

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

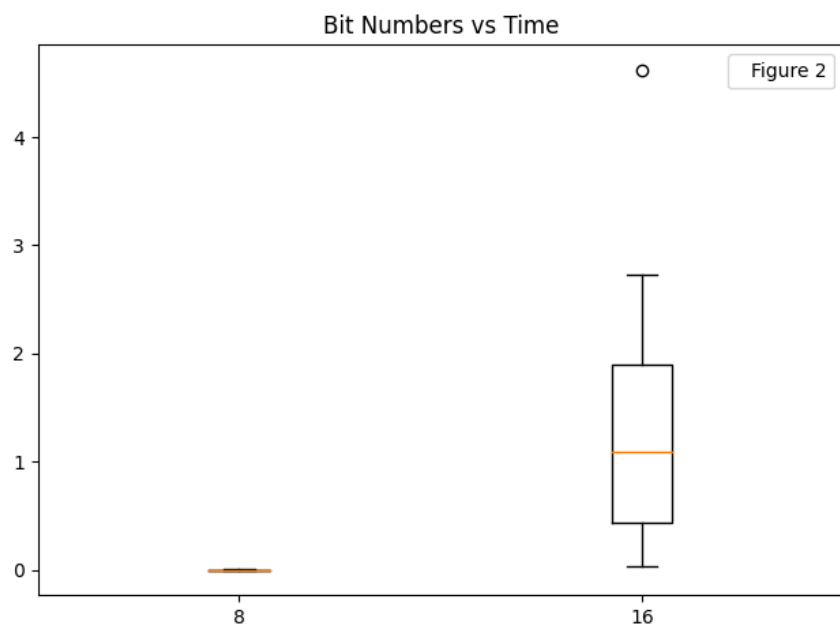
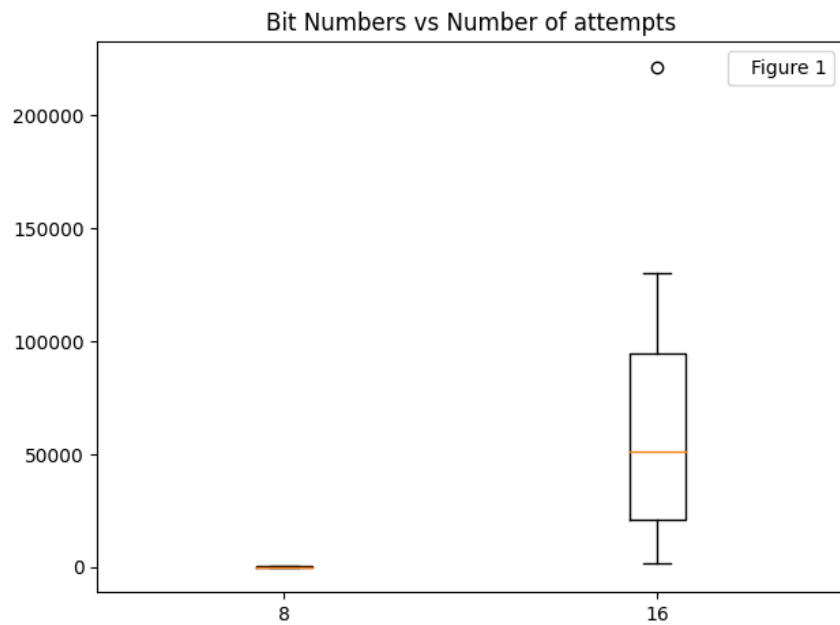
Todos os resultados apresentados correspondem à seed com o valor de 1611491668.914867, ao correr o código inteiro é apresentada a seed que vai ser utilizada. Para utilizar uma seed específica é necessário mudar a linha 18 utilizando a seed que for pretendida

## **Pergunta 1**

O código correspondente a esta pergunta é a função generate(x), que recebe um valor x que corresponde ao tamanho pretendido

