

Métodos numéricos para la solución de problemas de optimización

Jackson Vera Pineda

1. Introducción

John von Neumann, uno de los mas grandes matemáticos del siglo XX, expresó: *"la optimización es el arte de obtener lo mejor de lo disponible"* [5]. En esta frase se captura la esencia de un campo que ha demostrado ser crucial para resolver problemas complejos y de diversas disciplinas. En la actualidad, los algoritmos de optimización permiten abordar tareas intratables, obteniendo soluciones (o aproximaciones de estas), incluso en espacios de búsqueda vastos y multidimensionales.

En este trabajo, nos centramos en la aplicación de algoritmos de optimización para resolver problemas en los que la función objetivo presenta una estructura compleja, como el siguiente caso:

$$f(x) = -\frac{1}{D} \sum_{i=1}^D x_i \sin(\sqrt{|x_i|}) \quad (1)$$

donde x_i son las variables de decisión y D es la dimensión del espacio en que opera el algoritmo. Esta función posee mínimos locales múltiples y discontinuidades en el comportamiento de sus derivadas, lo que la convierte en un desafío para técnicas tradicionales de optimización.

Para abordar el problema (1) exploramos y comparamos el desempeño de diversos algoritmos de optimización incluyendo **Recocido Simulado**, **Evolución Diferencial** y el método **L-BFGS-B** (Cuasi-Newton). Estos métodos se seleccionaron por su eficacia reconocida en escenarios con diferentes características geométricas y sus capacidades para manejar restricciones en las variables de decisión, como límites en los valores de x_i en el rango de $[-500, 500]$.

Históricamente, la búsqueda de soluciones óptimas en sistemas complejos ha evolucionado desde enfoques analíticos hacia métodos computacionales. Un ejemplo destacado es el desarrollo del algo-

ritmo de recocido simulado en los años 80, inspirado en los procesos de enfriamiento lento de materiales. La simulación de este fenómeno natural permitió resolver problemas combinatorios extremadamente difíciles [3]. De manera similar, el desarrollo de técnicas como evolución diferencial en la década de los 90 proporcionó herramientas poderosas para encontrar soluciones en espacios de búsqueda no diferenciables y de alta dimensionalidad [4].

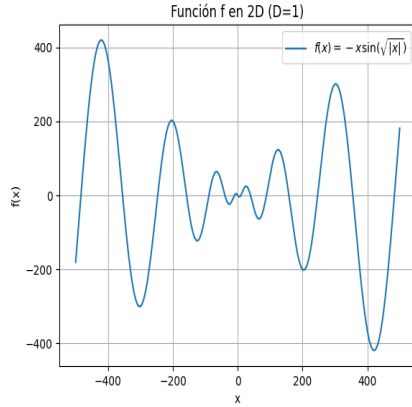
A lo largo de este trabajo, medimos y comparamos el costo computacional de los distintos métodos a medida que la dimensionalidad del problema crece. Nuestro objetivo es identificar qué algoritmo ofrece un balance óptimo entre precisión y costo temporal, permitiendo así una mejor toma de decisiones en problemas con restricciones complejas y funciones objetivo difíciles.

2. Desarrollo

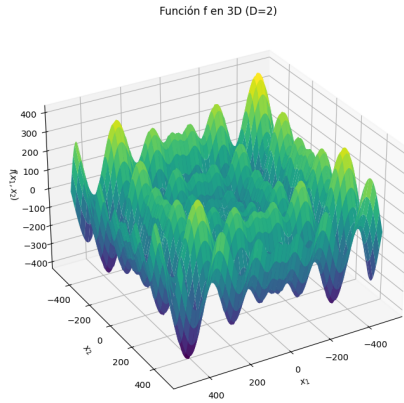
2.1. Función objetivo

En el presente, se toma la función número 128 propuesta en el estudio de literatura sobre funciones de referencia para problemas de optimización global [2], descrita en la sección anterior. En el estudio proveen datos de interés sobre (1), como mínimo $f(x^*) = -418,983$ situado en $x^* = \pm[\pi(0,5 + k)]^2$ que usaremos como puntos de referencia para analizar el desempeño de los algoritmos empleados.

