

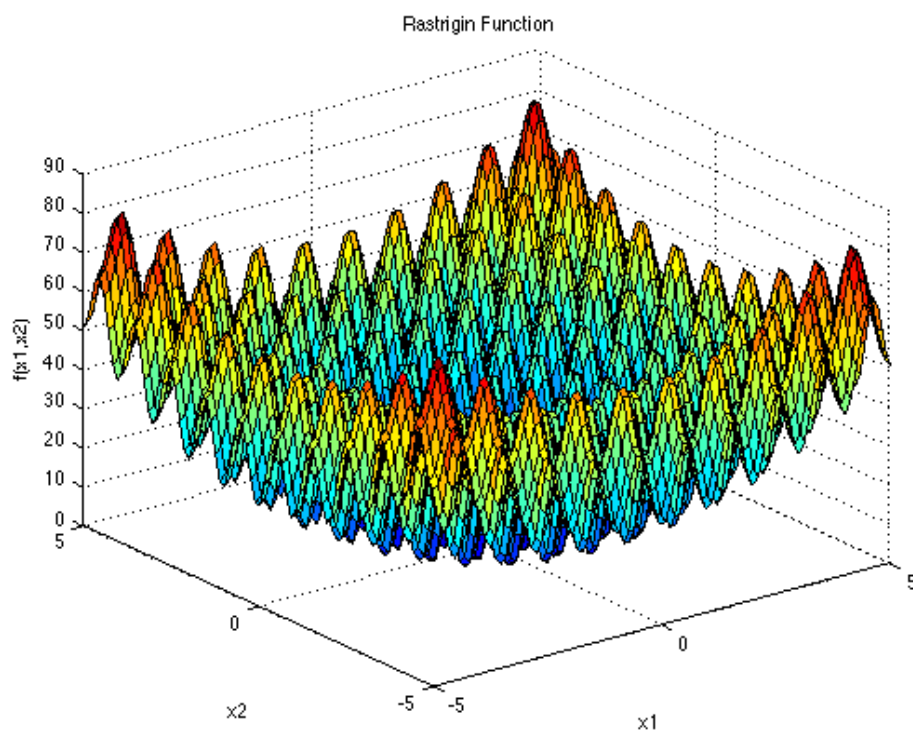
# Trabalho 2 Inteligência Computacional Aplicada

Leo Land Bairos Lomardo - 2020201  
Luigi Gosling de Alencastro - 1811556

Maio 2023

# 1 Função Rastrigin

A função Rastrigin é uma função matemática não linear multi-modal utilizada para testar otimização de algoritmo. Nesse caso buscaremos o mínimo desta função, com o menor número de iterações possível. Tal problema se torna extremamente desafiador, devido a grande quantidade de mínimos locais que essa função apresenta. A função Rastrigin pode ser analisada tendo 1,...,n dimensões. No caso onde a dimensão é 1, o problema se torna bem simples. Já no caso da segunda dimensão, o problema se torna mais complexo, visto que terá diversos mínimos locais.



## 2 Análise do Problema

O objetivo do Trabalho 2 sobre Algoritmo Genético é avaliar e testar diversos parâmetros, afim de encontrar o mínimo da função, no menor tempo possível. Pela prioridade ser reduzir ao máximo o tempo de processamento, devemos definir uma estratégia onde o número de gerações é o menor possível.

O maior desafio do problema se relaciona com a quantidade de parâmetros e a interpolação de seus valores. Os valores iniciais dos parâmetros do nosso trabalho, serão os valores *default* da biblioteca *geneticalgorithm2* ou então valores previamente definidos no código fornecido. O único valor que será definido arbitrariamente é o número máximo de iterações, que optamos por iniciar com um valor mais alto e otimiza-lo por último.

Outro detalhe importante, é que a própria biblioteca retirou o parâmetro *Probabilidade de Cruzamento*, e na documentação eles mostraram que uma maneira de contornar esse parâmetro era coloca-lo com valor = 1. Tais valores serão explicitados na tabela abaixo, junto com os gráficos plotados com os resultados obtidos.

Dimensão	2
Máximo de Iterações	500
População	100
Probabilidade de Mutação	0.1
Taxa de Elitismo	0.01
Probabilidade de Cruzamento	1.0
Porção de Parentes	0.3
Tipo de Cruzamento	One point
Tipo de Seleção	Roleta

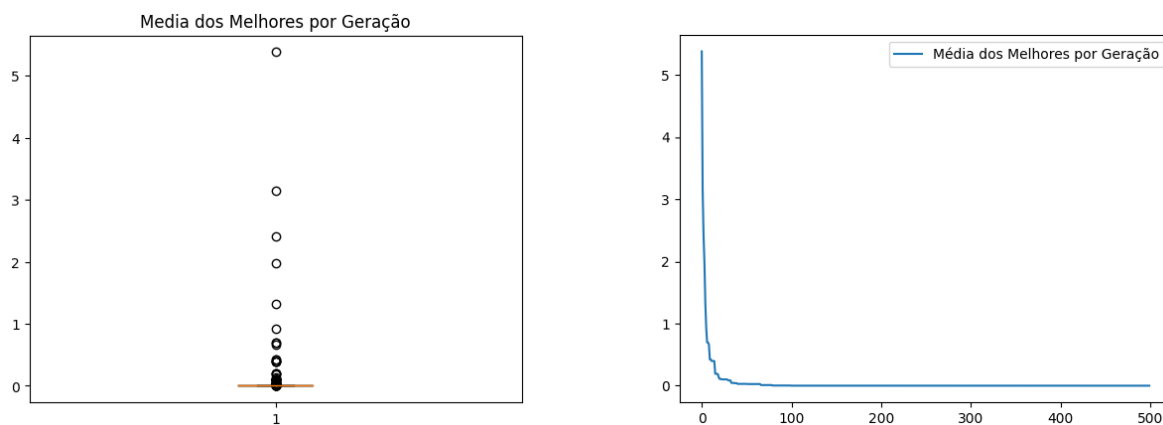


Figura 1: Resultado Inicial

O valor médio dos melhores por geração obtido foi de 0,0461686. Resultado bom, visto que está próximo do mínimo global (0,0). Podemos ver no segundo gráfico, que o gráfico apresentou uma lateralização próximo da geração 100, sendo um indicador que podemos diminuir tal valor. Nos experimentos seguintes, atualizaremos o número máximo de gerações para 250.