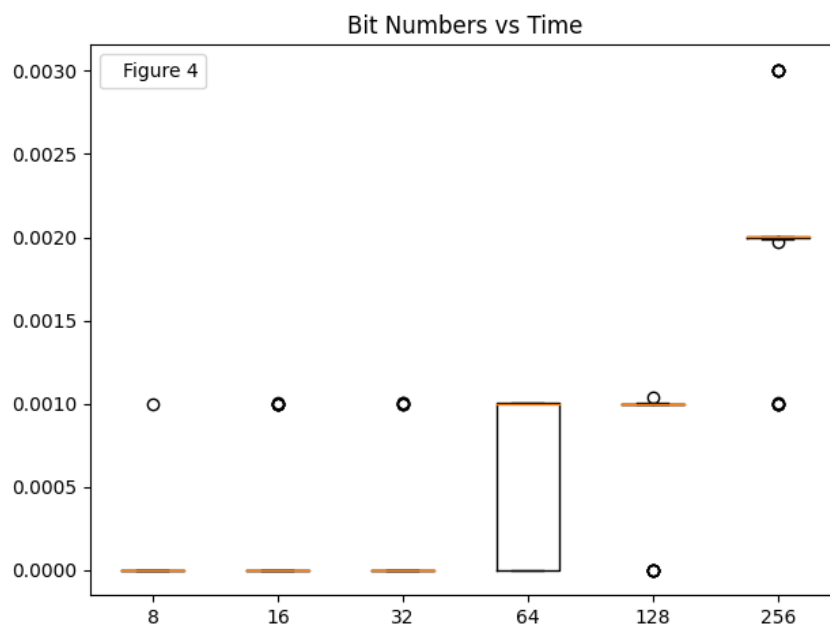
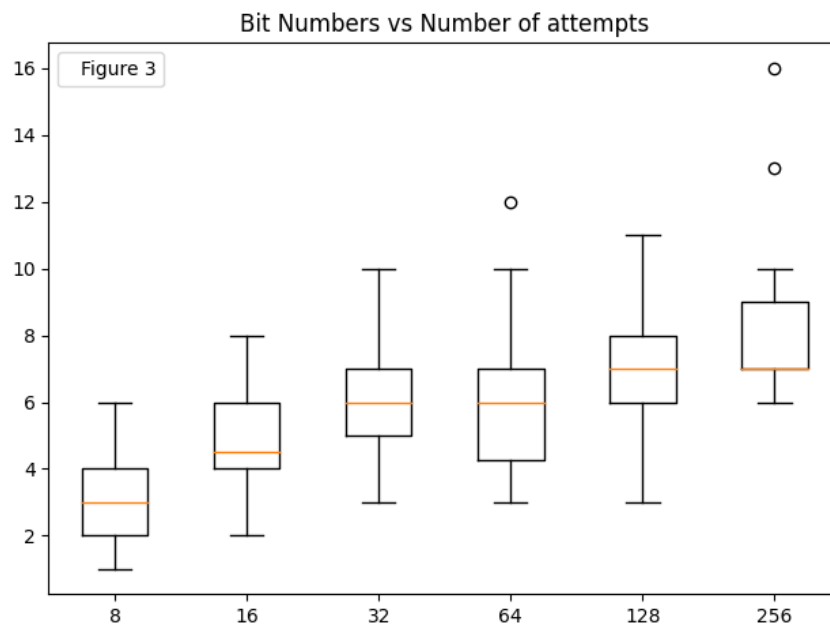
## Pergunta 2

Os resultados apresentados apenas correspondem a tamanhos de bits de 8 e 16, pois o tempo de processamento seria bastante grande. Para correr a função e obter os gráficos é necessário correr a função `new_start()`



