.182170963148288}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:24,750] Trial 36 finished with value: 1.974860496897849e-07 and parameters: {'num_particulas': 32, 'cant_iteraciones': 268, 'w': 0.15441752224633007, 'c1': 2.8257056495975377, 'c2':

2.3708339286719937, 'lb': -75.73274007707417, 'ub': 14.946551796523687}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:24,814] Trial 37 finished with value:

5.478114853042239e-10 and parameters: {'num_particulas': 43, 'cant_iteraciones': 237, 'w': 0.2623740469192416, 'c1': 2.9975112868490124, 'c2':

1.4537042224363288, 'lb': -92.99386490472014, 'ub': 6.196680779361962}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:24,879] Trial 38 finished with value:

3.56025001630702e-09 and parameters: {'num_particulas': 44, 'cant_iteraciones': 238, 'w': 0.2608059346449244, 'c1': 2.775046595723786, 'c2': 1.417416338826441, 'lb': -83.48882835103821, 'ub': 5.742827981536731}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:24,950] Trial 39 finished with value:

43.58732615499147 and parameters: {'num_particulas': 73, 'cant_iteraciones': 220, 'w': 0.2980251631378511, 'c1': 2.9836678723095176, 'c2': 2.147117250261863, 'lb': -86.19660579939224, 'ub': 1.026280345277776}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:25,007] Trial 40 finished with value:

6.155586429355212e-10 and parameters: {'num_particulas': 54, 'cant_iteraciones': 210, 'w': 0.10114817297387152, 'c1': 2.9876477092642317, 'c2':

1.2087412253785237, 'lb': -94.4457873113397, 'ub': 12.048107746271533}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:25,067] Trial 41 finished with value:

6.3041786521806446e-09 and parameters: {'num_particulas': 57, 'cant_iteraciones': 206, 'w': 0.1415618780855278, 'c1': 2.97428362809272, 'c2': 1.2176242790671665, 'lb': -94.89537988800284, 'ub': 10.122037787942363}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:25,116] Trial 42 finished with value:

1.5927792834168183e-08 and parameters: {'num_particulas': 41, 'cant_iteraciones': 239, 'w': 0.11029268590867006, 'c1': 2.8440927077497733, 'c2': 0.9879286871215377, 'lb': -95.46868503243, 'ub': 13.482140575135013}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:25,190] Trial 43 finished with value:

7.926139550209981e-10 and parameters: {'num_particulas': 60, 'cant_iteraciones': 255, 'w': 0.19330301769607328, 'c1': 2.6534316217811886, 'c2':

1.4422134145668932, 'lb': -87.13822206678626, 'ub': 7.612064410444531}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:25,251] Trial 44 finished with value:

1.030217380594795e-07 and parameters: {'num_particulas': 53, 'cant_iteraciones': 253, 'w': 0.18256076170037386, 'c1': 2.677059757562592, 'c2':

1.4349962206698892, 'lb': -79.50477481356991, 'ub': 4.889660486192803}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:25,314] Trial 45 finished with value:

89.22150583151341 and parameters: {'num_particulas': 70, 'cant_iteraciones': 281, 'w': 0.1240822327577487, 'c1': 2.591702300659422, 'c2': 1.3345389541738741, 'lb': -85.41956113147599, 'ub': 0.1254867896360734}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:25,375] Trial 46 finished with value:

1.9468877118999938e-07 and parameters: {'num_particulas': 41, 'cant_iteraciones': 260, 'w': 0.19260466270088636, 'c1': 2.855270430836956, 'c2': 1.0238699877352297, 'lb': -73.07448352912127, 'ub': 7.615968355810281}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:25,456] Trial 47 finished with value: 2.8096680613041274e-08 and parameters: {'num_particulas': 81, 'cant_iteraciones': 206, 'w': 0.24803441411665952, 'c1': 2.744679823722206, 'c2': 1.5831861447060345, 'lb': -78.38004443721775, 'ub': 21.423449953970866}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:25,504] Trial 48 finished with value: 4.704151876260349e-10 and parameters: {'num_particulas': 14, 'cant_iteraciones': 247, 'w': 0.16406897684625182, 'c1': 2.9936937409118025, 'c2': 1.3068080015778132, 'lb': -96.41890437992579, 'ub': 14.050869456308455}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:25,555] Trial 49 finished with value: 5.57119535069653e-08 and parameters: {'num_particulas': 16, 'cant_iteraciones': 192, 'w': 0.12634405426643286, 'c1': 2.858937465050038, 'c2': 1.1243844929346856, 'lb': -96.2403983428913, 'ub': 15.0101056544767}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:25,614] Trial 50 finished with value: 3.459227269283965e-08 and parameters: {'num_particulas': 12, 'cant_iteraciones': 225, 'w': 0.1637855449372429, 'c1': 2.964491288683413, 'c2': 1.31902983886181, 'lb': -99.04077677341424, 'ub': 13.209449280086147}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:25,671] Trial 51 finished with value: 1.1742334464341483e-09 and parameters: {'num_particulas': 29, 'cant_iteraciones': 243, 'w': 0.1026319836314483, 'c1': 2.8817248322336058, 'c2': 1.5418273176961244, 'lb': -86.86733691280591, 'ub': 4.230487177490452}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:25,746] Trial 52 finished with value: 4.974090566447848e-09 and parameters: {'num_particulas': 49, 'cant_iteraciones': 260, 'w': 0.18189036295222705, 'c1': 2.694463958239729, 'c2': 1.472905637887838, 'lb': -93.65650516631408, 'ub': 19.203400382404055}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:25,800] Trial 53 finished with value: 3.944653361756996e-09 and parameters: {'num_particulas': 23, 'cant_iteraciones': 283, 'w': 0.22193047600820776, 'c1': 2.560439289085086, 'c2': 1.6625942245822571, 'lb': -90.00506626249145, 'ub': 8.0589430115959}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:25,891] Trial 54 finished with value: 9.563511247449461e-08 and parameters: {'num_particulas': 58, 'cant_iteraciones': 231, 'w': 0.13822454014692967, 'c1': 2.989053706573694, 'c2': 1.3107802106642983, 'lb': -81.0086729734695, 'ub': 12.92863273433284}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:25,978] Trial 55 finished with value: 2.1859929937500343e-08 and parameters: {'num_particulas': 36, 'cant_iteraciones': 246, 'w': 0.2619658979638248, 'c1': 2.674635144292817, 'c2': 1.8416449022741117, 'lb': -99.83177738131721, 'ub': 23.22195709596125}. Best is

trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:26,035] Trial 56 finished with value: 3.855952894172911e-08 and parameters: {'num_particulas': 19, 'cant_iteraciones': 210, 'w': 0.31323913692103944, 'c1': 2.879493869595179, 'c2': 1.519011723632114, 'lb': -85.71811826549671, 'ub': 17.472368860568782}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:26,088] Trial 57 finished with value: 125.1339024137018 and parameters: {'num_particulas': 14, 'cant_iteraciones': 257, 'w': 0.16984173560337523, 'c1': 2.6057256973107714, 'c2': 1.9069678372152978, 'lb': -93.11522366254266, 'ub': 6.66118895494059}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:26,149] Trial 58 finished with value: 7.518971850301118e-09 and parameters: {'num_particulas': 25, 'cant_iteraciones': 80, 'w': 0.20049963058132886, 'c1': 2.7704183833019123, 'c2': 1.4119787349972954, 'lb': -88.95857998947278, 'ub': 11.156440958976894}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:26,223] Trial 59 finished with value: 6.899499255094869e-08 and parameters: {'num_particulas': 53, 'cant_iteraciones': 193, 'w': 0.23716610963321946, 'c1': 2.9235367044990963, 'c2': 1.1462494169418853, 'lb': -82.41445679844014, 'ub': 3.669814031687439}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:26,278] Trial 60 finished with value: 1.2335840186497126e-08 and parameters: {'num_particulas': 10, 'cant_iteraciones': 232, 'w': 0.13060166961689001, 'c1': 2.791355070628572, 'c2': 1.6702262226214877, 'lb': -69.7119541388219, 'ub': 26.3560478739339}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:26,344] Trial 61 finished with value: 4.2671554933639537e-10 and parameters: {'num_particulas': 31, 'cant_iteraciones': 240, 'w': 0.10713638655483075, 'c1': 2.8868921875412425, 'c2': 1.3812361766383763, 'lb': -87.24010079562855, 'ub': 3.727552823045907}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:26,392] Trial 62 finished with value: 1.3316642019262745 and parameters: {'num_particulas': 29, 'cant_iteraciones': 276, 'w': 0.16596371389326184, 'c1': 2.9053437341939294, 'c2': 1.2724469276307666, 'lb': -95.72943093745558, 'ub': 2.752096978790546}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:26,447] Trial 63 finished with value: 3.0242099270199595e-08 and parameters: {'num_particulas': 38, 'cant_iteraciones': 266, 'w': 0.13639626608133046, 'c1': 2.995648126774812, 'c2': 1.3959329737483415, 'lb': -91.33442716057229, 'ub': 6.871588947944934}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:26,501] Trial 64 finished with value: 2.7465861053408866e-09 and parameters: {'num_particulas': 32, 'cant_iteraciones': 250, 'w': 0.21558402184598374, 'c1': 2.6987697608207335, 'c2': 1.5144338627619343, 'lb': -78.35037681004896, 'ub': 8.501270526116977}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:26,573] Trial 65 finished with value: 1.988926786286553e-09 and parameters: {'num_particulas': 68, 'cant_iteraciones':

214, 'w': 0.10836602303059793, 'c1': 2.8282161019969356, 'c2': 1.221202986505966, 'lb': -89.70367428037609, 'ub': 15.705299179984955}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:26,644] Trial 66 finished with value: 2.7995393062275395e-09 and parameters: {'num_particulas': 60, 'cant_iteraciones': 289, 'w': 0.19112328667636835, 'c1': 2.6485347273145625, 'c2': 1.3491523138717691, 'lb': -85.5786171265144, 'ub': 20.74575809182337}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:26,709] Trial 67 finished with value: 2.203035188891846e-08 and parameters: {'num_particulas': 44, 'cant_iteraciones': 221, 'w': 0.15709414405125313, 'c1': 2.914877867671109, 'c2': 1.570799108928934, 'lb': -96.8975852911661, 'ub': 11.374683616077583}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:26,764] Trial 68 finished with value: 10.76411386370005 and parameters: {'num_particulas': 48, 'cant_iteraciones': 237, 'w': 0.22528071310406178, 'c1': 2.5440069512135324, 'c2': 1.466412292701681, 'lb': -73.7911487183595, 'ub': 2.078351146738717}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:26,838] Trial 69 finished with value: 3.9123091659562154e-09 and parameters: {'num_particulas': 94, 'cant_iteraciones': 255, 'w': 0.12486727868187511, 'c1': 2.4794016843521116, 'c2': 1.6807759459874039, 'lb': -83.17400991164702, 'ub': 17.164297445191472}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:26,945] Trial 70 finished with value: 93.10882439669622 and parameters: {'num_particulas': 24, 'cant_iteraciones': 164, 'w': 0.20245284691707333, 'c1': 2.7607321667747757, 'c2': 1.78754870603599, 'lb': -93.62774996105279, 'ub': 0.0706140655078098}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:26,991] Trial 71 finished with value: 2.7169133291287444e-07 and parameters: {'num_particulas': 30, 'cant_iteraciones': 244, 'w': 0.10203244302004708, 'c1': 2.8701694484203135, 'c2': 1.5666313606163915, 'lb': -85.25504286148852, 'ub': 4.544367797379325}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:27,037] Trial 72 finished with value: 3.338960313506873e-08 and parameters: {'num_particulas': 17, 'cant_iteraciones': 246, 'w': 0.10043304256641629, 'c1': 2.9061388942928135, 'c2': 1.3619669550483333, 'lb': -88.09318326292428, 'ub': 9.252915357406408}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:27,092] Trial 73 finished with value: 3.013988414716244e-08 and parameters: {'num_particulas': 27, 'cant_iteraciones': 224, 'w': 0.14316050317311899, 'c1': 2.99874731625562, 'c2': 1.270812300027464, 'lb': -88.22578245769868, 'ub': 3.895536476102017}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:27,153] Trial 74 finished with value: 1.1886790784843517e-09 and parameters: {'num_particulas': 21, 'cant_iteraciones': 242, 'w': 0.17125089005952593, 'c1': 2.8344672013461687, 'c2': 1.497103246710028, 'lb': -81.31102462649667, 'ub': 13.413089830826806}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:27,208] Trial 75 finished with value: 2.429257311853038e-08 and parameters: {'num_particulas': 35, 'cant_iteraciones': 274, 'w': 0.1229217159981541, 'c1': 2.7135907080433768, 'c2': 1.598517308969916, 'lb': -92.34389723464975, 'ub': 6.188414457185258}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:27,277] Trial 76 finished with value: 1.7086145110007764e-09 and parameters: {'num_particulas': 55, 'cant_iteraciones': 263, 'w': 0.1535024394870021, 'c1': 2.615922738884756, 'c2': 1.4179747016952917, 'lb': -76.66173572643862, 'ub': 9.920613306724679}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:27,332] Trial 77 finished with value: 1.608244887899857e-06 and parameters: {'num_particulas': 30, 'cant_iteraciones': 237, 'w': 0.19599376897366513, 'c1': 2.807473371717394, 'c2': 1.948866178285728, 'lb': -97.40316022057122, 'ub': 15.93422539916385}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:27,383] Trial 78 finished with value: 0.40102753717558753 and parameters: {'num_particulas': 41, 'cant_iteraciones': 230, 'w': 0.17192951206883084, 'c1': 2.911011748568071, 'c2': 1.1380849574192604, 'lb': -86.99575952670486, 'ub': 2.917044173580827}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:27,429] Trial 79 finished with value: 3.742985065848874e-10 and parameters: {'num_particulas': 19, 'cant_iteraciones': 254, 'w': 0.12032842968044388, 'c1': 2.7514650231233455, 'c2': 1.7027347960534422, 'lb': -92.97349389170124, 'ub': 19.304734076861372}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:27,481] Trial 80 finished with value: 1.5142644081630206e-09 and parameters: {'num_particulas': 19, 'cant_iteraciones': 252, 'w': 0.1448703423803207, 'c1': 2.6299221974722684, 'c2': 2.0342466693198866, 'lb': -90.67055736666336, 'ub': 20.47214331700109}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:27,532] Trial 81 finished with value: 1.1566318456242509e-07 and parameters: {'num_particulas': 22, 'cant_iteraciones': 266, 'w': 0.10087520044228218, 'c1': 2.7319356253168827, 'c2': 1.7520231548335357, 'lb': -94.19975835031437, 'ub': 13.326183028764476}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:27,583] Trial 82 finished with value: 3.4377602473811412e-09 and parameters: {'num_particulas': 26, 'cant_iteraciones': 300, 'w': 0.12390547582595429, 'c1': 2.922182600463783, 'c2': 1.8280490300875536, 'lb': -83.61596434837189, 'ub': 7.8396432613331255}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:27,642] Trial 83 finished with value: 4.4885300449617745e-08 and parameters: {'num_particulas': 14, 'cant_iteraciones': 250, 'w': 0.17817438941122316, 'c1': 2.8288707484671582, 'c2': 1.7191805854957423, 'lb': -96.8604963269186, 'ub': 10.917223649899883}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:27,704] Trial 84 finished with value: 3.840420895487899e-08 and parameters: {'num_particulas': 38, 'cant_iteraciones': 142, 'w': 0.1448783832697357, 'c1': 2.7512081515194815, 'c2':