### 3 População

O tamanho da população define a quantidade de soluções experimentais que teremos em cada iteração. De tal forma, não podemos escolher um valor muito pequeno, visto que tornaria mais difícil de achar a solução ótima. Por outro lado, um valor muito grande tornaria o algoritmo muito lento.

População	10	50	75	100	200
Resultado	0,10957	0,00388	0,00048	0,00042	$2.8508 \times 10^{-5}$

Podemos ver nas imagens abaixo, organizados da esquerda para a direita, a média dos melhores resultados obtidos por geração.

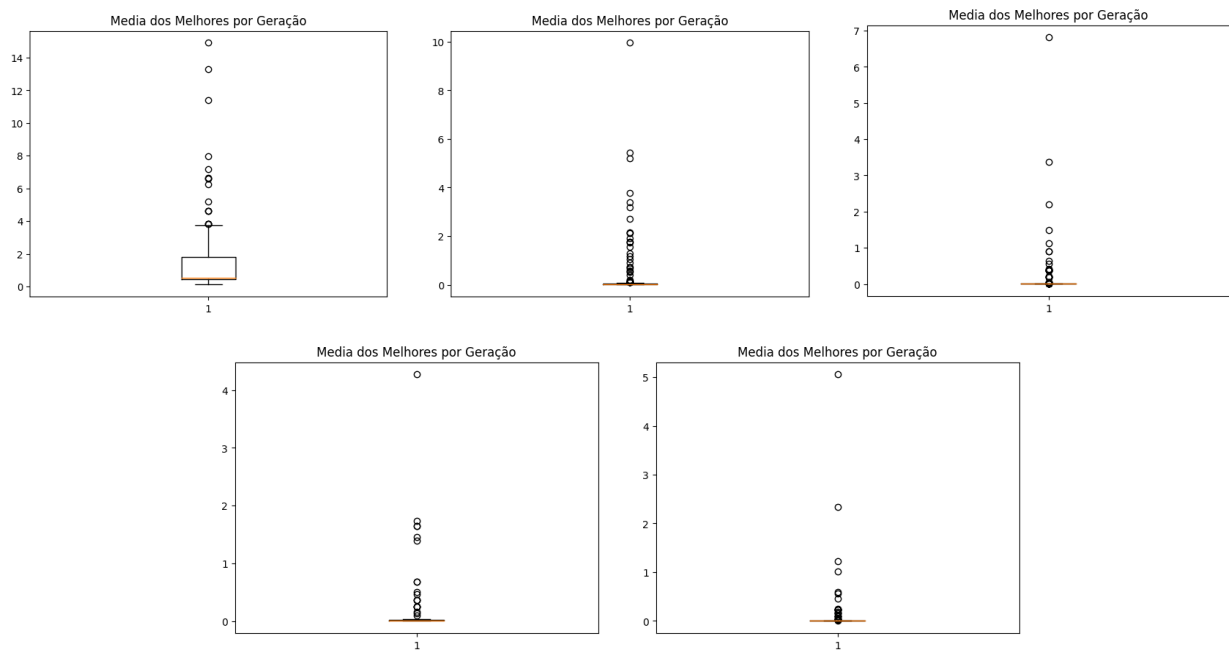


Figura 2: Média dos Melhores por Geração

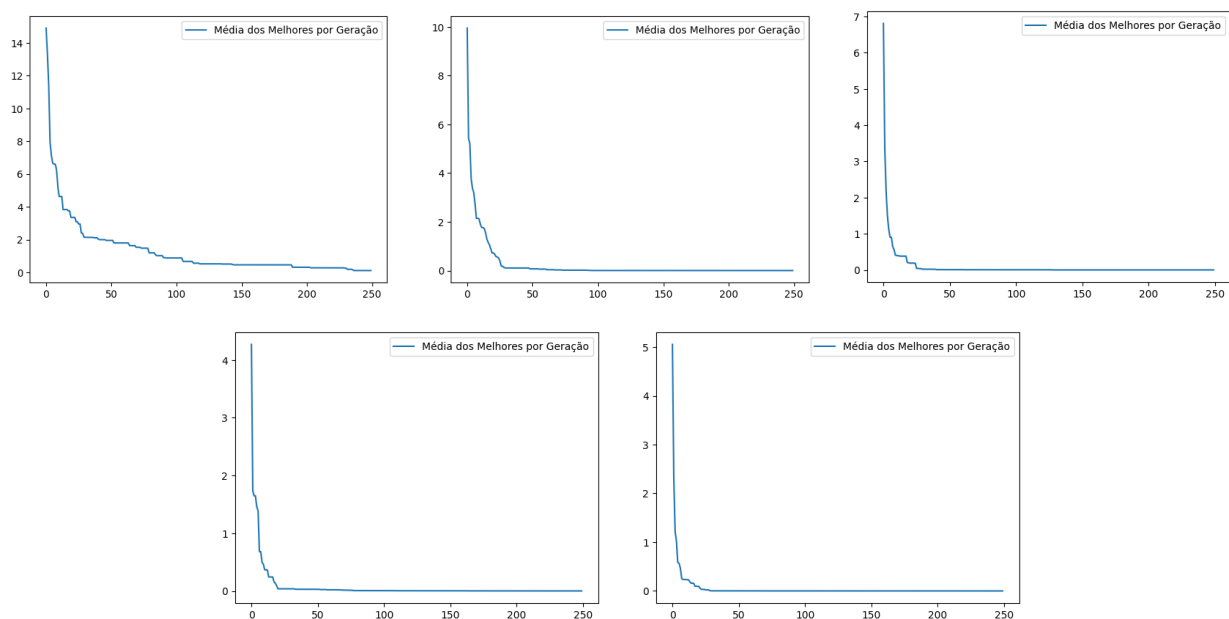


Figura 3: Resultado Médio por Geração

Analisando exclusivamente os resultados, escolheríamos o valor da população igual a 200. Porém como foi citado anteriormente, tal valor tornaria o algoritmo muito demorado. Percebe-se nos gráficos da figura 3 que a partir do tamanho da população igual a 50, todos os resultados iniciaram uma lateralização próximo do valor de 50. Devido a este fato, utilizaremos a população máxima igual a 75.

