



Ministério da Educação
Secretaria de Educação Profissional e Tecnológica
Instituto Federal do Norte de Minas Gerais
Campus Montes Claros
Bacharelado em Ciência da Computação



Relatório do Primeiro Trabalho Prático

Disciplina de Computação Natural

João Vitor Pereira Amorim

Relatório do trabalho apresentado como
pré-requisito para aprovação na
disciplina Computação Natural.

Montes Claros/MG
13 de março de 2022

1) Introdução

O presente documento trata-se do relatório dos resultados de um algoritmo genético, cuja implementação se dá para solucionar um problema onde busca-se maximizar o sinal de saída de uma caixa preta.

A caixa preta em questão possui 9 botões rotativos, sendo que a posição de cada botão pode variar entre 0 e 15, ou seja, 16 posições diferentes. Outra característica dessa caixa preta, é que não é possível analisar o processo interno de transformação do sinal, sendo possível apenas observar e comparar os resultados, para assim tentar-se descobrir qual a combinação de posições dos botões que melhor maximiza o sinal. Caso fosse optado pela utilização desse método para obter a melhor solução, teria-se um processo extremamente ineficaz e humanamente/analiticamente impossível de ser realizado por completo, já que existem 16^9 (aproximadamente $68,71 \times 10^9$) possíveis soluções. É neste ponto que entra em cena o algoritmo genético, eficaz e ideal para problemas onde não é necessário um “ótimo” perfeito, mas sim uma solução próxima do “ótimo” e que já satisfaça a necessidade do problema.

Os algoritmos genéticos funcionam através de estruturas e operadores inspirados no real processo de seleção natural. Desta forma, a principal característica dos algoritmos genéticos é justamente essa simulação da seleção natural, porém aplicada à solução de problemas dos mais diversos tipos. Essa inspiração é totalmente perceptível nas etapas desse tipo de algoritmo, sendo as principais: (1) inicialização de uma população de indivíduos, onde cada indivíduo é representado como um cromossomo; (2) seleção de indivíduos para a reprodução, através de algum método que calcula a adaptação natural (*fitness*) de cada um; (3) reprodução dos indivíduos, onde acontece recombinação de material genético e uma possível mutação em algum gene; (4) seleção dos indivíduos para a próxima geração; (5) repetição dos passos anteriores com a nova geração (população), até atingir-se algum resultado desejado.

Portanto, um algoritmo do tipo genético mostra-se bastante eficaz para o problema da caixa preta, já que através dele é possível processar múltiplas soluções simultaneamente (onde cada solução é um indivíduo da população), ao mesmo tempo em que se seleciona e evolui as melhores, ou seja, as que mais aproximam-se da solução “ótima”.

2) Resultados

Nesta seção são exibidos os resultados obtidos na execução do algoritmo genético. Esse tipo de algoritmo possui diversos parâmetros que devem ser ajustados de acordo com cada problema. Entre esses parâmetros, inclui-se: quantidade de execuções do algoritmo para cada configuração¹; quantidade de indivíduos na população; quantidade de gerações (iterações do algoritmo); método para a seleção de indivíduos para reprodução; taxa de mutação; método para a seleção de indivíduos para a próxima geração.

Os seguintes parâmetros foram fixos em todas as execuções do algoritmo:

- Quantidade de execuções do algoritmo para cada configuração: 1000;
- Quantidade de indivíduos na população: 30;
- Quantidade de gerações: 40;
- Método para a seleção de indivíduos para a próxima geração: elitismo².

O parâmetro de taxa de mutação foi alterado conforme necessário e indicado nos resultados. Quanto ao parâmetro “método de seleção de indivíduos para reprodução”, necessitou-se definir qual o mais adequado para o problema em questão através da realização de alguns testes exibidos na próxima subseção.

Uma informação a se notar em todos os resultados exibidos é a respeito da quantidade de vezes que o melhor *fitness* foi atingido em cada configuração do algoritmo. Esse valor, apresentado abaixo do título de cada gráfico, trata-se de quantas vezes algum dos indivíduos da última geração de cada execução atingiu o melhor *fitness* possível, que no caso deste problema é o valor 27³. Essa informação é importante para avaliar a eficiência de cada configuração do algoritmo.