1.6219313895635612, 'lb': -80.50995415418531, 'ub': 18.22130698949089}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:27,765] Trial 85 finished with value: 1.7120121627160445e-08 and parameters: {'num_particulas': 33, 'cant_iteraciones': 259, 'w': 0.12494730246019506, 'c1': 2.949479483975494, 'c2': 1.5305080076035562, 'lb': -91.53035988747203, 'ub': 5.3574886346699255}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:27,821] Trial 86 finished with value: 4.633175863671189e-09 and parameters: {'num_particulas': 12, 'cant_iteraciones': 229, 'w': 0.2077472882137041, 'c1': 2.863070997802438, 'c2': 1.4560242389388365, 'lb': -99.8483414157707, 'ub': 14.888721518895775}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:27,898] Trial 87 finished with value: 1.880503024016244e-08 and parameters: {'num_particulas': 61, 'cant_iteraciones': 242, 'w': 0.16170117798984734, 'c1': 2.7922644465801705, 'c2': 1.3679449374044674, 'lb': -88.74014392339363, 'ub': 24.356496951165322}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:27,959] Trial 88 finished with value: 1.2327384575972767e-09 and parameters: {'num_particulas': 16, 'cant_iteraciones': 271, 'w': 0.23986583957872315, 'c1': 2.4491262451412057, 'c2': 1.8835272822148288, 'lb': -94.5842574255983, 'ub': 9.176527212045055}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:28,013] Trial 89 finished with value: 3.885055540357473e-08 and parameters: {'num_particulas': 19, 'cant_iteraciones': 234, 'w': 0.11442443249652653, 'c1': 2.548286505809242, 'c2': 2.1640406549348916, 'lb': -87.38894743547282, 'ub': 30.634370660806635}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:28,082] Trial 90 finished with value: 1.5403662141954335 and parameters: {'num_particulas': 65, 'cant_iteraciones': 286, 'w': 0.17925485659447504, 'c1': 2.6664311493755153, 'c2': 1.2863974121716693, 'lb': -84.19163779711722, 'ub': 2.7244936286108086}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:28,143] Trial 91 finished with value: 1.8600199280498758e-07 and parameters: {'num_particulas': 22, 'cant_iteraciones': 243, 'w': 0.14601902164004027, 'c1': 2.85506316218132, 'c2': 1.625722295144138, 'lb': -80.63051135419474, 'ub': 14.07981362399421}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:28,195] Trial 92 finished with value: 7.793469831031403e-09 and parameters: {'num_particulas': 25, 'cant_iteraciones': 255, 'w': 0.1720922998548678, 'c1': 2.943375533571824, 'c2': 1.4543903555244257, 'lb': -90.27425964414087, 'ub': 12.24141438488543}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:28,249] Trial 93 finished with value: 1.1745432419551877e-08 and parameters: {'num_particulas': 29, 'cant_iteraciones': 248, 'w': 0.12101989645464828, 'c1': 2.9975435143841285, 'c2': 1.5303258545945848, 'lb': -81.60299315127914, 'ub': 18.246295632577283}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:28,304] Trial 94 finished with value:

2.1835267726467113e-09 and parameters: {'num_particulas': 18, 'cant_iteraciones': 215, 'w': 0.21105105725001116, 'c1': 2.807035271813597, 'c2': 1.3709871813978705, 'lb': -78.522393560335, 'ub': 7.4523126218820055}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:28,357] Trial 95 finished with value: 3.0230108647619485e-07 and parameters: {'num_particulas': 21, 'cant_iteraciones': 241, 'w': 0.18964253478994228, 'c1': 2.7215753987841316, 'c2': 1.6999313170118577, 'lb': -97.5546292173245, 'ub': 12.347557738125147}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:28,408] Trial 96 finished with value: 4.753877378174655e-10 and parameters: {'num_particulas': 15, 'cant_iteraciones': 262, 'w': 0.15630724403445212, 'c1': 2.863068636465827, 'c2': 1.220122699631171, 'lb': -92.8174313107829, 'ub': 6.081270805933996}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:28,456] Trial 97 finished with value: 5.1708228462127434e-08 and parameters: {'num_particulas': 14, 'cant_iteraciones': 278, 'w': 0.1496708706365113, 'c1': 2.894343488811217, 'c2': 1.2265880340617281, 'lb': -93.01066829536005, 'ub': 4.526687718728562}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:28,516] Trial 98 finished with value: 621.6219768205509 and parameters: {'num_particulas': 11, 'cant_iteraciones': 264, 'w': 0.13370194078629719, 'c1': 2.9396578636472226, 'c2': 1.1728959516971593, 'lb': -86.47478076741126, 'ub': 1.600448303390908}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:28,562] Trial 99 finished with value: 4.604689597275354e-08 and parameters: {'num_particulas': 15, 'cant_iteraciones': 257, 'w': 0.11246772377774017, 'c1': 2.752902880785847, 'c2': 1.3065403125896213, 'lb': -95.2446684502558, 'ub': 10.114115952422228}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:28,627] Trial 100 finished with value: 1.1879766190152672e-09 and parameters: {'num_particulas': 46, 'cant_iteraciones': 224, 'w': 0.16101298432644004, 'c1': 2.647302911159867, 'c2': 1.2275904118320193, 'lb': -90.0400495953689, 'ub': 6.609092441123025}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:28,689] Trial 101 finished with value: 2.650171030268242e-08 and parameters: {'num_particulas': 46, 'cant_iteraciones': 225, 'w': 0.15682951302171158, 'c1': 2.585373309334973, 'c2': 1.1059560363327352, 'lb': -91.85333302603483, 'ub': 6.396284422316819}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:28,758] Trial 102 finished with value: 6.011843989714853e-09 and parameters: {'num_particulas': 52, 'cant_iteraciones': 235, 'w': 0.2293011253299061, 'c1': 2.8819383542655745, 'c2': 1.2327884825636524, 'lb': -89.4520354206096, 'ub': 8.627616950717773}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:28,834] Trial 103 finished with value: 1.9942950739311355e-09 and parameters: {'num_particulas': 63, 'cant_iteraciones': 206, 'w': 0.10038940289321714, 'c1': 2.6998559471350365, 'c2': 1.329424606799164, 'lb': -93.72918138237479, 'ub': 4.979819360219803}.

Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:28,888] Trial 104 finished with value: 19.65516668963098 and parameters: {'num_particulas': 43, 'cant_iteraciones': 220, 'w': 0.13209814907746317, 'c1': 2.6448393890441584, 'c2': 1.1908677575160969, 'lb': -97.6831221761665, 'ub': 1.713254747600815}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:28,966] Trial 105 finished with value: 1.4990025832831608e-08 and parameters: {'num_particulas': 55, 'cant_iteraciones': 262, 'w': 0.19440620712515516, 'c1': 2.794128160758757, 'c2': 1.4180413397714746, 'lb': -85.0482058688507, 'ub': 16.39547457883321}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:29,029] Trial 106 finished with value: 5.73941540424916e-09 and parameters: {'num_particulas': 36, 'cant_iteraciones': 196, 'w': 0.11568313103088887, 'c1': 2.950035802250003, 'c2': 1.0433229060501372, 'lb': -90.7092812270112, 'ub': 11.491908646389835}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:29,104] Trial 107 finished with value: 4.2915279987733074e-10 and parameters: {'num_particulas': 47, 'cant_iteraciones': 271, 'w': 0.16232174546148098, 'c1': 2.8626114322562985, 'c2': 1.2523277574670955, 'lb': -88.09009522387448, 'ub': 6.563838399394197}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:29,186] Trial 108 finished with value: 1.0662291995225863e-08 and parameters: {'num_particulas': 57, 'cant_iteraciones': 274, 'w': 0.13672635337213998, 'c1': 2.9997709669085073, 'c2': 1.382969859256751, 'lb': -95.17486551767108, 'ub': 3.7226273988591956}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:29,268] Trial 109 finished with value: 3.4617483314127064e-09 and parameters: {'num_particulas': 50, 'cant_iteraciones': 271, 'w': 0.1755919979372248, 'c1': 2.8446081016960147, 'c2': 1.2804416263164176, 'lb': -86.68311995182897, 'ub': 14.664038216422751}. Best is trial 31 with value: 2.302273676510042e-11.

[I 2024-08-10 12:41:29,341] Trial 110 finished with value: 1.0430942124367217e-11 and parameters: {'num_particulas': 39, 'cant_iteraciones': 289, 'w': 0.21841625601899686, 'c1': 2.9018986677091667, 'c2': 1.4722160130770745, 'lb': -83.86920697758734, 'ub': 9.596565052531925}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:29,400] Trial 111 finished with value: 3.0798937406910725e-07 and parameters: {'num_particulas': 43, 'cant_iteraciones': 281, 'w': 0.21948573052747303, 'c1': 2.8820069007922724, 'c2': 1.4781881736791471, 'lb': -83.08104244168506, 'ub': 9.80044210084819}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:29,466] Trial 112 finished with value: 4.3616037868255226e-08 and parameters: {'num_particulas': 40, 'cant_iteraciones': 294, 'w': 0.25256319608483757, 'c1': 2.945745539805187, 'c2': 1.4377077809295977, 'lb': -87.489428049958, 'ub': 7.626156275232079}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:29,524] Trial 113 finished with value: 1.2880119188855995e-08 and parameters: {'num_particulas': 34,

'cant_iteraciones': 286, 'w': 0.18404106006150522, 'c1': 2.8128451334362046, 'c2': 1.5760971304899625, 'lb': -84.20838274372642, 'ub': 11.918121383044443}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:29,580] Trial 114 finished with value: 48.76752612629462 and parameters: {'num_particulas': 37, 'cant_iteraciones': 291, 'w': 0.27147979781923454, 'c1': 2.7575510718146483, 'c2': 1.3339990863064375, 'lb': -92.514232833281, 'ub': 0.9054926362888027}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:29,639] Trial 115 finished with value: 3.711234755180549e-09 and parameters: {'num_particulas': 31, 'cant_iteraciones': 267, 'w': 0.20800118156374287, 'c1': 2.8879813728427517, 'c2': 1.6519040389345692, 'lb': -98.06820672903937, 'ub': 20.243165034047937}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:29,706] Trial 116 finished with value: 8.06582921627312e-10 and parameters: {'num_particulas': 48, 'cant_iteraciones': 254, 'w': 0.15478548101429182, 'c1': 2.708669641064806, 'c2': 1.5525414335490064, 'lb': -76.98217585826296, 'ub': 5.8907217636482265}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:29,774] Trial 117 finished with value: 1.9686587737696313e-08 and parameters: {'num_particulas': 42, 'cant_iteraciones': 251, 'w': 0.22997252325560485, 'c1': 2.712149753037717, 'c2': 1.4831925891102617, 'lb': -78.39497048304833, 'ub': 16.857443673552943}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:29,833] Trial 118 finished with value: 1.2147175897990072e-07 and parameters: {'num_particulas': 39, 'cant_iteraciones': 259, 'w': 0.15418774307456268, 'c1': 2.8121919272909635, 'c2': 1.2734645180920532, 'lb': -75.6882315954797, 'ub': 8.890100067822427}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:29,899] Trial 119 finished with value: 1.5864928664192085e-08 and parameters: {'num_particulas': 48, 'cant_iteraciones': 276, 'w': 0.19378971185280586, 'c1': 2.9471613286847806, 'c2': 1.7701089681899402, 'lb': -88.20929821244574, 'ub': 13.212177217972343}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:29,973] Trial 120 finished with value: 4.4951256622943564e-08 and parameters: {'num_particulas': 50, 'cant_iteraciones': 254, 'w': 0.24077736450096282, 'c1': 2.7621559015675077, 'c2': 1.399572380834648, 'lb': -82.1326140339001, 'ub': 6.61749544328734}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:30,031] Trial 121 finished with value: 0.8604116189357753 and parameters: {'num_particulas': 28, 'cant_iteraciones': 174, 'w': 0.13472249442048223, 'c1': 2.8512273013160665, 'c2': 1.5490340315825424, 'lb': -84.98107496524807, 'ub': 3.4114818458648317}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:30,106] Trial 122 finished with value: 5.302270479311652e-09 and parameters: {'num_particulas': 45, 'cant_iteraciones': 249, 'w': 0.11615403712805085, 'c1': 2.905604271646518, 'c2': 1.6131904233455356, 'lb': -88.82053105711108, 'ub': 5.336840810093127}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:30,177] Trial 123 finished with value:
1.631373133088351e-08 and parameters: {'num_particulas': 53, 'cant_iteraciones':
245, 'w': 0.15794764052222948, 'c1': 2.994715309446736, 'c2':
1.4991369779754773, 'lb': -92.52002314619716, 'ub': 9.847286879427715}. Best is
trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:30,236] Trial 124 finished with value:
6.223948677244584e-11 and parameters: {'num_particulas': 34, 'cant_iteraciones':
268, 'w': 0.14612818891702575, 'c1': 2.6832774505881845, 'c2':
1.5524387888623992, 'lb': -79.5077143303895, 'ub': 10.846279124082528}. Best is
trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:30,306] Trial 125 finished with value:
1.8450922647529838e-09 and parameters: {'num_particulas': 47,
'cant_iteraciones': 266, 'w': 0.1704872970899713, 'c1': 2.5771522837797725,
'c2': 1.3569118265504938, 'lb': -80.30331212968515, 'ub': 11.135647259056427}.
Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:30,361] Trial 126 finished with value:
1.2942106767453425e-07 and parameters: {'num_particulas': 34,
'cant_iteraciones': 283, 'w': 0.14245722868080196, 'c1': 2.516101761229164,
'c2': 1.697771960287033, 'lb': -76.95765910524361, 'ub': 22.023105981865815}.
Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:30,422] Trial 127 finished with value:
2.048280899381541e-09 and parameters: {'num_particulas': 40, 'cant_iteraciones':
258, 'w': 0.1885233828518844, 'c1': 2.6821765071570978, 'c2': 1.584899301856688,
'lb': -96.3494819954218, 'ub': 14.86631940442053}. Best is trial 110 with value:
1.0430942124367217e-11.

