



Universidade Federal de São João del Rei
Departamento de Ciência da Computação
Curso de Ciência da Computação

Algoritmo Genético 4 - Caxeiro Viajante

Adélson de Oliveira Carmo Júnior
212050019

1 Como executar o programa:

O programa possibilita a passagem de valores para o número de gerações e o tamanho da população como parâmetros, o que facilita a realização de novos testes para uma compreensão mais aprofundada do seu funcionamento. Para utilizar o programa, basta fornecer a sua versão do Python, o nome do arquivo, o tamanho da população, o número de gerações desejado e se deseja mostrar (0 para não e 1 para sim) o gráfico ou não seguindo a ordem exemplificada abaixo:

```
1 python3.11 ag1.py 100 1000 1
```

Além disso, para o funcionamento adequado do programa, é essencial que o arquivo de entrada esteja localizado na mesma pasta do programa, devidamente nomeado e preenchido conforme demonstrado abaixo

- **instancia.txt:** Contém a matriz de distância das cidades;

2 Resultados esperados

Após a execução do algoritmo, serão gerados três arquivos de saída: `saida_1.txt`, `saida_2.txt` e `saida.txt`. Cada um desses arquivos tem uma função específica para os resultados. Os dois primeiros, `saida_1.txt` e `saida_2.txt`, salvam os dados dos melhores indivíduos por iteração utilizando os métodos de torneio e roleta, respectivamente. O último arquivo, `saida.txt`, apresenta a melhor combinação de cidades encontrada, assim como a distância total entre elas. Este resultado é baseado no último indivíduo de ambos os outros arquivos de texto, selecionando o menor entre eles.

As figuras 1, 2 e 3 ilustram esses três arquivos de saída. Se necessário, um gráfico adicional será gerado, conforme mostrado na figura 4, exibindo as curvas de convergência para encontrar o melhor caminho utilizando os dois métodos.

1991	291.0
1992	291.0
1993	291.0
1994	291.0
1995	291.0
1996	291.0
1997	291.0
1998	291.0
1999	291.0
2000	291.0
2001	

Figura 1: Melhores resultados usando Torneio

1995	307.0
1996	307.0
1997	307.0
1998	307.0
1999	307.0
2000	307.0
2001	

Figura 2: Melhores resultados usando Roleta

ag.py	1	9
instancia.txt	2	13
saida_1.txt	3	11
saida_2.txt	4	2
saida.txt	5	6
testes.py	6	4
	7	8
	8	14
	9	1
	10	12
	11	0
	12	10
	13	3
	14	5
	15	7
	16	Distancia = 291.0

Figura 3: Melhor resultado entre os dois testes

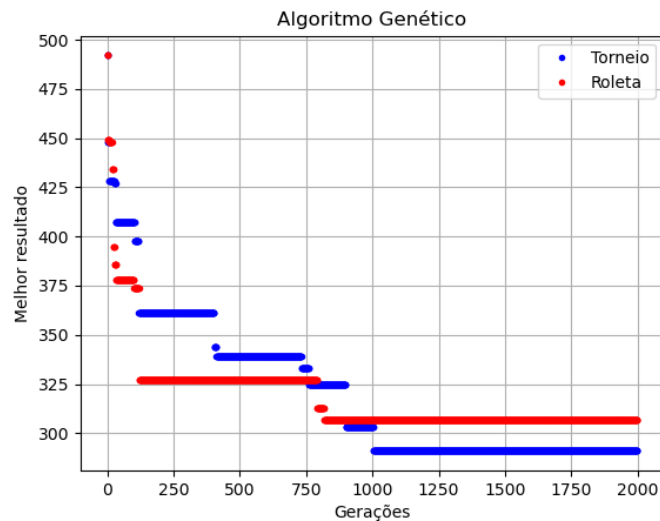


Figura 4: Gráfico do melhor indivíduo de cada geração

3 Testes e análise de resultados

Com o objetivo de compreender o desempenho do algoritmo genético em diferentes cenários, foram conduzidos diversos testes, nos quais variáveis cruciais, como a presença de elitismo e o tamanho da população, foram modificadas. Nesta seção, serão apresentados os resultados obtidos nestes experimentos e as conclusões que são possíveis de extrair sobre o algoritmo com base neles.