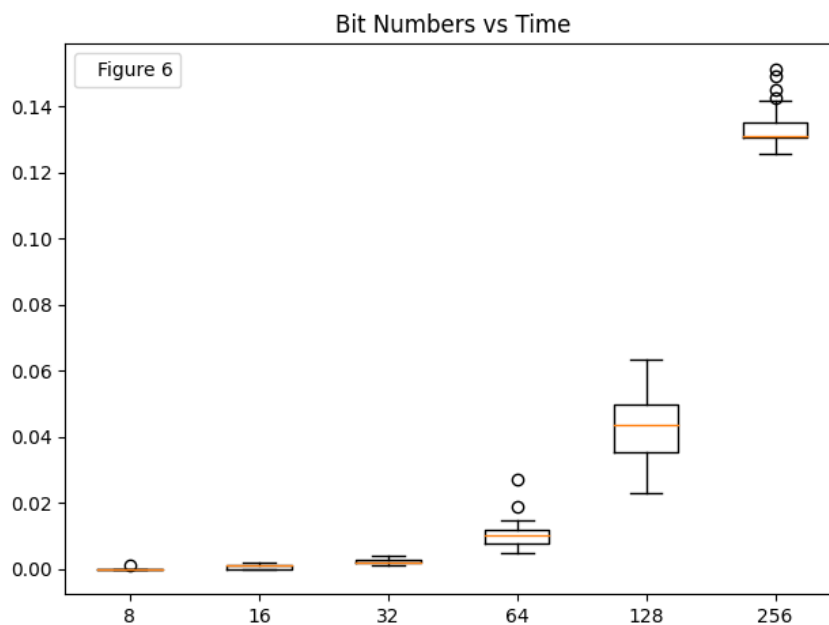
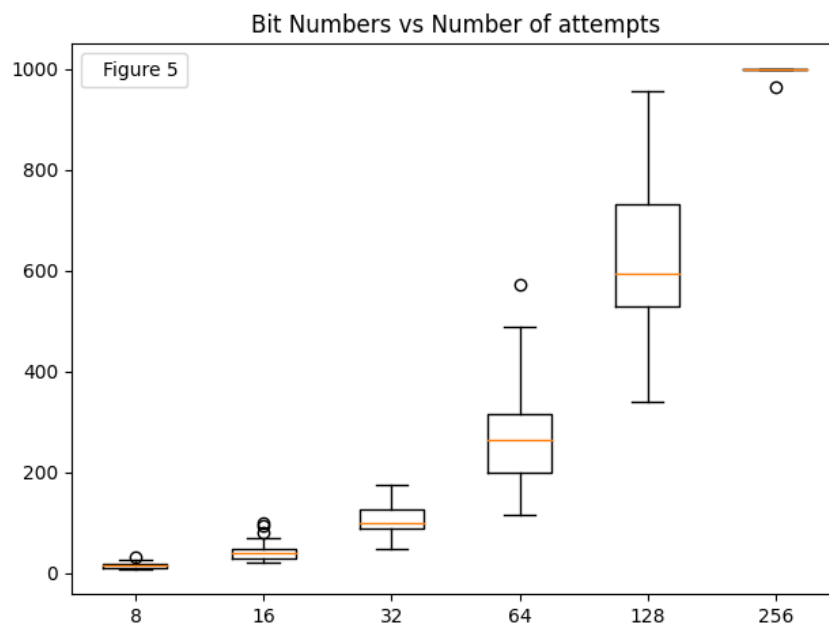
## Pergunta 3

Sozinha a função de avaliação não pode ser utilizada para melhorar os tempos de pesquisa, é necessário então juntar essa função a uma maneira de mudar os bits utilizando esta informação. Esta função está feita na função evaluation que recebe 2 strings uma que vai ser testada e outra que é a string com que vai ser comparada. Neste caso para implementar esta função o que utilizei foi retirar os bits que não são iguais nas duas strings, utilizando a função de evaluation como condição de paragem. A função devolve em valores decimais a proximidade dos padrões, se ambos forem iguais devolve 1.



