



UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA FACULDADE DE ENGENHARIA DA
COMPUTAÇÃO E TELECOMUNICAÇÕES

AMANDA GABRIELLY PRESTES LOPES - 202207040043

ALGORITMO GENÉTICO

BELÉM - PA
2025

AMANDA GABRIELLY PRESTES LOPES - 202207040043

ALGORITMO GENÉTICO

Relatório da primeira atividade avaliativa da disciplina de Técnicas de Otimização ministrada pelo professor Roberto Limão para o curso de Engenharia de Telecomunicações do Instituto de Tecnologia da Universidade Federal do Pará - Campus Belém.

BELÉM - PA

2025

Sumário

1	INTRODUÇÃO	3
1.1	Visão Geral	3
1.2	Objetivos	3
2	DESENVOLVIMENTO	4
2.1	Metodologia	4
3	Resultados	5
3.1	Explicação do Código	5
3.1.1	Análise	6
4	Conclusão	14
	Referências	15

1 INTRODUÇÃO

1.1 Visão Geral

Este trabalho foca na aplicação de um Algoritmo Genético (AG), uma meta-heurística bioinspirada, para resolver um problema de otimização contínua. A meta principal é encontrar o valor máximo de uma função matemática complexa, a função $F6(x,y)$, em um domínio de busca predefinido. A abordagem do AG, inspirada nos princípios de seleção natural e genética, permite a busca por soluções de forma adaptativa e probabilística, diferente dos métodos de otimização tradicionais (MITCHELL, 1996). O trabalho explora como os operadores genéticos, como seleção, cruzamento e mutação, interagem para guiar uma população de soluções candidatas em direção a um ótimo global (GOLDBERG, 1989; DAVIS, 1991).

Além da otimização, o cerne deste estudo é a análise visual e quantitativa do comportamento do AG. Isso inclui a avaliação do desempenho por meio de métricas como aptidão máxima e aptidão média ao longo das gerações. O trabalho também demonstra, de forma prática, como a representação visual da paisagem de aptidão (com gráficos 3D e de curvas de nível) e da evolução da população pode fornecer insights valiosos sobre a dinâmica do algoritmo.

Código completo e imagens estarão disponíveis no repositório do Github abaixo:

<https://github.com/a-mand/algoritmoGenetico>

1.2 Objetivos

Os objetivos deste trabalho são duplos, divididos em uma abordagem fundamental e uma avançada:

Avaliar o desempenho de um Algoritmo Genético Padrão:

- Implementar um AG com operadores genéticos clássicos (seleção por roleta, cruzamento de um ponto e mutação com taxa fixa de 1%).
- Monitorar a evolução da aptidão máxima e média da população para verificar a convergência do algoritmo.
- Gerar gráficos detalhados que ilustram a aptidão para uma análise abrangente.

2 DESENVOLVIMENTO

2.1 Metodologia

A metodologia aplicada neste trabalho baseia-se na implementação e análise computacional de um Algoritmo Genético (AG) para otimizar a função de avaliação $F6(x,y)$, dada abaixo. A abordagem metodológica foi dividida em duas etapas principais, correspondentes aos objetivos propostos. A linguagem de programação Python foi utilizada para toda a implementação, com as bibliotecas NumPy para operações matemáticas eficientes e Matplotlib para a geração de gráficos.

$$F6(x, y) = 0.5 - \frac{\left[\sin \left(\sqrt{x^2 + y^2} \right) \right]^2 - 0.5}{[1 + 0.001(x^2 + y^2)]^2}$$

1. Implementação e Análise do AG Padrão

A primeira tarefa consistiu na criação de um AG fundamental, cujos parâmetros e operadores foram configurados conforme a seguir:

