

# Aplicação de Algoritmos Genéticos na Otimização de Funções

Luiz Augusto Manfron Matias<sup>1</sup>

<sup>1</sup>Universidade Tuiuti do Paraná  
Curitiba – PR

luiz.matias@utp.edu.br

**Resumo.** *Este trabalho investiga a aplicação de Algoritmos Genéticos (AGs) na otimização de funções matemáticas, analisando o impacto de diferentes configurações de operadores genéticos (crossover, mutação, inicialização da população e critério de parada) sobre o desempenho do algoritmo. Implementado em Python, o AG foi testado com três métodos de crossover, três taxas de mutação e dois tipos de inicialização da população. Os resultados mostraram que a combinação de crossover de um ponto, alta taxa de mutação e inicialização heurística obteve os melhores desempenhos. A análise destaca a importância da escolha adequada dos parâmetros para maximizar a eficiência do AG em tarefas de otimização..*

## 1. Introdução

Os Algoritmos Genéticos (AGs) são técnicas de otimização inspiradas nos mecanismos de seleção natural e evolução biológica [DIO 2025]. Desenvolvidos por John Holland na década de 1960, os AGs utilizam operadores como seleção, cruzamento e mutação para evoluir soluções ao longo de gerações, buscando pontos de alta aptidão em espaços de busca complexos [Holland 1992]. Este trabalho propõe a implementação de um AG para otimização de funções matemáticas, com foco na análise de como diferentes configurações de operadores genéticos (métodos de crossover, taxa de mutação, inicialização da população e critério de parada) afetam o desempenho do algoritmo. O objetivo é identificar as configurações mais eficientes e compreender a influência de cada operador no processo evolutivo, contribuindo para o entendimento prático e teórico dos AGs na otimização.

## 2. Metodologia

A metodologia adotada neste trabalho consiste na implementação de um Algoritmo Genético (AG) para otimização de funções matemáticas, com o objetivo de comparar diferentes configurações dos operadores genéticos. Os principais operadores abordados são o crossover, a mutação, a inicialização da população e o critério de parada. O algoritmo genético foi implementado e testado em diferentes configurações para avaliar seu desempenho em relação ao problema específico de otimização proposto.

### 2.1. Crossover

O operador de crossover é fundamental para a combinação das informações genéticas de dois pais para gerar descendentes. Três métodos de crossover foram implementados neste estudo, sendo eles:

- **Crossover de um ponto:** Um ponto de corte é selecionado aleatoriamente ao longo do cromossomo, e as partes antes e depois do ponto de corte são trocadas entre os pais para gerar os filhos.
- **Crossover de dois pontos:** Dois pontos de corte são selecionados aleatoriamente, e a parte do cromossomo entre esses pontos é trocada entre os pais, gerando um filho com segmentos de ambos os pais.
- **Crossover uniforme:** Para cada gene do filho, é escolhido aleatoriamente se o gene será herdado do pai1 ou do pai2. Cada gene tem uma probabilidade de 50

## 2.2. Mutação

O operador de mutação é utilizado para introduzir diversidade genética na população, evitando a convergência prematura para soluções subótimas. A taxa de mutação foi configurada em três níveis distintos para analisar seu impacto no desempenho do algoritmo:

- **Baixa taxa de mutação (0.01):** Introduz pequenas variações nos indivíduos, garantindo que a maior parte da população preserve sua estrutura genética original.
- **Média taxa de mutação (0.05):** Promove um equilíbrio entre exploração e exploração do espaço de soluções.
- **Alta taxa de mutação (0.1):** Introduz modificações significativas nos indivíduos, favorecendo a exploração do espaço de soluções, porém com risco de perder boas soluções já encontradas.

## 2.3. Inicialização da População

A inicialização da população define como os indivíduos iniciais são gerados. Foram testadas duas abordagens para inicializar a população:

- **Aleatória:** Cada indivíduo é gerado aleatoriamente dentro de um intervalo pré-determinado para os genes. Esse método oferece uma ampla exploração do espaço de soluções.
- **Baseada em heurísticas:** Os indivíduos são gerados com base em uma distribuição mais estratégica de valores, tentando melhorar a inicialização e facilitar a convergência para boas soluções. Esse método usa uma distribuição linear dos valores dos genes.

## 2.4. Critério de Parada

O critério de parada determina quando o algoritmo deve ser interrompido. Foram implementados dois critérios:

- **Número fixo de gerações:** O algoritmo executa um número pré-definido de gerações (neste estudo, 100 gerações), independentemente da qualidade das soluções encontradas.
- **Convergência para uma solução ótima:** O algoritmo pode ser interrompido se a melhoria do melhor indivíduo nas últimas 5 gerações for inferior a um valor de limiar, definido como  $1e-5$ . Este critério indica que a população convergiu para uma solução ótima ou próxima disso, permitindo uma parada antecipada.

