

Tema 10-Computación evolutiva

Las técnicas que se usan en la computación evolutiva son:

- Algoritmos genéticos
- Programación genética
- Vida Artificial
- Otras técnicas como las Estrategias Evolutivas, que son similares a los algoritmos genéticos pero se centran en la adaptación continua en lugar de la optimización; el Hardware Evolutivo, que utiliza la evolución para diseñar circuitos electrónicos y sistemas hardware; los Learning Classifier Systems, que se basan en la evolución para aprender y mejorar la clasificación de patrones; la Optimización de Enjambre de Partículas, que imita el comportamiento de un enjambre para buscar soluciones óptimas; y las técnicas de computación inspiradas en la naturaleza como la Ant Colony Optimization, la Coevolución, los Sistemas Inmunes Artificiales y la Computación ADN.

Los algoritmos genéticos fueron diseñados específicamente para seguir la filosofía de la naturaleza en la solución de problemas. Como su nombre indica, se inspiran en la genética y la evolución natural para encontrar soluciones óptimas a problemas complejos. Estos algoritmos imitan el proceso de selección natural, donde las soluciones más aptas sobreviven y se reproducen para producir descendencia aún mejor adaptada.

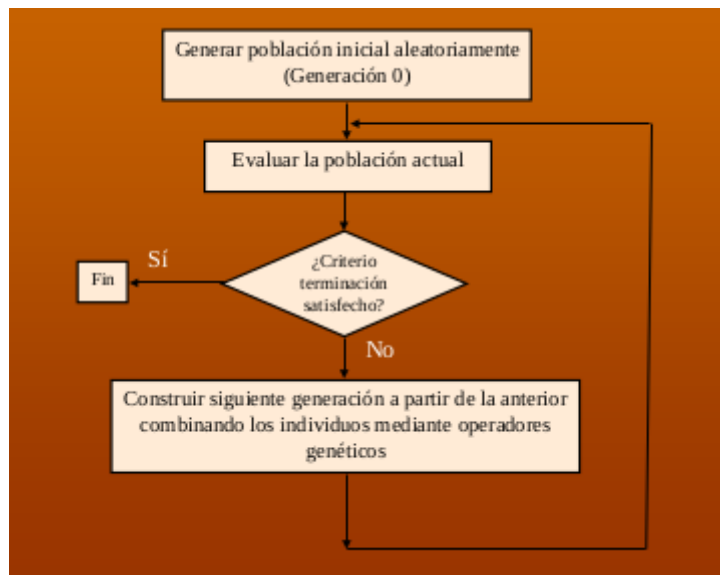
En lugar de trabajar con una única solución, los algoritmos genéticos trabajan con una población de soluciones, donde cada solución se representa por un cromosoma. Estos cromosomas se someten a operaciones de selección, cruzamiento y mutación, de manera similar a como funcionan los procesos biológicos. A través de estas operaciones, se generan nuevas soluciones que se evalúan en función de su aptitud para resolver el problema. Este proceso se repite varias veces hasta que se encuentra una solución que cumpla con los criterios de aptitud deseados.

Se suelen aplicar en problemas de optimización. Son eficaces en problemas de superficie compleja, es decir, problemas donde la solución óptima no es fácil de encontrar, debido a que hay múltiples mínimos locales y grandes espacios de búsqueda. También son útiles en problemas donde no hay una solución directa disponible, como los problemas NP-duros, que son problemas que no pueden ser resueltos en tiempo polinómico utilizando los algoritmos de búsqueda clásicos.

Componente de un AG

1. **Representación genética:** es la forma en que se codifican las soluciones del problema en una cadena de bits o caracteres. La representación debe ser adecuada para el problema en cuestión.
2. **Población inicial:** se requiere una población inicial de soluciones aleatorias para comenzar el proceso de búsqueda. La población debe ser lo suficientemente grande para cubrir el espacio de búsqueda.
3. **Función de evaluación:** se utiliza para medir la calidad de una solución en términos de su conveniencia o adaptación al problema. La función de evaluación debe ser definida de tal manera que las soluciones más aptas reciban una puntuación más alta.
4. **Operadores genéticos:** son las operaciones que se aplican a los individuos de la población para crear nuevos descendientes. Los operadores más comunes son la selección, el cruce y la mutación.
5. **Parámetros del algoritmo:** son valores que se ajustan para optimizar el rendimiento del algoritmo. Algunos de los parámetros más importantes son el tamaño de la población (N), la probabilidad de cruce (P_c), la probabilidad de mutación (P_m), etc. Estos valores deben ser cuidadosamente ajustados para cada problema específico.