Os testes foram realizados usando um outro código, também em Python, além do código principal. Esse código adicional permitia a realização de vários testes e a geração de gráficos a partir dos resultados.

Nesta seção, foram mantidos alguns parâmetros constantes para explorar o impacto de um em particular. O padrão utilizado foi o seguinte:

- Método da Torneio;
- Tamanho da população: 100 indivíduos;
- Taxa de mutação: 1% em condições normais e 10% caso o melhor elemento permaneça inalterado após 5 gerações;
- Taxa de cruzamento: 100%;
- Com elitismo;
- Número de gerações: 2000;
- Entradas de 15 cidades

3.1 Método de escolha de pais

Para uma análise inicial, optou-se por examinar o comportamento geral entre os dois métodos de seleção de pais, o Torneio e a Roleta. Assim, foram conduzidos 100 testes utilizando ambos os métodos, mantendo certos parâmetros constantes citados anteriormente.

Os resultados desses testes podem ser observados nas Figuras 5 e 6. Vale ressaltar que eles foram ordenados do menor ao melhor em ambos os métodos, proporcionando uma melhor visualização. Além disso, foram utilizadas sempre a mesma população inicial em ambos os métodos.

Ambas as imagens provam que num geral o método do torneio é melhor que o da roleta para essa aplicação, uma vez que tanto para encontrar um melhor indivíduo, quanto em número de gerações para encontrá-lo, o

torneio encontrou valores menores e mais satisfatórios que o outro método. Assim, é possível concluir que o torneio é a escolha ideal para essa aplicação.

