Problema das 8 Rainhas com Algoritmos Genéticos

Hugo Fernandes[1], Alexandre Melo Queiroga[1]

[1]Depto de Computação e Tecnologia - Universidade Federal do Rio Grande do Norte (UFRN)
Caixa Postal: 59.300-000 - Caicó - RN - Brazil

hugo.medeiros.fernandes@gmail.com, alexandrekurticao@gmail.com

Abstract. *This work seeks to solve the problem of 8 queens of chess with the application of Genetic Algorithms technique generating the fitness chart of each generation, and show the results achieved.*

Resumo. *Este trabalho busca resolver o problema das 8 rainhas do xadrez com a aplicação da técnica de Algoritmos Genéticos, gerando o gráfico de aptidão de cada gerações, além de mostrar o resultado atingido.*

1. Descrição da Implementação

O problema foi implementado utilizando a linguagem Python, com a ajuda das bibliotecas Matplotlib para geração dos gráficos e a Numpy para manipulação de arrays.

O principal componente na construção de um algoritmo genético é sem dúvida a representação de seu cromossomo. Dessa maneira, foi decidido uma representação em lista de tamanho 8, onde cada elemento seria a posição de uma rainha no tabuleiro de xadrez.

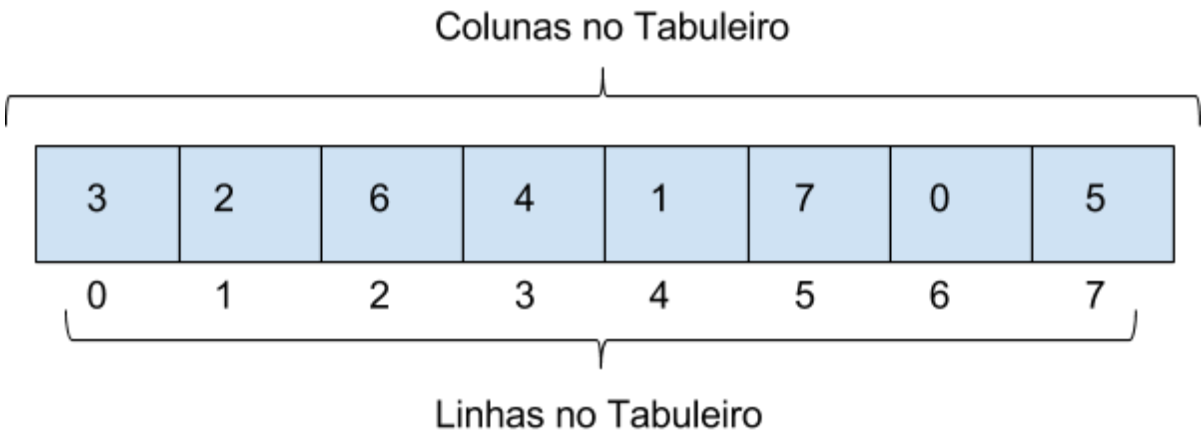

Figura 1: Exemplo de um Cromossomo

Para evitar perda de tempo com futuras verificações e melhor performance, os cromossomos são gerados em linhas diferentes, ou seja, é feita uma permutação nos itens para evitar que uma rainha fique na mesma linha que outra.

Logo em seguida, foi definida a função de crossover. Esta, utiliza uma taxa onde os cromossomos dos pais serão cortados para gerar um filho.

