

Aluno: Henrique Cesar Silva Soares

GCC128 – Inteligência Artificial Prof. Ahmed Ali Abdalla Esmin Relatório

Técnico – Algoritmos Genéticos

1. Definição e apresentação do algoritmo genético

O Algoritmo Genético (AG) é uma técnica de busca e otimização inspirada no processo de seleção natural, sendo amplamente utilizada na área de Inteligência Artificial para resolver problemas complexos que envolvem grandes espaços de busca. Baseado nos princípios da evolução biológica, o AG utiliza mecanismos como seleção, crossover (recombinação) e mutação para evoluir soluções candidatas ao longo de sucessivas gerações, buscando aprimorar sua aptidão com relação a uma função objetivo.

Neste trabalho, o Algoritmo Genético é empregado para encontrar o valor máximo da função real $f(x) = x^2 - 3x + 4$, definida no intervalo $X = [-10, +10]$. O objetivo é identificar o valor de x dentro desse intervalo que resulta no maior valor possível da função. Para isso, cada indivíduo na população representa uma solução candidata, codificada de forma binária, inteiro ou em ponto flutuante, dependendo da implementação adotada. O processo evolutivo inicia-se com a geração de uma população inicial composta por um número definido de indivíduos, cujos valores de x são escolhidos aleatoriamente dentro do intervalo estabelecido. A cada geração, os indivíduos são avaliados com base em sua aptidão, determinada pelo valor da função $f(x)$. Em seguida, os melhores indivíduos são selecionados para reprodução, gerando novos indivíduos por meio de operadores genéticos. O operador de crossover combina características de dois indivíduos para produzir descendentes, promovendo a exploração do espaço de busca. Já o operador de mutação introduz pequenas variações aleatórias nos descendentes, garantindo diversidade genética e evitando a convergência prematura para soluções subótimas. A implementação do AG neste experimento utiliza como parâmetros variáveis o número de indivíduos da população, o número total de gerações, a taxa de crossover e a taxa de mutação. Esses parâmetros são ajustados ao longo dos testes para avaliar sua influência sobre a qualidade das soluções obtidas e o desempenho do algoritmo. O critério de parada é atingido quando o número de gerações é alcançado ou

quando não há melhora significativa na aptidão da melhor solução ao longo de várias gerações consecutivas.

O uso de Algoritmos Genéticos nesse tipo de problema contínuo demonstra a capacidade dos métodos evolutivos de lidar com funções que podem apresentar múltiplos máximos e mínimos locais. Assim, o AG se mostra uma alternativa robusta para encontrar soluções aproximadas em espaços de busca complexos, mesmo sem conhecimento prévio da estrutura da função a ser otimizada.

2. Comparação entre os parâmetros variáveis

População Inicial

No projeto, a população inicial é composta por indivíduos gerados aleatoriamente, cada um representando um valor binário no intervalo $[-10, +10]$. Testes com diferentes tamanhos de população (de 4 até 30 indivíduos) mostraram que populações maiores tendem a aumentar a diversidade genética, o que pode ajudar o algoritmo a escapar de ótimos locais e encontrar soluções melhores. No entanto, populações pequenas tornam o algoritmo mais rápido, mas podem limitar a exploração do espaço de busca.

Mutação

A taxa de mutação foi definida inicialmente em 1%, mas pode ser ajustada em atualizações futuras. Taxas muito baixas podem levar à convergência prematura, pois a diversidade genética diminui rapidamente. Por outro lado, taxas muito altas podem transformar o algoritmo em uma busca quase aleatória, dificultando a convergência. No projeto, a taxa de 1% mostrou-se suficiente para manter a diversidade sem prejudicar a convergência.

Crossover

O crossover de um ponto foi utilizado com taxa padrão de 70%. Essa taxa permite que a maior parte dos indivíduos seja recombinada a cada geração, promovendo a exploração de novas combinações de genes. Taxas muito baixas podem tornar o algoritmo lento para explorar o espaço de busca, enquanto taxas muito altas podem prejudicar a preservação de boas soluções já encontradas.

Seleção

A seleção por torneio foi escolhida por sua simplicidade e eficiência, especialmente

em populações pequenas. Esse método favorece indivíduos mais aptos, mas ainda permite que indivíduos menos aptos sejam selecionados ocasionalmente, mantendo a diversidade.

Outras técnicas, como roleta, poderiam ser testadas, mas o torneio mostrou bom equilíbrio entre exploração e exploração.

Gerações

O número de gerações foi definido como variável, permitindo testes de 5 até 20 gerações. Mais gerações aumentam a chance de encontrar soluções melhores, mas também aumentam o tempo de execução. Em geral, poucas gerações já são suficientes para que o algoritmo encontre valores próximos do ótimo global para a função proposta.

3. Resultados e limitações

Resultados

O algoritmo genético foi capaz de encontrar rapidamente valores próximos ao máximo da função $f(x) = x^2 - 3x + 4$ no intervalo $[-10, +10]$. Em diversos testes, o melhor valor encontrado variou devido à natureza estocástica do algoritmo, mas frequentemente esteve próximo do máximo global dentro do intervalo. Como o processo envolve aleatoriedade na geração da população inicial, seleção, crossover e mutação, o melhor x pode mudar a cada execução. A cada geração, o algoritmo exibe o melhor indivíduo, permitindo acompanhar a evolução da população.

Limitações

Apesar do bom desempenho, o algoritmo apresenta algumas limitações:

• **Ótimos Locais:** Em funções mais complexas, pode convergir para ótimos locais, especialmente com populações pequenas ou poucas gerações.

- **Representação Binária:** O mapeamento binário limita a precisão dos valores representados, o que pode ser um problema em funções contínuas ou com intervalos maiores.
- **Parâmetros Fixos:** A escolha dos parâmetros (taxas de crossover, mutação, tamanho da população) influencia fortemente o desempenho e pode exigir ajuste manual para cada problema.

Considerações Finais

O projeto demonstrou de forma clara o funcionamento de um algoritmo genético simples e flexível, permitindo ajustes nos principais parâmetros. O método mostrou-se eficiente para otimização de funções simples, mas pode exigir adaptações para problemas mais complexos. O uso de parâmetros variáveis foi fundamental para entender o impacto de

cada componente no desempenho do algoritmo, reforçando a importância da experimentação em inteligência artificial.

Link do vídeo:  Projeto05-IA.mp4