## 2.5. Execução do Algoritmo

A implementação do algoritmo genético foi realizada em Python, utilizando as bibliotecas numpy para operações matemáticas e pandas para manipulação de dados. O algoritmo foi projetado para otimizar funções matemáticas representadas por coeficientes, os quais são carregados a partir de arquivos CSV. Para cada execução do AG, os parâmetros de crossover, mutação, inicialização da população e critério de parada foram variáveis, permitindo a análise comparativa de diferentes configurações. O algoritmo realiza as seguintes etapas em cada geração:

1. Avaliação de resultado dos indivíduos, que é calculada com base na função objetivo representada pelos coeficientes carregados.
2. Seleção de pais para o crossover com base no resultado dos indivíduos, utilizando o método de seleção por classificação.
3. Realização do crossover entre os pais selecionados para gerar novos filhos.
4. Aplicação da mutação nos filhos gerados, com base na taxa de mutação escolhida.
5. Substituição da população atual pelos filhos gerados para a próxima geração.

A execução é repetida até atingir o critério de parada, seja pelo número fixo de gerações ou pela convergência para uma solução ótima.

## 2.6. Avaliação de Desempenho

Para avaliar o desempenho do algoritmo genético, foram realizadas comparações entre as diferentes configurações dos operadores genéticos. Os critérios de avaliação incluem:

- **Tempo de execução:** O tempo total gasto para a execução do algoritmo em cada configuração.
- **Qualidade da solução:** O valor do resultado do melhor indivíduo encontrado ao final das gerações, comparado entre as diferentes configurações de crossover, mutação e inicialização.

A análise dos resultados permite verificar o impacto de cada configuração no desempenho do algoritmo e a identificação da combinação mais eficaz para a otimização do problema abordado.

## 3. Resultados

Foram realizados testes utilizando diferentes combinações de métodos, mutação, e inicialização para otimizar a função. A tabela abaixo apresenta os melhores resultados obtidos, considerando a configuração de cada teste.

**Tabela 1. Tabela de Resultados**

Método	Mutação	Inicialização	Melhor Resultado (mín)	Melhor Resultado (máx)
Um Ponto	Baixa	Aleatória	-87.2917	-120.9014
Um Ponto	Baixa	Heurística	-117.0697	-107.8767
Um Ponto	Média	Aleatória	-239.6706	-248.5483
Um Ponto	Média	Heurística	-257.2050	-286.5089
Um Ponto	Alta	Aleatória	-377.8131	-331.4306
Um Ponto	Alta	Heurística	-332.2894	-337.9946
Dois Pontos	Baixa	Aleatória	-93.6636	-102.0610
Dois Pontos	Baixa	Heurística	-92.3808	-101.7089
Dois Pontos	Média	Aleatória	-259.0472	-255.5967
Dois Pontos	Média	Heurística	-233.6348	-225.7622
Dois Pontos	Alta	Aleatória	-332.9072	-318.2682
Dois Pontos	Alta	Heurística	-321.8059	-342.4647
Uniforme	Baixa	Aleatória	-90.7872	-107.6201
Uniforme	Baixa	Heurística	-79.5039	-106.1572
Uniforme	Média	Aleatória	-258.2251	-248.1426
Uniforme	Média	Heurística	-233.2404	-268.0043
Uniforme	Alta	Aleatória	-339.5584	-357.2458
Uniforme	Alta	Heurística	-329.7127	-340.1873

## 4. Discussão

Os resultados obtidos em relação ao método de otimização utilizado e seus respectivos desempenhos podem ser analisados a partir de algumas perspectivas importantes, como a escolha do operador de mutação, a inicialização dos parâmetros e a comparação entre as melhores soluções encontradas (mínimas e máximas). Em geral, podemos observar tendências e padrões que são indicativos da eficácia de diferentes configurações para resolver o problema de otimização em questão.

### 4.1. Métodos de Mutação

O estudo envolveu três métodos de mutação: "um ponto", "dois pontos" e "uniforme". Em termos gerais, os resultados indicam que a mutação "um ponto" tende a fornecer melhores resultados em comparação com as mutações "dois pontos" e "uniforme", especialmente quando associada à inicialização "baixa" e "heurística".