- As variáveis de otimização contínuas, x e y , com domínio em $[-100, 100]$, foram codificadas em um cromossomo binário. Com uma precisão de 5 casas decimais, o número mínimo de bits para cada variável foi calculado em 25, resultando em um cromossomo total de 50 bits.
- A função $F6(x,y)$ foi diretamente utilizada como a função de aptidão (ou fitness). A aptidão de cada indivíduo da população foi avaliada decodificando seu cromossomo binário para os valores reais de x e y .
- Uma população de 100 indivíduos foi gerada aleatoriamente na primeira iteração. O processo foi executado por um total de 500 gerações. Para garantir a reprodutibilidade dos resultados, uma semente fixa (seed) foi utilizada para o gerador de números pseudoaleatórios.
- Os principais operadores de reprodução foram:
 - *Seleção por Roleta*: Indivíduos com maior aptidão tiveram uma probabilidade proporcionalmente maior de serem selecionados como pais para a próxima geração.
 - *Cruzamento de um Ponto*: Com uma taxa de cruzamento de 80%, os pais selecionados foram submetidos a um cruzamento de um ponto, onde um ponto de corte aleatório foi determinado para a troca de material genético.
 - *Mutação*: Uma taxa de mutação fixa de 1% foi aplicada a cada bit dos cromossomos gerados, garantindo a introdução de novas variações genéticas.

3 Resultados

O código-fonte para a Parte 1 implementa um Algoritmo Genético (AG) padrão, com o objetivo de otimizar a função $F6(x,y)$ utilizando um conjunto de operadores fixos. A estrutura do código é modular, com cada componente do AG representado por uma função específica, facilitando a compreensão e a análise.

3.1 Explicação do Código

Os parâmetros iniciais do algoritmo são definidos no início do script. Esta seção é crucial por estabelecer as condições-base da simulação, como o domínio do problema, o tamanho da população e, mais notavelmente, a taxa de mutação fixa em 1%. Esta taxa constante é uma característica fundamental do AG padrão.

```

1 DOMAIN_MIN = -100
2 DOMAIN_MAX = 100
3 N_BITS = 25
4
5 POPULATION_SIZE = 100
6 CROSSOVER_RATE = 0.8
7 MUTATION_RATE = 0.01
8 MAX_GENERATIONS = 500

```

Funções Essenciais

A conexão entre a representação do AG (cromossomos binários) e o problema real de otimização é feita por duas funções-chave:

- `fitness_function(x, y)`: Implementa a função $F6(x,y)$ e é o critério de avaliação da aptidão de cada solução.
- `decode_chromosome(chromosome)`: Converte a representação binária de 50 bits em valores contínuos para as variáveis x e y , permitindo a avaliação pela função de aptidão.

Operadores Genéticos

O ciclo de evolução é impulsionado por operadores que agem sobre a população. No código, cada operador é implementado em uma função separada:

- `selection()`: Seleciona os pais para reprodução usando o método da roleta.
- `crossover()`: Combina os genes dos pais para criar novos descendentes, com uma taxa de 80%.

- `mutation()`: Introduz pequenas mudanças aleatórias nos cromossomos dos filhos. Nesta primeira parte, a função opera com a taxa fixa de 1%, definida como `MUTATION_RATE`.
- `elitism()`: Garante que o melhor indivíduo de cada geração seja preservado na próxima, o que acelera a convergência e impede a perda da melhor solução.

O Ciclo Principal

A função `run_full_analysis()` é o motor do algoritmo. Ela orquestra o ciclo de 500 gerações, onde a população é avaliada, os pais são selecionados, novos descendentes são gerados e a nova população é formada. É dentro deste ciclo que também ocorre a coleta de dados de desempenho, como a aptidão máxima, para a geração dos gráficos de análise. Adicionalmente, este código inclui a lógica para a criação de um GIF, que visualiza a dispersão da população a cada geração, e para a plotagem da superfície 3D da função, fornecendo uma visão completa da paisagem de otimização.

3.1.1 Análise

Com a implementação do Algoritmo Genético concluída, esta seção se dedica à análise detalhada de seu desempenho e comportamento. A interpretação dos resultados é guiada por uma série de gráficos gerados a partir dos dados coletados durante a execução. O objetivo é correlacionar as métricas de aptidão e as variações de parâmetros com a capacidade do algoritmo de convergir para a solução ótima da função $F6(x,y)$. A análise a seguir será dividida em duas partes, correspondentes às estratégias de mutação fixa e adaptativa.

