Variantes y Extensiones

Los Algoritmos Genéticos (AG) básicos son poderosas herramientas de optimización, pero a lo largo del tiempo, han surgido numerosas variantes y extensiones para abordar problemas más complejos y desafiantes.

Algoritmos genéticos paralelos

Los algoritmos genéticos pueden ser paralelizados para ganar en performance y escalabilidad. Pueden ser fácilmente implementados en redes de computadoras heterogéneas o en mainframes paralelos.

La forma de paralelización de un algoritmo genético depende de los siguientes elementos:

- Cómo el fitness es evaluado y la mutación aplicada.
- Si son usadas poblaciones simples o múltiples subpoblaciones.
- Si se utilizan múltiples poblaciones, cómo los individuos son intercambiados.
- Cómo es aplicada la selección (global o local).

Dependiendo de cómo cada uno de estos elementos es implementado, se pueden obtener variados métodos de paralelización, los cuales pueden ser clasificados en las siguientes clases:

1. **Paralelización maestro-esclavo**:
    - **Sincrónica**: En este enfoque, un maestro coordina el proceso y asigna trabajos a esclavos. Los esclavos realizan la evaluación de fitness de las soluciones en paralelo y luego envían los resultados al maestro para la evolución de la población.
    - **Asincrónica**: A diferencia del enfoque sincrónico, aquí los esclavos pueden realizar evaluaciones de forma independiente sin esperar al maestro. Los resultados se actualizan en la población en tiempo real.
2. **Subpoblaciones estáticas con migración**:
    - Se divide la población en varias subpoblaciones, y cada una de ellas se ejecuta en un procesador o hilo independiente. De manera periódica, los individuos de las subpoblaciones pueden migrar entre sí para intercambiar información genética.
3. **Subpoblaciones estáticas solapadas**:

- Similar al enfoque anterior, pero en este caso, los individuos pueden pertenecer a más de una subpoblación al mismo tiempo. Esto permite una mayor exploración del espacio de soluciones.

4. **Algoritmos genéticos masivamente paralelos**:
    - En este enfoque, se utilizan muchos procesadores o hilos para evaluar múltiples soluciones en paralelo. Esto es especialmente útil para problemas con un gran número de soluciones posibles.
5. **Subpoblaciones dinámicas solapadas**:
    - Aquí, las subpoblaciones pueden cambiar dinámicamente durante la ejecución del algoritmo. Los individuos pueden migrar entre subpoblaciones de acuerdo con ciertas reglas o criterios.
6. **Algoritmos genéticos paralelos de estado seguro (steady-state)**:
    - En lugar de reemplazar toda la población en cada generación, estos algoritmos reemplazan solo a un subconjunto de la población en cada iteración, lo que facilita la convergencia hacia una solución óptima.
7. **Algoritmos genéticos paralelos desordenados (messy)**:
    - Estos algoritmos utilizan una representación desordenada de los individuos, lo que significa que los genes pueden estar en cualquier posición. Esto permite una mayor diversidad en la población y una exploración más efectiva del espacio de soluciones.

Algoritmos genéticos multiobjetivo

Los Algoritmos Genéticos Multiobjetivo (AGMO) son una extensión de los algoritmos genéticos tradicionales que se utilizan para resolver problemas de optimización en los que existen múltiples objetivos a optimizar simultáneamente.

Fundamentos de los AGMO

1. Problema de Optimización Multiobjetivo

En los problemas de optimización multiobjetivo, se busca encontrar un conjunto de soluciones que optimicen múltiples objetivos, los cuales a menudo son conflictivos entre sí. Por ejemplo, en la optimización de un producto, se pueden tener objetivos como minimizar el costo y maximizar la calidad, y no es posible encontrar una solución que sea óptima para ambos objetivos de manera simultánea.

2. Representación de Soluciones

Las soluciones en un AGMO se representan generalmente como vectores en un espacio multidimensional, donde cada dimensión corresponde a uno de los objetivos a optimizar. Por ejemplo, si tenemos dos objetivos (costo y calidad), una solución se representaría como un vector (x, y), donde x es el costo y y es la calidad.

3. Dominancia y Frente de Pareto

En AGMO, se utiliza el concepto de dominancia para evaluar la calidad relativa de las soluciones. Una solución A domina a otra solución B si es igual o mejor en al menos un objetivo y mejor en al menos un objetivo. El conjunto de soluciones no dominadas se conoce como el frente de Pareto, y contiene las soluciones óptimas desde un punto de vista multiobjetivo.

Ejemplo:

Supongamos que tenemos dos objetivos: maximizar el valor (V) y minimizar el costo (C). Si tenemos dos soluciones A con (V=10, C=5) y B con (V=12, C=4), entonces A domina a B, ya que A es igual o mejor en ambos objetivos.

Estrategias de Resolución

Métodos de Agregación

Los métodos de agregación combinan los objetivos en uno solo para convertir el problema multiobjetivo en un problema de optimización monoobjetivo. El método más común es la suma ponderada.

Fórmula:

Maximizar Z(x) = w1 * f1(x) + w2 * f2(x) + ... + wk * fk(x), donde w1, w2, ..., wk son los pesos asignados a cada objetivo.

