



Ministério da Educação
Secretaria de Educação Profissional e Tecnológica
Instituto Federal do Norte de Minas Gerais
Campus Montes Claros- Bacharel em Ciência da
computação



Relatório do Primeiro Trabalho Prático da Disciplina Computação Natural

Filipi Maciel Rodrigues Jardim

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

1) Introdução

O problema da caixa preta consiste em uma caixa com nove botões. Cada botão pode ser posicionado em 16 valores indo de 0 até 15. Devemos gerar uma combinação de valores para cada botão, que após aplicar um cálculo matemático(função fitness) irá gerar um valor inteiro, visando se aproximar do valor ótimo já calculado de 27.

2) Metodologia

Para resolver o problema foi feito o desenvolvimento de um algoritmo genético básico, que vai utilizar três métodos diferentes de seleção e quatro taxas diferentes de mutação. De forma geral, a sequência de operações ficou da seguinte forma:

- 1) Gera 30 indivíduos
 - a) Cria uma matriz 30 x 9
 - b) Preencho a matriz com valores aleatórios de 0 à 15
 - c) Converto esses valores para binário, armazenando em outra matriz 30 x 36
- 2) Entro em um loop de repetição que corresponde as 40 gerações solicitadas
 - a) Realizo a seleção (torneio binário, uniforme ou roleta)
 - i) O resultado da seleção é um vetor de ganhadores, que contem as posições na matriz de indivíduos
 - b) Realizo o cruzamento e gero os filhos, guardo numa matriz 30 x 36
 - c) Testo para saber se vai ter mutação, se tiver, realizo a inversão de um bit aleatório em algum individuo aleatório
 - d) Faço a troca do pior filho da geração atual pelo melhor pai da geração anterior
 - e) A geração de filhos se torna a nova geração de pais
- 3) Salvo alguns valores referentes aos melhores individuos para serem usados no boxplot

Dessa forma, o algoritmo será executado 10 vezes dentro de um loop de repetição.

3) Resultados

Os resultados serão apresentados da seguinte forma:

- 1) Comparação de métodos de seleção:
 - a) 10 execuções com torneio binário, com mutação em 3%
 - b) 10 execuções com torneio roleta, com mutação em 3%

- c) 10 execuções com seleção uniforme, com mutação em 3%
- 2) Identifico o melhor método de seleção do passo anterior, comparo:
 - a) 10 execuções com melhor seleção, com mutação em 3%
 - b) 10 execuções com melhor seleção, sem mutação
- 3) Testo outras três taxas de mutação diferentes no melhor método de seleção:
 - a) 10 execuções com melhor seleção, com mutação em 5%
 - b) 10 execuções com melhor seleção, com mutação em 10%
 - c) 10 execuções com melhor seleção, com mutação em 40%

Parte I

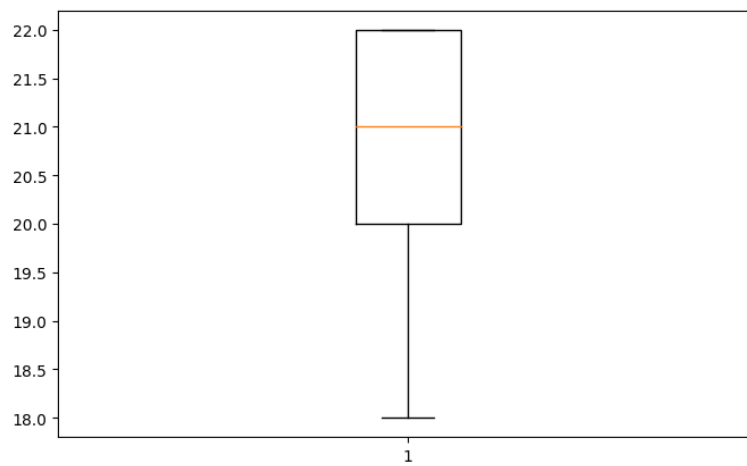
- a) 10 execuções com torneio binário, com mutação em 3%

Para entendermos melhor o gráfico vamos olhar primeiro para o resultado do algoritmo nas suas dez execuções.