2.1) Definição do melhor método de seleção de indivíduos

Para definir qual o melhor método de seleção de indivíduos para a reprodução, a taxa de mutação foi fixada no valor 0,03 e foram gerados resultados para cada um dos três

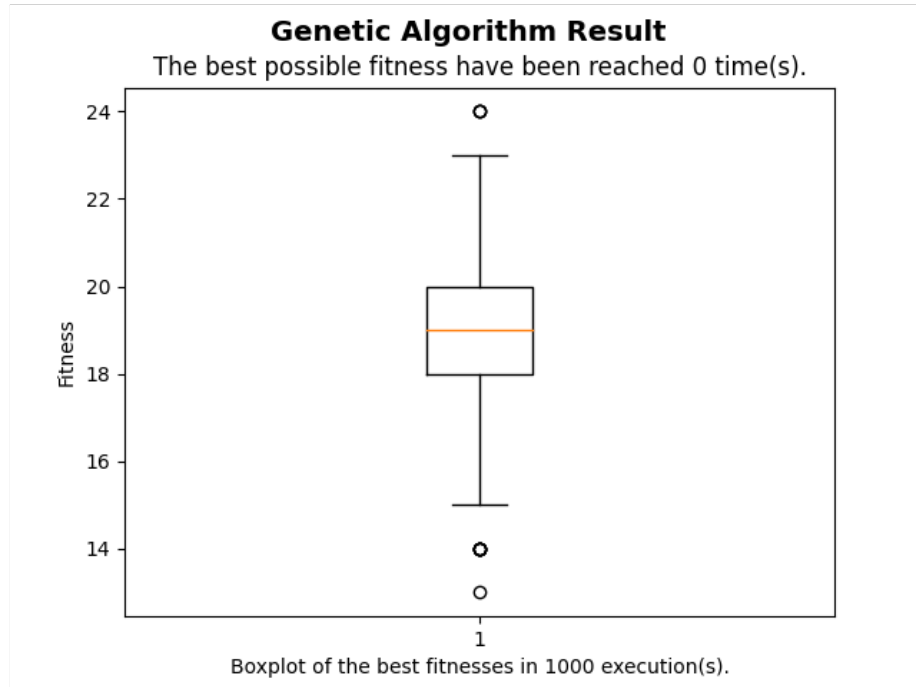
¹ Neste relatório, o termo “configuração” é utilizado para se referir a combinação de parâmetros utilizada na geração de cada um dos gráficos de resultado.

² Selecionou-se o melhor indivíduo da geração atual e os outros N - 1 filhos gerados durante a reprodução da geração atual, sendo que o “- 1” refere-se à não inclusão do filho com pior *fitness*.

³ O valor 27, melhor *fitness* possível para o problema em questão, corresponde aos seguintes valores para os 9 botões da caixa preta, respectivamente: 15, 11, 09, 11, 15, 11, 15, 02 e 15.

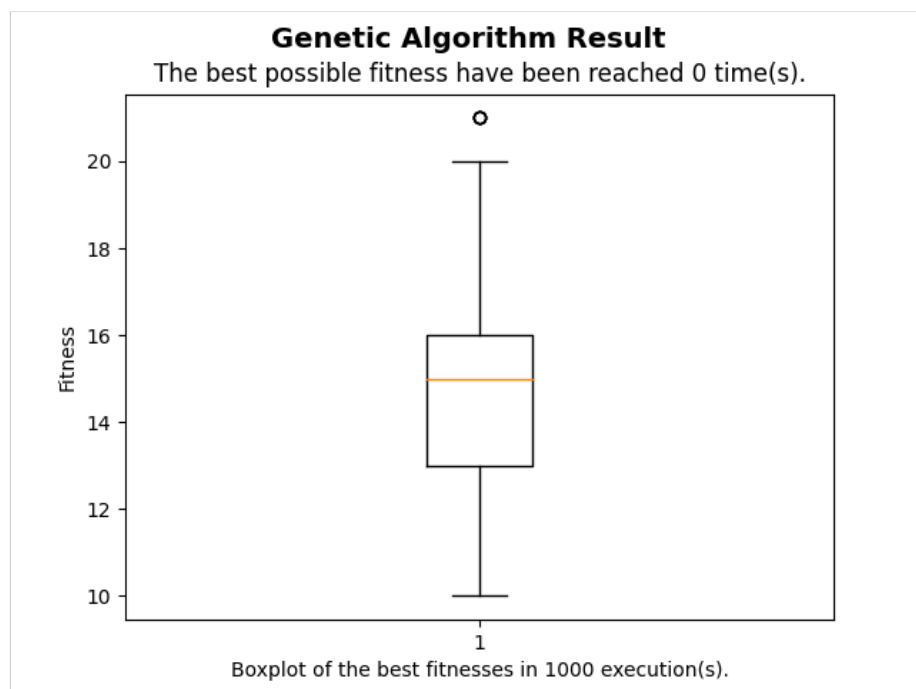
métodos selecionados como possíveis neste algoritmo: *roulette wheel*, uniforme e torneio binário.