Métodos de Rango

Los métodos de rango asignan a cada solución un valor de rango basado en su dominancia en relación con otras soluciones. Luego, se busca el conjunto Pareto de soluciones de rango más bajo.

Algoritmos genéticos con restricciones

Los Algoritmos Genéticos con restricciones se utilizan para resolver problemas de optimización en los que las soluciones factibles deben satisfacer un conjunto de restricciones. Estas restricciones pueden ser igualdades o desigualdades y pueden ser lineales o no lineales.

Función de Aptitud Penalizada

En lugar de utilizar una función de aptitud convencional, se utiliza una función de aptitud penalizada que penaliza las soluciones que no cumplen con las restricciones. La función de aptitud penalizada se define de la siguiente manera:

$$F(x) = f(x) + \sum_{i}^{k} \lambda_i * h_i(x)$$

Donde:

$F(x)$ es la función de aptitud penalizada.

$f(x)$es la función objetivo que se busca minimizar o maximizar.

λ_i son factores de penalización que determinan la magnitud de la penalización por violar cada restricción.

$h_i(x)$ son las funciones que evalúan si se cumplen las restricciones ($h_i(x)$≤0 para restricciones de desigualdad y $h_i(x)$=0 para restricciones de igualdad).

Operadores Genéticos Modificados

En Algoritmos Genéticos con Restricciones, los operadores genéticos como la selección, el cruce y la mutación se adaptan para garantizar que las soluciones generadas sigan siendo factibles. Por ejemplo, en la selección, se pueden preferir individuos que cumplan con más restricciones. En el cruce, se deben aplicar operadores que combinen soluciones de manera que las restricciones se mantengan válidas. En la mutación, se deben realizar cambios de manera que las restricciones no se violen.

Ejemplo

Supongamos que tenemos el siguiente problema de optimización con restricciones:

$$f(x) = 2x_1 + 3x_2$$

Sujeto a las siguientes restricciones:

$x_1 + 2x_2 \leq 6$

$2x_1 + x_2 \leq 8$

$x_1, x_2 \geq 0$

Podemos representar una solución x como un par ordenado x_1, x_2 y utilizar Algoritmos Genéticos con Restricciones para encontrar la solución óptima que maximice $f(x)$ y cumpla con todas las restricciones.

Enfoques Avanzados

Estrategias de Penalización

Las estrategias de penalización involucran ajustar los factores de penalización λ_i de manera dinámica durante la ejecución del algoritmo para encontrar un equilibrio entre la búsqueda de soluciones factibles y la mejora de la función objetivo.

Algoritmos Meméticos con Restricciones

Los Algoritmos Meméticos combinan Algoritmos Genéticos con búsquedas locales, lo que puede ser especialmente útil en problemas con restricciones para mejorar la calidad de las soluciones.

Estrategias evolutivas y programación genética

Las Estrategias Evolutivas (EE) son una variante de los Algoritmos Genéticos que se enfocan en la adaptación de la estrategia de búsqueda en sí misma, en lugar de solo modificar las soluciones individuales. Esto significa que las EE ajustan parámetros de búsqueda, como las tasas de mutación, en función de la eficacia de las soluciones encontradas previamente.

Características Clave de las Estrategias Evolutivas:

- **Control de Parámetros**: Las EE ajustan las tasas de mutación y otros parámetros durante la ejecución para adaptarse a la topología del espacio de búsqueda.
- **Diversidad**: Mantienen la diversidad en la población de soluciones, lo que puede ser beneficioso en problemas con múltiples óptimos locales.

Supongamos que estamos utilizando Estrategias Evolutivas para ajustar hiperparámetros de una red neuronal profunda. La estrategia adaptaría dinámicamente la tasa de aprendizaje y la probabilidad de dropout en función del rendimiento de la red en conjuntos de validación. Si el rendimiento mejora, las tasas de aprendizaje y dropout aumentan ligeramente; si empeora, disminuyen.

sigma_nueva = sigma_actual * exp(tau * N(0,1))

Donde **sigma_nueva** es el nuevo tamaño de paso, **sigma_actual** es el tamaño de paso actual, **tau** es un parámetro de ajuste y **N(0,1)** es una muestra aleatoria de una distribución normal estándar.

Programación Genética

La Programación Genética (PG) es otra extensión de los Algoritmos Genéticos que se utiliza para evolucionar programas informáticos en lugar de estructuras de datos. En la PG, las soluciones son representadas como árboles sintácticos que representan programas, y se aplican operadores genéticos para evolucionar estos programas hacia soluciones mejores.

Características Clave de la Programación Genética:

- **Representación de Programas**: Utiliza árboles sintácticos para representar programas, lo que permite la creación de soluciones más complejas.
- **Evolución de Algoritmos**: En lugar de encontrar soluciones a problemas, la PG busca encontrar algoritmos o programas que resuelvan problemas específicos.

Ejemplo de Aplicación:

Imaginemos que queremos encontrar un algoritmo que resuelva un problema de clasificación de imágenes. En lugar de ajustar parámetros de una red neuronal existente, la PG evolucionaría programas informáticos completos, incluyendo la lógica de procesamiento de imágenes y la decisión de clasificación. Los operadores genéticos permitirían la creación y adaptación de estos programas a lo largo de las generaciones.