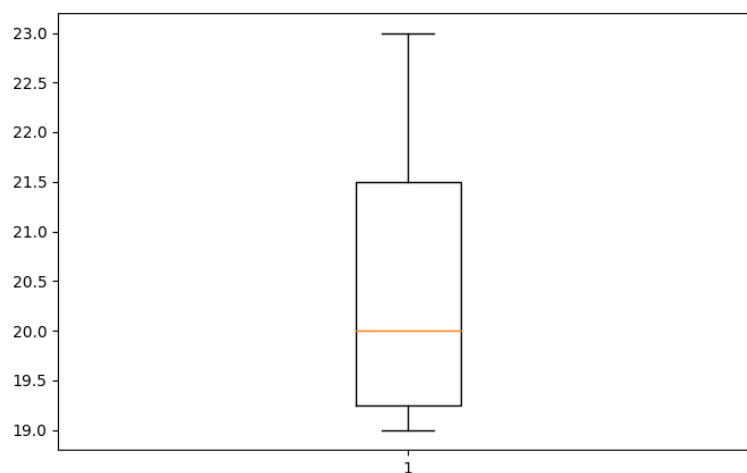
----- Iteração - 0 melhor -> 18 -----	----- Iteração - 5 melhor -> 22 -----
----- Iteração - 1 melhor -> 20 -----	----- Iteração - 6 melhor -> 22 -----
----- Iteração - 2 melhor -> 20 -----	----- Iteração - 7 melhor -> 22 -----
----- Iteração - 3 melhor -> 20 -----	----- Iteração - 8 melhor -> 22 -----
----- Iteração - 4 melhor -> 20 -----	----- Iteração - 9 melhor -> 22 -----

Na primeira execução o melhor fitness obtido foi o 18. Na segunda, o 20, na terceira manteve o 20, e assim por diante até chegar na última execução com o valor 22. Ou seja, de dez execuções realizadas, na sexta execução (iteração 5), foi atingido o valor fitness 22, e após isso não teve nenhum valor melhor que ele nas outras 4 iterações restantes.

Olhando agora para o gráfico gerado, percebemos que a concentração de valores está entre 20 e 22, o que faz sentido, pois o valor 18, apesar de ter sido considerado pelo gráfico na base da barra, foi gerado na primeira iteração e considerado o melhor somente naquele momento. Outros valores, como 18, 19, e até menores como 16 e 15 podem ter sido gerados, mas como o melhor fitness de valor 20 tinha sido encontrado eles são descartados.



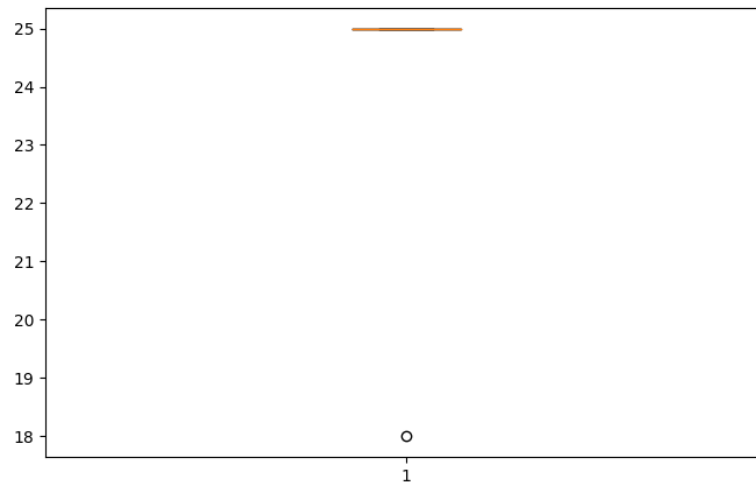
b) 10 execuções com torneio roleta, com mutação em 3%



Aqui podemos notar que a concentração de valores ficou entre 20 e 22, o valor 23 foi obtido na última iteração, o 19 foi mantido nas três primeiras rodadas, por isso esses valores estão nos extremos do gráfico.

Uma observação a ser feita é que o algoritmo com torneio roleta de mostrou mais lento na execução.