[I 2024-08-10 12:41:30,537] Trial 128 finished with value:
5.694469303057876e-08 and parameters: {'num_particulas': 59, 'cant_iteraciones':
54, 'w': 0.2157133497116508, 'c1': 2.785420580424854, 'c2': 1.433302617963326,
'lb': -79.5580653002381, 'ub': 7.990057168788164}. Best is trial 110 with value:
1.0430942124367217e-11.

[I 2024-08-10 12:41:30,636] Trial 129 finished with value:
2.8174249029072783e-08 and parameters: {'num_particulas': 56,
'cant_iteraciones': 264, 'w': 0.16895935316660385, 'c1': 2.632316859815392,
'c2': 1.8049498080910467, 'lb': -73.6034260696334, 'ub': 19.511455232279893}.
Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:30,725] Trial 130 finished with value:
0.4724690397204857 and parameters: {'num_particulas': 10, 'cant_iteraciones':
270, 'w': 0.12858426441290374, 'c1': 2.730186736835783, 'c2':
1.1735989375330393, 'lb': -83.08591147725056, 'ub': 13.296573315738948}. Best is
trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:30,804] Trial 131 finished with value:
3.5039844702900252e-09 and parameters: {'num_particulas': 32,
'cant_iteraciones': 239, 'w': 0.11451689667445943, 'c1': 2.8503048206346366,
'c2': 1.5293288761112773, 'lb': -86.57627977631125, 'ub': 5.292032141947585}.
Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:30,867] Trial 132 finished with value:
1.2617704909459173 and parameters: {'num_particulas': 17, 'cant_iteraciones':
253, 'w': 0.10070780720644339, 'c1': 2.9181487233346894, 'c2':

1.6568449103691636, 'lb': -90.5645540000045, 'ub': 2.761818944195035}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:30,932] Trial 133 finished with value: 1.1262341149212932e-08 and parameters: {'num_particulas': 27, 'cant_iteraciones': 247, 'w': 0.14928562414571697, 'c1': 2.815798588264682, 'c2': 1.55292809282282, 'lb': -85.35724373326347, 'ub': 10.981055558105508}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:31,186] Trial 134 finished with value: 81.80522913329224 and parameters: {'num_particulas': 36, 'cant_iteraciones': 259, 'w': 0.20098860772259133, 'c1': 2.953127687664218, 'c2': 2.095462823139579, 'lb': -94.09061408873906, 'ub': 0.34718754852975886}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:31,240] Trial 135 finished with value: 2.6222665035637146e-08 and parameters: {'num_particulas': 12, 'cant_iteraciones': 279, 'w': 0.13822509290539448, 'c1': 2.883918673571866, 'c2': 1.4888195780836428, 'lb': -87.89434177159283, 'ub': 6.883549738705014}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:31,298] Trial 136 finished with value: 7.743486346948065e-08 and parameters: {'num_particulas': 25, 'cant_iteraciones': 232, 'w': 0.17921779125564352, 'c1': 2.7283112990373364, 'c2': 1.3132780127021895, 'lb': -91.76861982970526, 'ub': 9.170558464452098}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:31,356] Trial 137 finished with value: 3.9109910328993045e-08 and parameters: {'num_particulas': 23, 'cant_iteraciones': 109, 'w': 0.1566809808870127, 'c1': 2.077555848418135, 'c2': 1.4098339737795578, 'lb': -82.03720078331914, 'ub': 16.230976024115158}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:31,419] Trial 138 finished with value: 1.0309927582507894e-08 and parameters: {'num_particulas': 30, 'cant_iteraciones': 240, 'w': 0.11811115751647427, 'c1': 2.8350167724205377, 'c2': 1.7294424694553996, 'lb': -99.91184913879091, 'ub': 4.124777858986899}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:31,475] Trial 139 finished with value: 3.031142632349496e-09 and parameters: {'num_particulas': 20, 'cant_iteraciones': 272, 'w': 0.13596713089248832, 'c1': 2.6780422097337104, 'c2': 1.6261310016025539, 'lb': -90.17660436569335, 'ub': 12.237197192953387}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:31,554] Trial 140 finished with value: 4.366545494422943e-09 and parameters: {'num_particulas': 44, 'cant_iteraciones': 261, 'w': 0.28476375874877935, 'c1': 2.595441754457546, 'c2': 1.2589303964433523, 'lb': -95.13232409303671, 'ub': 7.835536168900372}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:31,620] Trial 141 finished with value: 1.3023417887506432e-07 and parameters: {'num_particulas': 47, 'cant_iteraciones': 229, 'w': 0.1546609369827978, 'c1': 2.666693716308055, 'c2': 1.2005906794436507, 'lb': -89.81162126694942, 'ub': 5.650055711942195}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:31,695] Trial 142 finished with value:

