



**TECNOLÓGICO
NACIONAL DE MÉXICO**

Tecnológico nacional de México

Instituto Tecnológico de Culiacán

Tarea Extra Algoritmos Genéticos

Alumno: Ojeda Bermúdez Leonel Alonso

Profesor: ZURIEL DATHAN MORA FELIX

Ing. Sistemas Computacionales

14/02/25

Investigación sobre los Algoritmos Genéticos

Un algoritmo es una serie de pasos organizados que describe el proceso que se debe seguir para dar solución a un problema específico. Un algoritmo genético (AG para abreviar) es una técnica de programación inspirada en la reproducción de los seres vivos y que imita a la evolución biológica como estrategia para resolver problemas de optimización. Los algoritmos genéticos son una de las herramientas más importantes dentro de la inteligencia artificial, que es la resolución de problemas mediante el uso de programas informáticos que intentan imitar las capacidades cognitivas de los seres humanos y otros seres vivos.

Los AGs fueron formulados por el científico norteamericano John Holland en los años 1970 y fueron presentados en 1989 por David Goldberg, quien los promovió como una herramienta útil para la optimización en problemas complejos. La idea de los AGs surgió a partir del deseo de imitar los procesos biológicos que permiten la adaptación y supervivencia de los organismos en su entorno. Así, los AGs se basan en los principios de la evolución biológica, que incluyen la selección natural, el cruzamiento y la mutación.

1. Optimización y su Relación con los AGs

Para entender los algoritmos genéticos, es esencial comprender el concepto de optimización. La optimización es un proceso en el cual se busca la mejor solución posible entre un conjunto de alternativas. En muchos problemas de la vida real, existe una cantidad infinita de soluciones posibles, pero solo algunas de ellas cumplen con las condiciones de ser eficientes o deseables. La tarea de los algoritmos de optimización es encontrar la mejor opción, o lo que se denomina el **óptimo global**.

El desafío de la optimización radica en que, a menudo, existen soluciones que parecen buenas, pero que no son las mejores posibles. Estas soluciones se conocen como **óptimos locales**, y un buen algoritmo debe ser capaz de evitar caer en ellos para poder encontrar el óptimo global. Es aquí donde los algoritmos genéticos se destacan, ya que pueden explorar un espacio de soluciones de forma global y salir de los óptimos locales gracias a sus mecanismos de selección, mutación y cruzamiento.

2. El Proceso de Evolución en los AGs

Un algoritmo genético imita el proceso evolutivo de los seres vivos, donde las especies sobreviven y se reproducen si tienen características ventajosas. Este proceso involucra tres componentes clave: **selección**, **mutación** y **cruzamiento**. A través de estos operadores genéticos, un AG mejora la población de soluciones de manera progresiva, generando nuevas soluciones cada vez más cercanas al óptimo global.

Selección: En un AG, los individuos (posibles soluciones) son evaluados mediante una función de aptitud. Esta función asigna un valor numérico que mide cuán adecuada es una solución frente a los objetivos del problema. A los individuos con una aptitud más alta se les da más probabilidades de ser seleccionados para reproducirse y generar nuevas soluciones. La selección puede ser aleatoria, aunque comúnmente se emplean métodos como el **torneo** o la **ruleta**, donde los mejores individuos tienen más oportunidades de ser elegidos.

Cruzamiento: El cruzamiento, también conocido como recombinación, es el proceso de tomar dos soluciones de la población y combinar sus características para crear nuevos descendientes. Este proceso se asemeja a la reproducción sexual en los seres vivos, en la cual los padres transmiten sus genes a sus hijos. Existen diversas formas de cruzamiento, como el **cruzamiento de un punto**, **dos puntos** o **uniforme**, dependiendo de cómo se divide la información genética entre los padres.

Mutación: La mutación introduce cambios aleatorios en los cromosomas de los individuos. Al igual que en la biología, donde las mutaciones pueden generar variaciones en las características de un organismo, en los AGs las mutaciones permiten explorar soluciones diferentes y evitar que la población se quede atrapada en un óptimo local. La mutación suele aplicarse con una baja probabilidad, ya que, aunque puede generar buenas soluciones, también tiene el riesgo de generar soluciones no deseadas.

3. Pasos de un Algoritmo Genético

El ciclo de vida de un algoritmo genético se puede dividir en varios pasos. A continuación, describimos los principales componentes y fases de un AG:

3.1. Inicialización de la Población

El primer paso en un AG es la creación de una población inicial de individuos, que son soluciones potenciales al problema. Estas soluciones se generan de forma aleatoria, o bien mediante un conjunto de soluciones conocidas. La representación de cada individuo se realiza mediante un cromosoma, que es una estructura codificada que contiene los parámetros del problema. Los cromosomas pueden representarse en diversas formas, como cadenas binarias, números reales, cadenas de caracteres, etc.

3.2. Evaluación de la Aptitud

Una vez que la población inicial ha sido generada, cada individuo se somete a una evaluación mediante una **función de aptitud**. Esta función es la encargada de medir cuán buena es una solución en relación con el problema que se está resolviendo. La función de aptitud puede ser cualquier ecuación matemática que cuantifique el rendimiento de cada solución. Las soluciones que obtienen valores más altos en la función de aptitud son consideradas mejores que aquellas que tienen valores bajos.

