

NOME: VANDO CARLOS DINIZ REIS - 2019057195

NOME: RAFAEL MOURA MARQUES - 2019089305

TÍTULO: Problema das N-Rainhas

ANÁLISE DOS RESULTADOS:

Para obtermos resultados plausíveis, rodamos cada caso 100 vezes antes de analisá-los.

n = 8:



Quando temos uma população inicial de 8 indivíduos em um tabuleiro de xadrez 8x8, podemos observar que em aproximadamente 75% dos casos encontramos uma solução perfeita, porém é mais difícil de começar com uma dessas. Além disso, é possível ver que na maioria dos casos a solução é encontrada com menos de 50 iterações. Acreditamos que isso acontece pois quando chamamos a função 'def_selection' é muito provável que tenham indivíduos 'bons' entre os 5 candidatos selecionados que terão seus genes transmitidos para a próxima geração.

n=20:

Histograma do NUMERO DE ITERAÇÕES

n = 20

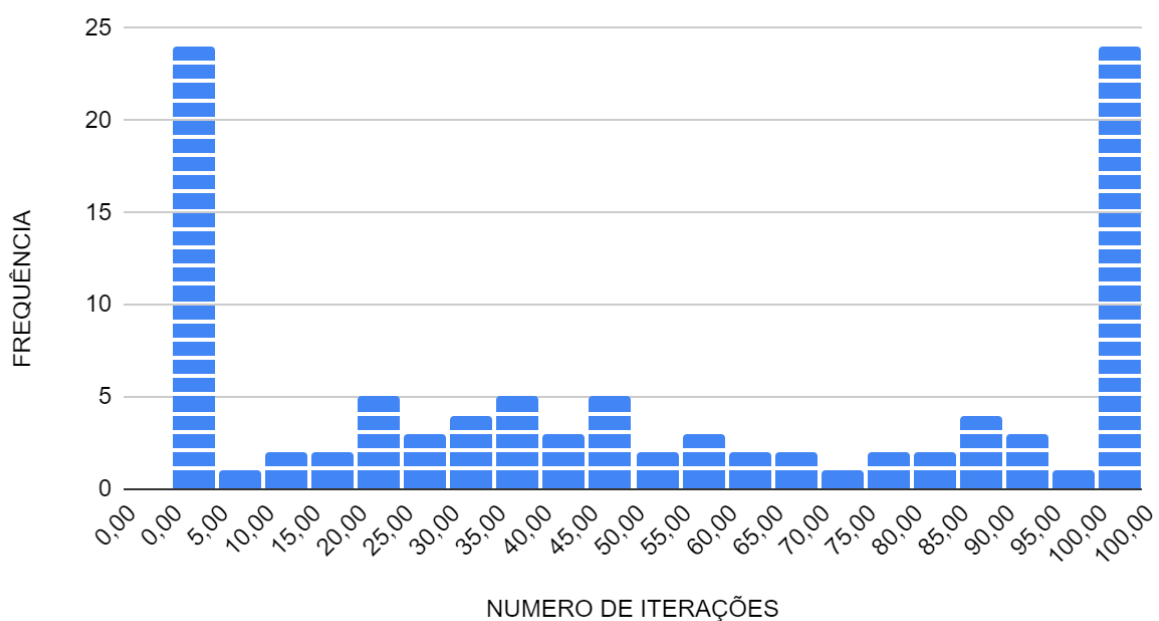


Quando temos uma população inicial de 20 indivíduos podemos perceber que tendemos a encontrar uma solução perfeita também em 75% dos casos, entretanto, nesse caso, é mais possível de se iniciar com uma solução perfeita e também vemos que é mais bem distribuída a quantidade de iterações feitas.

n=50:

Histograma de NUMERO DE ITERAÇÕES

n = 50



Já com 50 indivíduos, temos que em também 75% das vezes encontramos uma solução para o problema. Mas também é notável que há várias simulações em que já começamos com uma solução ideal 25% das vezes. Acreditamos que nesse caso seja mais difícil gerar uma solução perfeita pois quando chamamos a função 'def_selection', a chance de selecionar indivíduos 'ruins' é maior.

Considerações finais:

- Notamos também que nos 3 casos a população tende a convergir para um indivíduo específico, principalmente quando um elevado número de iterações acontece. Acreditamos que isso acontece pois em dado ponto do processo, começa a ocorrer a permutação de indivíduos muito parecidos. Assim, os filhos gerados por eles serão praticamente iguais aos pais. Além disso, eles substituirão os piores indivíduos, que possuem um genótipo diferente, e esse genótipo igual perpetuará. É nesse momento que vemos que a mutação é um fator importantíssimo para conseguirmos variar nossa população e chegar ao resultado desejado.

Exemplo:

```
Menor número de cheques no início: 2
População final: [array([5, 2, 0, 6, 3, 7, 1, 4]), array([5, 2, 0, 6, 3, 7, 4, 1]), array([5, 1, 0, 6, 3, 7, 2, 4]), array([5, 2, 0, 6, 3, 7, 1, 4]), array([2, 6, 1, 3, 0, 5, 7, 4]), array([5, 2, 0, 6, 3, 1, 7, 4]), array([5, 2, 0, 6, 3, 1, 4, 7]), array([5, 2, 0, 6, 3, 7, 4, 1]), array([5, 2, 0, 6, 3, 1, 7, 4]), array([5, 2, 0, 6, 3, 7, 4, 1]), array([5, 2, 0, 6, 3, 1, 4, 7]), array([5, 2, 0, 6, 3, 7, 4, 1]), array([5, 0, 2, 6, 3, 7, 4, 1]), array([5, 0, 2, 6, 3, 7, 4, 1]), array([5, 0, 2, 6, 3, 1, 7, 4]), array([5, 2, 0, 6, 3, 1, 7, 4]), array([5, 2, 0, 6, 3, 7, 1, 4]), array([5, 2, 0, 6, 3, 1, 4, 7]), array([5, 2, 0, 6, 3, 7, 4, 1]), array([5, 2, 0, 6, 4, 7, 1, 3])]
Número de iterações: 85
Menor numero de cheques no final = 0
Melhor solução: [5 2 0 6 4 7 1 3]
```

- As funções criadas por nós tem seu funcionamento e lógica explicadas no arquivo .pynb