Figura 1 - Resultados utilizando o método de seleção *roulette wheel*, com taxa de mutação de 0,03.



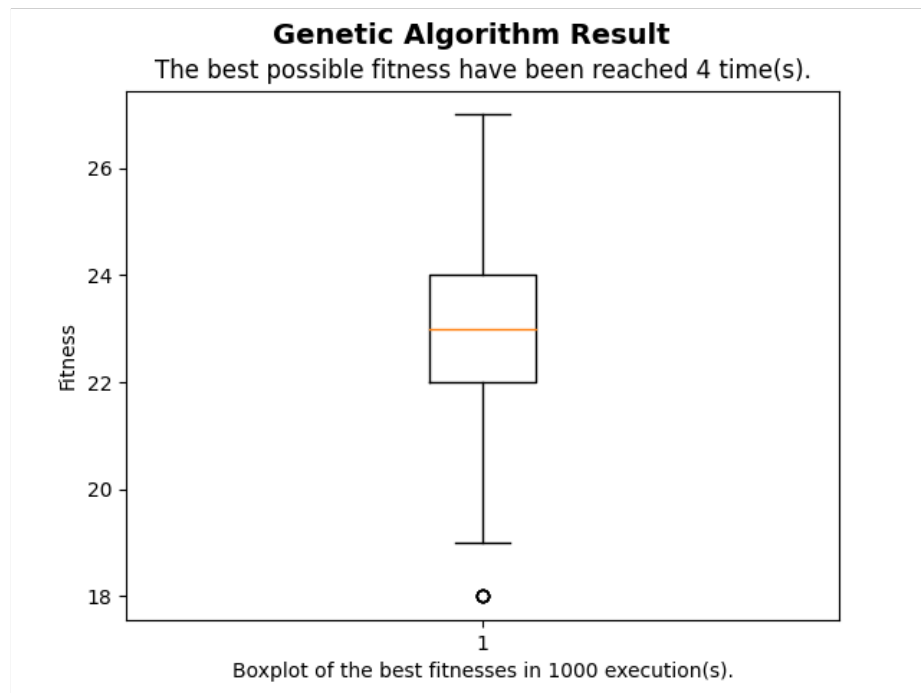
Fonte: produzido pelo autor, 2022.

Figura 2 - Resultados utilizando o método de seleção uniforme, com taxa de mutação de 0,03.



Fonte: produzido pelo autor, 2022.

Figura 3 - Resultados utilizando o método de seleção de torneio binário, com taxa de mutação de 0,03.



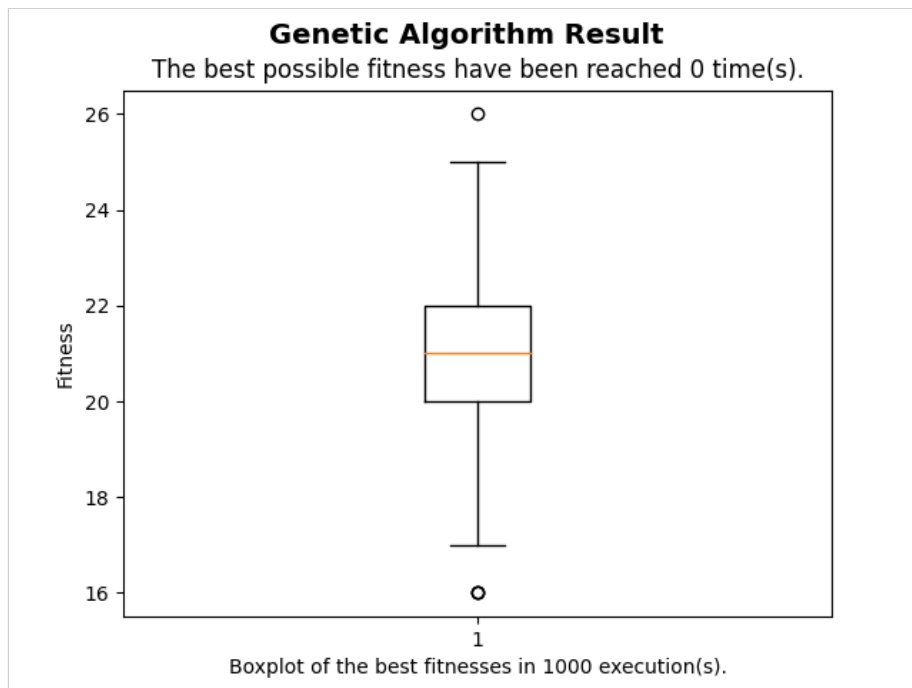
Fonte: produzido pelo autor, 2022.