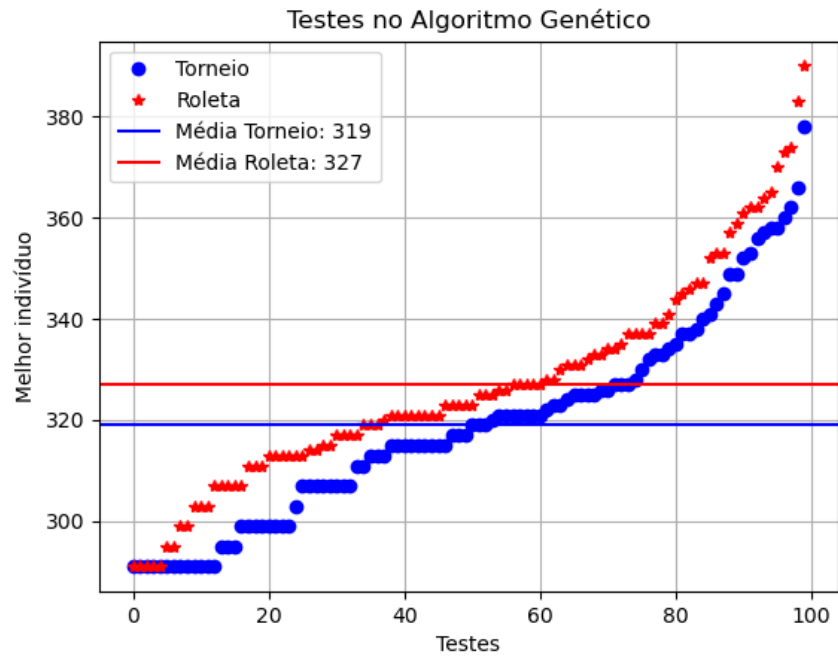


Figura 5: Melhores valores em cada teste

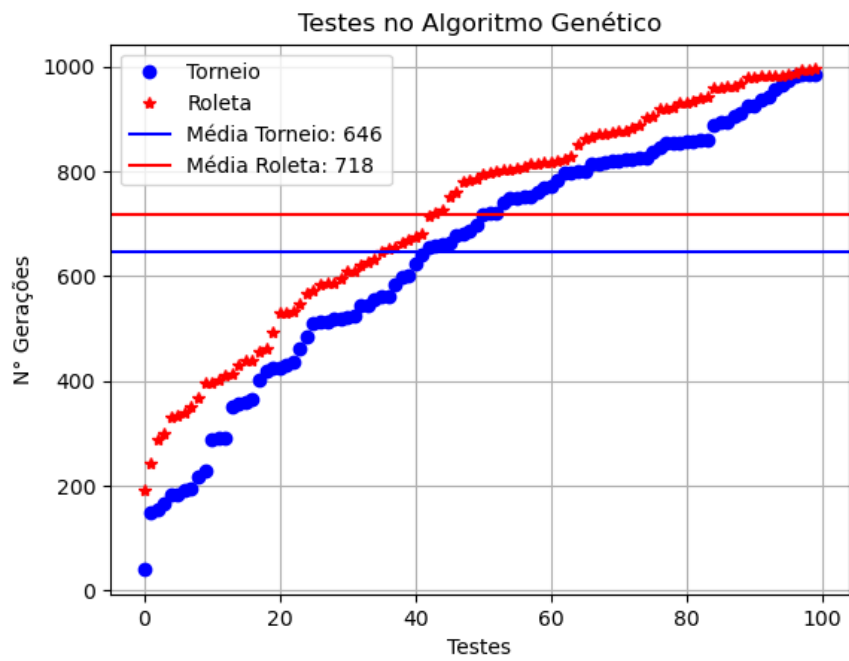


Figura 6: Gerações para encontra o melhor indivíduo

3.2 Tamanho da população

Foram realizados mais 100 testes, desta vez variando o tamanho da população para 30, 50 e 100 indivíduos. Os demais parâmetros foram mantidos constantes e, nesta sessão, o foco foi apenas no tamanho da população.

A Figura 7 ilustra claramente os efeitos da diminuição do tamanho da população, mostrando que a eficiência em encontrar um melhor indivíduo foi comprometida. Além disso, a Figura 8 evidencia uma deterioração na quantidade de gerações necessárias para encontrar o melhor indivíduo ao diminuir a população.

