

PROYECTO FINAL

En muchos problemas reales de ingeniería, ciencias naturales y diseño computacional, las funciones objetivo presentan múltiples mínimos locales, regiones planas, valles estrechos o gradientes irregulares que dificultan la convergencia hacia el óptimo global. Para resolver este tipo de problemas, las metaheurísticas inspiradas en procesos naturales como los Algoritmos Genéticos, Particle Swarm Optimization, Evolución Diferencial, Recocido Simulado, entre otras se han convertido en herramientas fundamentales debido a su robustez y flexibilidad.

Sin embargo, ninguna metaheurística es universalmente superior. Cada una presenta ventajas particulares:

- Algunas exploran muy bien el espacio pero convergen lentamente.
- Otras realizan una explotación fina pero pueden estancarse en óptimos locales.
- Algunas son muy rápidas pero sensibles a sus parámetros.
- Otras son robustas pero costosas computacionalmente.

Por esta razón, en la última década ha surgido un enfoque prometedor: **la hibridización de metaheurísticas**.

La idea es **combinar dos algoritmos de manera secuencial o cooperativa** para aprovechar las fortalezas de ambos. Por ejemplo:

- un algoritmo fuerte en exploración → encuentra buenas regiones,
- un algoritmo fuerte en explotación → refina y converge con precisión.

El objetivo de este proyecto es que los estudiantes **implementen su propia hibridización entre dos metaheurísticas**, y evalúen su desempeño frente a una función particularmente desafiante: **la función de Michalewicz**.

La función de Michalewicz es un benchmark ampliamente utilizado para medir la capacidad de los algoritmos de optimización para escapar de mínimos locales y realizar exploración inteligente.

Para d dimensiones:

$$f(\vec{x}) = - \sum_{i=1}^d \sin(x_i) \left[\sin\left(\frac{ix_i^2}{\pi}\right) \right]^{2m}$$

La función es usualmente evaluada en el hipercubo $x_i \in [0, \pi]$ $\forall i = 1, 2, \dots, d$. Utilice $d = 5$.

Características importantes

- Altamente multimodal → muchos mínimos locales.
- Extrema irregularidad del paisaje de optimización.
- Óptimo global conocido en baja dimensión, útil para ver si el algoritmo se acerca.

Se deberá:

1. Seleccionar dos metaheurísticas vistas en clase.
2. Diseñar una hibridación entre ambas.
 - Secuencial
 - Por etapas
 - Iterativa
 - Cooperativa
 - Multiarranque híbrido
3. Implementar en Python el algoritmo híbrido.
4. Comparar el desempeño.
 - Metaheurística A
 - Metaheurística B
 - Híbrida
5. Analizar los resultados mediante:
 - Curva de convergencia
 - Estadística de 30 corridas (media, desviación estándar, mínimo, máximo).
 - Costo computacional

Ejemplos de recomendaciones de hibridación:

- DE → PSO
- GA → SA
- UMDA → PSO
- PSO → GA
- PSO → SA
- DE → SA
- DE → GA