

# Optimization-Exercise

## Informe sobre la Evaluación del Algoritmo PSO para la Función de Mishra

### Introducción:

En este informe se evalúa el comportamiento del algoritmo **Particle Swarm Optimization (PSO)** aplicado a la minimización de la función **Mishra 2**. El objetivo es determinar la calidad de los puntos hallados, su proximidad a la solución conocida, y cómo varían los resultados con el incremento de la dimensión del problema y otros parámetros como el tamaño del enjambre y el número de iteraciones.

### Selección del Algoritmo PSO:

El **Particle Swarm Optimization (PSO)** fue seleccionado para este ejercicio de optimización debido a varias razones. En primer lugar, PSO es un algoritmo de optimización basado en la simulación del comportamiento de enjambres, lo que le permite equilibrar la exploración del espacio de búsqueda y la explotación de áreas prometedoras. Este balance es crucial en problemas multimodales como el presentado en la función **Mishra 2**, donde es importante evitar caer en óptimos locales.

PSO es altamente eficiente en problemas continuos, donde las soluciones pueden ser representadas como vectores, lo que es ideal para el problema de alta dimensionalidad planteado. Además, su simplicidad de implementación y capacidad de ajuste a través de parámetros como el tamaño del enjambre y el número de iteraciones lo hacen una opción versátil y ajustable a diferentes complejidades del problema.

Finalmente, se eligió PSO debido a su buena convergencia en experimentos preliminares y por su adaptabilidad a problemas donde el gradiente de la función objetivo no está disponible, como es el caso de la función Mishra 2, una función no diferenciable y multimodal.

### Algoritmo de Solución:

Se ha implementado el algoritmo PSO, ajustando el número de partículas en el enjambre y el número de iteraciones en diferentes pruebas. Los experimentos se realizaron para dimensiones del problema de (  $D = 10$  ), (  $D = 30$  ) y (  $D = 50$  ), variando tanto el tamaño del enjambre como el número de iteraciones.

### Resultados:

Se presenta una tabla resumen de los resultados obtenidos para las diferentes configuraciones. Los parámetros principales que se registraron incluyen el tamaño del enjambre, el número de iteraciones, la mejor posición encontrada, el valor de la función objetivo, el tiempo de ejecución y la proximidad al valor óptimo conocido (  $f(x^*) = 2$  ).

#### Dimensión 10:

Enjambre	Iteraciones	Mejor Punto	Mejor Valor	Tiempo Ejecución (s)	Cercanía
100	200	[1,1,...,1]	2	0.55	0
200	300	[1,1,...,1]	2	1.55	0

300	500	[1,1,...,1]	2	3.78	0
400	1000	[1,1,...,1]	2	10.09	0

En todas las configuraciones de la dimensión (  $D = 10$  ), el algoritmo PSO logró encontrar consistentemente el valor óptimo de (  $f(x) = 2$  ) en diferentes tiempos de ejecución. Los tiempos de ejecución aumentaron conforme se incrementaron las iteraciones y el tamaño del enjambre.

### Dimensión 30:

Enjambre	Iteraciones	Mejor Punto	Mejor Valor	Tiempo Ejecución (s)	Cercanía
100	300	[0.5, 0.7, ... 1]	1.21902	1.22	79.6
200	500	[0.7, 1, ... 1]	4.03508	4.04	41.5
300	700	[1, 1, ... 1]	8.4923	8.49	0.6

Para (  $D = 30$  ), se observó que con tamaños de enjambre y número de iteraciones más bajos, el valor de la función objetivo fue subóptimo, lo que indica que el enjambre no exploró lo suficiente el espacio de búsqueda. A medida que se incrementaron las iteraciones y el tamaño del enjambre, los resultados mejoraron significativamente en términos de proximidad al valor óptimo.

### Dimensión 50:

Enjambre	Iteraciones	Mejor Punto	Mejor Valor	Tiempo Ejecución (s)	Cercanía
100	300	[0.1, 1, ... 1]	1.44476	1.44	4.29603e+08
200	500	[1, 0.7, ... 1]	4.97626	4.98	3.2618e+08
300	1000	[0.9, 0.8, ... 1]	12.9003	12.90	3.08677e+06

Para (  $D = 50$  ), se encontraron mayores dificultades en alcanzar el valor óptimo global debido al crecimiento del espacio de búsqueda. Aunque el algoritmo PSO continuó mejorando con un mayor tamaño del enjambre y número de iteraciones, los valores de la función objetivo estuvieron lejos del óptimo global en varios experimentos, sugiriendo que el enjambre se estancó en óptimos locales.

### Evaluación de la calidad:

La calidad de los puntos hallados se evaluó en función de la proximidad al valor óptimo conocido (  $f(x^*) = 2$  ). Para la dimensión (  $D = 10$  ), el algoritmo PSO logró converger consistentemente al valor óptimo. Sin embargo, para (  $D = 30$  ) y (  $D = 50$  ), la cercanía a la solución óptima fue menor, indicando que el PSO tuvo dificultades para explorar completamente el espacio de búsqueda en dimensiones más altas.

### Variación de resultados con el incremento de la dimensión:

El tiempo computacional aumentó de manera significativa con el incremento de la dimensión del problema, el tamaño del enjambre, y el número de iteraciones. Para (  $D = 10$  ), el tiempo de ejecución fue bajo incluso en configuraciones con muchas iteraciones, mientras que para (  $D = 50$  ), los tiempos se incrementaron de manera considerable.

La calidad de las soluciones también disminuyó a medida que aumentó la dimensión, lo cual es típico en problemas de alta dimensionalidad donde el algoritmo PSO puede quedarse atrapado en óptimos locales.

## Conclusiones:

El algoritmo PSO demostró ser efectivo para problemas de menor dimensión ( $D = 10$ ), encontrando el óptimo global en todos los casos evaluados. Sin embargo, para problemas de mayor dimensión ( $D = 30$ ) y ( $D = 50$ ), la calidad de las soluciones decreció, y fue necesario ajustar el tamaño del enjambre y el número de iteraciones para obtener mejores resultados.

Se recomienda utilizar parámetros más agresivos (mayor tamaño del enjambre y número de iteraciones) en problemas de alta dimensionalidad para mejorar la exploración y evitar óptimos locales.

Este análisis ha mostrado que el rendimiento de PSO varía considerablemente con la dimensión del problema y requiere ajustes finos de sus parámetros para alcanzar soluciones óptimas en problemas complejos.