

## Trabajo Práctico N° 2

### Algoritmos Evolutivos (2024) – CEIA

Por: Sevann Radhak Triztan  
sevann.rahdak@gmail.com

1. Escribir un algoritmo PSO para la maximización de la función  $y = \sin(x) + \sin(x^2)$ , en el intervalo de  $0 \leq x \leq 10$ . Ejecutar el algoritmo en Python con los siguientes parámetros: número de partículas = 2, máximo número de iteraciones = 30, coeficientes de aceleración  $c_1 = c_2 = 1.49$ , peso de inercia  $w = 0.5$ . De acuerdo a los requisitos anteriores realizar las siguientes consignas:

- a. Transcribir en el informe la solución óptima encontrada (dominio) y el valor óptimo (imagen).

**SOLUCIÓN:**

Optimal Solution (x): 1.2946711614227775 --- Objective Value (y): [1.95657202]

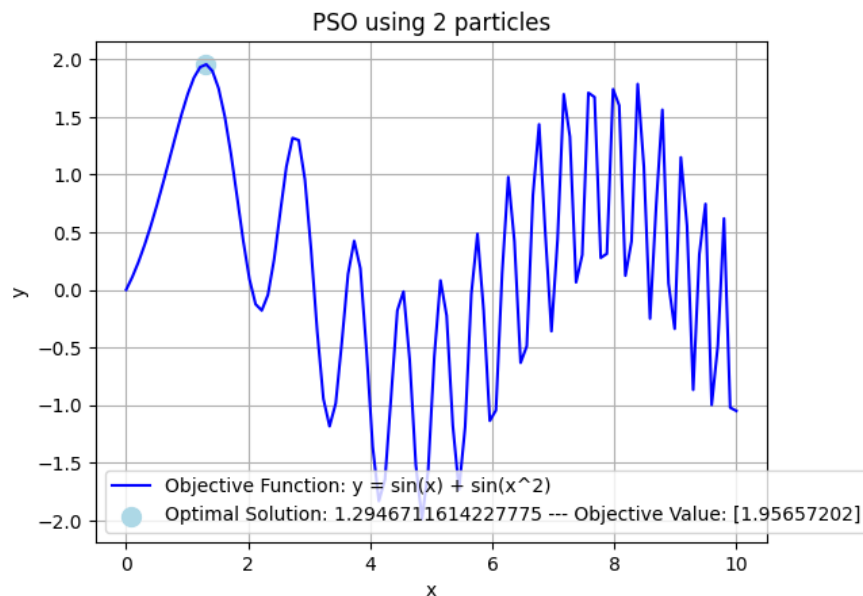
- b. Indicar la URL del repositorio en donde se encuentra el algoritmo PSO.

**SOLUCIÓN:**

<https://github.com/sevann-radhak/UBA-AE/blob/main/TP2/TP2-ej1.ipynb>

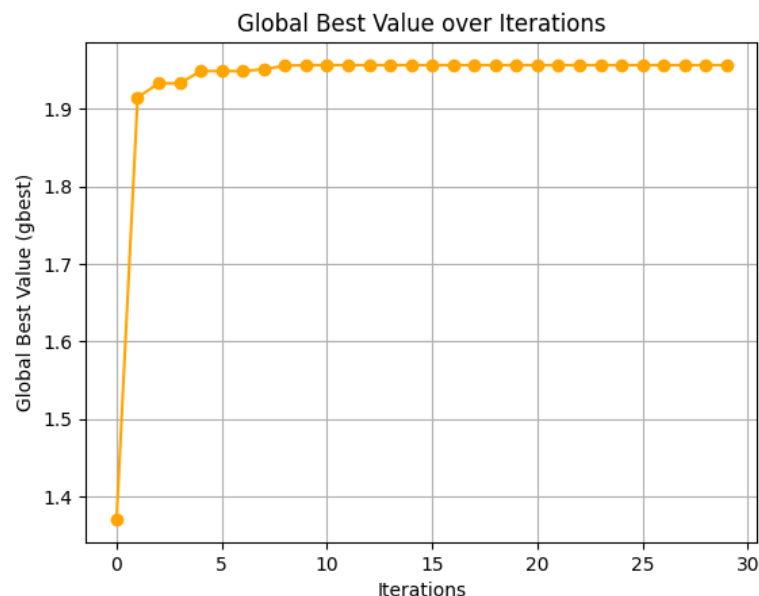
- c. Graficar usando matplotlib la función objetivo y agregar un punto celeste en donde el algoritmo haya encontrado el valor máximo. El gráfico debe contener etiquetas en los ejes, leyenda y un título.