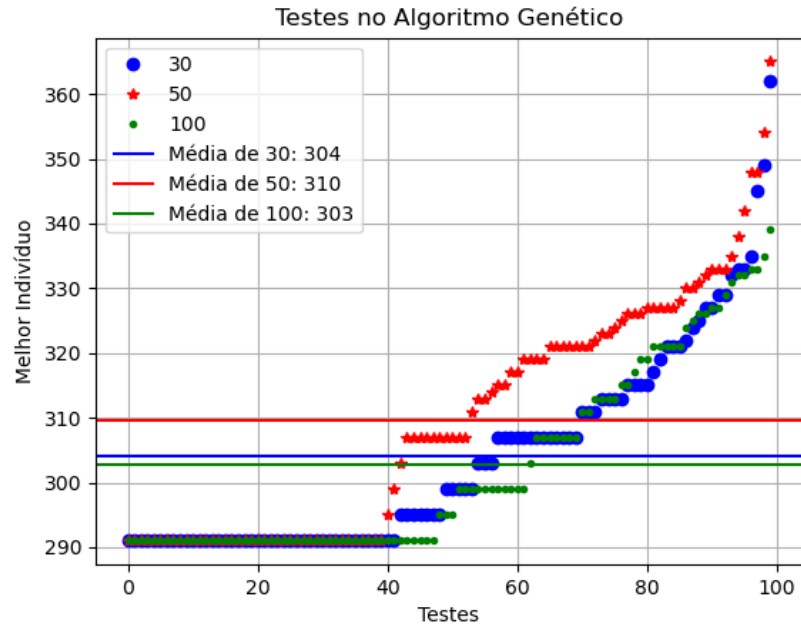


Figura 7: Melhores valores em cada teste

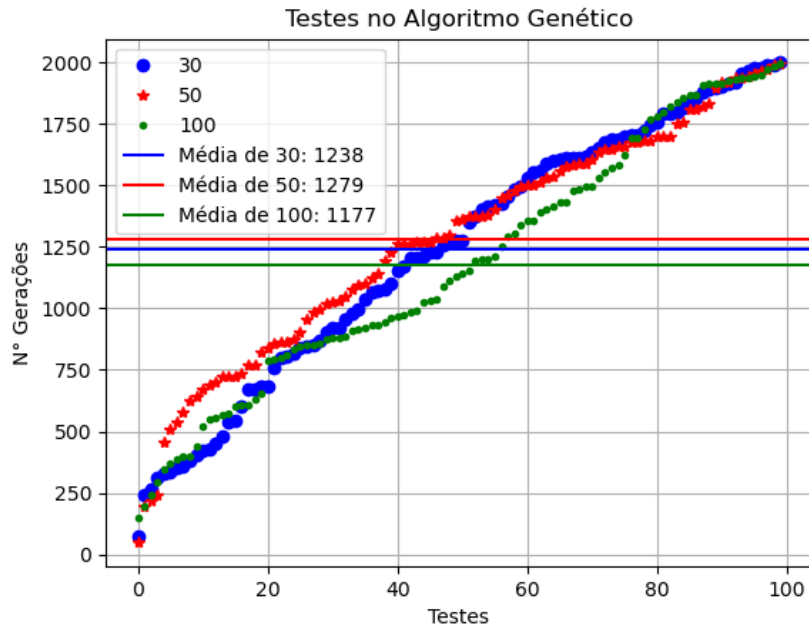


Figura 8: Gerações para encontra o melhor indivíduo

Vale destacar que, embora as populações de 30 e 50 indivíduos tenham um desempenho inferior ao de 100, a população de 30 se saiu melhor que a de 50. Isso indica que não é tão simples afirmar que apenas diminuir ou aumentar a população garante eficiência. Testes adicionais são necessários para determinar qual tamanho de população é mais adequado para cada situação específica.

Portanto, é possível concluir que é preciso analisar cuidadosamente o tamanho da população. Nesta aplicação, houve tanto melhorias quanto pioras ao ajustar esse parâmetro, indicando que a eficiência depende de um equilíbrio adequado no tamanho da população.

3.3 Taxa de Mutação

Semelhante ao processo realizado na seção anterior, alguns parâmetros foram mantidos constantes enquanto variava-se o que deveria ser estudado. Desta vez, o foco foi na taxa de mutação, mantendo constante uma população de 100 indivíduos.

Para a variação, foram adotadas três porcentagens diferentes: 1%, 5% e 10%. Os resultados dos 100 testes realizados podem ser observados nas Figuras 9 e 10. Analisando a Figura 9, nota-se que a melhor taxa de mutação foi de 1%, seguida por 5% e 10%, nessa ordem. A diminuição da taxa de mutação proporcionou melhores resultados, comportamento que também foi observado na Figura 10 com relação ao número de gerações.

