

Método de búsqueda aleatorio para búsqueda de mínimos globales con tunelamiento

Brayhan Pérez Rendon*

Universidad de Antioquia

(FACOM)

(Dated: March 27, 2025)

El documento presenta un método de búsqueda global de mínimos que combina técnicas Monte Carlo con un mecanismo de tunelamiento para evitar estancamientos en mínimos locales. Se utiliza un algoritmo base en el que se lanzan N partículas, se seleccionan las M mejores y se redefine la zona de muestreo, complementado con un método de tunelamiento que reorienta la búsqueda hacia nuevos candidatos mediante una función $T(x)$. Las pruebas realizadas en seis funciones benchmark y el análisis de parámetros como η y el “*Size_{ite}*” muestran que, bajo las condiciones evaluadas, la variación de η no afecta significativamente la velocidad ni la calidad de la convergencia. En conjunto, el método se destaca por su robustez, rapidez y aplicabilidad en contextos exigentes, consolidándose como una alternativa prometedora para la optimización en problemas complejos.

I. INTRODUCCIÓN

La teoría de la optimización es un campo ampliamente estudiado durante varias décadas, en gran parte por la diversidad de aplicaciones que tiene en ámbitos industriales y comerciales. En este documento se analiza un algoritmo de optimización desarrollado e implementado en el presente trabajo. Aunque hasta el momento se ha encontrado escaso material específico sobre este algoritmo, existen implementaciones similares [1] que utilizan una idea comparable, aunque no idéntica, en la selección de las mejores partículas.

En trabajos previos, concretamente en la tesis de maestría de Jorge Hernán Cárdenas y en el proyecto de grado del autor, se buscó optimizar un híbrido en cuadratura, dispositivo electrónico que divide una señal entrante en dos señales de igual potencia pero con un desfase de 90° . Con ese contexto, se procedió a optimizar la estructura del aparato, penalizando comportamientos indeseados y priorizando las características deseadas. La función a optimizar en dichos trabajos era una función de penalización que variaba según la geometría del híbrido, donde el comportamiento electrónico se simulaba mediante el software [2], utilizando diferencias finitas temporales.

Partiendo de esos antecedentes, y basándose en el trabajo realizado en la tesis, se propuso una alternativa al algoritmo de optimización PSO [3], dando origen al algoritmo denominado School Fish Search [4]. Asimismo, se estudió el comportamiento de un algoritmo de optimización diseñado desde cero, basado en distribuciones aleatorias, al que se le incorporó una mejora posterior para evitar mínimos locales mediante un método de tunelamiento. Esta aproximación permite comprender en profundidad el funcionamiento de cada algoritmo, identificar qué técnicas resultan útiles en cada caso y evitar la aplicación mecánica de métodos sin considerar el tiempo que puede demorar cada paso del proceso.

El objetivo del algoritmo es ofrecer una alternativa más rápida y paralelizable, que se adapte a cada función e incorpore mejoras para incrementar la efectividad en la búsqueda de mínimos globales, evitando estancamientos gracias al método de tunelamiento [5].

II. IMPLEMENTACIÓN DEL ALGORITMO

A. algoritmo base

La idea del algoritmo es optimizar mediante métodos Monte Carlo. Es decir, se lanzan N partículas a la muestra y, de manera inteligente, se seleccionan las M partículas cuya evaluación en la función a optimizar es la menor, para luego volver a muestrear N partículas. Esta estrategia se adopta para reducir la cantidad de llamadas a la función a optimizar. Posteriormente, se repite el proceso de forma similar al algoritmo posterior.

- Despliegan N partículas en todo el muestreo
- Se escogen las M partículas con el mejor resultado
- Se ubica una nueva zona alrededor de la partícula con un valor V , ya sea una zona N -cúbica
- Se repite lo anterior en la nueva zona

B. Tunelamiento

La idea principal del tunelamiento fue desarrollada por A. V. Levy and A. Montalvo en 1985 [5] y tiene como objetivo ayudar a los algoritmos de optimización a converger hacia un mínimo global. Este tipo de problemas es muy común y resulta complejo en ciertos casos, por lo que este método asiste a aquellos algoritmos que se estancan en mínimos locales e impiden una búsqueda más amplia. El algoritmo construye una función que relaciona múltiples mínimos locales y selecciona el mejor de

* Also at Instituto de Física, Maestría en Física

ellos para reorientar la búsqueda alrededor de nuevos candidatos. Además, permite controlar qué tan lejos se desea explorar y puede interrumpir los ciclos al obtener el mejor resultado.