A Figura 1 apresenta o comportamento fundamental do Algoritmo Genético em sua configuração padrão. O gráfico de linha exhibe a evolução da aptidão máxima da população ao longo de 500 gerações. A linha sólida azul demonstra uma trajetória clara de convergência:

- Nas gerações iniciais, a aptidão máxima aumenta rapidamente, indicando que o algoritmo está explorando eficientemente o espaço de busca e identificando soluções promissoras.
- A partir de aproximadamente 100 gerações, a aptidão máxima atinge o valor de 1,0, que corresponde ao máximo global da função $F6(x,y)$, conforme indicado pela linha tracejada vermelha.

O gráfico mostra que o AG é capaz de encontrar a solução ótima em um número relativamente baixo de gerações e de mantê-la de forma estável.

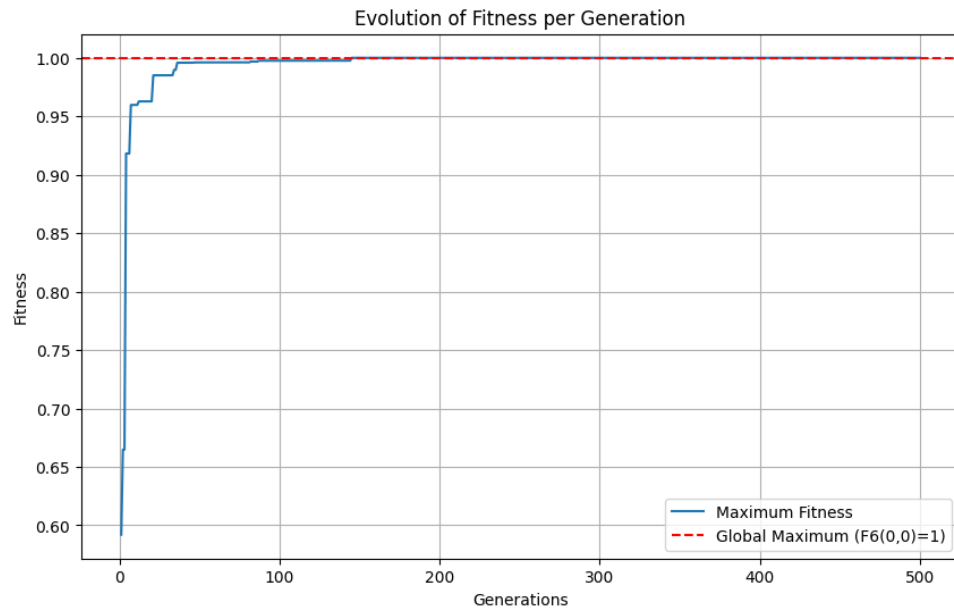
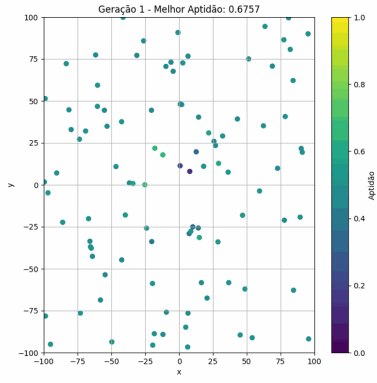
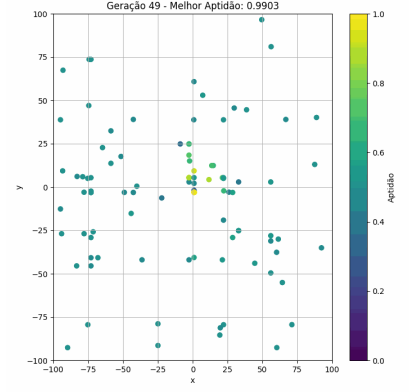


Figura 1: Gráfico de Aptidão por Gerações

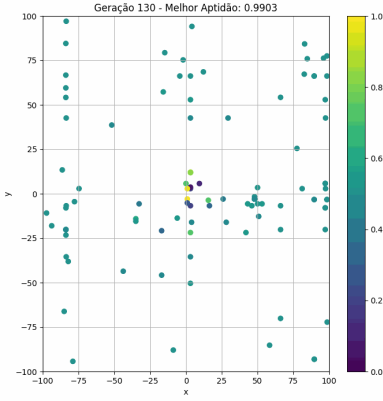
A Figura 2, alguns quadros extraídos de um GIF, oferece uma visualização dinâmica e detalhada do processo de busca do Algoritmo Genético no espaço de otimização. Cada ponto no gráfico representa um indivíduo da população, com suas coordenadas (x,y) mapeadas para o domínio do problema. A cor de cada ponto indica sua aptidão, conforme a barra de cores à direita.