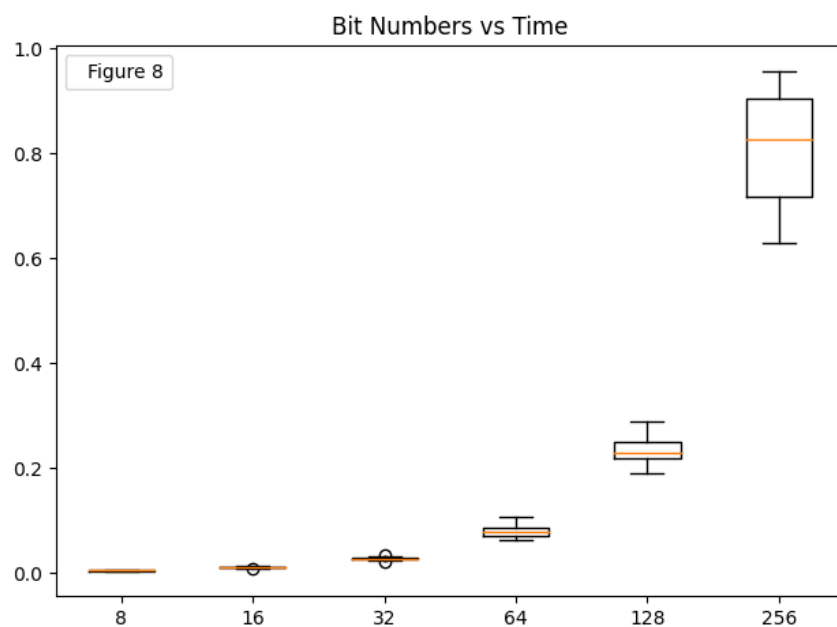
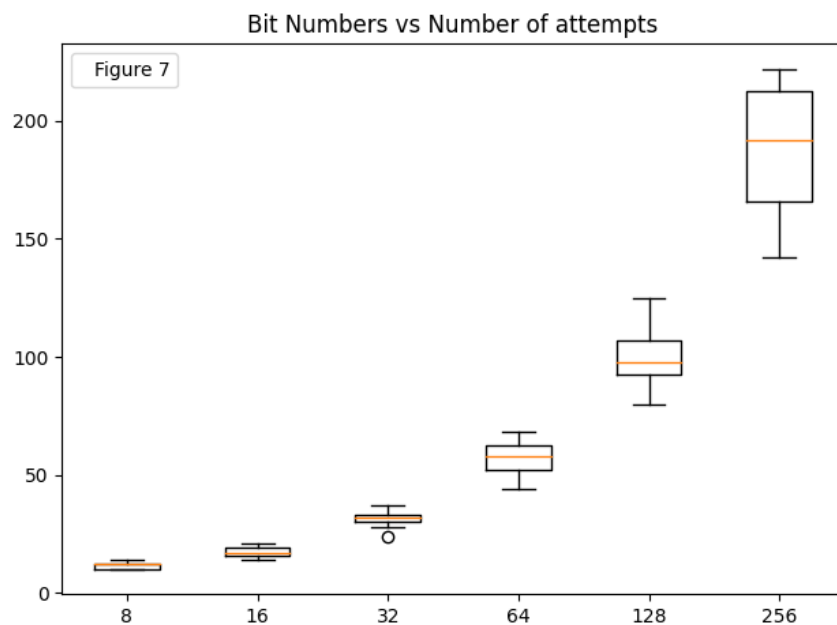
## Pergunta 4(1)

Utilizar mutação nos padrões grandes impede de chegar ao resultado esperado como se pode ver no gráfico em que para 256 bits a função chega às 1000 tentativas e não consegue chegar ao valor necessário. Isto deve-se ao facto de como se trata de um grande número de bits é necessário também um grande número de tentativas para ser possível chegar ao valor pretendido



## Pergunta 5

Neste caso os tempos de execução são bastante maiores comparando com os da pergunta anterior, especialmente considerando os valores em que o número de bits é mais elevado especialmente o 256. No entanto o número de tentativas ficou bastante reduzido, comparando com os valores do exercício anterior, enquanto no exercício anterior os valores atingiam o máximo de 1000 gerações, utilizando uma população é possível chegar ao valor pretendido em menos tempo. No entanto no exercício 3 o número de tentativas é ainda menor porque é conhecido quais os bits que é necessário modificar para chegar ao valor pretendido reduzindo a complexidade do problema.



## Pergunta 6

Semelhante ao que aconteceu com a pergunta anterior em que foi aplicada a mutação a uma população. Aplicar o crossover reduz ainda mais o número de tentativas necessárias para chegar ao resultado pretendido, diminuindo também o tempo necessário para chegar ao resultado pretendido. É de notar que existe um grande desvio com a crossover