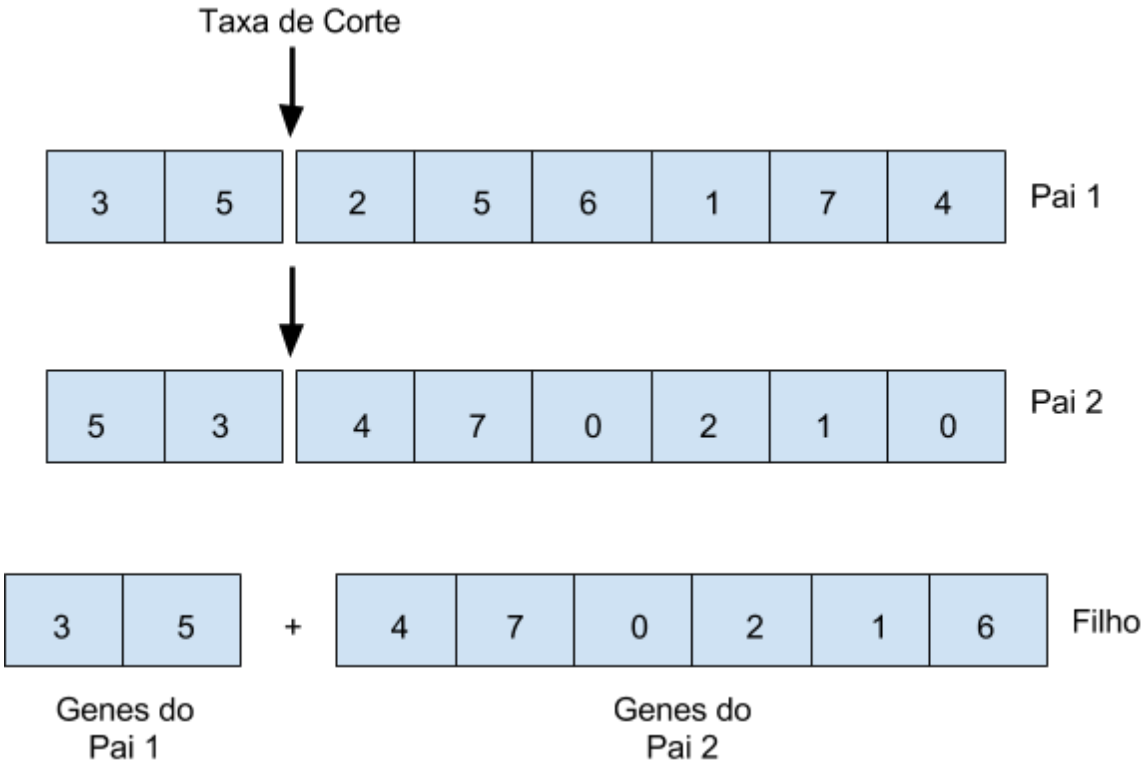

Figura 2: Exemplo de um Crossover

O próximo passo é implementar a função de mutação. Essa função, busca criar uma mudança em um cromossomo de forma a atingir uma variação na população que talvez não seria atingida por estratégias convencionais. Ela também utiliza uma taxa para ocorrer.