3.3. Selección de los Padres

Después de la evaluación, se seleccionan los individuos que se van a reproducir. La selección es un paso crítico, ya que garantiza que las mejores soluciones se trasladen a la siguiente generación. Existen varios métodos de selección, como la **selección por torneo**, donde se elige al ganador de un pequeño concurso entre varios individuos, o la **selección por ruleta**, donde la probabilidad de ser seleccionado es proporcional a la aptitud de cada individuo.

3.4. Cruzamiento y Mutación

Una vez seleccionados los padres, se aplica el operador de cruzamiento para combinar sus características y crear nuevos individuos. Los descendientes generados pueden ser modificados por mutación, un proceso aleatorio que introduce variaciones en los cromosomas de los hijos. Este proceso asegura que no se pierda diversidad en la población, lo cual es crucial para explorar eficientemente el espacio de soluciones.

3.5. Evaluación y Reemplazo

Una vez generada la nueva población de descendientes, se evalúan de nuevo y se comparan con la población anterior. Los mejores individuos son seleccionados para sobrevivir y formar parte de la siguiente generación. Dependiendo del algoritmo, la nueva población puede reemplazar completamente a la anterior o solo una parte de ella.

3.6. Condiciones de Parada

El AG continuará ejecutándose hasta que se alcance una condición de parada. Esta condición puede estar relacionada con el número de generaciones o con un umbral de aptitud predefinido. En algunos casos, el AG puede detenerse si no se observan mejoras significativas en la aptitud durante varias generaciones consecutivas.

4. Codificación de los Cromosomas

Los cromosomas, que representan las soluciones en un AG, deben ser codificados de manera adecuada para el tipo de problema que se está resolviendo. La forma más común de codificación es la **codificación binaria**, en la que cada cromosoma se representa como una secuencia de bits (0s y 1s). Sin embargo, también existen otras formas de codificación, como la **codificación de valores reales** o la **codificación de números enteros**.

La codificación binaria es especialmente útil cuando se trata de problemas discretos, como la optimización de combinaciones. En este tipo de codificación, cada gen en el cromosoma representa un parámetro del problema, y su valor se define por una secuencia de bits. Por ejemplo, si un cromosoma tiene 5 bits, puede representar los valores entre 0 y 31, dependiendo de cómo se interpreten los bits.

5. Aplicaciones de los Algoritmos Genéticos

Los algoritmos genéticos se aplican a una gran variedad de problemas de optimización en diferentes campos. Algunas de las aplicaciones más comunes incluyen:

- **Optimización de funciones:** Mejorar el rendimiento de sistemas y procesos mediante la búsqueda de parámetros óptimos.
- **Problemas de optimización combinatoria:** Como el problema del viajante de comercio (TSP), en el cual se busca la ruta más corta que recorra un conjunto de ciudades.
- **Diseño de redes y sistemas:** Mejorar la eficiencia de redes de telecomunicaciones o sistemas de control.
- **Inteligencia artificial:** Aprendizaje automático y redes neuronales, donde los AGs pueden ayudar a ajustar parámetros.
- **Biología y medicina:** Modelado de procesos biológicos o el diseño de fármacos.

Los **algoritmos genéticos** (AG) se usan en muchas áreas como optimización, aprendizaje automático y simulaciones. Algunos ejemplos actuales incluyen:

1. AutoML y Optimización de Modelos de IA

- **AutoML-Zero (Google Research)** → Usa AG para diseñar arquitecturas de modelos de machine learning desde cero.
- **NEAT (NeuroEvolution of Augmenting Topologies)** → Evoluciona redes neuronales automáticamente.

2. Optimización en Juegos y Simulaciones

- **AlphaStar (DeepMind)** → Usó AG para mejorar su estrategia en StarCraft II.
- **Evolución en Juegos como Mario o Flappy Bird** → AG ha sido utilizado para crear bots que aprenden a jugar mejor con generaciones sucesivas.

3. Aplicaciones en Robótica y Automatización

- **Optimización de Movimiento en Robots** → Se han aplicado AG para mejorar la locomoción en robots autónomos.
- **Diseño de Estructuras Aerodinámicas** → Se usan para optimizar alas de aviones o drones.

4. Biomedicina y Descubrimiento de Fármacos

- **Optimización de Moléculas** → Algunas farmacéuticas usan AG para buscar la mejor combinación molecular en nuevos medicamentos.
- **Diagnóstico Médico** → Modelos evolucionan patrones de diagnóstico en enfermedades como cáncer o COVID-19.

6. Conclusión

Los algoritmos genéticos son una poderosa herramienta para resolver problemas complejos de optimización. A través de la simulación de procesos evolutivos, los AGs pueden explorar eficientemente grandes espacios de soluciones y encontrar resultados cercanos al óptimo global. Su flexibilidad y capacidad para evitar óptimos locales han hecho que los AGs sean utilizados en una amplia gama de aplicaciones, desde la ingeniería hasta la biología. A medida que los problemas se hacen más complejos, el uso de los algoritmos genéticos seguirá siendo una opción valiosa para la búsqueda de soluciones.

Referencias:

Conogasi. (2018). **Algoritmos genéticos**. Conogasi.org. <https://conogasi.org/articulos/algoritmos-geneticos/>

Goldberg, D. E. (1989). **Algoritmos genéticos en búsqueda, optimización y aprendizaje automático**. Addison-Wesley.

Holland, J. H. (1975). **Adaptación en sistemas naturales y artificiales**. University of Michigan Press.