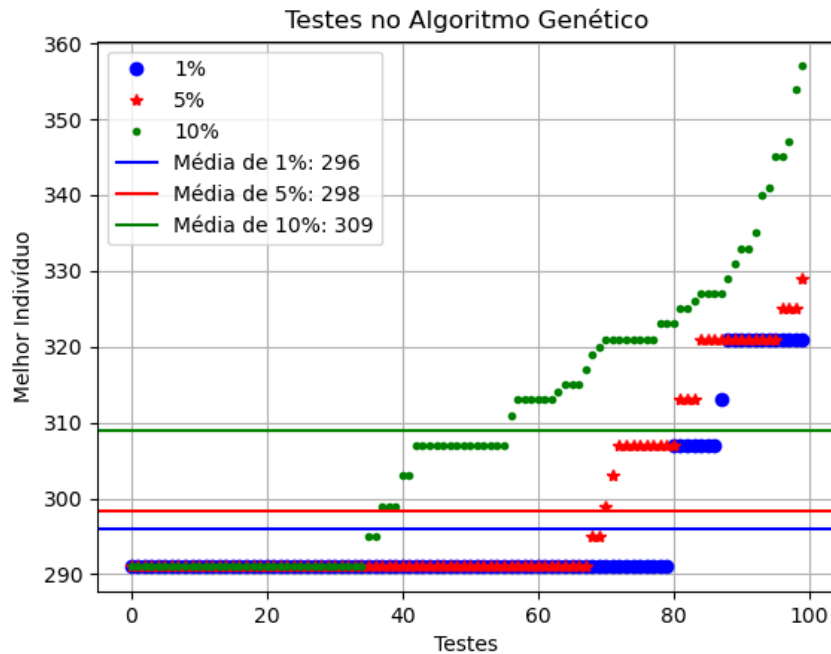


Figura 9: Melhores valores em cada teste

Portanto, foi possível constatar que o aumento na taxa de mutação provoca uma queda na eficiência. Para esta aplicação, uma taxa de mutação menor previne que o desempenho seja comprometido.

3.4 Taxa de cruzamento

Mantendo a ideia de conservar alguns parâmetros, pode-se observar o comportamento de um parâmetro específico por meio de sua variação. Neste tópico, será estudada a taxa de cruzamento, seguindo o padrão anterior.

As porcentagens utilizadas para os 100 testes foram as seguintes: 60%, 80% e 100%. Ao analisar a Figura 12, percebe-se uma nítida diminuição nas gerações necessárias para encontrar o melhor indivíduo à medida que a taxa de cruzamento diminui. Portanto, a taxa de 60% demonstrou ser a mais eficiente, seguida por

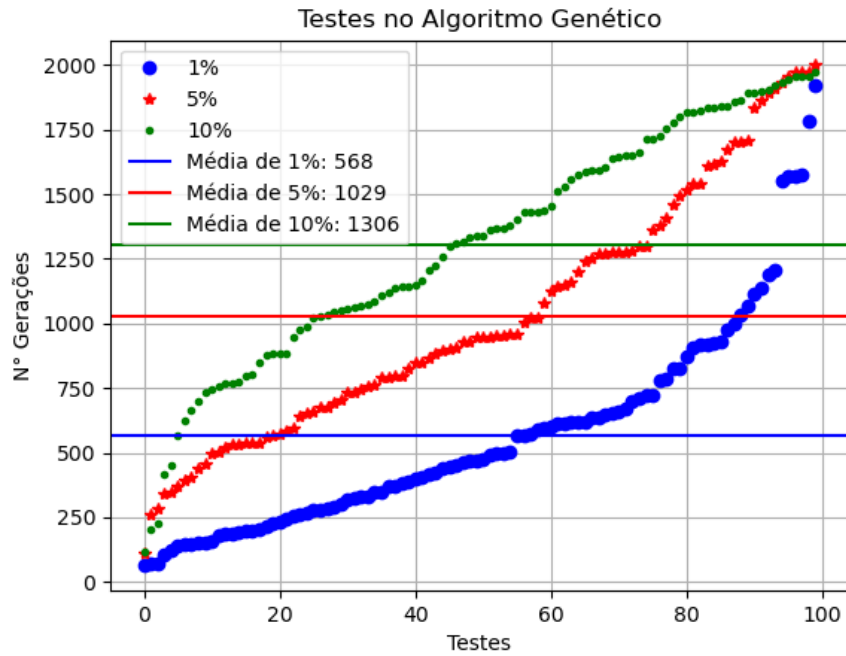


Figura 10: Gerações para encontra o melhor indivíduo

80% e, por fim, 100%. Essa tendência é invertida na Figura 11, onde uma taxa de 100% resultou em mais indivíduos com o valor desejado para o melhor elemento, seguida por 80% e 60%, sendo esta última a menos eficaz.