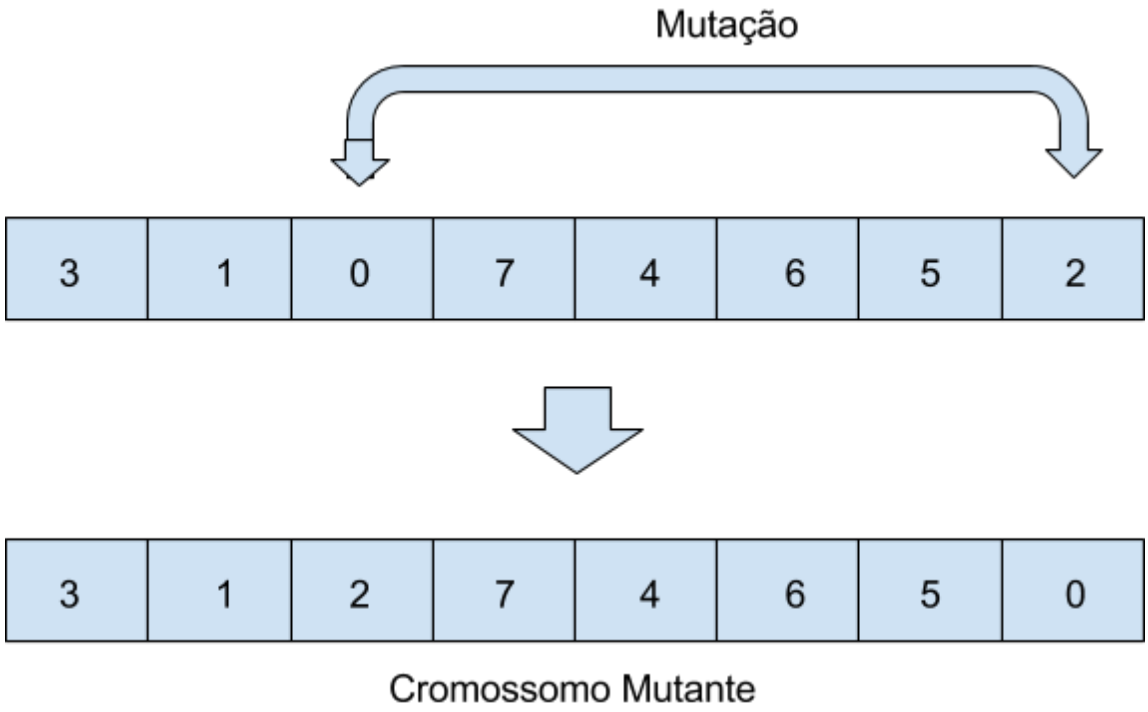

Figura 3: Exemplo de Mutação

Outro ponto bastante importante no desenvolvimento de um algoritmo genético é a criação de uma função de avaliação. Ela implementa um índice na qual pode-se mensurar a qualidade de cada cromossomo. No caso do problema das rainhas, foi incrementado o número de ataques que cada rainhas cometia. Sendo assim, o melhor cromossomo seria aquele que obtivesse avaliação igual a zero, o que representava que o mesmo não atacava nenhuma outra rainha.

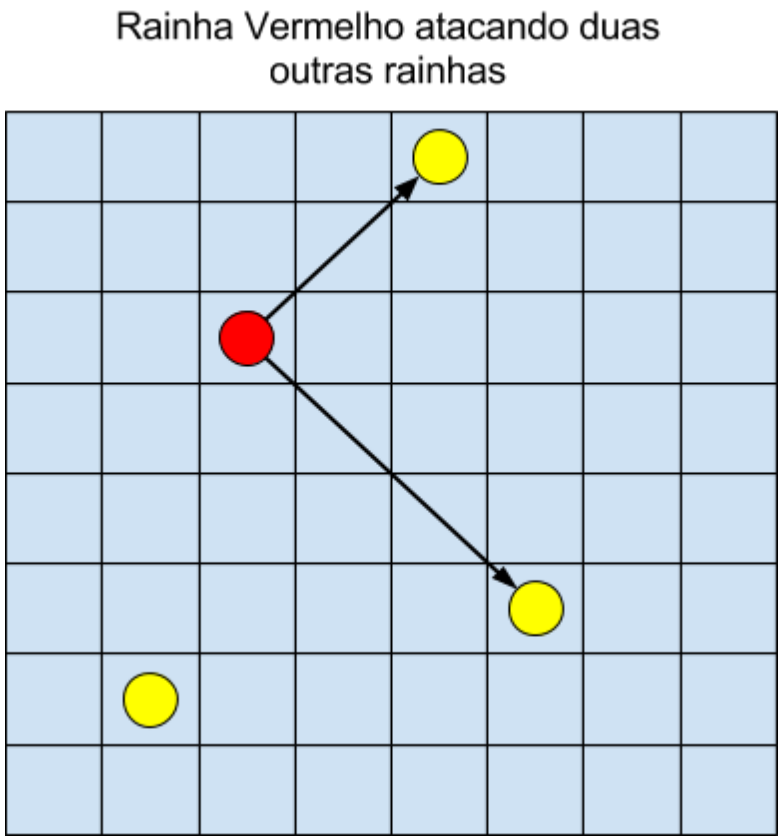

Figura 4: Exemplo de Ataque de uma Rainha

Mais um componente que merece destaque na construção de um algoritmo genético é uma função de seleção. Ela trabalha em conjunto com a avaliação, onde permite aos cromossomos melhor avaliados, um maior destaque.