**SOLUCIÓN:**



- d. Realizar un gráfico de línea que muestre gbest en función de las iteraciones realizadas.

**SOLUCIÓN:**



e. Realizar observaciones/comentarios/conclusiones sobre los resultados obtenidos.

**SOLUCIÓN:**

- El valor máximo de la función ocurre alrededor de  $x = 1.3$  con un valor de aproximadamente 1.956.
- La función exhibe un comportamiento oscilatorio, con valores tanto positivos como negativos. Alcanza máximos y mínimos locales, característico de funciones que involucran seno.
- El valor óptimo global (gbest) comienza en 1.3707 y mejora a lo largo de las iteraciones, alcanzando un máximo de 1.9566 al final del proceso de optimización.
- La convergencia parece ser bastante estable, ya que el valor óptimo global no cambia después de la iteración 10.

2. 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:

$$-100 \leq x \leq 100; x \in \mathbb{R}$$

$$-100 \leq y \leq 100; y \in \mathbb{R}$$

$$-50 \leq a \leq 50; a \in \mathbb{R}$$

$$-50 \leq b \leq 50; b \in \mathbb{R}$$

escribir en Python un algoritmo PSO para la minimización de la función (1) con los siguientes parámetros: número de partículas = 20, máximo número de iteraciones = 10, coeficientes de aceleración  $c1 = c2 = 2$ , peso de inercia  $w = 0.7$ , y que cumpla con las siguientes consignas:

a. Transcribir en el informe la solución óptima encontrada (dominio) y el valor óptimo (imagen).

**SOLUCIÓN:**

Optimal Solution (x, y): [ 30.38654285 -27.04189322] --- Optimal Value (f(x, y)): 4.31874331572478

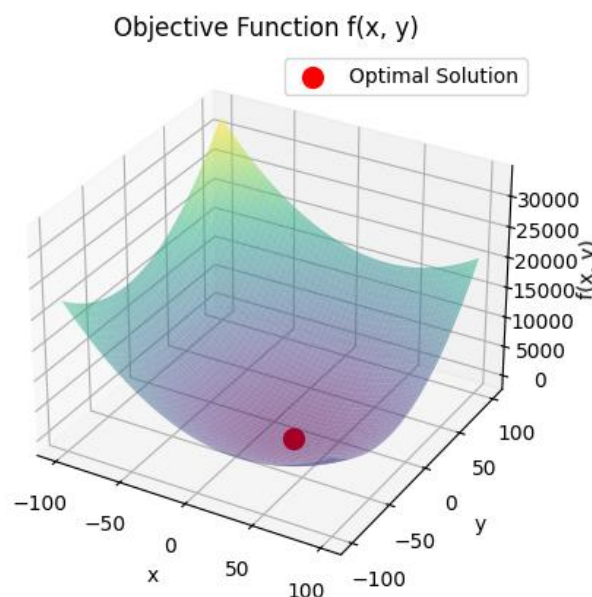
b. Indicar la URL del repositorio en donde se encuentra el algoritmo PSO.