Como é possível observar, o método de torneio binário obteve os melhores resultados, até mesmo atingindo o melhor *fitness* possível em 4 das 1000 execuções. Devido a isso, esse método será utilizado em todos os resultados exibidos na próxima subseção.

2.2) Influência da taxa de mutação nos resultados

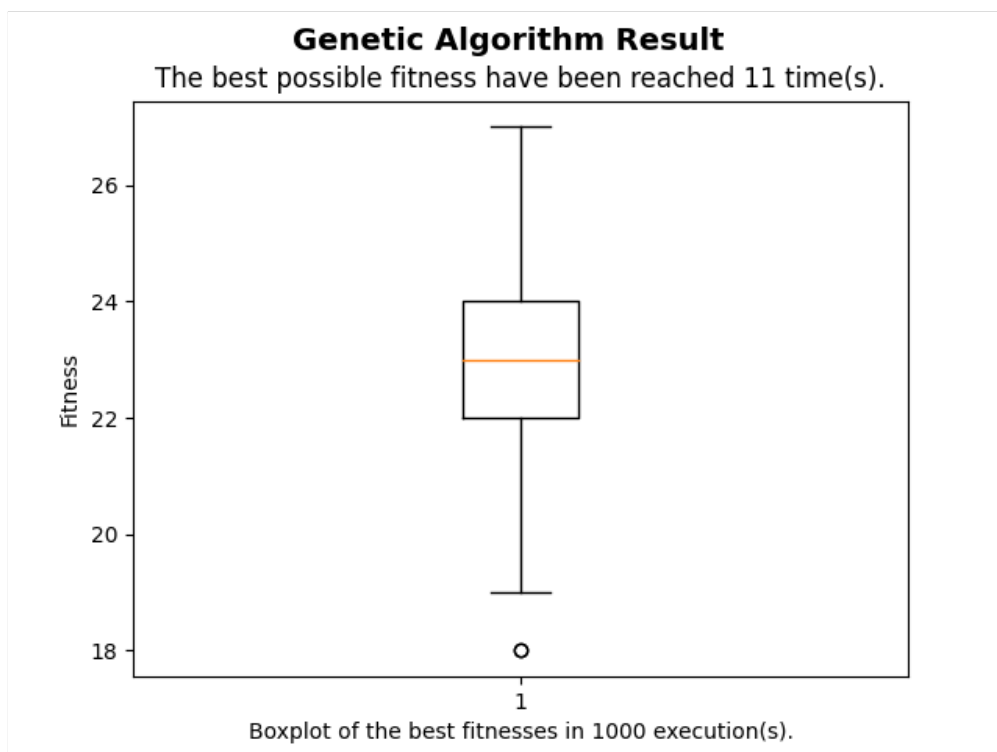
Definido o método de torneio binário como o melhor para o problema atual, nesta subseção é analisada a influência da taxa de mutação nos resultados utilizando o método citado.

Figura 4 - Resultados utilizando o método de seleção de torneio binário, sem mutação.