8.52984919474453e-10 and parameters: {'num_particulas': 51, 'cant_iteraciones': 237, 'w': 0.16457283963734926, 'c1': 2.6343582066772435, 'c2': 1.257301480013156, 'lb': -86.29114718723802, 'ub': 6.636465894293522}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:31,760] Trial 143 finished with value: 3.6077401494063612 and parameters: {'num_particulas': 50, 'cant_iteraciones': 254, 'w': 0.16928862199406736, 'c1': 2.7736158558210886, 'c2': 1.3705325173980532, 'lb': -84.9486075881216, 'ub': 2.5159641126769943}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:31,832] Trial 144 finished with value: 6.224763206471792e-09 and parameters: {'num_particulas': 52, 'cant_iteraciones': 245, 'w': 0.11741401806678126, 'c1': 2.997104655565081, 'c2': 1.0958503268397704, 'lb': -87.29653388856771, 'ub': 10.038752861822957}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:31,895] Trial 145 finished with value: 94.69924034748917 and parameters: {'num_particulas': 38, 'cant_iteraciones': 237, 'w': 0.19877017606435615, 'c1': 2.9249587502560637, 'c2': 1.311072702423401, 'lb': -83.52022778443563, 'ub': 0.035003398280156084}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:31,965] Trial 146 finished with value: 2.7050611255257776e-09 and parameters: {'num_particulas': 54, 'cant_iteraciones': 248, 'w': 0.10107346112419448, 'c1': 2.4584688678662725, 'c2': 1.9751903493676766, 'lb': -93.4262180146936, 'ub': 15.038930373386044}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:32,020] Trial 147 finished with value: 2.1213740051223657e-09 and parameters: {'num_particulas': 15, 'cant_iteraciones': 266, 'w': 0.12725239402446162, 'c1': 2.3917788920762058, 'c2': 2.2090774876687727, 'lb': -78.76640547550427, 'ub': 7.925443677125286}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:32,077] Trial 148 finished with value: 5.4407875817908986e-08 and parameters: {'num_particulas': 33, 'cant_iteraciones': 255, 'w': 0.18262335418732523, 'c1': 2.8740144149199365, 'c2': 1.440121312404043, 'lb': -66.68440747225773, 'ub': 4.125481034288203}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:32,140] Trial 149 finished with value: 4.757015951699342e-10 and parameters: {'num_particulas': 42, 'cant_iteraciones': 234, 'w': 0.14254515727083242, 'c1': 2.7795887143444338, 'c2': 1.258154061723292, 'lb': -81.70772091265756, 'ub': 13.197586543038451}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:32,215] Trial 150 finished with value: 1.1564564306323843e-09 and parameters: {'num_particulas': 42, 'cant_iteraciones': 234, 'w': 0.16550151989431078, 'c1': 2.547530678185854, 'c2': 1.2590113598630193, 'lb': -80.29768247789949, 'ub': 18.584798206687797}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:32,282] Trial 151 finished with value: 1.3834083639960583e-09 and parameters: {'num_particulas': 41, 'cant_iteraciones': 183, 'w': 0.14626418179156425, 'c1': 2.5690569935385663, 'c2': 1.2447650770427874, 'lb': -80.24420218347633, 'ub': 17.546606198084685}.

Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:32,363] Trial 152 finished with value: 5.7200684819072677e-08 and parameters: {'num_particulas': 43, 'cant_iteraciones': 234, 'w': 0.1606561478172685, 'c1': 2.764087467267003, 'c2': 1.1386540596648815, 'lb': -75.49916999130939, 'ub': 14.224050496840482}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:32,437] Trial 153 finished with value: 1.1758068518771343e-09 and parameters: {'num_particulas': 45, 'cant_iteraciones': 250, 'w': 0.18667574997375422, 'c1': 2.7000008088432823, 'c2': 1.2658205045128625, 'lb': -81.77779177110523, 'ub': 18.552917155152247}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:32,500] Trial 154 finished with value: 3.461126284514168e-10 and parameters: {'num_particulas': 38, 'cant_iteraciones': 242, 'w': 0.21756509737282814, 'c1': 2.5283449059733965, 'c2': 1.1871586139723695, 'lb': -84.34047629270538, 'ub': 12.297384424446046}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:32,574] Trial 155 finished with value: 6.22255485279025e-08 and parameters: {'num_particulas': 49, 'cant_iteraciones': 237, 'w': 0.25922405964223144, 'c1': 2.6287012309630255, 'c2': 1.1892991072616084, 'lb': -85.76663794636065, 'ub': 11.947786792505577}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:32,656] Trial 156 finished with value: 2.1794183685417582e-09 and parameters: {'num_particulas': 37, 'cant_iteraciones': 212, 'w': 0.22296495749334153, 'c1': 2.8092341067949653, 'c2': 1.3371877913905297, 'lb': -84.02550149054574, 'ub': 10.912195890433946}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:32,722] Trial 157 finished with value: 1.1332439297133427e-08 and parameters: {'num_particulas': 38, 'cant_iteraciones': 243, 'w': 0.2120879109891328, 'c1': 2.710119290193494, 'c2': 1.0595506707335816, 'lb': -57.49449352642088, 'ub': 13.485912791216284}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:32,789] Trial 158 finished with value: 8.520108654953404e-09 and parameters: {'num_particulas': 35, 'cant_iteraciones': 227, 'w': 0.234345133420009, 'c1': 2.4968702317959885, 'c2': 1.1862568632390529, 'lb': -77.14927391455669, 'ub': 9.247172461598105}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:32,855] Trial 159 finished with value: 1.2523016328772836e-09 and parameters: {'num_particulas': 40, 'cant_iteraciones': 260, 'w': 0.31988379273761053, 'c1': 2.751588871461463, 'c2': 0.956577537412433, 'lb': -88.64879827380946, 'ub': 6.241576270337148}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:32,927] Trial 160 finished with value: 2.4953853053726e-08 and parameters: {'num_particulas': 48, 'cant_iteraciones': 269, 'w': 0.20149219170387128, 'c1': 2.6175147020805216, 'c2': 1.9143456921012745, 'lb': -72.5363194490199, 'ub': 15.922342610201545}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:32,995] Trial 161 finished with value: 5.5304139647244e-08 and parameters: {'num_particulas': 42, 'cant_iteraciones':