## 4 Probabilidade de Mutação

A probabilidade de mutação tem um peso muito grande no resultado do algoritmo. Um valor muito baixo pode acarretar na estagnação do algoritmo em um ótimo local e caso o valor seja muito alto, pode causar o mesmo problema. A tabela abaixo mostra os valor que analisamos nessa etapa do trabalho.

Mutação	0,001	0,01	0,1	0,5	0,75	1,0
Resultado	0,03413	0,00859	0,00128	$1.1939 \times 10^{-6}$	$6.0301 \times 10^{-8}$	$1.2244 \times 10^{-8}$

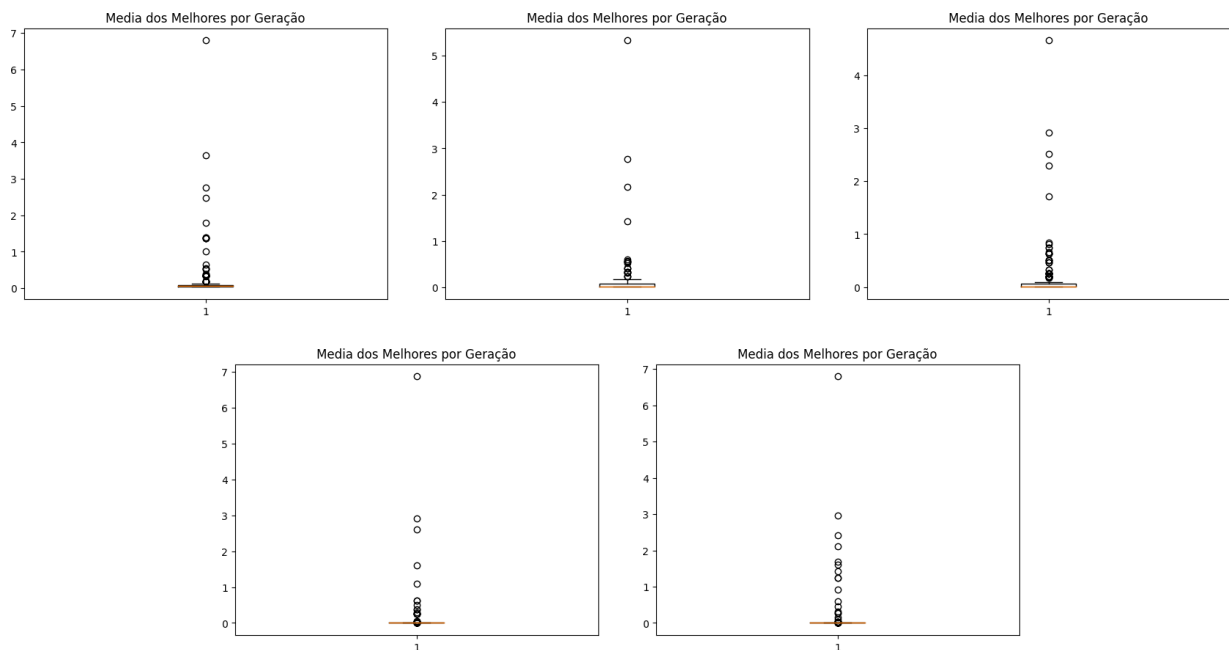


Figura 4: Média dos Melhores por Geração

O algoritmo com a maior taxa de mutação apresentou o melhor resultado, e tal fato se justifica pela característica do problema. Como estamos buscando o mínimo da função, uma taxa de mutação mais alta permite a maior exploração do espaço de busca. Apesar do melhor resultado, o algoritmo que apresentou a lateralização do gráfico na menor geração foi o algoritmo com taxa de mutação igual a 0,75. Por agora manteremos a taxa de mutação igual a 1,0.