- **Mutação "um ponto":** A mutação de um ponto foi eficaz em alcançar os melhores resultados para vários cenários, principalmente com inicialização "baixa" e "heurística". Isso sugere que a mutação de um ponto pode ser mais eficiente para explorar o espaço de soluções quando combinada com uma inicialização otimizada.
- **Mutação "dois pontos":** Embora tenha mostrado resultados razoáveis, a mutação de dois pontos geralmente não foi tão eficaz quanto a de um ponto, especialmente nas inicializações "média" e "alta". Isso pode indicar que a troca de dois pontos nas soluções não contribui tanto para melhorar a exploração do espaço de busca quanto a mutação de um único ponto.

- **Mutação "uniforme"**: A mutação uniforme, por sua vez, mostrou um desempenho misto, mas não se destacou tanto quanto os outros métodos, especialmente nas inicializações de maior complexidade (como "alta").

## 4.2. Inicialização dos Parâmetros

O impacto da inicialização dos parâmetros foi uma das variáveis mais significativas nos resultados. Inicializações mais simples, como "baixa", frequentemente levaram a melhores resultados, particularmente quando combinadas com mutações de um ponto e heurísticas.

- **Inicialização "baixa"**: Quando a inicialização foi configurada como baixa, observou-se que tanto a mutação de um ponto quanto a heurística ajudaram a obter os melhores resultados. Isso sugere que uma inicialização simples pode ser mais eficaz quando a exploração de novas soluções é realizada com precisão.
- **Inicialização "media" e "alta"**: As inicializações "media" e "alta" frequentemente levaram a resultados mais distantes dos ótimos, principalmente em termos de melhores resultados máximos e mínimos. Isso pode ocorrer devido à maior complexidade envolvida em configurações iniciais mais avançadas, que podem exigir mais tempo de busca para encontrar soluções melhores ou levar a uma convergência para mínimos locais.

## 4.3. Comparação entre as Inicializações

Em comparação entre as inicializações "baixa", "media" e "alta", foi possível observar que a inicialização "baixa" apresentou os melhores desempenhos em termos de valores mínimos obtidos. No entanto, em alguns casos específicos, como nas configurações de "alta" e "heurística", a inicialização "alta" proporcionou um equilíbrio entre a exploração do espaço de soluções e a profundidade de busca.

## 4.4. Análise das Melhores Soluções

Os melhores resultados mínimos (exibidos nas colunas "Melhor Resultado (Min)") variaram consideravelmente com a combinação dos parâmetros. Por exemplo:

- No arquivo `function_opt_2.csv`, as combinações com "um ponto", "baixa" e "aleatória" produziram o melhor resultado mínimo de -87.2917, destacando a eficácia dessa configuração.
- No arquivo `function_opt_10.csv`, as configurações de "um ponto", "baixa", "aleatória" também se destacaram, com um melhor resultado de -767.8521, apesar dos valores em geral serem mais altos (indicando um problema mais difícil).

Esses resultados sugerem que a combinação de mutação de um ponto com inicialização baixa pode ser vantajosa para a obtenção de boas soluções em problemas de otimização de função.

## 4.5. Padrões de Desempenho nos Arquivos

A análise também revela que, à medida que a complexidade do problema aumenta (como observado nos arquivos de funções mais complexas como `function_opt_10.csv` e `function_opt_9.csv`), a diferença de desempenho entre as diferentes configurações se torna mais pronunciada. Com a variação da complexidade do problema, os métodos de otimização mostraram uma sensibilidade maior, e as soluções ideais foram mais difíceis de alcançar.

## 5. Conclusão

Em geral, a configuração mais eficiente para a maioria dos problemas é a mutação "um ponto" com inicialização "baixa" e heurística. Esta combinação apresentou bom desempenho, alcançando os melhores resultados em termos de valores mínimos. Contudo, problemas mais complexos podem exigir ajustes nas estratégias de inicialização e mutação. Em cenários mais desafiadores, técnicas de inicialização mais elaboradas e abordagens híbridas de mutação, como a combinação de múltiplos pontos ou métodos baseados em aprendizado de máquina, poderiam melhorar os resultados. Sugestões de melhorias incluem testar populações maiores e mais iterações para uma melhor convergência. Também seria útil implementar um mecanismo adaptativo para ajustar dinamicamente os parâmetros durante a execução. Além disso, explorar heurísticas mais avançadas, como aprendizado reforçado, pode ser uma direção promissora para otimizar ainda mais o desempenho.

## Referências

- DIO (2025). Algoritmos genéticos: Uma abordagem inspirada na evolução para resolver problemas complexos. Disponível em: <https://www.dio.me/articles/algoritmos-geneticos-uma-abordagem-inspirada-na-evolucao-para-resolver>. Acesso em: 10 maio 2025.
- Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence* (Bradford Book). The MIT Press.