Trabajo Práctico Nº2

Ejercicio 1

Escribir un algoritmo PSO para la maximización de la función:

$$f(x) = 2\sin(x) - \frac{x^2}{2}$$

En el intervalo de $0 \leq x \leq 4$ y que cumpla con las siguientes consignas:

A. Transcribir el algoritmo en Python con los siguientes parámetros: número de partículas = 2, máximo número de iteraciones 80, coeficientes de aceleración c1 = c2 = 2, peso de inercia w = 0.7.

B. Transcribir la solución óptima encontrada (dominio) y el valor objetivo óptimo (imagen).

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

D. Graficar usando matplotlib la función objetivo 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.

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

F. Sobre el gráfico anterior superponer (con colores diferentes) 5 gráficos de línea de gbest en función de las iteraciones realizadas para ejecuciones con 4, 10, 100, 200 y 400 partículas respectivamente.

G. Realizar observaciones/comentarios/conclusiones sobre los resultados obtenidos en el ítem anterior.

Ejercicio 2

Escribir un algoritmo PSO para la maximización de la función:

$$y = \sin(x) + \sin(x^2)$$

En el intervalo de $0 \leq x \leq 10$ y que cumpla con las siguientes consignas:

A. Transcribir 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.

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

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

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

E. Transcribir la solución óptima encontrada (dominio) y el valor objetivo óptimo (imagen).

F. Incrementar el número de partículas a 4, ejecutar la rutina, transcribir la solución óptima encontrada, transcribir el valor objetivo óptimo y realizar nuevamente los gráficos solicitados en C y D.

G. Incrementar el número de partículas a 6, ejecutar la rutina, transcribir la solución óptima encontrada, transcribir el valor objetivo óptimo y realizar nuevamente los gráficos solicitados en C y D.

H. Incrementar el número de partículas a 10, ejecutar la rutina, transcribir la solución óptima encontrada, transcribir el valor objetivo óptimo y realizar nuevamente los gráficos solicitados en C y D.

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

Ejercicio 3 Dada la siguiente función perteneciente a un paraboloide elíptico de la forma:

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

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) que cumpla con las siguientes consignas:

A. 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.

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

C. 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.

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

E. Transcribir la solución óptima encontrada (dominio) y el valor objetivo óptimo (imagen).

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

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

H. Realizar observaciones/comentarios/conclusiones comparando los resultados obtenidos sin pyswarm y con pyswarm.

Ejercicio 4 Mediante PSO es posible resolver **en forma aproximada** un sistema de ***n*** ecuaciones con ***n*** incógnitas clásico del tipo:

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

Por ejemplo, el siguiente es un sistema de 2 ecuaciones con 2 incógnitas (x1 y x2) que puede ser resuelto con PSO:

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

Utilizando la biblioteca pyswarm:

A. Escribir un algoritmo PSO con parámetros a elección (c1, c2, w, número de partículas, máximo número de iteraciones) que encuentre x1 y x2 para el sistema de ecuaciones anterior (3). Transcribir el código fuente.

B. Transcribir los valores de x1 y x2 encontrados por el algoritmo.

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

D. Realizar observaciones/comentarios/conclusiones sobre: (i) ¿Cómo eligió los limites superior e inferior de x1 y x2?. (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.