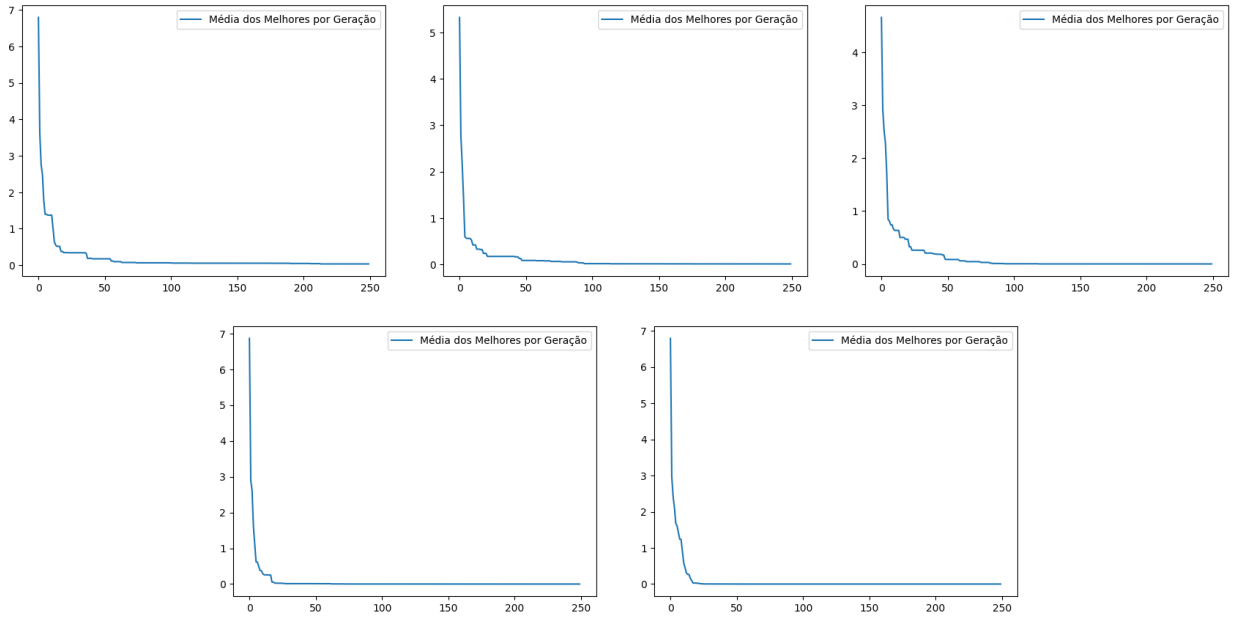


Figura 5: Média dos Melhores por Geração

## 5 Taxa de Elitismo

A taxa de elitismo define a fração dos melhores indivíduos de uma população que irão se reproduzir. Como estamos buscando o ponto mínimo, ter uma taxa de elitismo agrega para o algoritmo, podendo acelerar a convergência do algoritmo. Entretanto, uma taxa de elitismo muito alta pode resultar em uma convergência prematura do algoritmo, nos mostrando um resultado referente a um mínimo local.

A escolha dos valores a serem testados, depende diretamente do parâmetro *população de parentes*, que veremos a seguir. Tal parâmetro tem um comportamento similar ao elitismo, e ele acaba limitando o valor máximo de elitismo.

Elitismo	0,005	0,01	0,05	0,1	0,3
Resultado	$1.0424 \times 10^{-8}$	$2.2925 \times 10^{-8}$	$1.1748 \times 10^{-7}$	$8.9953 \times 10^{-5}$	0,00024

Os resultados obtidos mostraram uma piora constante dos resultados a medida que a taxa de elitismo aumentava, sendo essa piora significativa quando a taxa era 0,3. Analisando os gráficos abaixo, pudemos perceber que o algoritmo com taxa de mutação igual a 0,05 apresentou a lateralização na menor geração, agregando bastante para o algoritmo, visto que queremos minimizar o número de gerações.

Analisando os resultados dos casos, pudemos perceber que sempre que houve a lateralização, o algoritmo já estava próximo do mínimo global, e como o elitismo acaba direcionando o algoritmo a solução ótima. Devido a isso, utilizaremos a taxa de elitismo sendo igual a 0,05 nos demais testes.