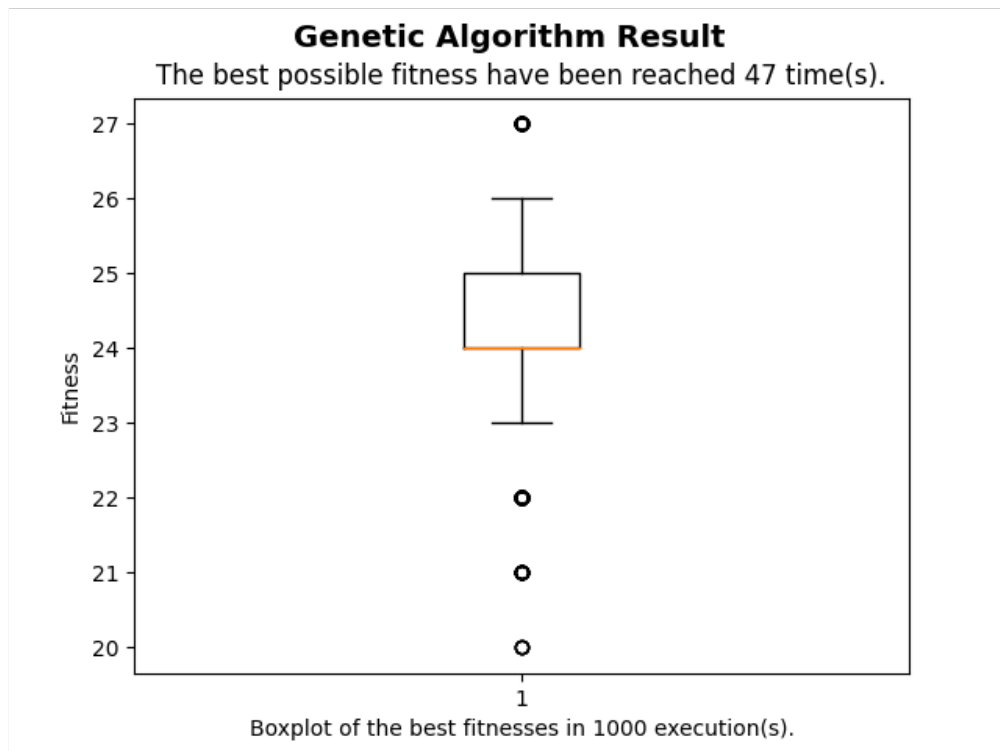
Fonte: produzido pelo autor, 2022.

Figura 5 - Resultados utilizando o método de seleção de torneio binário, com taxa de mutação de 0,05.



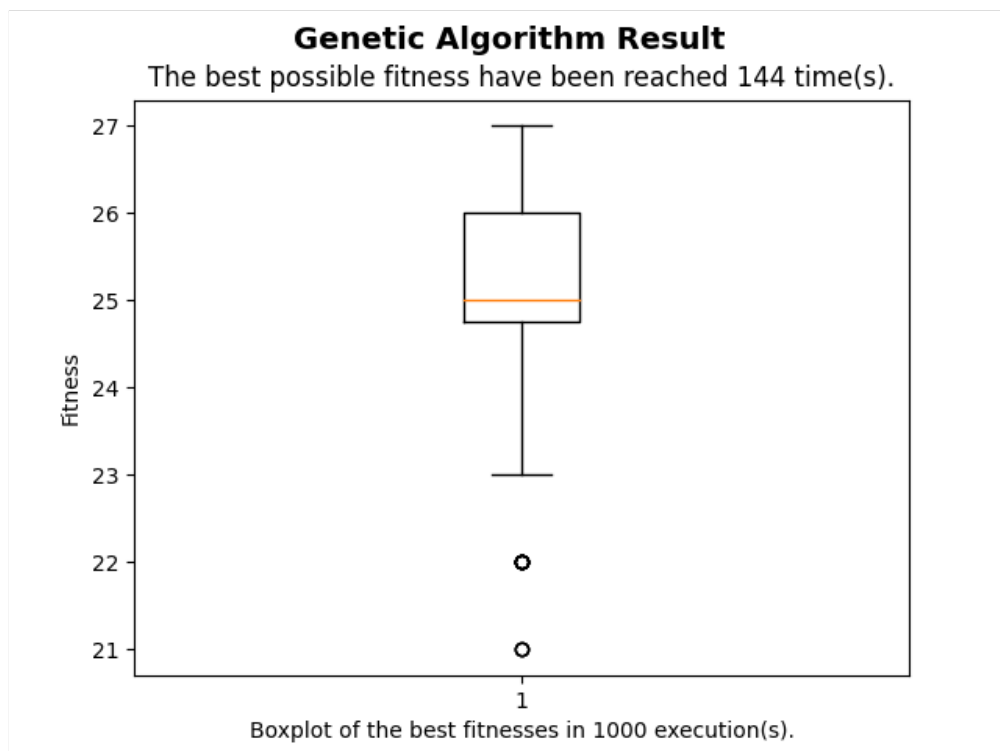
Fonte: produzido pelo autor, 2022.

Figura 6 - Resultados utilizando o método de seleção de torneio binário, com taxa de mutação de 0,1.



Fonte: produzido pelo autor, 2022.

Figura 7 - Resultados utilizando o método de seleção de torneio binário, com taxa de mutação de 0,4.