c) 10 execuções com seleção uniforme, com mutação em 3%

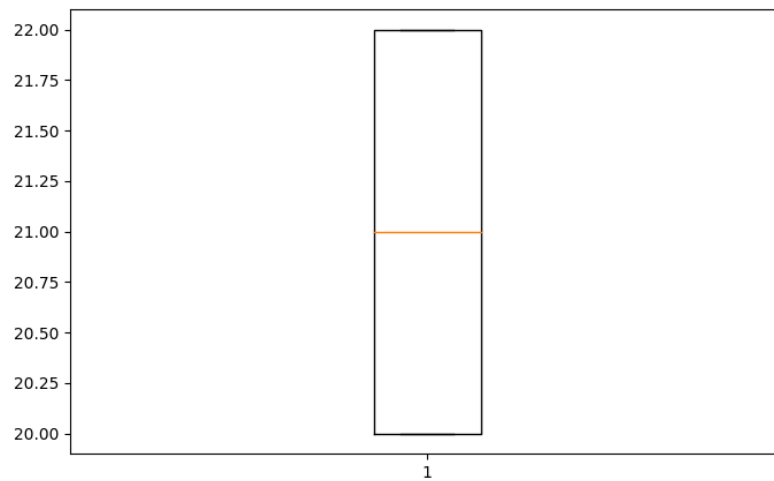


Apesar de parecer estranho o gráfico está correto, pois nessa execução o valor da primeira iteração foi o fitness 18, mas na segunda iteração foi encontrado o fitness 25, de modo que nenhuma outra iteração encontrou um valor maior que esse. Sendo assim, a concentração de valores ficou no 25, e a bolinha branco informa que um valor 18 foi obtido mas está distante de onde temos a aglomeração de resultados.

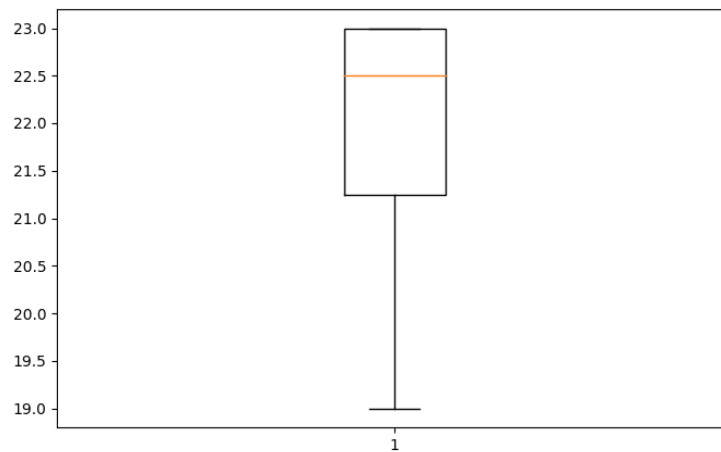
Parte II

O melhor método de seleção foi o uniforme, que chegou a atingir o valor 25. Mas esse não foi o único fator de escolha, durante os testes foi observado que as seleções por torneio binário e roleta ficavam sempre em torno de fitness 20, tendo mais dificuldade para atingir valores superiores. Já no método uniforme a frequência de valores altos foi maior, como 23, 24, 25 e até um 26, encontrado em testes anteriores a escrita desse relatório.

a) 10 execuções com melhor seleção, com mutação em 3%



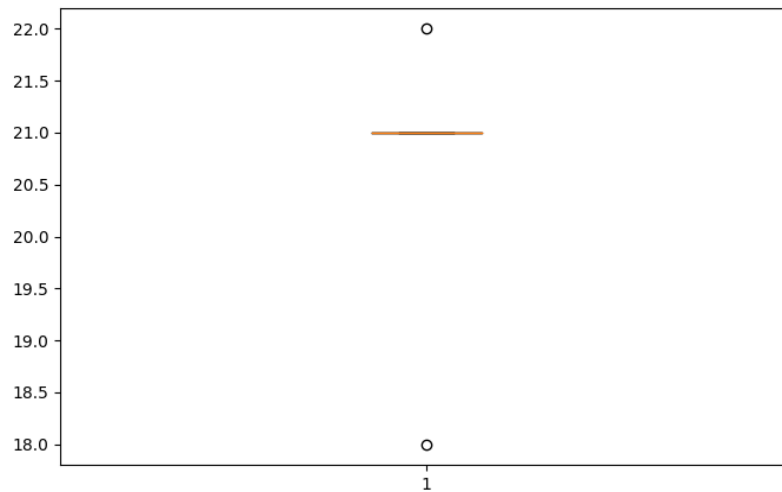
b) 10 execuções com melhor seleção, sem mutação



Demonstrando o que foi descrito acima, valor 23 foi encontrado como o melhor fitness, tendo o 19 sido encontrado apenas na primeira iteração.