La función de tunelamiento se define como:

$$T(x) = \frac{f(x) - f(x^*)}{[(x - x^*)'(x - x^*)]^\eta} \quad (1)$$

donde x^* es el candidato anterior y η es una constante, cuya importancia se estudiará posteriormente.

Implementando el tunelamiento, el código sigue el siguiente orden:

- Se realiza una primera minimización y se escoge el primer candidato.
- Se inicia un ciclo en el que se buscan nuevos candidatos que se comparan con el candidato anterior mediante la función T
 - se buscan nuevos candidatos que se comparan con el anterior candidato con la función T
 - Si es un nuevo candidato, se selecciona para próximos ciclos
 - Si no, se minimiza la función T
- Si no se encuentra nuevo candidato se rompe todos los ciclos
- Si se encuentra candidato Se almacena como candidato general
- Se inicia un nuevo ciclo

III. MÉTODOS

Los algoritmos, tanto con como sin tunelamiento, serán puestos a prueba con las siguientes seis funciones: Rastigrin, Sphere, Ackley, Schwefel, Griewank y Rosenbrock. Cada una de estas funciones es N-dimensional y posee un mínimo global bien definido, además de presentar una cantidad considerable de mínimos locales. En el presente proyecto se busca evaluar el rendimiento del algoritmo utilizando el menor número posible de evaluaciones, ya que cada una puede demorar entre 5 y 15 minutos. Por ello, resulta fundamental que el algoritmo sea rápido en la convergencia y eficaz para evitar quedar atrapado en mínimos locales dentro de la región seleccionada.

Posteriormente, se analizarán las constantes numéricas, como lo es η , y el parámetro que define la reducción de la región en cada iteración del algoritmo base, denominado “*Size_ite*”. Este parámetro se establece con valores entre 0 y 1. Para estudiar el efecto del *Size_ite* se crea un arreglo con 100 elementos en dicho rango. Se simula cada valor de *Size_ite* en 20 ciclos, generando 10 partículas de forma aleatoria y seleccionando 2 de ellas. Luego, se calcula la distancia

al punto del mínimo global, definiéndose de esta forma un error en la proximidad del mínimo encontrado. Este proceso se repite 100 veces (debido a la aleatoriedad del método), registrándose el promedio de esos 100 ciclos, el *Size_ite* que obtuvo el mejor resultado y la desviación correspondiente. Con ello, se podrá observar el comportamiento del algoritmo frente a diferentes valores de *Size_ite* y cuantificar el error inherente a la aleatoriedad del método. Se espera obtener una gráfica de forma cóncava con algún pico, ya que reducir el tamaño demasiado poco impediría la convergencia, mientras que una reducción excesiva no daría tiempo suficiente entre ciclo y ciclo para seleccionar de manera adecuada una nueva región de muestreo.

Además, se realiza un muestreo similar utilizando 100 valores distintos de η comprendidos entre 1 y 20, con la misma intención de identificar los mejores resultados y, a su vez, analizar cómo η afecta la cantidad de ciclos y su relación con la rápida convergencia del algoritmo.

IV. RESULTADO Y DISCUSIÓN

A. Implementación base

En la figura 1 se ilustra el comportamiento del algoritmo base aplicado a dos funciones con diferentes niveles de dificultad. Se observa que, dependiendo de la función, el algoritmo puede estancarse en mínimos locales, especialmente en aquellas de alta dificultad (entendiéndose como funciones que poseen numerosos mínimos locales con profundidades similares a la del mínimo global).

