

Ejercicio 1 (Valor 1 punto)

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 $c1 = c2 = 1.49$, peso de inercia $w = 0.5$.

De acuerdo a los requisitos anteriores realizar las siguientes consignas:

- (Valor 0.1 puntos) Transcribir en el informe la solución óptima encontrada (dominio) y el valor óptimo (imagen).
- (Valor 0.1 puntos) Indicar la URL del repositorio en donde se encuentra el algoritmo PSO.
- (Valor 0.2 puntos) 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.
- (Valor 0.2 puntos) Realizar un gráfico de línea que muestre gbest en función de las iteraciones realizadas.
- (Valor 0.4 puntos) Realizar observaciones/comentarios/conclusiones sobre los resultados obtenidos.

Ejercicio 2 (Valor 2 puntos)

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

$$f(x, y) = (x - a)^2 + (y + b)^2 \quad (1)$$

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:

- (Valor 0.1 puntos) Transcribir en el informe la solución óptima encontrada (dominio) y el valor óptimo (imagen).
- (Valor 0.1 puntos) Indicar la URL del repositorio en donde se encuentra el algoritmo PSO.
- (Valor 0.3 puntos) 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.
- (Valor 0.3 puntos) Realizar un gráfico de línea que muestre g_{best} en función de las iteraciones realizadas.
- (Valor 0.4 puntos) Establecer el coeficiente de inercia w en 0, ejecutar el algoritmo y realizar observaciones/comentarios/conclusiones sobre los resultados observados.
- (Valor 0.4 puntos) Reescribir el algoritmo PSO para que cumpla nuevamente con los ítems A hasta F pero usando la biblioteca pswarm (`from pswarm import pso`).
- (Valor 0.4 puntos) Realizar observaciones/comentarios/conclusiones comparando los resultados obtenidos sin pswarm y con pswarm.

Ejercicio 3 (Valor 4 puntos)

Maximizar mediante PSO en Python y con parámetros a elección la función:

$$z = e^{-0.1(x^2+y^2)} \cdot \cos(x) \cdot \sin(y)$$

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:

- (Valor 0.1 puntos) Transcribir en el informe la solución óptima encontrada (dominio) y el valor óptimo (imagen).
- (Valor 0.1 puntos) Indicar la URL del repositorio en donde se encuentra el algoritmo PSO.
- (Valor 0.4 puntos) 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.
- (Valor 0.4 puntos) Realizar un gráfico de línea que muestre gbest en función de las iteraciones realizadas.
- (Valor 1 punto) Establecer el coeficiente de inercia w en 0, ejecutar el algoritmo y realizar observaciones/comentarios/conclusiones sobre los resultados observados.
- (Valor 1 punto) Graficar 3 boxplots usando gbest, uno para $w = 0.8$, $w = 0.5$ y el otro para $w = 0$.
- (Valor 1 punto) Explicar interpretando la información que proporcionan los 3 boxplots.

Ejercicio 4 (Valor 3 puntos)

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

$$\begin{cases} f_1(x_1, x_2, \dots, x_n) = 0 \\ f_2(x_1, x_2, \dots, x_n) = 0 \\ \vdots \\ f_n(x_1, x_2, \dots, x_n) = 0 \end{cases} \quad (2)$$




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

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

Utilizando la biblioteca pyswarm:

- (Valor 0.3 puntos) 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 en el informe el código fuente.

- b. (Valor 0.1 puntos) Transcribir en el informe los valores de x_1 y x_2 encontrados por el algoritmo.
- c. (Valor 0.1 puntos) Indicar la URL del repositorio en donde se encuentra el algoritmo PSO.
- d. (Valor 2.5 puntos) 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.

-  Subir la resolución de este TP al campus en formato .pdf.
-  Indicar en el .pdf la URL del repositorio en donde se encuentran los códigos fuentes.
-  Fecha límite de entrega: 07/10/2024.