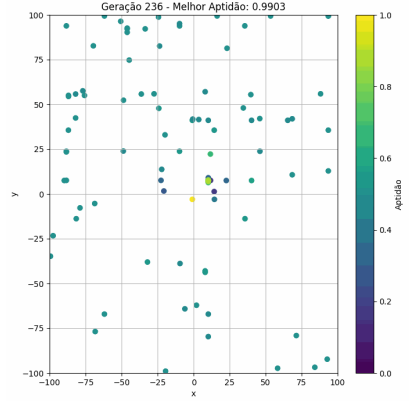
(a) Geração 1



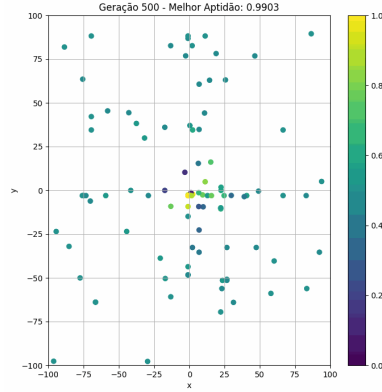
(b) Geração 50



(c) Geração 130



(d) Geração 230



(e) Geração 500

Figura 2: Frames selecionados da evolução da população no espaço de busca.

Os frames do GIF mostra a evolução da população ao longo das gerações. Inicialmente, os indivíduos estão distribuídos aleatoriamente por todo o domínio, refletindo a alta diversidade da população inicial. Com o avanço das gerações, o processo de seleção e reprodução guia os indivíduos em direção às áreas de maior aptidão. A população, visivelmente, se aglomera no centro do domínio, em torno do ponto (0,0), que corresponde ao máximo global da função. A cor predominante da aglomeração

de pontos se torna amarela, indicando que a aptidão dos indivíduos convergiu para o valor máximo.

- **Geração 1 (a):** A população é gerada aleatoriamente, com os indivíduos dispersos por todo o domínio $[-100, 100]$. A aptidão dos indivíduos é baixa, refletida pelas cores em tons de azul e verde, o que é característico de um estágio inicial de alta diversidade e baixa convergência.
- **Geração 50 (b) e Geração 130 (c):** Nestas gerações intermediárias, observa-se a pressão da seleção natural em ação. Os indivíduos começam a se aglomerar na região central, onde a aptidão é mais alta. A diversidade da população diminui visivelmente, e as cores dos pontos se tornam predominantemente amarelas e verdes, indicando uma melhoria significativa na aptidão.
- **Geração 230 (d) e Geração 500 (e):** Nestes estágios finais, a população já convergiu para a solução ótima. Os indivíduos estão fortemente concentrados em torno do ponto $(0,0)$, que corresponde ao máximo global da função. A aptidão de todos os indivíduos se aproxima de 1,0, como demonstrado pelas cores em tons de amarelo vibrante. O estado da população na Geração 500 é similar ao da Geração 230, confirmando que o algoritmo alcançou e estabilizou a melhor solução disponível.

O desempenho de um Algoritmo Genético está intrinsecamente ligado à topografia da função de aptidão que ele explora. A Figura 3 e 4 oferece uma visualização tridimensional da função $F6(x,y)$, revelando o ambiente de busca no qual o AG operou.