232, 'w': 0.1714145486044184, 'c1': 2.5351875064848226, 'c2': 1.2731716691974175, 'lb': -80.46348340659631, 'ub': 12.692565416370519}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:33,063] Trial 162 finished with value: 9.716513304854298e-09 and parameters: {'num_particulas': 51, 'cant_iteraciones': 243, 'w': 0.14390932615007315, 'c1': 2.5406200381479516, 'c2': 1.2310785617012556, 'lb': -82.6001290562937, 'ub': 23.896996535998923}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:33,151] Trial 163 finished with value: 1.9679112111726608e-09 and parameters: {'num_particulas': 46, 'cant_iteraciones': 251, 'w': 0.40710893955716987, 'c1': 2.67971991068141, 'c2': 1.3701474902016122, 'lb': -86.32655670117266, 'ub': 20.506854948216077}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:33,216] Trial 164 finished with value: 5.845538461387538e-09 and parameters: {'num_particulas': 42, 'cant_iteraciones': 221, 'w': 0.1621008588197201, 'c1': 2.9489985882005207, 'c2': 1.3252020334369259, 'lb': -77.99746486835764, 'ub': 10.137912337690164}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:33,286] Trial 165 finished with value: 1.2104353281743032e-10 and parameters: {'num_particulas': 39, 'cant_iteraciones': 238, 'w': 0.1846156936703185, 'c1': 2.268161257257205, 'c2': 1.1453702815925186, 'lb': -91.21685320399048, 'ub': 17.90232879942068}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:33,354] Trial 166 finished with value: 1.3047960165200051e-08 and parameters: {'num_particulas': 58, 'cant_iteraciones': 239, 'w': 0.18766124181555077, 'c1': 2.361281215899634, 'c2': 1.1006966161842642, 'lb': -91.79647519620903, 'ub': 47.082999074389164}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:33,420] Trial 167 finished with value: 6.391416171476973e-08 and parameters: {'num_particulas': 36, 'cant_iteraciones': 247, 'w': 0.1328485795519898, 'c1': 2.840099894377539, 'c2': 1.1707856779592838, 'lb': -96.8216236692262, 'ub': 7.6197359626313075}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:33,486] Trial 168 finished with value: 1.5149292082249645e-08 and parameters: {'num_particulas': 40, 'cant_iteraciones': 257, 'w': 0.24194223126713746, 'c1': 2.783142661565553, 'c2': 1.1440197312381222, 'lb': -89.25908591743243, 'ub': 25.49206416349932}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:33,541] Trial 169 finished with value: 1.9005604101442675e-08 and parameters: {'num_particulas': 18, 'cant_iteraciones': 295, 'w': 0.20997015203249433, 'c1': 2.8949880236156513, 'c2': 1.4010287153602994, 'lb': -93.60185040482607, 'ub': 14.499166509108552}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:33,598] Trial 170 finished with value: 1.3468464812172963e-08 and parameters: {'num_particulas': 13, 'cant_iteraciones': 264, 'w': 0.17704323003033817, 'c1': 2.610313826336184, 'c2': 1.4781227184955057, 'lb': -91.37446701294729, 'ub': 16.966728644988496}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:33,660] Trial 171 finished with value:
2.6368921357437065e-08 and parameters: {'num_particulas': 42,
'cant_iteraciones': 234, 'w': 0.15190906834604825, 'c1': 2.4354124823030046,
'c2': 1.2318801037878295, 'lb': -84.40186701889235, 'ub': 17.831734820553653}.
Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:33,726] Trial 172 finished with value:
1.3242627448586928e-08 and parameters: {'num_particulas': 44,
'cant_iteraciones': 240, 'w': 0.16758706595788692, 'c1': 2.2832012986835974,
'c2': 1.2793792759921572, 'lb': -87.13774455785398, 'ub': 11.569790366447782}.
Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:33,793] Trial 173 finished with value:
2.080240426751732e-10 and parameters: {'num_particulas': 38, 'cant_iteraciones':
229, 'w': 0.12267998463147065, 'c1': 2.492365585825281, 'c2':
2.2618346552226325, 'lb': -81.72137177138869, 'ub': 52.47173545025056}. Best is
trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:33,859] Trial 174 finished with value:
5.370636438370834e-08 and parameters: {'num_particulas': 39, 'cant_iteraciones':
223, 'w': 0.12172246171142573, 'c1': 2.7313899052132675, 'c2':
2.054721328998417, 'lb': -82.91126960087142, 'ub': 9.05446331321061}. Best is
trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:33,962] Trial 175 finished with value:
6.488505126516373e-08 and parameters: {'num_particulas': 34, 'cant_iteraciones':
229, 'w': 0.11067560137148176, 'c1': 2.481385589015246, 'c2':
2.3817700092470444, 'lb': -48.64488594718752, 'ub': 52.29856304078395}. Best is
trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:34,049] Trial 176 finished with value:
1.7926262839554817e-07 and parameters: {'num_particulas': 37,
'cant_iteraciones': 246, 'w': 0.1335821651088493, 'c1': 2.4109234127859054,
'c2': 2.302331099948052, 'lb': -88.88729180131935, 'ub': 54.920690481507826}.
Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:34,121] Trial 177 finished with value:
3.899399072852578e-06 and parameters: {'num_particulas': 16, 'cant_iteraciones':
253, 'w': 0.14386255568362322, 'c1': 2.6578116450768285, 'c2':
2.124248892742367, 'lb': -85.60481537022687, 'ub': 40.55733702549503}. Best is
trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:34,191] Trial 178 finished with value:
80.62493458036086 and parameters: {'num_particulas': 32, 'cant_iteraciones':
241, 'w': 0.18937316260482298, 'c1': 1.862962065755439, 'c2': 2.284894216839711,
'lb': -95.90860409658328, 'ub': 5.962008385951435}. Best is trial 110 with
value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:34,272] Trial 179 finished with value:
3.839160740923368e-09 and parameters: {'num_particulas': 39, 'cant_iteraciones':
274, 'w': 0.12525183922458957, 'c1': 2.8301174930888995, 'c2':
1.856327524753569, 'lb': -91.80441739724787, 'ub': 13.316807585486732}. Best is
trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:34,351] Trial 180 finished with value:
3.602052326803216e-05 and parameters: {'num_particulas': 36, 'cant_iteraciones':
250, 'w': 0.22317070230881278, 'c1': 2.9305678576722682, 'c2':

2.218013907671908, 'lb': -94.57031853496295, 'ub': 22.53570310592879}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:34,454] Trial 181 finished with value: 6.803633854766275e-11 and parameters: {'num_particulas': 46, 'cant_iteraciones': 236, 'w': 0.16189161948528136, 'c1': 2.5260028127510825, 'c2': 2.00771523084551, 'lb': -78.18259771110505, 'ub': 27.432542535424698}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:34,544] Trial 182 finished with value: 1.5543959593085239e-09 and parameters: {'num_particulas': 46, 'cant_iteraciones': 226, 'w': 0.14714920781231292, 'c1': 2.599930019906438, 'c2': 1.9741177444667262, 'lb': -78.56660880728298, 'ub': 28.88921975808463}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:34,726] Trial 183 finished with value: 2.6019064855788303e-07 and parameters: {'num_particulas': 49, 'cant_iteraciones': 217, 'w': 0.5540673839486839, 'c1': 2.568889727710798, 'c2': 2.146278612815689, 'lb': -70.8977885326485, 'ub': 15.030937426899026}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:34,794] Trial 184 finished with value: 9.812215028824453e-09 and parameters: {'num_particulas': 44, 'cant_iteraciones': 238, 'w': 0.19937382110988575, 'c1': 2.50701382877364, 'c2': 1.9469466285490937, 'lb': -82.36128798100118, 'ub': 10.354129916364405}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:34,867] Trial 185 finished with value: 4.697634537790992e-08 and parameters: {'num_particulas': 54, 'cant_iteraciones': 231, 'w': 0.17581323923620606, 'c1': 2.6458372255516776, 'c2': 1.8798547337301568, 'lb': -87.27840297192077, 'ub': 7.6639280182577165}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:34,934] Trial 186 finished with value: 5.5044047806242294e-08 and parameters: {'num_particulas': 47, 'cant_iteraciones': 245, 'w': 0.1565387998466552, 'c1': 2.9994205119909405, 'c2': 2.093485243814333, 'lb': -74.63790677361199, 'ub': 22.07052046345187}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:34,994] Trial 187 finished with value: 7.254076992354113e-07 and parameters: {'num_particulas': 38, 'cant_iteraciones': 256, 'w': 0.13494909253303902, 'c1': 2.4430154666164294, 'c2': 2.002892173317702, 'lb': -77.00851750897944, 'ub': 11.811849026949549}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:35,078] Trial 188 finished with value: 4.418415353551471e-08 and parameters: {'num_particulas': 56, 'cant_iteraciones': 289, 'w': 0.25211889836046436, 'c1': 2.269289934965121, 'c2': 2.1838510258622827, 'lb': -83.7119616129581, 'ub': 5.6718616504774335}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:35,145] Trial 189 finished with value: 2.5682345842137075 and parameters: {'num_particulas': 51, 'cant_iteraciones': 236, 'w': 0.11824991706504714, 'c1': 2.3419736621624714, 'c2': 2.038575542269391, 'lb': -79.45853138536776, 'ub': 2.6099935146548328}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:35,209] Trial 190 finished with value:

8.839801076146169e-09 and parameters: {'num_particulas': 34, 'cant_iteraciones': 262, 'w': 0.18591096321772935, 'c1': 2.718809726049777, 'c2': 1.546792188118173, 'lb': -89.03989944863729, 'ub': 16.399826551258347}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:35,278] Trial 191 finished with value:

1.7598951946608015e-08 and parameters: {'num_particulas': 43, 'cant_iteraciones': 233, 'w': 0.16581014035778674, 'c1': 2.535189942164738, 'c2': 1.205602167017622, 'lb': -80.38920606099991, 'ub': 20.180957559068958}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:35,342] Trial 192 finished with value:

4.46316239491507e-08 and parameters: {'num_particulas': 41, 'cant_iteraciones': 242, 'w': 0.160480819392522, 'c1': 2.4913949467251175, 'c2': 1.326304862774567, 'lb': -84.34756533891716, 'ub': 18.916242438307787}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:35,407] Trial 193 finished with value:

3.236314261677636e-08 and parameters: {'num_particulas': 45, 'cant_iteraciones': 227, 'w': 0.1428212727775278, 'c1': 2.563934341709304, 'c2': 1.3802849133444943, 'lb': -81.45473782783453, 'ub': 13.842490804616684}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:35,472] Trial 194 finished with value:

9.558694832104532e-09 and parameters: {'num_particulas': 41, 'cant_iteraciones': 236, 'w': 0.1763416978796549, 'c1': 2.1386102100130224, 'c2': 1.7850959315750745, 'lb': -76.49102991881152, 'ub': 25.571791088254283}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:35,555] Trial 195 finished with value:

4.767805156151672e-09 and parameters: {'num_particulas': 62, 'cant_iteraciones': 247, 'w': 0.2020031431264251, 'c1': 2.791662007504438, 'c2': 1.4548258366783935, 'lb': -86.52374957383456, 'ub': 18.168021061138422}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:35,618] Trial 196 finished with value:

2.1490881255765617e-09 and parameters: {'num_particulas': 39, 'cant_iteraciones': 253, 'w': 0.10860985358037545, 'c1': 2.8710915473130876, 'c2': 1.303397679900638, 'lb': -81.24847718825187, 'ub': 9.332316298070424}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:35,690] Trial 197 finished with value:

2.4843147393476114e-10 and parameters: {'num_particulas': 48, 'cant_iteraciones': 240, 'w': 0.15369276551405137, 'c1': 2.618572689671523, 'c2': 1.1384332818945437, 'lb': -90.48694545203595, 'ub': 15.716856960776823}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:35,774] Trial 198 finished with value:

7.010479039980827e-08 and parameters: {'num_particulas': 53, 'cant_iteraciones': 278, 'w': 0.34264885189761224, 'c1': 2.7439309164712635, 'c2': 1.0590913016622827, 'lb': -90.45550956349751, 'ub': 11.680137878957018}. Best is trial 110 with value: 1.0430942124367217e-11.

[I 2024-08-10 12:41:35,843] Trial 199 finished with value:

2.177149917636281e-10 and parameters: {'num_particulas': 49, 'cant_iteraciones': 242, 'w': 0.10032767866216602, 'c1': 2.676434014640137, 'c2': 1.1375107819884056, 'lb': -98.64063694521293, 'ub': 33.95098792691052}. Best is

trial 110 with value: 1.0430942124367217e-11.

Mejor valor objetivo: 1.0430942124367217e-11

Mejores hiperparámetros: {'num_particulas': 39, 'cant_iteraciones': 289, 'w': 0.21841625601899686, 'c1': 2.9018986677091667, 'c2': 1.4722160130770745, 'lb': -83.86920697758734, 'ub': 9.596565052531925}

```
[497]: # Mejores hiperparámetros
study.best_params
```

```
[497]: {'num_particulas': 39,
        'cant_iteraciones': 289,
        'w': 0.21841625601899686,
        'c1': 2.9018986677091667,
        'c2': 1.4722160130770745,
        'lb': -83.86920697758734,
        'ub': 9.596565052531925}
```

```
[498]: # Con los mejores hiperparámetros hallados, resuelvo con PSO
best_position, best_value = pso(
    funcion_objetivo,
    lb=[study.best_params['lb']]*2,
    ub=[study.best_params['ub']]*2,
    swarmsize=study.best_params['num_particulas'],
    maxiter=study.best_params['cant_iteraciones'],
    omega=study.best_params['w'],
    phip=study.best_params['c1'],
    phig=study.best_params['c2'],
    debug=False)
```

Stopping search: Swarm best objective change less than 1e-08

```
[499]: # Verifico las igualdades de mis dos ecuaciones
x_1 = best_position[0]
x_2 = best_position[1]

print(np.round(3*x_1+2*x_2, 3) == 9)
print(np.round(x_1-5*x_2, 3) == 4)
```

True
False

El algoritmo PSO puede resolver un sistema de n ecuaciones no lineales con n incógnitas. Esto se logra formulando el problema como una minimización de la suma de los errores cuadráticos para cada una de las ecuaciones. La función objetivo suma los cuadrados de las diferencias de cada ecuación con respecto a cero. Al minimizar esta función, se busca encontrar un conjunto de valores para x_1, x_2, \dots, x_n que hagan que todas las ecuaciones se acerquen a cero simultáneamente.

Los resultados obtenidos mediante PSO están fuertemente influenciados por los valores de los hiperparámetros seleccionados, como el número de partículas, el número de iteraciones, el factor

de inercia w , el coeficiente $c1$ y el coeficiente $c2$.

- Número de partículas: Un número mayor de partículas puede explorar un espacio de soluciones más amplio, lo que potencialmente puede llevar a una mejor solución. Sin embargo, un número excesivo de partículas aumenta el costo computacional.
- Número de iteraciones: Un mayor número de iteraciones permite que las partículas tengan más tiempo para converger hacia la solución óptima, pero más iteraciones también incrementan el tiempo de cómputo.
- Factor de inercia w : Controla la tendencia de las partículas a mantener su velocidad. Un valor alto de w fomenta la exploración global, mientras que un valor bajo favorece la explotación local.
- Coeficiente $c1$: Representa la confianza en la experiencia propia de la partícula. Un valor alto de $c1$ hace que las partículas se dirijan hacia sus mejores posiciones encontradas, aumentando la explotación.
- Coeficiente $c2$: Representa la confianza en la experiencia del enjambre. Un valor alto de $c2$ hace que las partículas tiendan a moverse hacia las mejores posiciones encontradas por el enjambre, promoviendo la explotación.

```
[500]: # CÓDIGO PARA EXPORTAR NOTEBOOK
from nbconvert import PDFExporter
import nbformat

# NOMBRE DEL NOTEBOOK
notebook_filename = "C:/Users/Leandro/Desktop/IoT/5_Bimestre/Algoritmos_
↳genéticos/TP2/TP2.ipynb"

# LEO EL NOTEBOOK
with open(notebook_filename, 'r', encoding='utf-8') as nb_file:
    notebook = nbformat.read(nb_file, as_version=4)

# CONFIGURO EL EXPORTADOR PDF
pdf_exporter = PDFExporter()

# CONVIERTO A PDF
pdf_data, _ = pdf_exporter.from_notebook_node(notebook)

# GUARDO EL PDF
pdf_filename = "C:/Users/Leandro/Desktop/IoT/5_Bimestre/Algoritmos genéticos/
↳TP2/TP2.pdf"
with open(pdf_filename, 'wb') as pdf_file:
    pdf_file.write(pdf_data)
```