

# Algoritmos genéticos

Los algoritmos genéticos (Genetic Algorithms - GA) son parte del campo de la computación evolutiva (EC), un campo que se inspira en el proceso biológico de la evolución darwiniana. Al imitar el proceso de selección natural y reproducción, los algoritmos genéticos pueden producir soluciones de alta calidad para diversos problemas relacionados con la búsqueda, la optimización y el aprendizaje. Al mismo tiempo, su analogía con la evolución natural permite a los algoritmos genéticos superar algunos de los obstáculos que encuentran los algoritmos tradicionales de búsqueda y optimización.

## ***Diferencias con los algoritmos tradicionales.***

Las características clave de los algoritmos genéticos que los distinguen de los algoritmos tradicionales son:

- La búsqueda genética se lleva a cabo sobre una población de soluciones candidatas (**individuos**) en lugar de un solo candidato. Por lo que, el algoritmo conserva un conjunto de individuos que forman la generación actual. Y en cada iteración del algoritmo genético crea la siguiente generación de nuevos individuos. Por el contrario, la mayoría de los otros algoritmos de búsqueda mantienen una única solución y la modifican iterativamente en busca de la mejor solución (ver figura 1). El algoritmo de descenso de gradiente, por ejemplo, mueve iterativamente la solución actual en la dirección de descenso más pronunciado, que se define por el negativo del gradiente de la función dada.

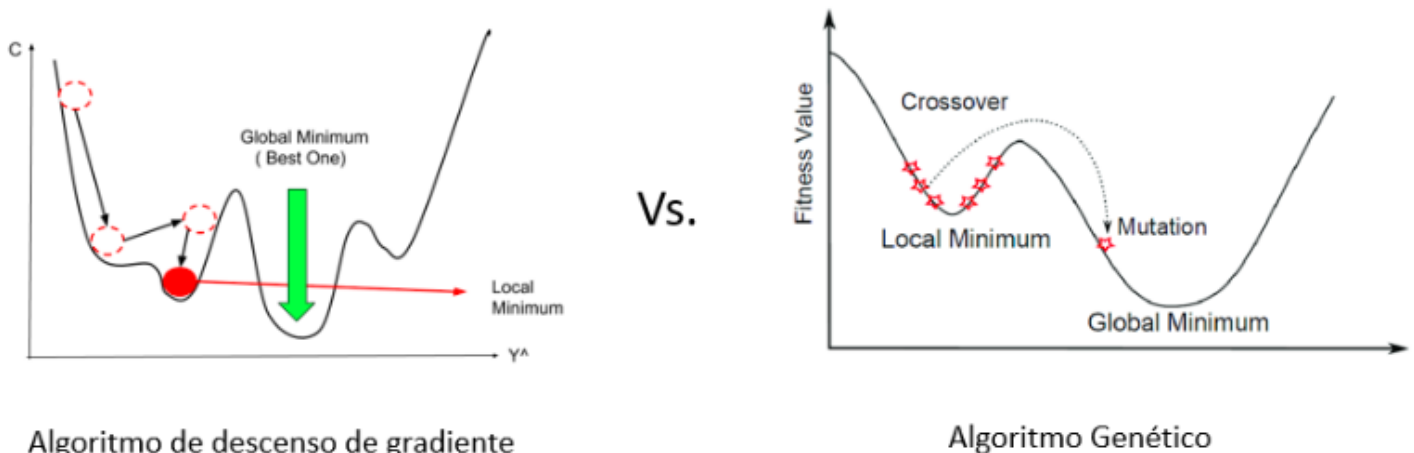
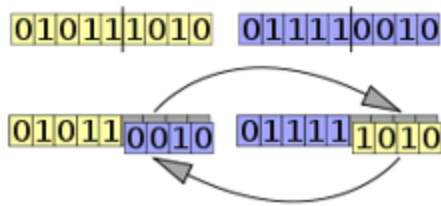
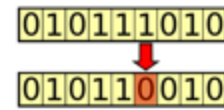


Figura 1. Comparación del algoritmo de Gradiente Descendente y GA.

- En lugar de operar directamente en soluciones candidatas, los algoritmos genéticos operan en sus representaciones (o codificación), a menudo denominadas **cromosomas**. Un ejemplo de un cromosoma simple es una cadena binaria de longitud fija. Los cromosomas nos permiten facilitar las operaciones genéticas de cruce y mutación. El **cruce** se implementa mediante el intercambio de partes cromosómicas entre dos padres, mientras que la **mutación** se implementa modificando partes del cromosoma.



Operación cruzada entre dos cromosomas (Crossover).



Una mutación es un cambio aleatorio en un gen para actualizar la población.

Figura 2. Operación de cruce y mutación para cromosomas binarios.

- La **función de fitness** representa el problema que nos gustaría resolver. El objetivo de los algoritmos genéticos es encontrar los individuos que obtienen la puntuación más alta cuando se calcula esta función para ellos. A diferencia de muchos de los algoritmos de búsqueda tradicionales, los algoritmos genéticos solo consideran el valor obtenido por la función de aptitud y no dependen de derivados ni de ninguna otra información. Esto los hace adecuados para manejar funciones que son difíciles o imposibles de diferenciar matemáticamente.
- Los GA son de **naturaleza probabilística**, la búsqueda basada en algoritmos genéticos no es aleatoria; en su lugar, utiliza el aspecto aleatorio para dirigir la búsqueda hacia áreas en el espacio de búsqueda donde hay una mejor oportunidad de mejorar los resultados.