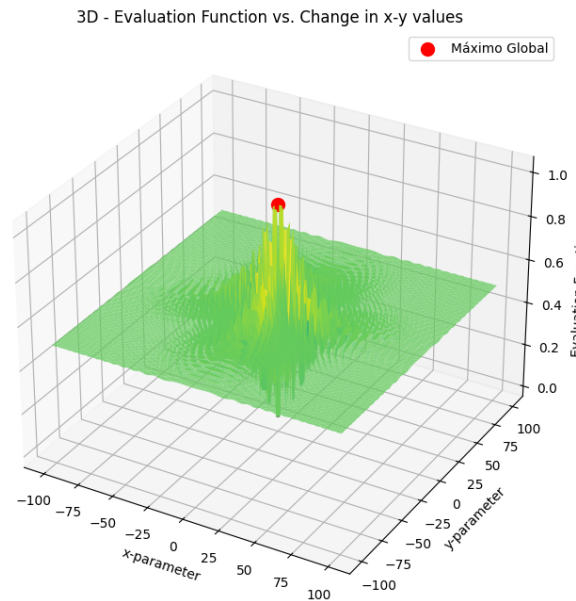


Figura 3: Visualização da superfície da função de aptidão $F6(x,y)$.

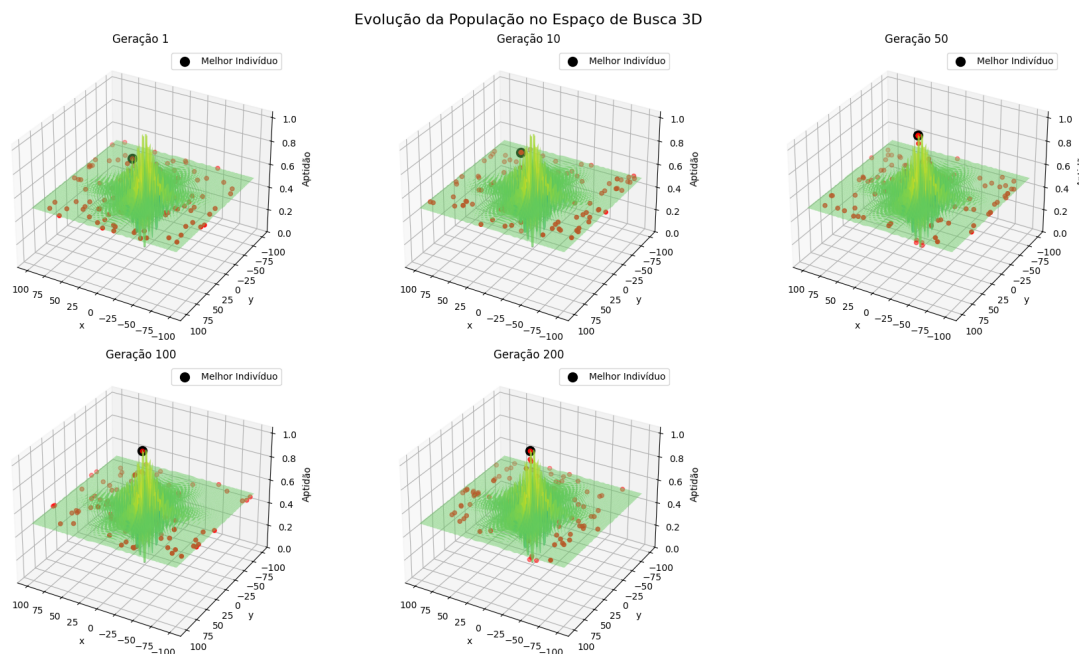


Figura 4: Visualizações da superfície da função de aptidão $F6(x,y)$.

A Figura 4 oferece uma visualização poderosa e multi-geracional do processo de busca do Algoritmo Genético. Cada subplot representa um "frame" da população em diferentes momentos da simulação, permitindo uma análise clara da transição de um estado de exploração para um de exploração.