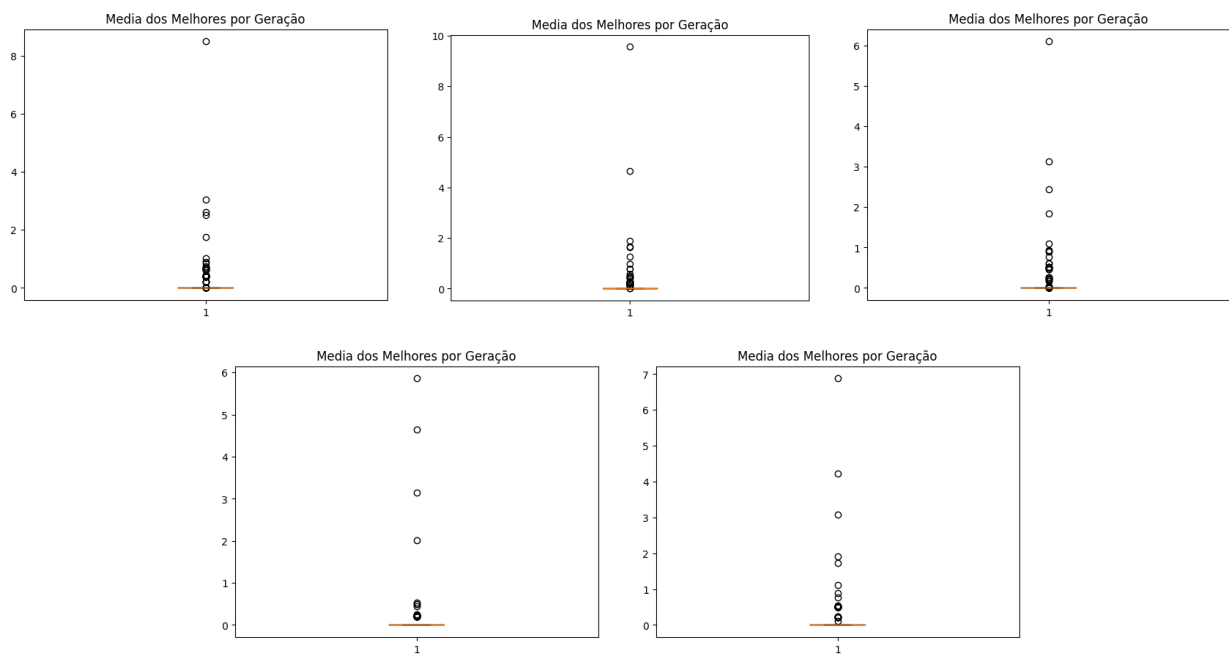


Figura 6: Média dos Melhores por Geração

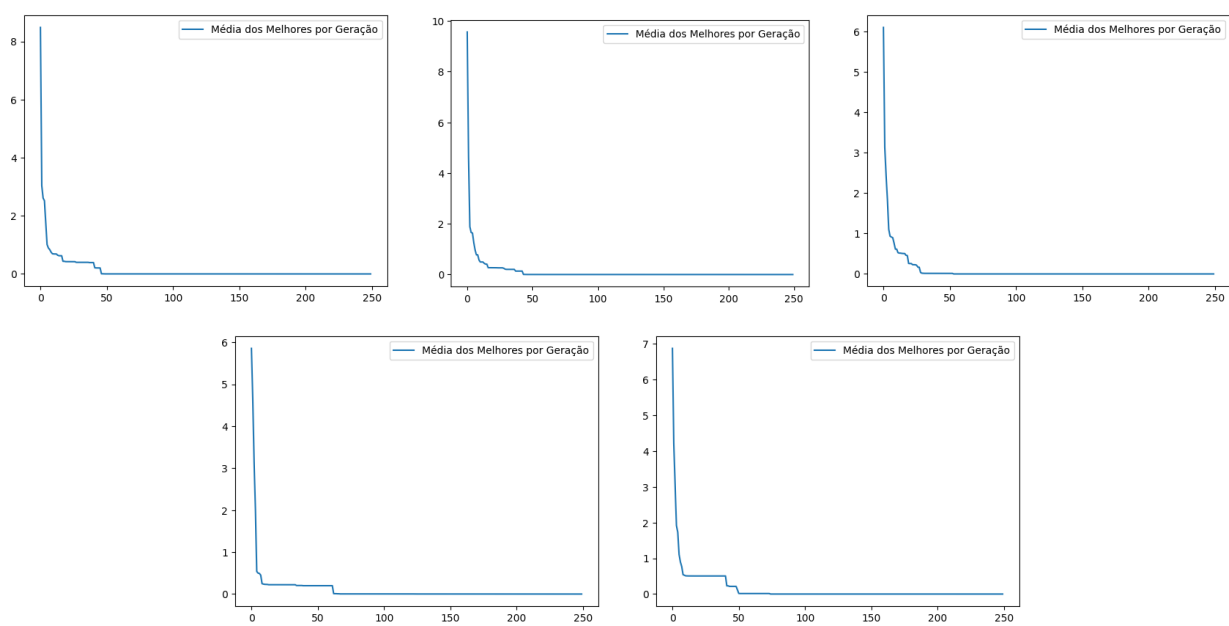


Figura 7: Média dos Melhores por Geração

## 6 Porção de Parentes

Essa variável define a quantidade de soluções da geração anterior que serão mantidas na geração seguinte. Um valor muito alto implica no surgimento reduzido de novas soluções( 0 caso taxa seja 1), já um valor muito baixo faz com que tenham muitas soluções novas.

Parentes	0,055	0,09	0,15	0,3	0,5
Resultado	0,00040	$4.1247 \times 10^{-5}$	$6.6805 \times 10^{-7}$	$3.8675 \times 10^{-7}$	$1.9525 \times 10^{-7}$