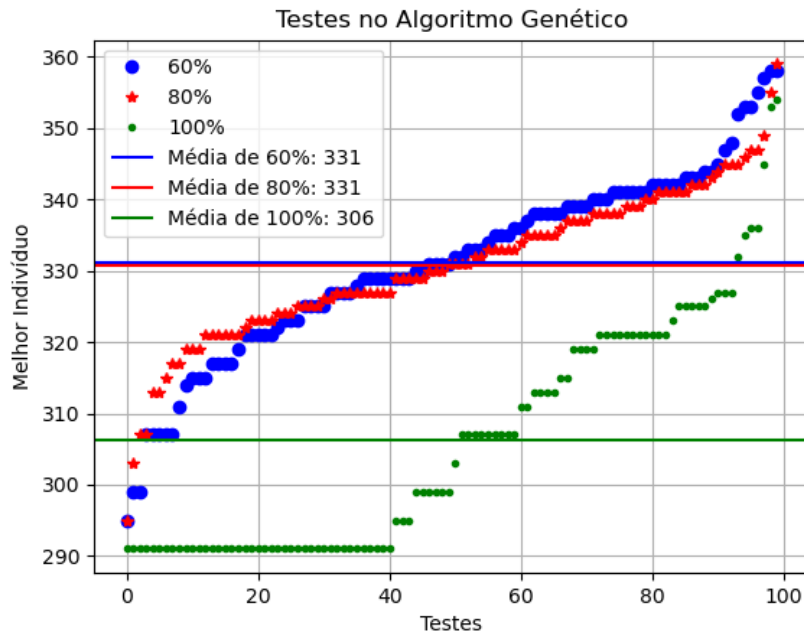


Figura 11: Melhores valores em cada teste

Portanto, embora uma taxa de cruzamento de 100% possa demorar mais gerações para encontrar o melhor valor, ela é a mais indicada, pois garante maior consistência e menores valores para o caminho solução.

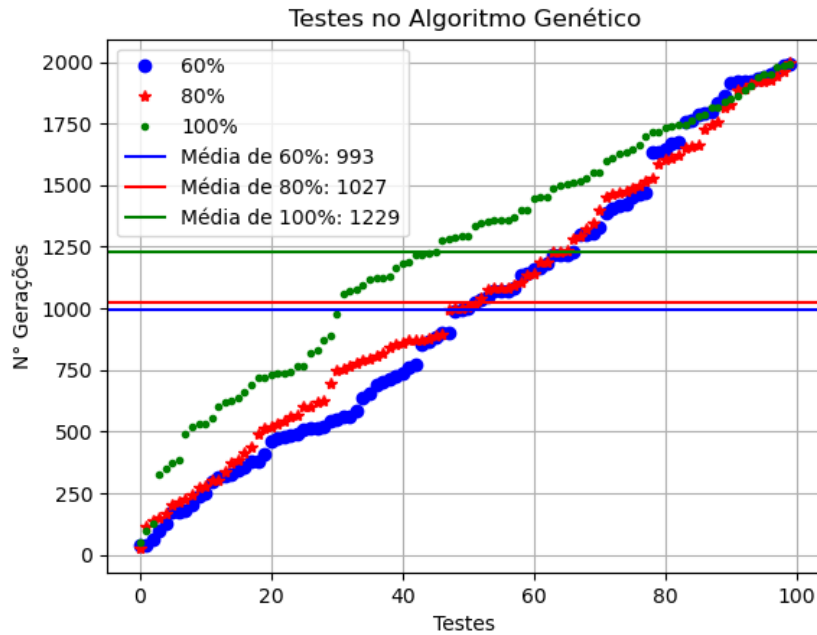


Figura 12: Gerações para encontra o melhor indivíduo

3.5 Presença de elitismo

A abordagem de teste nesta seção se concentra em avaliar o impacto do elitismo no desempenho do algoritmo genético. Ao analisar as figuras 13 e 14, fica evidente que o elitismo desempenha um papel crucial na estabilidade e na eficiência do algoritmo.

Na figura 13, observa-se uma dispersão significativa dos pontos quando o elitismo não é empregado. Muitos pontos estão abaixo do eixo x, indicando flutuações nos resultados ao longo das gerações. Essa instabilidade é causada pela ausência de uma manutenção do melhor valor, resultando em mudanças frequentes no melhor indivíduo encontrado.

Por outro lado, na figura 14, onde o elitismo é aplicado, observa-se uma estabilidade notável ao longo das gerações. Os pontos se mantêm consistentemente próximos aos valores desejados, sem flutuações significativas. Isso evidencia que a preservação do melhor indivíduo ao longo das gerações proporcionada pelo elitismo contribui para a consistência e a eficácia do algoritmo genético.