Esquema de un AG



Generaciones y selección

Cada ciclo de funcionamiento se conoce como generación. Durante cada generación, se llevan a cabo tres procesos principales: selección, cruce y sustitución, y mutación. En la selección, se seleccionan los individuos más aptos de la población actual para que se reproduzcan y produzcan descendencia. El proceso de cruce implica tomar dos o más soluciones y combinarlos para producir nuevos individuos. La sustitución implica reemplazar los individuos menos aptos de la población actual con los nuevos individuos producidos a través del proceso de cruce.

La mutación se utiliza para introducir variabilidad en la población, lo que ayuda a evitar la convergencia prematura a soluciones subóptimas. Durante la mutación, se cambia aleatoriamente una parte de la cadena de bits o valores de un individuo.

El proceso de generación se repite hasta que se alcanza un nivel de error o un número de generaciones predefinido. En general, los algoritmos genéticos se utilizan para encontrar soluciones óptimas a problemas de optimización en los que la solución exacta no se puede obtener directamente o en un tiempo razonable.

La selección se realiza de manera probabilística, es decir, individuos con mejores valores de adaptación (mayor conveniencia) tienen más posibilidades de ser seleccionados. El objetivo de la selección es mantener y mejorar la calidad de la población en el tiempo.

Existen diversas técnicas de selección, y algunas de las más comunes son:

- **Ruleta:** se seleccionan individuos de manera aleatoria, pero con probabilidad proporcional a su adaptación relativa. Es decir, se le asigna una probabilidad a cada individuo de ser elegido, proporcional a su aptitud o valor de adaptación en comparación con la aptitud de los otros individuos. Luego, se realiza un sorteo y los individuos con mayor probabilidad tienen mayor probabilidad de ser elegidos.
- **Selección de torneo:** se eligen aleatoriamente un número determinado de individuos (por ejemplo, 2 o 3) y se selecciona el mejor de ellos como uno de los padres.
- **Basado en el rango:** se ordenan los individuos en función de su valor de adaptación, y se les asigna una probabilidad de selección proporcional a su posición en la lista ordenada. Los individuos en las posiciones superiores tienen una mayor probabilidad de ser seleccionados que los ubicados en posiciones inferiores.

Cruce

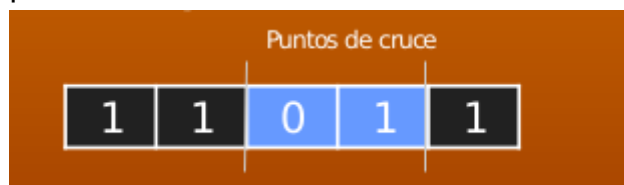
El cruce es uno de los operadores genéticos clave en los algoritmos genéticos, que se utiliza para generar nuevos individuos a partir de los padres seleccionados. Existen varias técnicas de cruce, y todas ellas involucran la combinación de partes de los cromosomas de los padres para crear uno o más descendientes:

- **Técnica de cruce básico de un punto:** se elige un punto aleatorio en el cromosoma de los padres y se intercambian las partes de los cromosomas antes y después del punto

para crear dos descendientes.



- **Técnica de cruce multipunto:** es similar a la técnica de cruce básico de un punto, pero en lugar de un solo punto, se eligen múltiples puntos de corte en el cromosoma de los padres.



- **Cruce uniforme:** implica la selección aleatoria de bits de los cromosomas de los padres para construir el cromosoma del descendiente.



- **Cruce segmentado:** se seleccionan segmentos aleatorios de los cromosomas de los padres y se combinan para crear los descendientes.