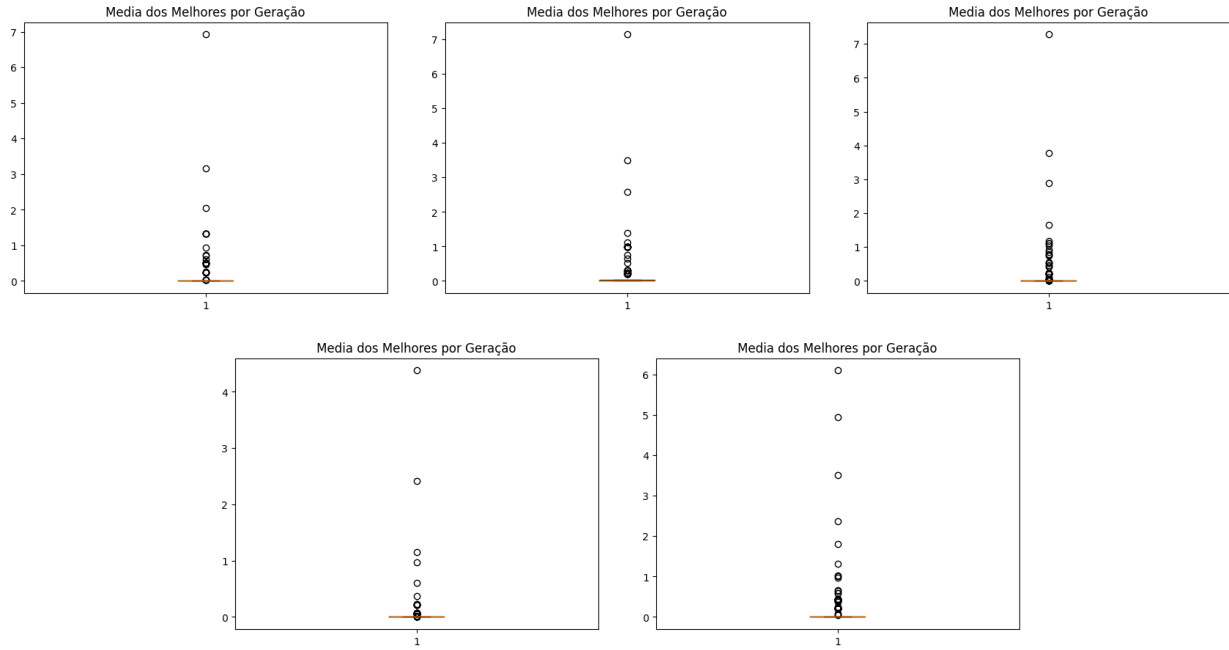


Figura 8: Média dos Melhores por Geração

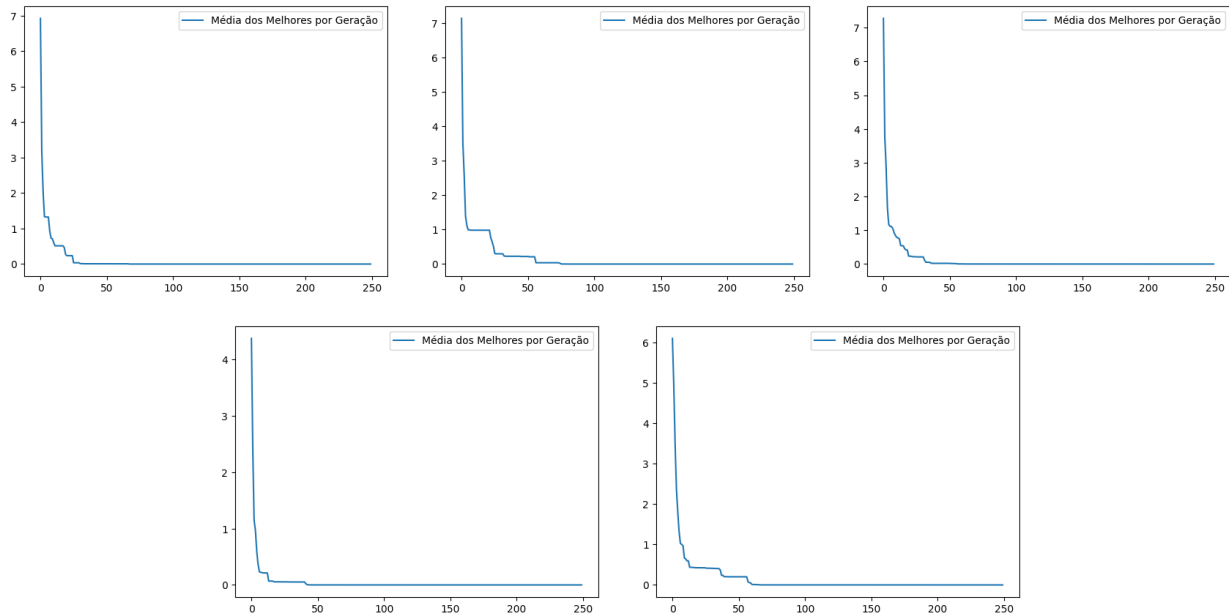


Figura 9: Média dos Melhores por Geração

O valor que apresentou o melhor resultado final foi o com a taxa igual a 0,5. Entretanto, seu resultado não foi significativamente superior ao com a taxa igual a 0,3. Analisando a lateralização do gráfico, o algoritmo com taxa igual a 0,3 apresentou um resultado expressivamente melhor, logo, utilizaremos o valor 0,3 nos próximos testes.



## 7 Tipo de Cruzamento

Na biblioteca utilizada no trabalho, existem 5 tipos de cruzamento. Realizamos os testes dos 5.

Cruzamento	<i>One-Point</i>	<i>Two-Point</i>	<i>Uniform</i>	<i>Segment</i>	<i>Shuffle</i>
Resultado	$5.9699 \times 10^{-8}$	$1.2055 \times 10^{-6}$	$1.4433 \times 10^{-7}$	$6.7708 \times 10^{-9}$	$9.7610 \times 10^{-7}$