Dicha función es una suma de términos que involucran relaciones no lineales (raíces cuadradas y senos), de ahí que presente comportamientos peculiares y múltiples mínimos locales como puede observarse en la figura 1 para valores de $D = 1, 2$.



(a)



(b)

Figura 1: Representación gráfica de la función f (1) para varios valores de D .

2.2. Métodos

Teniendo en cuenta las características de la función y los algoritmos encontrados en bibliografías clásicas, se utilizan los siguientes métodos para la resolución de f . Nótese que estos algoritmos son dependientes de la dimensión del problema, lo cual en algunos casos es un factor crítico en su rendimiento. En la figura 2 se presenta una tabla con un resumen comparativo de los algoritmos teniendo en cuenta su documentación, sirviendo de punto de partida, para su posterior análisis empírico.

2.2.1. Recocido Simulado

Es un algoritmo de optimización global que simula el proceso de enfriamiento de los metales. Requiere

un número grande de evaluaciones de la función objetivo y explora el espacio de búsqueda con cierto grado de aleatoriedad, permitiendo escapar de mínimos locales; fue introducido por Kirkpatrick et al. en 1983 [3] y es un enfoque bien conocido para evitar mínimos locales en problemas complejos. Su eficiencia temporal depende de la configuración del enfriamiento y del tamaño del espacio de búsqueda.

Costo asintótico: Aproximadamente $O(k \cdot D)$, donde k es el número de iteraciones, que a menudo debe ser grande para un espacio de búsqueda amplio.

2.2.2. Evolución Diferencial

Es un algoritmo de optimización global basado en la evolución de una población de soluciones candidatas, utilizando mutaciones y recombinaciones. Similar a Simulated Annealing, pero basado en métodos evolutivos; fue introducida por Storn y Price en 1997 [4]. Se ha reconocido como un método robusto para la optimización global, particularmente en problemas de alta dimensión y no diferenciables. La eficiencia temporal depende del número de evaluaciones de la función objetivo y del tamaño de la población.

Costo asintótico: Aproximadamente $O(P \cdot k \cdot D)$, donde P es el tamaño de la población y k es el número de iteraciones.

2.2.3. L-BFGS-B

Es un método cuasi-Newton que utiliza una aproximación a la matriz Hessiana (segunda derivada) de la función objetivo en lugar de calcularla directamente, con uso limitado de memoria, lo que puede ser computacionalmente más eficiente que el método de Newton. Este algoritmo es especialmente útil para problemas grandes y maneja restricciones de caja de manera natural. Fue desarrollado por Byrd et al. en 1995 [1].

Costo asintótico: Aproximadamente $O(n \cdot m \cdot D)$, donde n es el número de iteraciones y m el número de pares de vectores almacenados (generalmente pequeño).

2.3. Metodología

Para comparar el desempeño de cada método en nuestra función, se tuvieron en cuenta el tiempo de

Algoritmo	Costo Asintótico Aproximado	Comportamiento para Dimensiones Grandes (D)	Optimización Global	Optimización Local
Recocido Simulado	$O(k \cdot D)$	Aumenta rápidamente con D	Sí	No
Evolución Diferencial	$O(P \cdot k \cdot D)$	Escala mal con D	Sí	No
L-BFGS-B	$O(n \cdot m \cdot D)$	Muy eficiente para D grande	No	Sí

Figura 2: Resumen comparativo de los algoritmos empleados.

ejecución de cada uno, la evaluación mínima obtenida, y el vector de dicha evaluación para valores de $1 \leq D \leq 29$. Para cada valor de D , por cada algoritmo utilizado se simularon 30 ejecuciones, estas son aproximadamente $30 \cdot 30 \cdot 3 = 2700$ ejecuciones. De esta muestra se toma la media del tiempo por cada método y valor de dimensión, así como mejor mínimo encontrado, y punto en el que se alcanza.

En la figura 3 se puede observar el coste temporal de los métodos, nótese como a partir de $D = 10$ explota la curva que describe el rendimiento del algoritmo Evolución Diferencial, mostrando su mala escalabilidad. Nótese además el buen desempeño de Recocido Simulado (Simulated Annealing) y L-BGS-B para resolver nuestra problema.