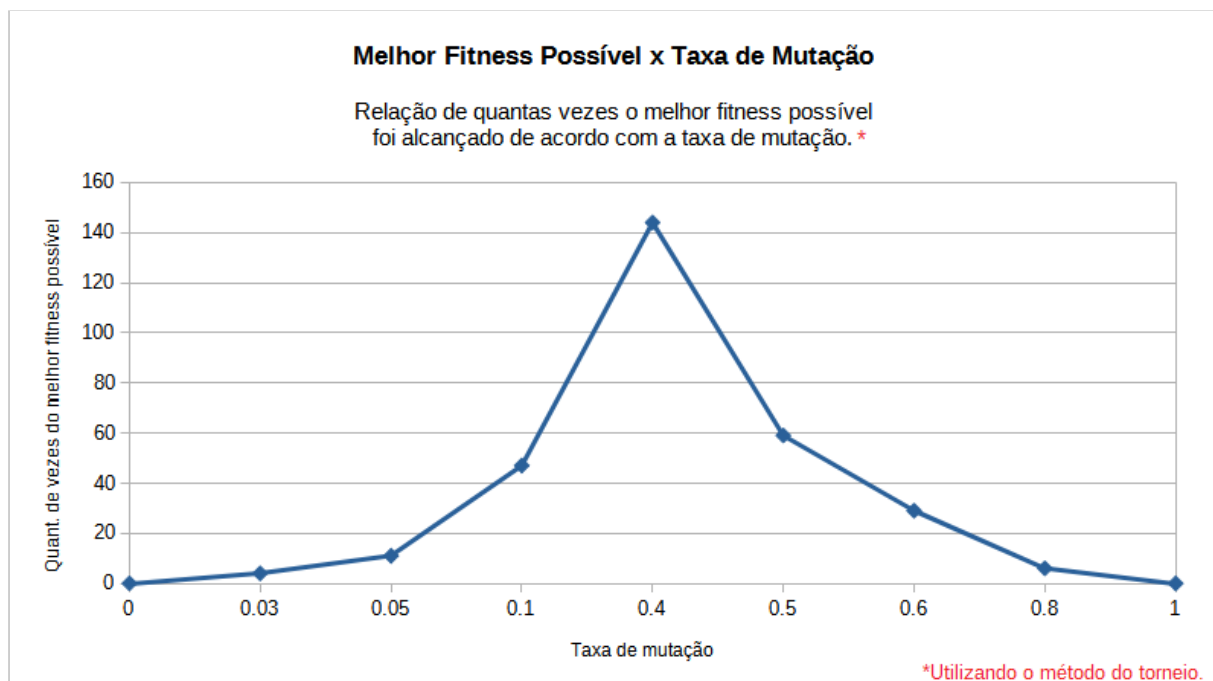
Fonte: produzido pelo autor, 2022.

Observando os resultados acima, percebe-se uma clara influência do aumento da taxa de mutação na quantidade de vezes que o melhor *fitness* foi atingido. Saiu-se de 4 vezes do melhor *fitness*, na taxa de mutação igual a 0,03, para 144 vezes, na taxa de mutação igual a 0,4. Mas será que existe algum ponto de inflexão, onde a alta taxa de mutação passa a influenciar negativamente no algoritmo? A resposta é sim, conforme exibida na próxima subseção.

2.3) Relação entre a quantidade de vezes do melhor *fitness* possível e a taxa de mutação

O melhor valor para a taxa de mutação para o problema em questão é por volta de 0,4. Para valores inferiores a este, a quantidade de vezes que o melhor *fitness* possível é obtido é inferior, conforme demonstrado na subseção anterior. Da mesma forma, para valores acima de 0,4, a alta taxa de mutação passa a influenciar negativamente no algoritmo, conforme testes conduzidos. No gráfico abaixo, é possível observar como a taxa de mutação influencia diretamente nos resultados do algoritmo genético para o problema em questão.

Figura 8 - Quantidade do melhor *fitness* possível vs. taxa de mutação, utilizando o método de torneio binário.



Fonte: produzido pelo autor, 2022.

3) Conclusão

Após diversas execuções do algoritmo, para diferentes métodos de seleção e valores para a taxa de mutação, chegou-se à melhor configuração para o problema em questão. A melhor configuração obtida é utilizando o método de torneio binário e a taxa de mutação de 0,4. Com essa configuração, é possível obter o melhor *fitness* possível, ou seja, a solução que mais se aproxima do “ótimo”, em cerca de 15% das execuções do algoritmo.