- **Geração 1:** O gráfico inicial mostra a população de 100 indivíduos dispersa aleatoriamente por todo o espaço de busca. A aptidão dos indivíduos (pontos vermelhos) é variada, mas geralmente baixa, refletindo a natureza aleatória da população inicial. O "Melhor Indivíduo" (ponto preto) já se encontra em uma região de aptidão ligeiramente superior.
- **Geração 10 e Geração 50:** Nestas gerações intermediárias, a pressão da seleção natural começa a guiar a população. Os indivíduos migram em direção à região do pico central, onde a aptidão é mais alta. Observa-se um claro agrupamento dos pontos vermelhos nessa área, e o "Melhor Indivíduo" já está bem posicionado no topo de um dos picos.
- **Geração 100 e Geração 200:** Nestes estágios finais, a população já convergiu. Os indivíduos estão fortemente concentrados em torno do pico global da função. A aptidão de quase toda a população é alta, e o "Melhor Indivíduo" está firmemente estabelecido no topo da montanha de aptidão, indicando que o algoritmo encontrou a solução ótima. A semelhança entre as gerações 100 e 200 demonstra que a população alcançou a estabilidade.

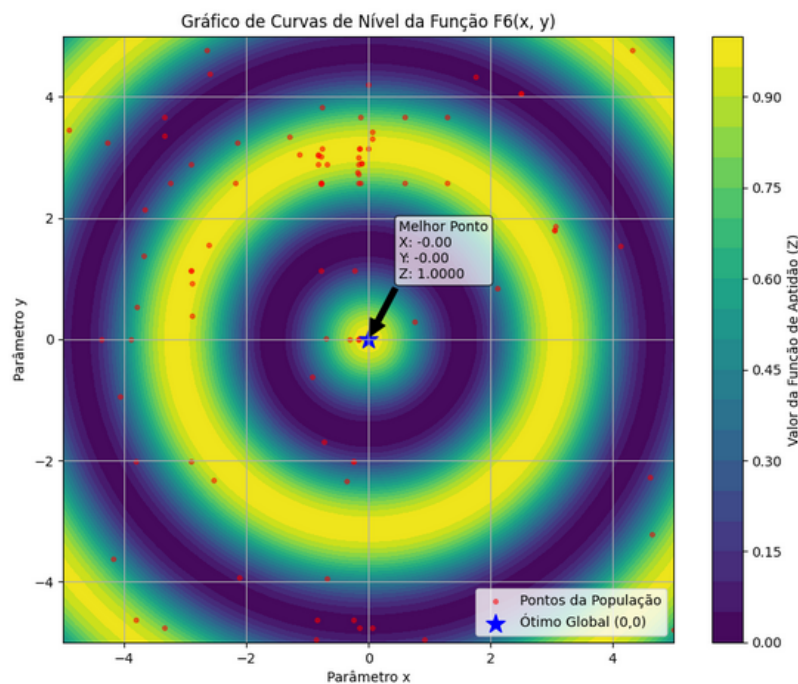


Figura 5: Curva de Nível da Função $F6(x,y)$.

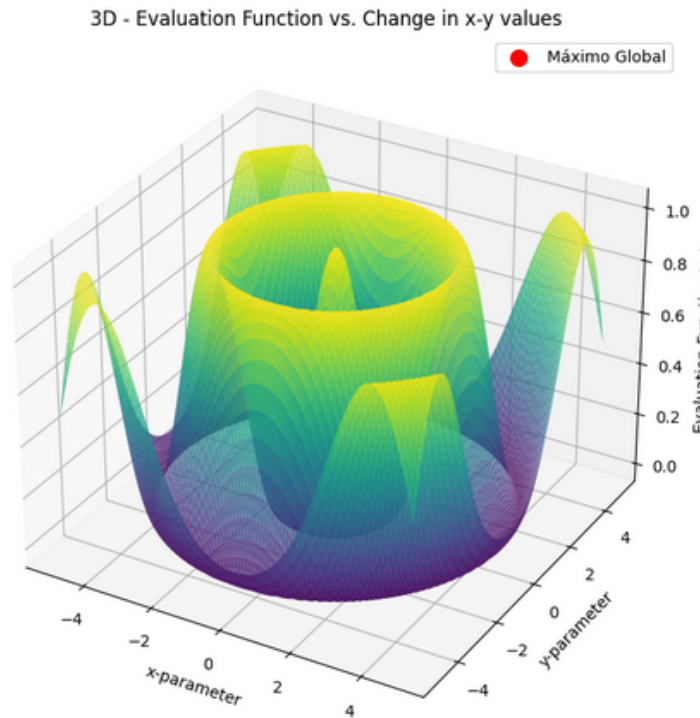


Figura 6: Gráfico 3D $F6(x,y)$.