En los problemas de permutación, se utilizan técnicas especiales de cruce, como el cruce de mapeo parcial y el cruce de orden. En el cruce de mapeo parcial, se selecciona una sección del primer padre y se mapea en el segundo padre para crear el descendiente. En el cruce de orden, se selecciona una subsecuencia ordenada de genes de un padre y se coloca en el mismo orden en el descendiente.

La elección de la técnica de cruce y sus parámetros afecta la diversidad y calidad de la población de la siguiente generación.

Mutación

La mutación es un operador genético que se utiliza en algoritmos genéticos y otras técnicas de computación evolutiva. Consiste en hacer cambios aleatorios en el genoma de un individuo en una posición determinada. El objetivo de la mutación es introducir variabilidad en la población, para que no todos los individuos tengan los mismos genes y se puedan explorar diferentes soluciones.

La mutación es importante en los algoritmos genéticos porque permite escapar de mínimos/máximos locales y, por lo tanto, explorar más completamente el espacio de búsqueda. También contribuye a la diversidad genética de la población, lo que aumenta la probabilidad de encontrar una solución óptima.

La frecuencia de mutación es un parámetro que se puede ajustar para controlar la tasa de cambio genético en la población. Si la frecuencia de mutación es demasiado baja, la población puede estancarse en un óptimo local y no explorar nuevas soluciones. Si la frecuencia de mutación es demasiado alta, puede producirse una degradación genética y perderse la información valiosa. Por lo tanto, encontrar una frecuencia adecuada de mutación es una tarea importante en la configuración de un algoritmo genético.

Existen varias técnicas de mutación. La mutación de bit es la más común y consiste en cambiar un bit aleatorio en el genoma del individuo. La mutación multibit es similar, pero cambia varios bits en lugar de uno solo. La mutación de gen implica cambiar todo el valor de un gen en particular, mientras que la mutación multigen cambia varios genes al mismo tiempo.

La mutación de intercambio es una técnica utilizada en problemas de permutación, en los que la solución es una secuencia ordenada de elementos. En este caso, la mutación de intercambio selecciona dos posiciones aleatorias en la secuencia y las intercambia. La mutación de barajado es similar, pero en lugar de intercambiar dos posiciones, se baraja un subconjunto aleatorio de posiciones en la secuencia.

Exploración y explotación

El balance entre la exploración y la explotación en los algoritmos genéticos es importante para lograr una búsqueda efectiva de la solución óptima a un problema. La exploración se refiere a la capacidad del algoritmo para buscar en diferentes áreas del espacio de solución, lo que aumenta la posibilidad de encontrar una solución óptima. La explotación se refiere a la capacidad del algoritmo para mejorar una solución que ya es prometedora, moviéndose hacia un óptimo local o global.

El cruce y la mutación son los principales operadores utilizados para equilibrar la exploración y la explotación en los algoritmos genéticos. El cruce tiende a generar soluciones similares a los padres, mientras que la mutación introduce variaciones aleatorias en los descendientes. La exploración se promueve a través de la mutación, ya que esta operación puede generar soluciones radicalmente diferentes a las presentes en la población actual. La explotación se promueve a través del cruce, ya que esta operación permite mantener ciertas características prometedoras de los padres en la descendencia.

En problemas como la búsqueda de máximos de una función, se requiere un equilibrio entre la exploración y la explotación para evitar quedarse atrapado en un máximo local subóptimo. En el ajedrez, por ejemplo, la exploración y la explotación están estrechamente relacionadas con

la estrategia a utilizar. En general, la exploración se asocia con una estrategia más agresiva, mientras que la explotación se asocia con una estrategia más conservadora.

Sustitución

La sustitución en un algoritmo genético tiene como objetivo mantener el tamaño de la población. Es decir, una vez que se han aplicado los operadores genéticos de selección, cruce y mutación a la población actual, se necesita un mecanismo para reemplazar a los individuos menos aptos por otros nuevos. Esto se hace mediante un proceso de selección de individuos de la población actual que serán reemplazados por los descendientes generados.

