

Érika Mara de Moraes Machado
Maria Luiza Bernardo Madeira

Matrícula: 202310058
Matrícula: 202310981

RELATÓRIO TÉCNICO - ALGORITMO GENÉTICO

Foram desenvolvidos dois componentes principais para a implementação de um Algoritmo Genético (AG): a estrutura do próprio algoritmo, programado manualmente em Python, e o módulo de execução responsável por testar diferentes parâmetros, gerar gráficos das evoluções e registrar os resultados. Toda a construção foi feita do zero, sem auxílio de bibliotecas especializadas em otimização, garantindo total controle sobre o funcionamento interno do AG.

Na implementação, a representação dos indivíduos é feita em formato binário, por meio de funções próprias que convertem inteiros para vetores de bits e vice-versa. A função objetivo utilizada foi $f(x) = x^2 - 3x + 4$, aplicada diretamente sobre cada indivíduo convertido para inteiro. As fronteiras mínimas e máximas (x_{\min} e x_{\max}) determinam o intervalo de busca, e o número de bits de representação é calculado dinamicamente com base nesses limites, permitindo melhor ajuste ao espaço de solução.

A população inicial é gerada aleatoriamente dentro dos limites definidos, e cada indivíduo é avaliado por meio da função objetivo. O algoritmo implementa seleção por torneio, escolhendo dois indivíduos aleatórios e selecionando aquele com maior fitness. Para produzir novos indivíduos, utiliza-se crossover de um ponto, que combina partes do material genético do pai e da mãe. Já a mutação, também implementada manualmente, altera um único bit do indivíduo com probabilidade definida pela taxa de mutação. Em ambas as operações, há uma função interna dedicada a garantir que os valores mutados ou cruzados permaneçam dentro dos limites permitidos, evitando soluções inválidas.

Durante cada geração, o algoritmo registra o melhor indivíduo encontrado, armazenando a evolução do fitness para posterior visualização. Ao final de cada execução, é gerado um gráfico utilizando a biblioteca Matplotlib, mostrando claramente como o valor do melhor fitness evolui ao longo das gerações. Esses gráficos são salvos automaticamente, permitindo análise posterior dos resultados obtidos com diferentes parâmetros.

O módulo principal possibilita a realização de múltiplas execuções. Inicialmente, o programa roda com parâmetros padrão, população de 4 indivíduos, 5 gerações, 1% de mutação e 70% de crossover. Em seguida, o usuário pode ajustar esses valores e realizar quantas execuções desejar, permitindo observar como diferentes combinações de parâmetros afetam o desempenho do algoritmo.

A implementação oferece total controle sobre cada etapa do processo evolutivo, permitindo compreender profundamente como seleção, mutação e crossover influenciam o resultado final. No entanto, essa abordagem tende a ser menos eficiente do que frameworks especializados, pois operações como comparação binária, geração de indivíduos e avaliação não possuem otimizações internas avançadas..

Complementando essa análise, a geração automática dos gráficos permitiu observar que diferentes parâmetros como tamanho da população, taxa de mutação e taxa de crossover influenciam diretamente a convergência do AG. Isso reforça a importância do ajuste fino dos hiperparâmetros, que pode ser explorado através das múltiplas execuções disponibilizadas no sistema.

Link para o vídeo de apresentação no YouTube: <https://youtu.be/BPa5jwY3YXI>