**SOLUCIÓN:**

<https://github.com/sevann-radhak/UBA-AE/blob/main/TP2/TP2-ej2.ipynb>

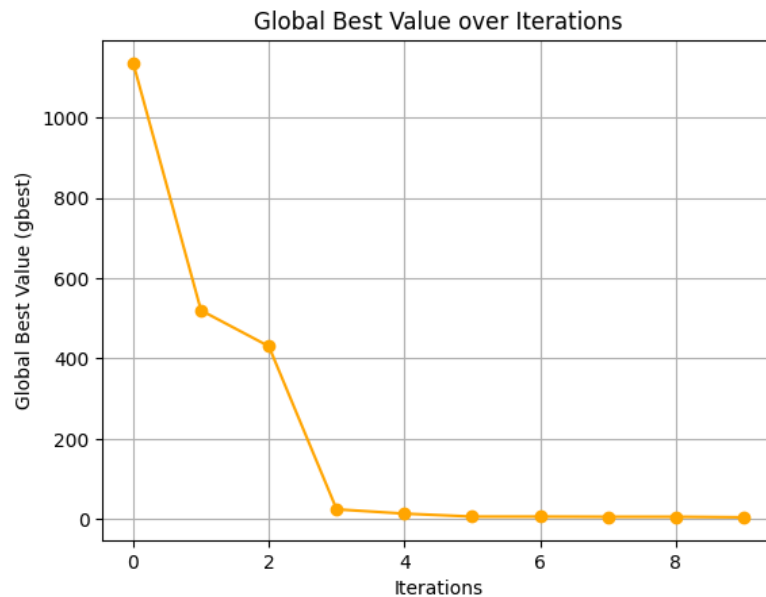
c. Graficar usando matplotlib la función objetivo  $f(x, y)$  en 3D y agregar un punto rojo en donde el algoritmo haya encontrado el valor mínimo. El gráfico debe contener etiquetas en los ejes, leyenda y un título.

**SOLUCIÓN:**



- d. Realizar un gráfico de línea que muestre gbest en función de las iteraciones realizadas.

**SOLUCIÓN:**



- e. Establecer el coeficiente de inercia  $w$  en 0, ejecutar el algoritmo y realizar observaciones, comentarios y conclusiones sobre los resultados observados.

**SOLUCIÓN:**

Optimal Solution with  $w=0$  (x, y): [ 29.79595196 -24.85382075]

Optimal Value: 0.06300397642559323

El algoritmo es eficiente al encontrar una solución óptima de  $f(x, y) = 4.3187$  tras varias iteraciones, con una mejora desde un valor inicial de 1135.5052. La estabilización de los valores de gbest en iteraciones posteriores es producto de que el algoritmo ha alcanzado un equilibrio.

- f. Reescribir el algoritmo PSO para que cumpla nuevamente con los ítems A hasta F pero usando la biblioteca pswarm (from pswarm import pso).

**SOLUCIÓN:**

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

Optimal solution (x, y): [ 30.00018206 -26.99986256]

Optimal value (f(x, y)): 4.000000052035346

- g. Realizar observaciones/comentarios/conclusiones comparando los resultados obtenidos sin pswarm y con pswarm.

**SOLUCIÓN:**

La solución óptima encontrada manualmente fue (x, y) = [30.3865, -27.0419] con un valor de  $f(x, y) = 4.3187$ , mientras que al utilizar pswarm, se logró una solución más precisa de (x, y) = [30.0002, -26.9999] y un valor óptimo de  $f(x, y) = 4.0000$ . Esto indica que pswarm mejora la precisión de los resultados y facilita la convergencia al alcanzar un cambio en el mejor objetivo por debajo de  $1e-08$ .

3. Maximizar mediante PSO en Python y con parámetros a elección la función:  $z = e^{-0.1(x^2+y^2)} \cos(x) \sin(y)$  donde el intervalo de las variables de decisión se encuentra en el rango  $-50 \leq (x, y) \leq 50$ . En base a las especificaciones mencionadas realizar las siguientes consignas:

- a. Transcribir en el informe la solución óptima encontrada (dominio) y el valor óptimo (imagen).