### **Limitaciones de los Algoritmos Genéticos.**

- Es necesario crear una **representación adecuada** y definir la función de aptitud y la estructura cromosómica, así como los operadores de selección, cruce y mutación que funcionarán para un problema determinado. Esto a menudo puede resultar desafiante y llevar mucho tiempo, afortunadamente, los algoritmos genéticos ya se han aplicado a innumerables tipos de problemas, y muchos de estos se han estandarizado.
- El comportamiento de los algoritmos genéticos está controlado por un **conjunto de hiperparámetros**, como el tamaño de la población y la tasa de mutación. Al aplicar algoritmos genéticos al problema en cuestión, no hay reglas exactas para tomar estas decisiones. Sin embargo, este es el caso de prácticamente todos los algoritmos de búsqueda y optimización.
- Operan en poblaciones (potencialmente grandes) y la naturaleza repetitiva de los algoritmos genéticos puede ser **computacionalmente intensivo**, así como consumir mucho tiempo antes de alcanzar un buen resultado. Estos se pueden aliviar con una buena elección de hiperparámetros, implementando el procesamiento paralelo y, en algunos casos, almacenando en caché los resultados intermedios.
- Si la aptitud de un individuo es mucho mayor que la del resto de la población, puede duplicarse lo suficiente como para que se apodere de toda la población. Esto puede llevar a que el algoritmo genético se **atasque prematuramente** en un máximo local, en lugar de encontrar el global. Para evitar que esto ocurra, es importante mantener la diversidad de la población.

- **No hay solución garantizada**, el uso de algoritmos genéticos no garantiza que se encuentre el máximo global para el problema en cuestión. Sin embargo, este es el caso para cualquier algoritmo de búsqueda y optimización, a menos que sea una solución analítica para un tipo particular de problema.

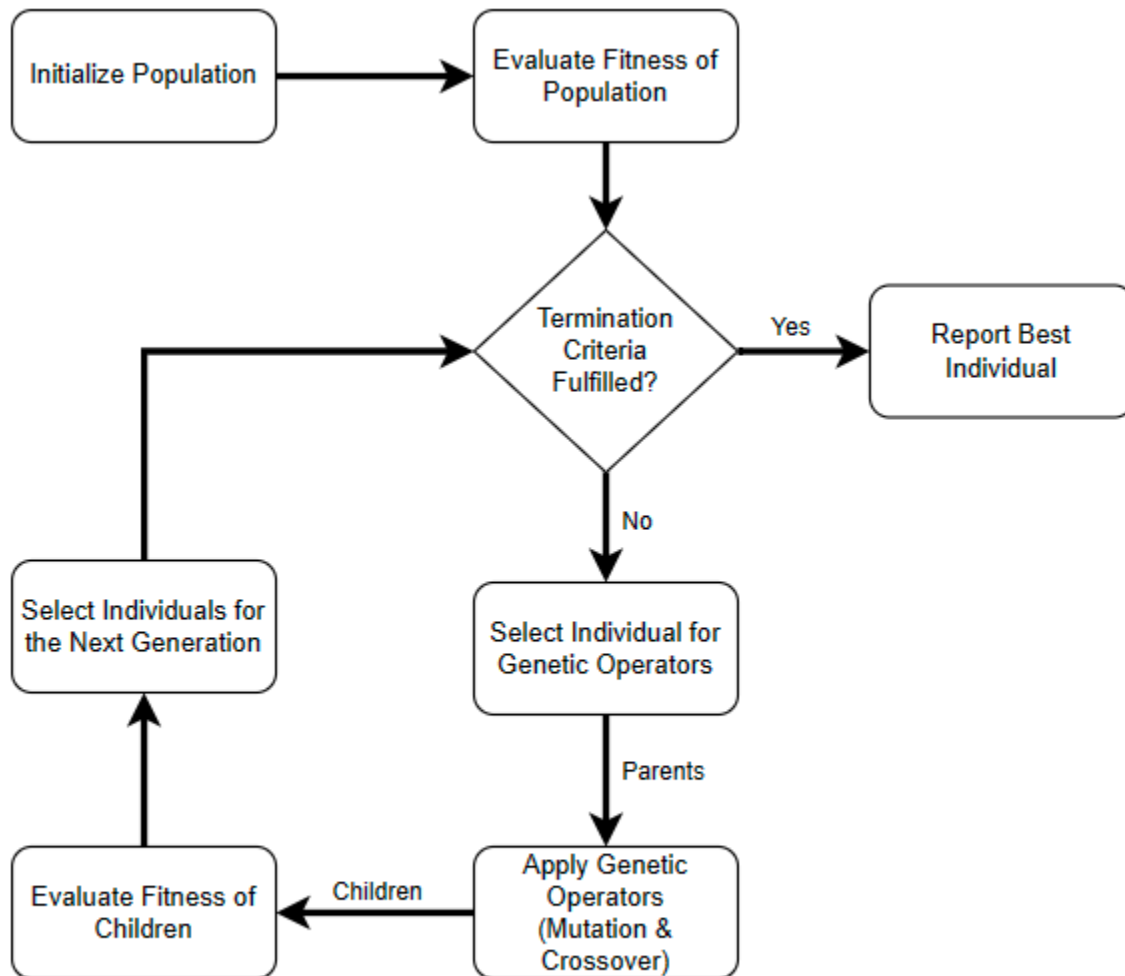


Figura 3. Diagrama de flujo generico para implementar un GA [1].

*Lecturas extra recomendadas.*

[1] A. Dockhorn and S. Lucas, "[Choosing Representation, Mutation, and Crossover in Genetic Algorithms](https://ieeexplore.ieee.org/document/9942691)," in IEEE Computational Intelligence Magazine, vol. 17, no. 4, pp. 52-53, 1 Nov. 2022, doi: 10.1109/MCI.2022.3199626.

<https://ieeexplore.ieee.org/document/9942691>