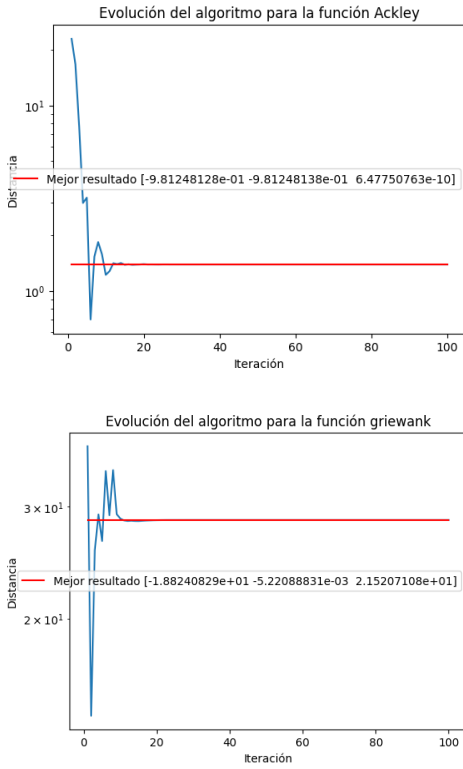


FIG. 1: La figura superior e inferior se generaron analizando la distancia euclidiana al mínimo global. La gráfica superior muestra una convergencia secuencial que alcanza un punto de estancamiento, mientras que la gráfica inferior ilustra una búsqueda aleatoria que recorre varios puntos distantes hasta detenerse en un mínimo local.

El comportamiento del parámetro *Size_ite* se ajusta a lo previsto figura 2, mostrando una dinámica similar para cada una de las funciones evaluadas. Además, este comportamiento depende en gran medida del número de partículas muestreadas y de las partículas seleccionadas como las mejores. En la mayoría de los casos, el promedio de las soluciones obtenidas a partir de 100 experimentos, realizados bajo las mismas condiciones de muestreo y selección, resulta similar entre función y función. Sin embargo, la dispersión de los resultados sí varía según la complejidad de la función: cuanto más compleja sea la función, mayor será la dispersión, lo cual es coherente, ya que alcanzar una solución adecuada resulta más difícil en esos casos. Esto es importante, ya que el estudio del parámetro *Size_ite* resulta independiente de la función específica, lo que facilita la obtención de resultados consistentes para casi cualquier función. Es decir, al poner en práctica el algoritmo, se puede anticipar que bajo ciertas condiciones específicas se obtendrán resultados predecibles.

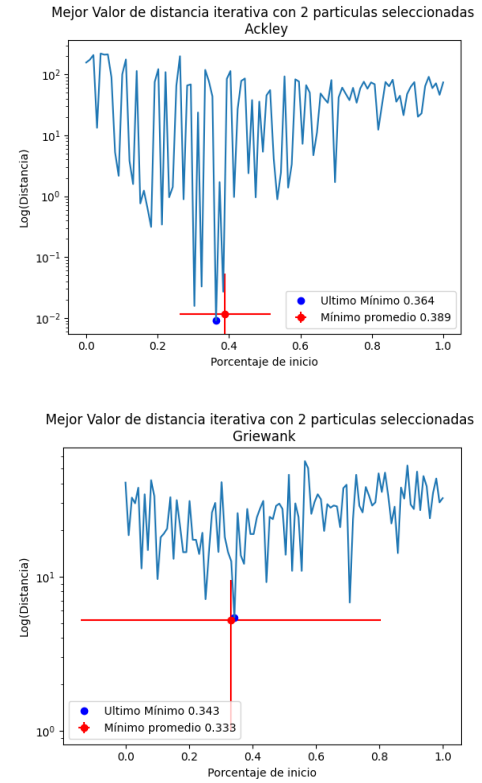


FIG. 2: Ambas figuras comparten las mismas condiciones iniciales. Se realizó una simulación para cada valor de *Size_ite*, obteniéndose el promedio de 100 experimentos junto con la desviación estándar de cada uno. Además, se graficaron dos desviaciones tanto en la distancia como en *Size_ite*, lo que evidencia claramente la similitud en los promedios, incluso entre funciones tan diferentes como Griewank y Ackley.

