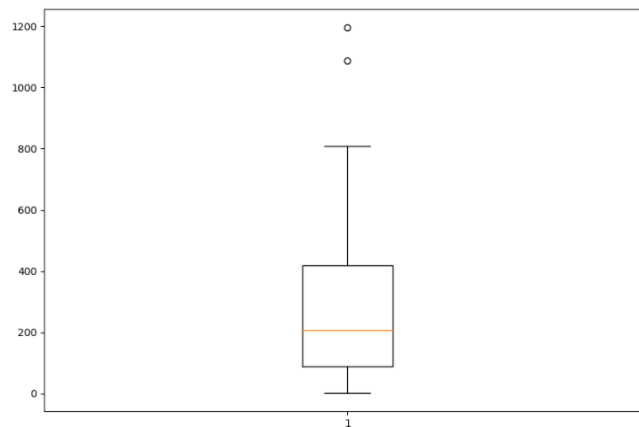


Relatório Ficha 2 – Evolutionary Algorithms Exercise
Carlos Guerra – nº78299 METI

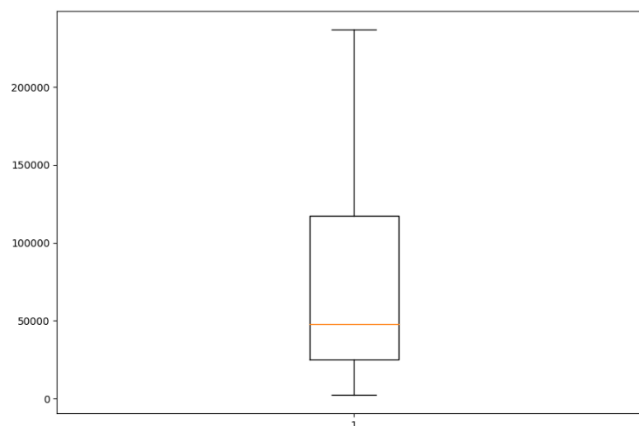
- 1) Esta função está feita na generate(x) que recebe o tamanho pretendido
- 2) Para utilizar a função na sua forma mais básica de ser aleatória é necessário chamar a função start(), devido a limitações de software e hardware e para evitar ter de esperar grandes períodos de tempo para cada obtenção de resultados, apenas irei utilizar para 8 e 16 bits. Obtendo assim os seguintes resultados:

a. Para 8 bits:



```
DONE 8  
TENTATIVAS MÉDIAS PARA 8 BITS: 7.131756286096566
```

b. Para 16 bits:



```
DONE 16  
TENTATIVAS MÉDIAS PARA 16 BITS: 2251.171510297718
```

- 3) A função de avaliação é a chamada `evaluation()`. Podemos utilizar esta função para melhorar os tempos de pesquisa se medirmos a distância até à combinação de objetivo e através dela medir os bits que faltam para atingir o valor desejado.
- 4)
- 5) A função pedida está como `mutation`, nesta função é nos pedido um `target` e uma lista para mutação, a função vai dar flip de um bit num sitio ao calhas e vai avaliar, se a avaliação com a mutação for superior ao melhor valor até ao momento esse passa a ser o valor do bit e o seu score, se o valor da avaliação for igual à melhor avaliação significa que já temos o melhor valor
- 6) Esta função tem como objetivo gerar uma população e de ela escolher os 30% melhor e gerar uma nova população apartir de mutação. Esta função está definida como `generation_mutation`.
- 7) Esta função está definida como `crossoverITALL()`, ao correr esta função deparei-me com um problema que não consegui resolver, chegando de uma certa geração o software (Pycharm) com mais de 16 bits dá erro de memória mesmo utilizando meu hardware 1 (ver [READ.ME](#)) algo que não está relacionado com o hardware, no entanto utilizando para 8 bits obtenho estes resultados:

```
10101010
MELHOR DA GERAÇÃO 1: 10001010

SEGUNDO MELHOR DA GERAÇÃO 1: 10111010

MELHOR DA GERAÇÃO 2: 10101010

SEGUNDO MELHOR DA GERAÇÃO 2: 10001010

10101010
stop
2
```

Pelos resultados podemos observar que através deste método é possível obter bastante rapidez e precisão nos resultados obtidos, utilizando apenas 2 gerações em que logo na primeira geração os valores eram bastantes próximos do objetivo

- 8) Se o tamanho do padrão alvo for desconhecido as funções teriam de mudar da seguinte forma:
 - a. Avaliação teria de se ver bit a bit a proximidade com o alvo e apartir desse valor gerar uma avaliação segundo uma regra
 - b. Mutação não teria de ser muito mudada visto que a mutação apenas tem como objetivo mudar um bit no padrão que está a ser dado como input e não o bit alvo
 - c. O crossover também na minha opinião não sofreria muitas alterações visto que utiliza valores como input e não vai ver os valores alvos