Existen diferentes tipos de gestión de población en los algoritmos genéticos, siendo la generacional la más común. En este tipo de sustitución, se genera una nueva población a partir de los descendientes generados por los operadores genéticos, y se seleccionan los individuos más aptos para la siguiente generación. Los individuos que no son seleccionados son eliminados.

Existen diferentes tipos de gestión de población en los algoritmos genéticos. Uno de ellos es el llamado "generacional", que consiste en generar una nueva población completa en cada generación. Para ello, se realiza una selección de individuos mediante técnicas como la selección por ruleta, y luego se aplican operadores de cruce y mutación para crear nuevos individuos. Finalmente, se reemplaza la población anterior con la nueva.

Otro tipo de gestión de población es el "steady-state", que en lugar de generar una nueva población completa, simplemente reemplaza algunos individuos de la población actual. Para ello, se pueden utilizar diferentes criterios de selección, como seleccionar los peores individuos, los padres de los nuevos individuos generados, individuos parecidos a los nuevos, o aplicar elitismo, que consiste en mantener los mejores individuos de la población actual sin modificarlos.

Función de evaluación

No se garantiza que las técnicas de CE encuentren una solución óptima. En general, las técnicas de CE comienzan con una distribución inicial uniforme sobre el espacio de soluciones. A medida que se realizan las operaciones de selección, cruce y mutación, la población evoluciona hacia soluciones mejores. Si todo funciona según lo previsto, las soluciones evolucionarán rápidamente hacia la zona del óptimo global.

Sin embargo, la evolución puede quedar atrapada en un máximo local, que es una solución que es óptima solo dentro de una pequeña región del espacio de búsqueda, pero no es la mejor solución global. Para superar este problema, se pueden combinar las técnicas de CE con técnicas de optimización local, como el escalado de colinas (Hill climbing), que buscan mejorar una solución localmente.

Criterio de paro

El criterio de paro es un método utilizado en la computación evolutiva para decidir cuándo se detendrá el algoritmo. En general, se busca un equilibrio entre el tiempo de ejecución y la calidad de la solución encontrada.

Un criterio común de paro es cuando un porcentaje alto de la población converge a un valor. Esto significa que los individuos en la población son muy similares entre sí, lo que indica que se ha alcanzado un estancamiento en la búsqueda. En este caso, el algoritmo puede detenerse y devolver la mejor solución encontrada hasta el momento.

Si la solución encontrada no es lo suficientemente buena, se pueden aplicar diferentes técnicas para inyectar diversidad genética en la población. Una forma común de hacerlo es tomar una pequeña proporción de la población y aplicar mutaciones para generar nuevos individuos con características diferentes. También es posible reemplazar completamente la población con una nueva generación de individuos generados aleatoriamente.

El criterio de paro debe elegirse cuidadosamente para garantizar que se obtenga una solución satisfactoria en un tiempo razonable. Si el criterio de paro se establece demasiado pronto, es posible que el algoritmo no haya tenido tiempo suficiente para explorar todo el espacio de búsqueda y encontrar la mejor solución. Por otro lado, si el criterio de paro se establece demasiado tarde, el algoritmo puede ejecutarse durante demasiado tiempo sin producir mejoras significativas en la solución.

Criterios sobre los parámetros

- **Tamaño de población:** se busca encontrar un equilibrio entre una población muy pequeña, que puede limitar la capacidad de exploración del espacio de búsqueda, y una población muy grande, que puede ser computacionalmente costosa. Normalmente se elige una población de tamaño fijo, que puede oscilar entre varias decenas a cientos de individuos, aunque también existen esquemas de poblaciones de tamaño variable.
- **Porcentaje de cruce y mutación:** un porcentaje demasiado alto de cruces puede llevar a una rápida convergencia, pero puede hacer que se pierda la diversidad genética en la población. Por otro lado, un porcentaje demasiado alto de mutación puede hacer que la población no converja lo suficientemente rápido. Normalmente se utiliza un porcentaje de cruce elevado, alrededor del 90%, y un porcentaje de mutación del 5-10%. Sin embargo, estos valores pueden variar según las fases del algoritmo, por ejemplo, al inicio y al final de la simulación se puede utilizar un porcentaje más alto de mutación para explorar mejor el espacio de búsqueda.