Portanto, fica claro que o elitismo é um fator crucial para o bom funcionamento do algoritmo genético, garantindo uma busca mais eficiente e estável pelo melhor indivíduo. Sua presença proporciona uma base sólida para o algoritmo, ajudando a evitar flutuações indesejadas nos resultados e aumentando a probabilidade de convergência para soluções ótimas.

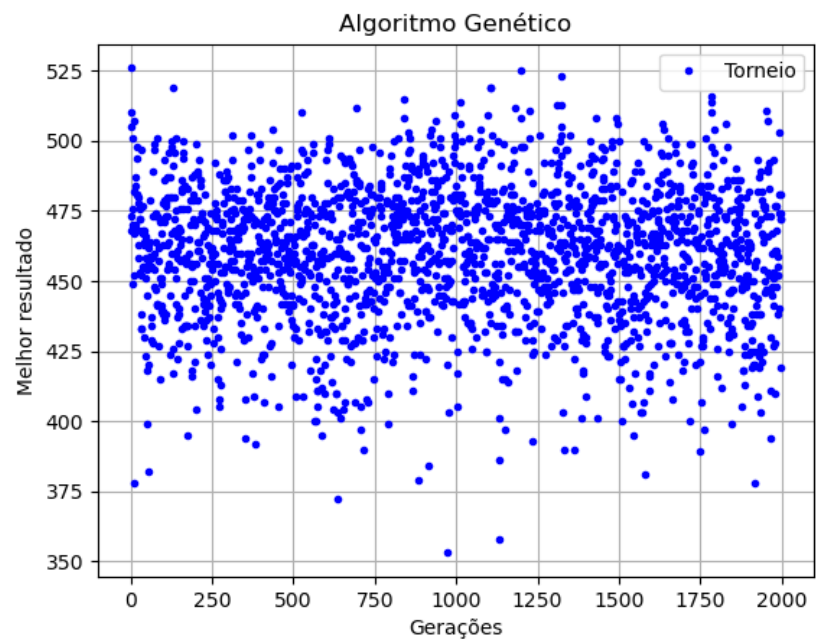


Figura 13: Sem elitismo

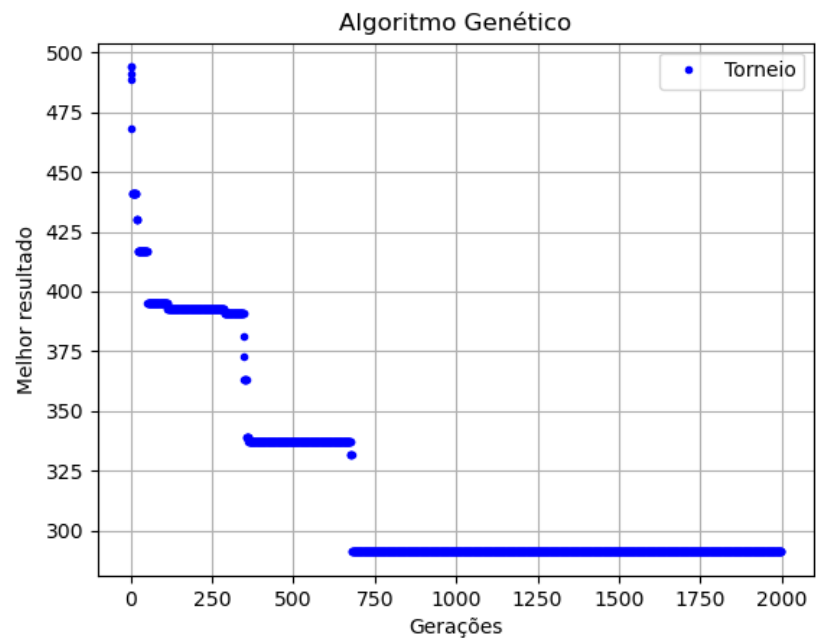


Figura 14: Com elitismo