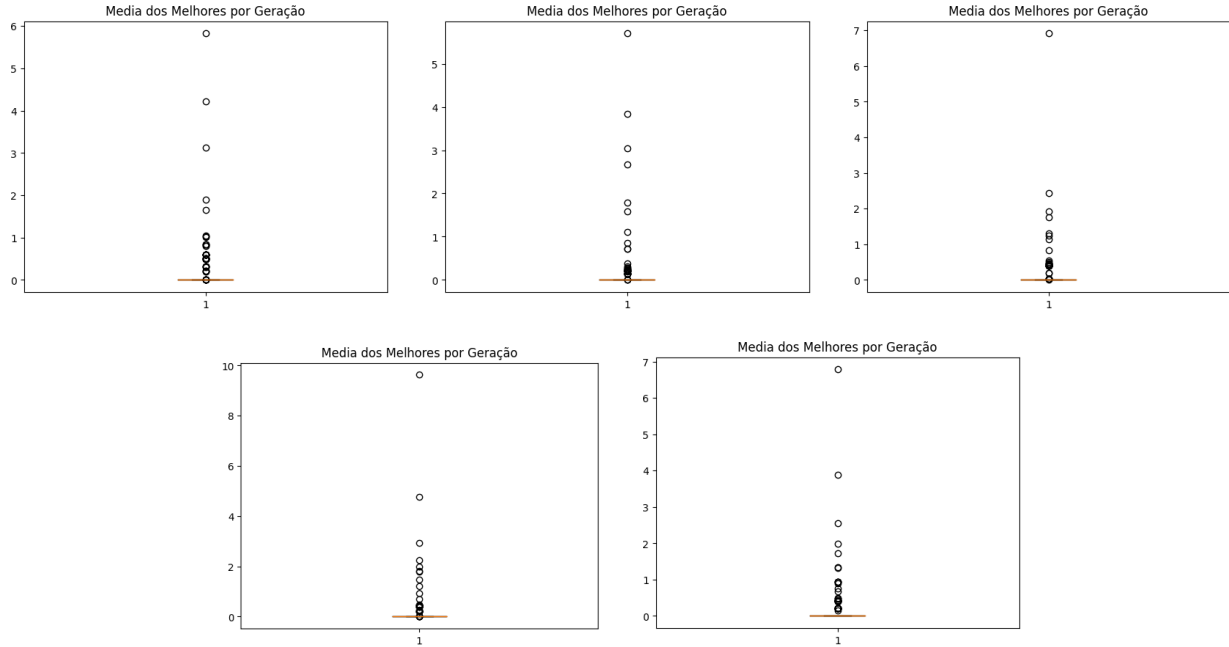


Figura 10: Média dos Melhores por Geração

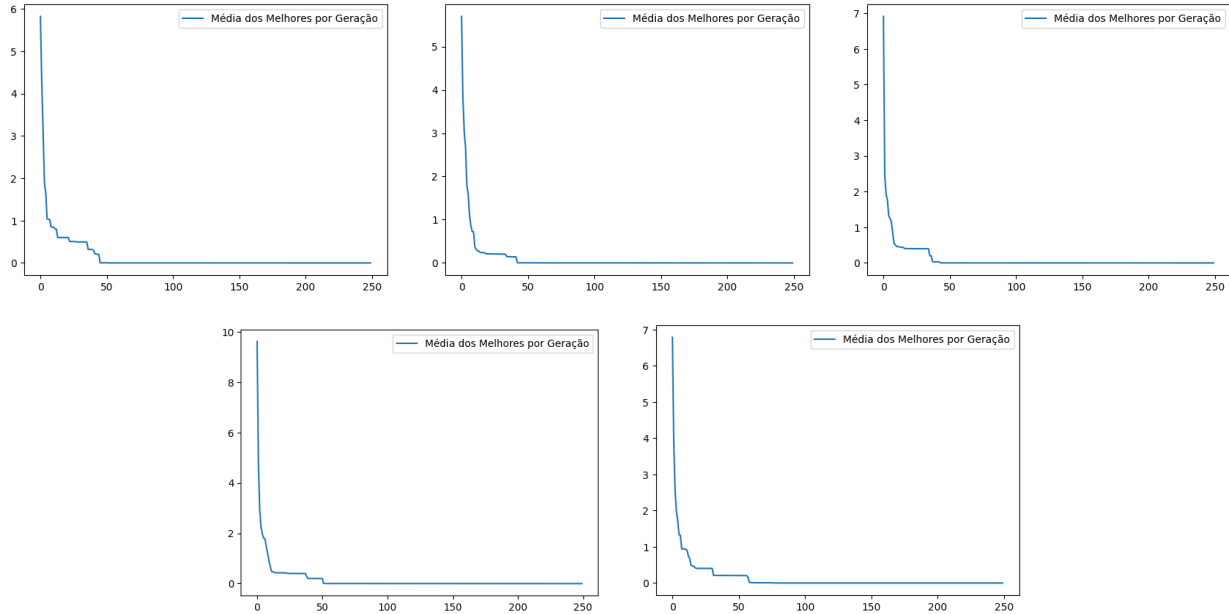


Figura 11: Média dos Melhores por Geração

É possível observar que o tipo de cruzamento com melhor desempenho foi o *Segment* com um valor 10 vezes menor que o penúltimo colocado (*One-Point*), além de possuir uma lateralização melhor que o *One-Point*, chegando ao menor valor mais rápido. Prosseguimos os testes, então, com o tipo de cruzamento *Segment*.

## 8 Tipo de Selecao

Por fim, o último parâmetro textado foi tipo de seleção dos indivíduos, sendo testados os 5 tipos disponibilizados pela biblioteca fornecida.

Seleção	<i>Fully-Random</i>	<i>Roulette</i>	<i>Stochastic</i>	<i>Ranking</i>	<i>Tournament</i>
Resultado	0,00043	$2.3325 \times 10^{-8}$	$8.8984 \times 10^{-5}$	0,00391	0,00181