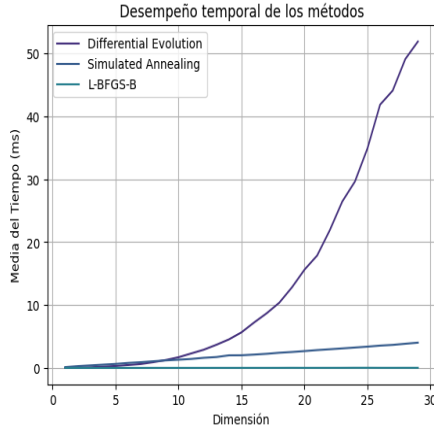


Figura 3: Comparación del tiempo medio de ejecución de cada método al aumentar la dimensión del problema.

Luego de observar el desempeño temporal surge, naturalmente, la pregunta ¿qué tanto se aproximan al mínimo en sus evaluaciones? Nótese en la figura 4 como, a pesar del buen desempeño de L-BGS-B, al aumentar la dimensión el mínimo encontrado se aleja de la repuesta, mostrando su sensibilidad a

mínimos locales; en cambio Recocido Simulado y Evolución Diferencial, se muestran robustos al aumento dimensional. Comportamineto similar se observa en la figura 5 de cuanto se alejan los puntos encontrados del punto en el cual f alcanza su mínimo (en el intervalo $-500 \leq x_i \leq 500$), teniendo como medida la raíz del error cuadrático medio.

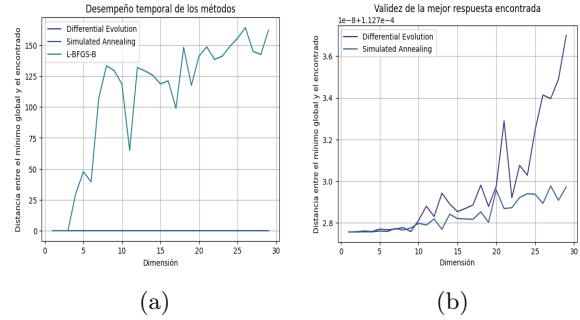


Figura 4: Comparación del error entre el mínimo encontrado y el mínimo global de la función $f(x^*) = -418,983$, al aumentar la dimensión del problema.

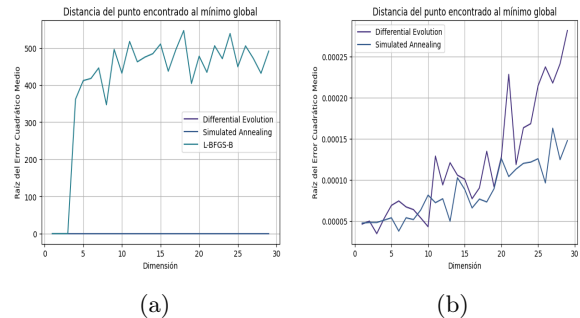


Figura 5: Comparación de la raíz del error cuadrático medio entre el punto de mínimo y el punto de mínimo global de la función $x^* = [\pi(0,5 + k)]^2$ para $k = 6$, al aumentar la dimensión del problema.

3. Conclusiones

Teniendo en cuenta el comportamiento de los algoritmos para el problema no lineal presentado, Recocido Simulado mostro tener un buen balance entre costo temporal y calidad de la solución, en comparación con las alternativas analizadas. Se probó lo costoso que puede llegar a ser Evolución Diferencial para problemas de alta dimencionalidad, criterio que se alinea con lo enunciado sobre este en la bibliografía, así como, aun siendo costoso mantiene resultados fiables. A diferencia de L-BFGS-B, cuya eficiencia es incomparable, sin embargo, su sensibilidad a mínimos locales juega un papel crítico en problemas con características similares a la función analizada, por lo que sus resultados deben ser validados e interpretados cuidadosamente.

Referencias

- [1] Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyu Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on scientific computing*, 16(5):1190–1208, 1995.
- [2] Momin Jamil and Xin-She Yang. A literature survey of benchmark functions for global optimisation problems. *International Journal of Mathematical Modelling and Numerical Optimization*, 4(2):150–194, 2013.
- [3] Scott Kirkpatrick, C Daniel Gelatt Jr, and Mario P Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.
- [4] Rainer Storn and Kenneth Price. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11:341–359, 1997.
- [5] John Von Neumann. *Mathematical foundations of quantum mechanics: New edition*, volume 53. Princeton university press, 2018.