B. Tunelamiento

Utilizando un número constante de partículas para el muestreo y la selección, se realizaron 100 experimentos con la implementación del método de tunelamiento para analizar el comportamiento de η y determinar cuántos ciclos se requieren para obtener el resultado óptimo. Esto es fundamental para ejecutar la mayor cantidad de experimentos en el contexto de la radioastronomía.

La gráfica 3 presenta el desempeño de diferentes valores de η (entre 1 y 20), evidenciando que su variación no afecta significativamente la calidad del resultado, el cual se mantiene relativamente estable. Además, el número de ciclos necesarios para la convergencia también se muestra estable.

En resumen, el gráfico indica que η no tiene una relación directa ni con la velocidad de convergencia ni con la calidad final del resultado; es decir, en este caso particular, emplear valores de η de 1 o 1.5 produce resultados muy similares.

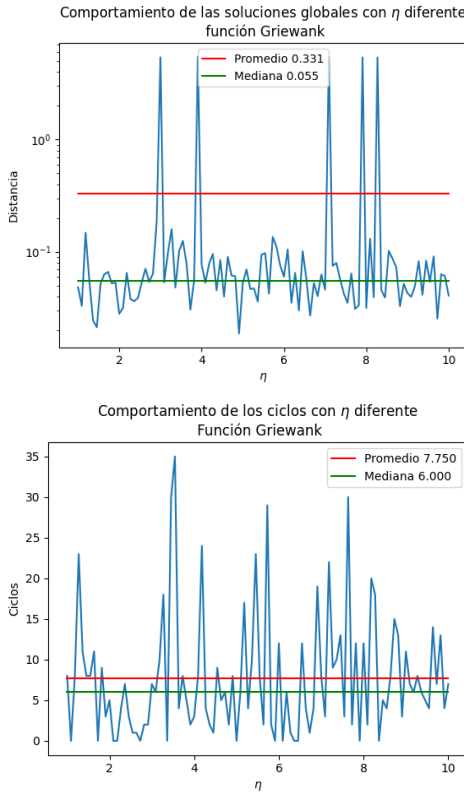


FIG. 3: Gráficos simulados con las mismas condiciones iniciales, con medias bien comportadas, quiere decir que descartando los pocos casos en que el algoritmo no tuvo un buen resultado, es un muy buen resultado, que solo toma 6 pasos en la mayoría de casos, con distancia menores a 0.1 al mínimo global

El gráfico indica que η no influye de forma directa ni en la velocidad de convergencia ni en la calidad del resultado. En este caso particular, emplear valores de η de 1 o 1.5 no modifica significativamente el resultado.

V. CONCLUSIÓN

El método resultó efectivo en combinación con el tunelamiento, ya que resolvió el problema de los mínimos locales. Se integra de manera óptima con el algoritmo base, logrando una convergencia rápida y eficaz hacia cualquier mínimo, ya sea cercano o lejano, dependiendo de la capacidad de búsqueda aleatoria. Aunque es necesario realizar pruebas adicionales, tal como está ideado, el algoritmo puede estudiarse de forma independiente y se garantiza que funcionará en la mayoría de los casos. Por ello, se presenta como un sólido candidato para ser utilizado como algoritmo de optimización alternativo y de fácil análisis.

-
- [1] R. Y. Rubinstein and D. P. Kroese, *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning* (Springer, 2004).
 - [2] Ansys, <https://www.ansys.com/> (2025), accedido: 25 de marzo de 2025.
 - [3] J. Kennedy and R. Eberhart, Particle swarm optimization, in *Proceedings of ICNN'95 - International Conference on Neural Networks*, Vol. 4 (1995) pp. 1942–1948 vol.4.
 - [4] C. Bastos-Filho, F. Lima Neto, A. Lins, A. Nascimento, and M. Lima, A novel search algorithm based on fish school behavior (2008) pp. 2646 – 2651.
 - [5] A. V. Levy and A. Montalvo, The tunneling algorithm for the global minimization of functions, *SIAM Journal on Scientific and Statistical Computing* **6**, 15 (1985), <https://doi.org/10.1137/0906002>.