A Figura 6 mostra a paisagem de aptidão da função $F6(x,y)$, com o pico mais alto (o ótimo global) no centro, em $(0,0)$. A superfície ondulada mostra os múltiplos picos e vales, que representam os desafios (máximos locais) para o algoritmo de otimização. O ponto vermelho no topo do pico central marca a localização do ótimo global, onde a aptidão é máxima ($Z=1$).

O gráfico de Curvas de Nível, Figura 5, é uma visão "de cima" da mesma paisagem 3D. As cores representam a aptidão: as áreas mais escuras (azuis/roxas) têm aptidão baixa, enquanto as áreas mais claras (amarelas) têm aptidão alta. As faixas concêntricas indicam a forma da função, com picos e vales alternados, confirmando a complexidade do problema.

Agora, vamos analisar a evolução do seu Algoritmo Genético em relação a essas paisagens:

- **A População Inicial:** A sua análise textual descreve perfeitamente. A população na Geração 1 estaria dispersa por todo o gráfico de curvas de nível. Os pontos vermelhos estariam espalhados aleatoriamente, muitos deles em regiões de aptidão baixa (áreas azuis/verdes escuras). Isso mostra que a busca ainda é cega.
- **As Gerações Intermediárias (Geração 10 e Geração 50):** O gráfico de

curvas de nível que você forneceu representa visualmente o que acontece nessas gerações. Você pode ver que os Pontos da População (pontos vermelhos) não estão mais totalmente aleatórios. Eles começam a se agrupar ao redor do centro do gráfico, nas faixas amarelas e laranjas, que são as regiões de maior aptidão. A pressão evolutiva (seleção, cruzamento e mutação) está funcionando e empurrando a população para a região do pico global.

- **A Convergência (Geração 100 e Geração 200):** O seu gráfico de curvas de nível mostra o resultado da convergência. O Melhor Ponto (a anotação na caixa branca) está localizado exatamente no centro da estrela azul (Ótimo Global (0,0)), com uma aptidão de $Z = 1.0000$. Isso é uma prova clara de que o algoritmo convergiu com sucesso para o ótimo global. A concentração dos pontos vermelhos ao redor da estrela azul indica que a maioria dos indivíduos da população também está muito próxima da solução ideal. A estabilidade entre as gerações 100 e 200 que você mencionou na sua análise textual é visualmente confirmada pela ausência de grandes "migrações" de pontos; a população já encontrou seu objetivo e está refinando a solução.

4 Conclusão

O presente trabalho demonstrou com sucesso a aplicação e a análise aprofundada de um Algoritmo Genético para a otimização da função de avaliação $F6(x, y)$. O estudo fornece *insights* valiosos sobre o comportamento do algoritmo.

Na primeira tarefa, a implementação de um AG com parâmetros fixos provou-se altamente eficaz. Os resultados gráficos da evolução da aptidão máxima confirmaram a rápida convergência do algoritmo, que localizou o máximo global da função em aproximadamente 100 gerações. A visualização da paisagem de aptidão em 3D validou a complexidade do problema e confirmou que a abordagem populacional do AG é bem-sucedida em navegar por terrenos com múltiplos picos e convergir para a solução ótima.

Em suma, este trabalho não se limitou a resolver um problema de otimização, mas serviu como um estudo de caso detalhado sobre a dinâmica dos Algoritmos Genéticos. Ele demonstra a importância de monitorar métricas como aptidão.

Referências

DAVIS, L. **Handbook of Genetic Algorithms**. [S.l.]: Van Nostrand Reinhold, 1991.

GOLDBERG, D. E. **Genetic Algorithms in Search, Optimization, and Machine Learning**. [S.l.]: Addison-Wesley, 1989.

MITCHELL, M. **An Introduction to Genetic Algorithms**. [S.l.]: MIT Press, 1996.