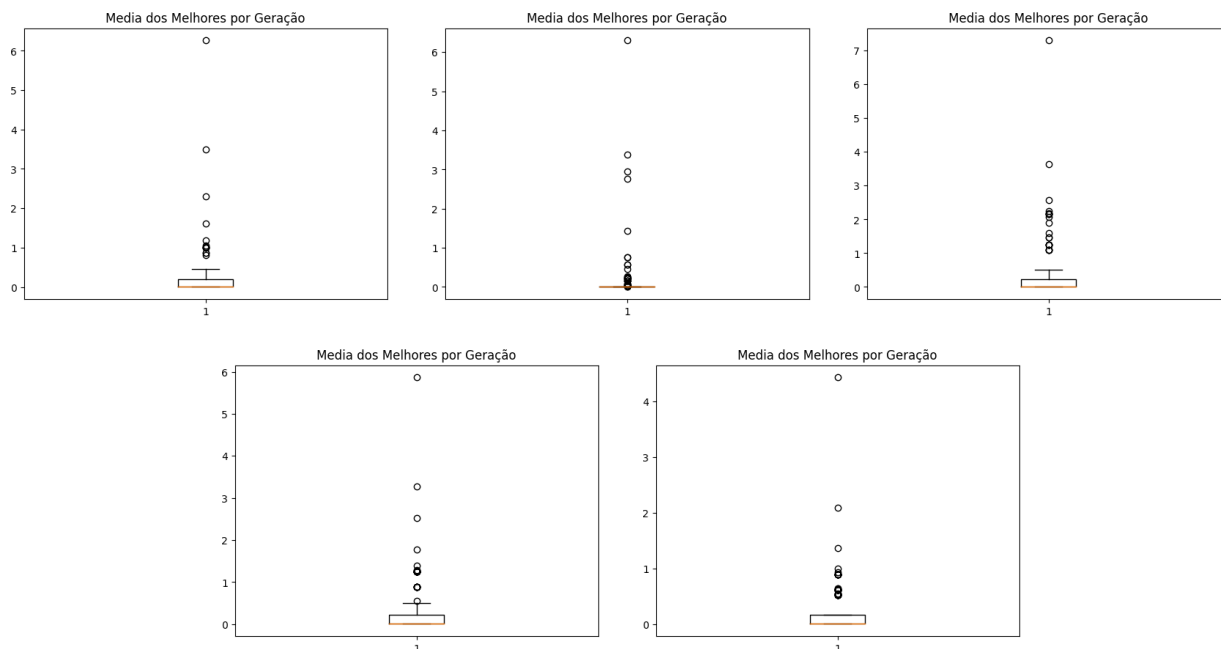


Figura 12: Média dos Melhores por Geração

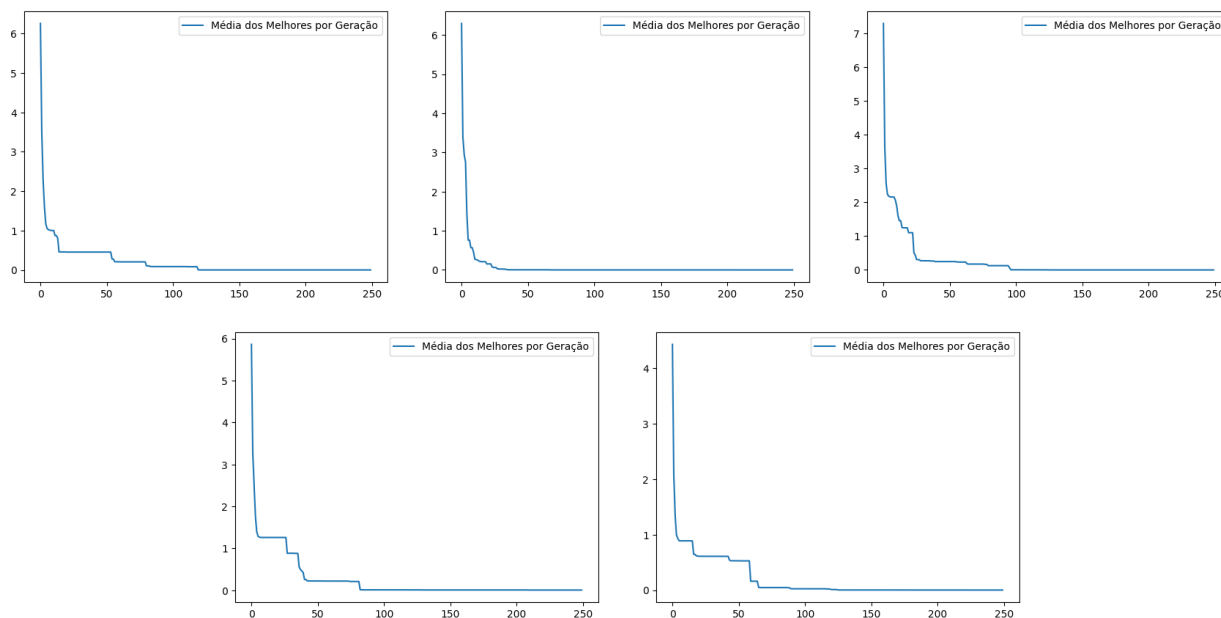


Figura 13: Média dos Melhores por Geração

É observado que o tipo *Roulette* possui um resultado muito mais satisfatório que todos os outros gráficos analisados, tanto por lateralização quanto por resultado final. Esse tipo de seleção agrega muito para o algoritmo genético, visto que ele favorece parametros que agregam para a solucao, ao mesmo tempo que mantém características aleatórias no processo, evitando convergencias em regioes de mínimo local. Portanto, será mantido como um parâmetro final.

## 9 Resultado Final

Após analisar todos os parametros, conseguimos perceber que poderíamos diminuir drasticamente a quantidade máxima de geracoes, aumentando a populacao total. Após realizar análises pontuais, chegamos a esses parâmetros finais. Vale ressaltar que após a definicao de todos os parametros, diminuimos o valor do número máximo de iteracoes, onde um resultado que julgamos aceitável, é  $\leq 0,001$ . Nessa etapada final de escolha dos parametros, aumentamos o número de experimentos para 50, buscando maior precisao nos resultados obtidos, contornando o caráter estocástico do algoritmo. Nessa etapa, também refizemos testes pontuais buscando diminuir o valor mínimo e, conseqüentemente o número máximo de geracoes também. Após essa última etapa de testes mais pontuais, chegamos aos valores finais, que estao explicitados na tabela abaixo.

Dimensão	2
Máximo de Iterações	25
População	250
Probabilidade de Mutação	1.0
Taxa de Elitismo	0.005
Probabilidade de Cruzamento	1.0
Porção de Parentes	0.3
Tipo de Cruzamento	One-Point
Tipo de Seleção	<i>Roulette</i>

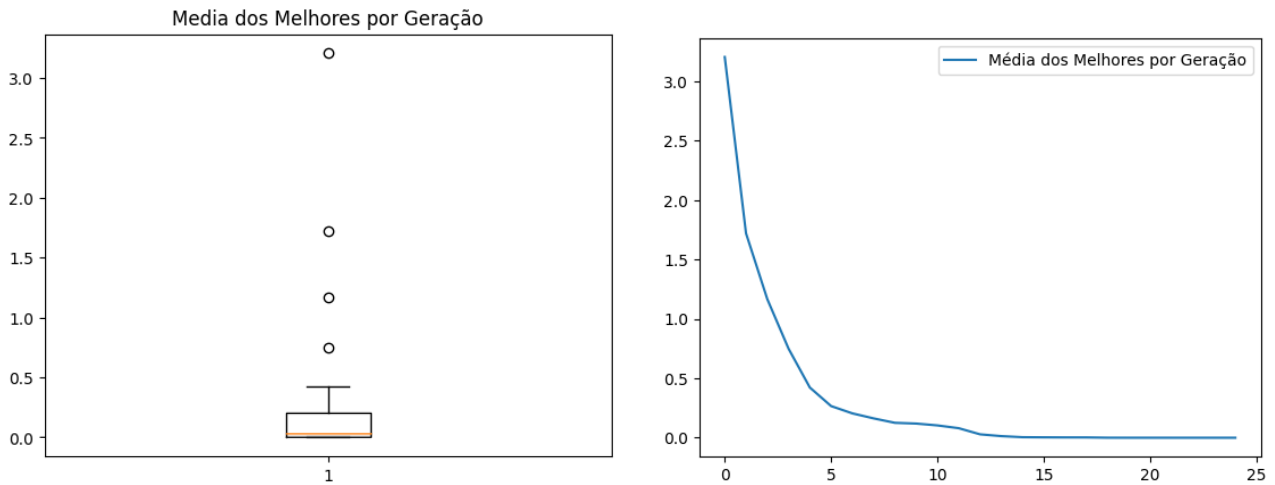


Figura 14: Média dos Melhores por Geração

Com esses valores nos parâmetros, obtivemos o menor valor igual a  $7.4679 \times 10^{-5}$ . Um valor excelente quando comparado com o valor máximo que estipulamos.