**SOLUCIÓN:**

Optimal Solution (x, y): [-2.36479001e-05 1.31374039e+00]

Optimal Value (z): 0.8138325411670783

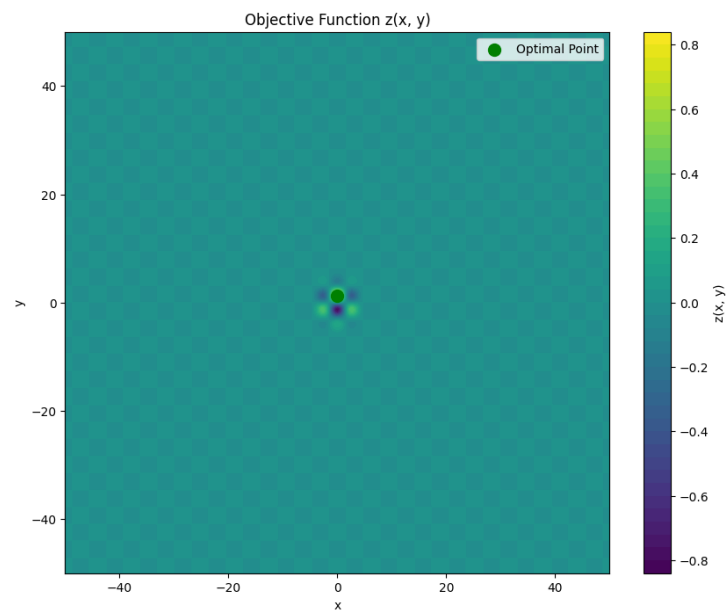
- b. Indicar la URL del repositorio en donde se encuentra el algoritmo PSO.

**SOLUCIÓN:**

Insertar url ----- insertar url -----

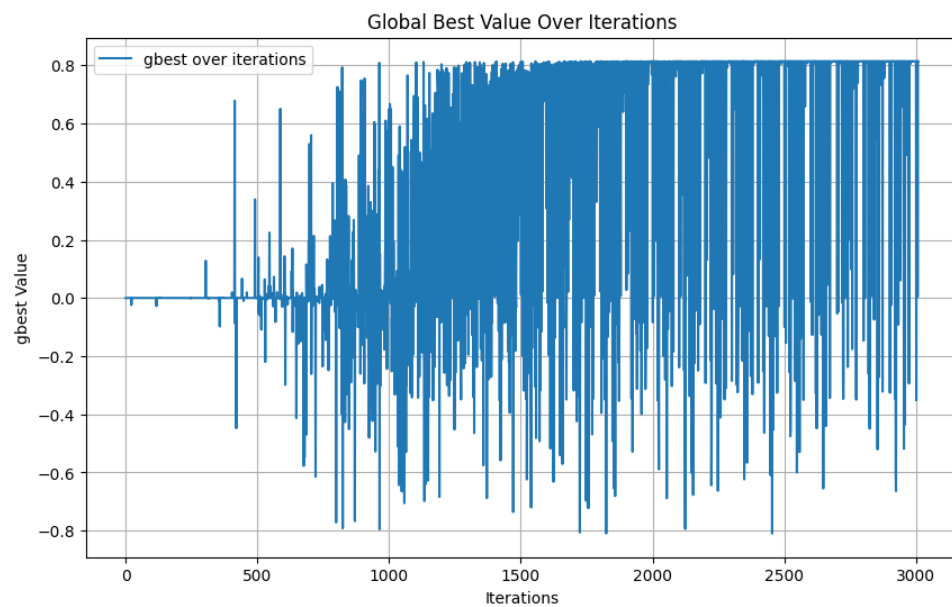
- c. Graficar usando matplotlib la función objetivo  $z(x, y)$  y agregar un punto verde en donde el algoritmo haya encontrado el valor máximo. El gráfico debe contener etiquetas en los ejes, leyenda y un título.

**SOLUCIÓN:**



- d. Realizar un gráfico de línea que muestre gbest en función de las iteraciones realizadas.

**SOLUCIÓN:**



**4.**