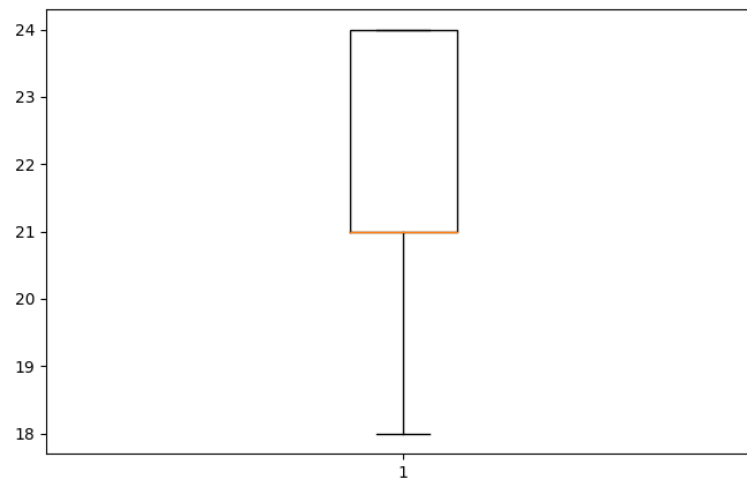
Parte III

- a) 10 execuções com melhor seleção, com mutação em 5%

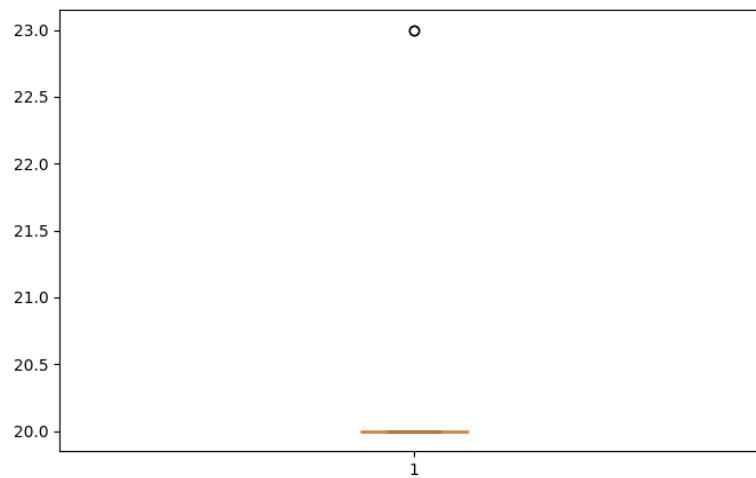


Comportamento dentro do que se imaginava, sem a mutação provavelmente o resultado melhor seria 21, mas em uma das iterações provavelmente alguma mutação ocorreu, fazendo um indivíduo gerar um fitness de 22.

- b) 10 execuções com melhor seleção, com mutação em 10%



c) 10 execuções com melhor seleção, com mutação em 40%



Resultado interessante, pois tudo indicava uma permanência em torno de 20, mas uma provavelmente uma mutação fez um indivíduo obter um fitness melhor.

Outra observação é de que esse comportamento foi frequentemente mais presente em execuções com mutações.

4) Conclusão e Discussão

Podemos concluir que, para essa série de execuções e testes, o algoritmo de seleção uniforme se saiu melhor nos resultados. De forma geral foi possível chegar em um valor satisfatório que foi o fitness 25, mas não foi possível atingir o 27.

Devido a quantidade de vetores e matrizes utilizadas o algoritmo tem uma velocidade reduzida, mas foi plenamente possível realizar diversos testes, inclusive fazendo um certo estresse na minha máquina, na tentativa de obter o melhor fitness com iterações extremamente elevadas.

Indo um pouco além dos testes propostos pelo trabalho, resolvi testar outras condições para observar melhor o comportamento do AG desenvolvido.

Testando uma execução com 1000 iterações, onde cada uma delas vai testar 40 gerações, sem taxa de mutação, o melhor fitness encontrado para seleção uniforme foi de 25, para torneio binário foi 24, e para roleta foi de 24. Em outras palavras, aumentar as iterações vai ajudar a chegar no ótimo, pois quanto mais testes mais chances de se encontrar ele.

Testei aumentar o número de indivíduos, colocando 100 e com taxa de mutação em 40%, o que ocasiona de convergi pra 24 ou 25 e até o 26 de forma mais rápida, onde esse último foi encontrado via seleção uniforme, porém o 27 não foi encontrado.

Minha hipótese sobre o motivo do 27 ser tão difícil de se encontrar na implementação, já que alguns colegas indagaram que o 27 estava sendo encontrado muito facilmente no algoritmo deles, é de que o AG está simulando um comportamento bem próximo do real dado suas condições. Mutações na natureza são raras, como descrito nesse artigo da [Scientific American](#). Logo, o AG não vai se pendurar em mutações. Encontrar o indivíduo que resulte em fitness 27 vai depender de uma situação realmente aleatória, pois 30 indivíduos é pouco. O que reforça esse último ponto é justamente o teste anterior, que resultou em 26 na iteração 858.

Ademais, o trabalho foi muito interessante de ser desenvolvido, estimulou bastante a pesquisa e realização de testes por curiosidade, que contribui bastante para a absorção de conhecimento.