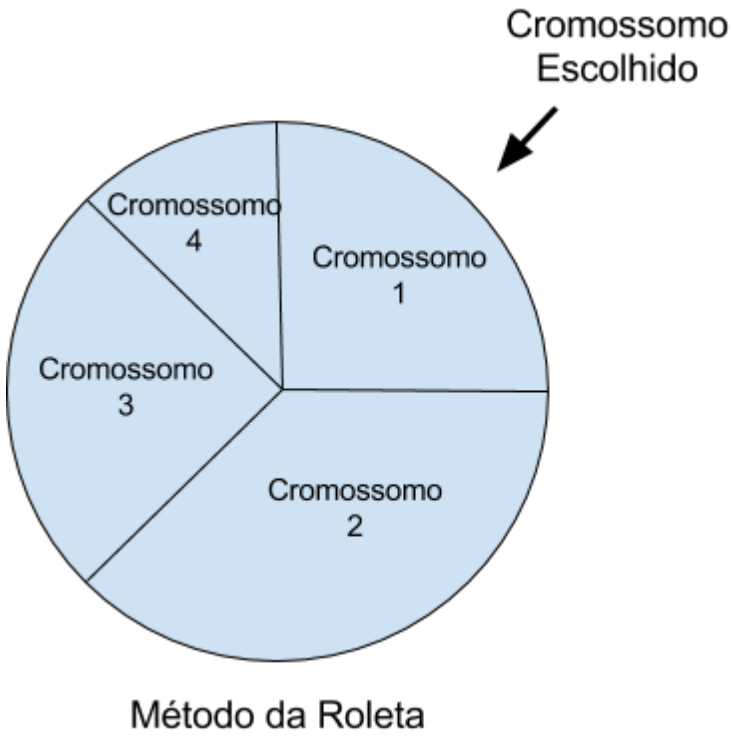

Figura 5: Exemplo de método de seleção

Finalmente, aplicando essas cinco propriedades o algoritmo genético pode ser utilizado. O primeiro passo consiste em gerar uma população inicial aleatória e avaliar cada cromossomo desta população. Em seguida, aplica-se à função de seleção, assim selecionando os cromossomos por sua aptidão. Depois disso, utiliza-se os operadores de Crossover e Mutação nos cromossomos selecionados e os avalia. Concluindo, se o cromossomo que se espera atingir for encontrado, o processo encerra, caso contrário, o processo é repetido por diversas gerações.

2. Primeiro caso de teste

Nesse caso o algoritmo foi executado com os seguintes parâmetros: População = 3, Taxa de Crossover = 0.5 e Taxa de Mutação = 0.1.

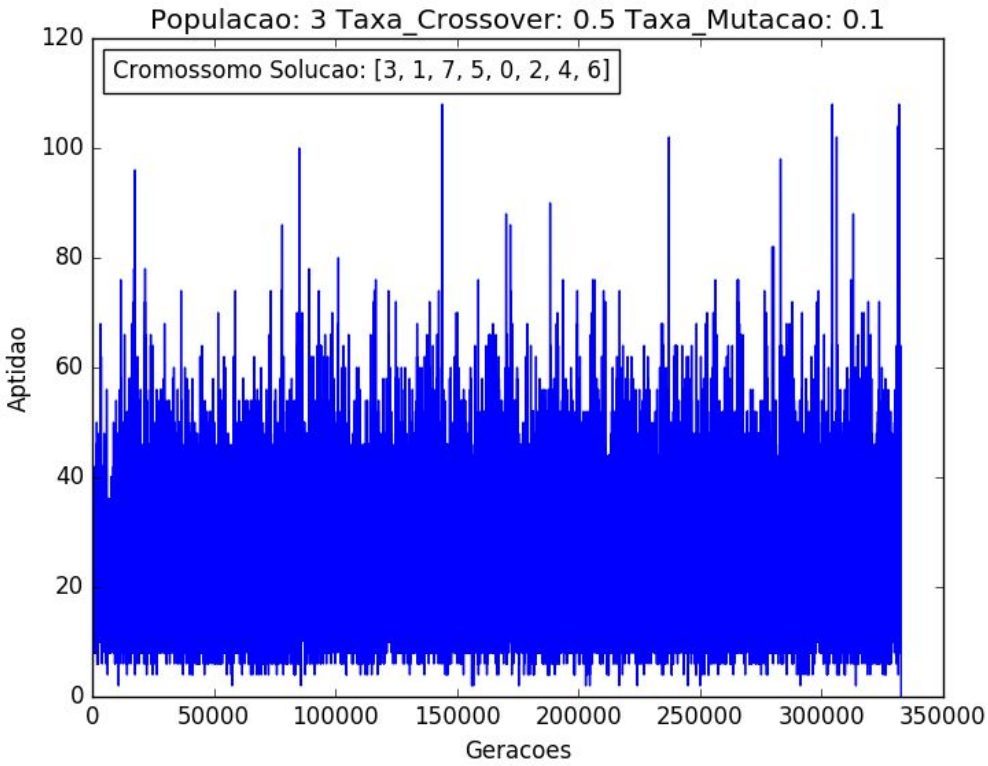

Figura 6: Gráfico Gerações por Aptidão no Primeiro Caso

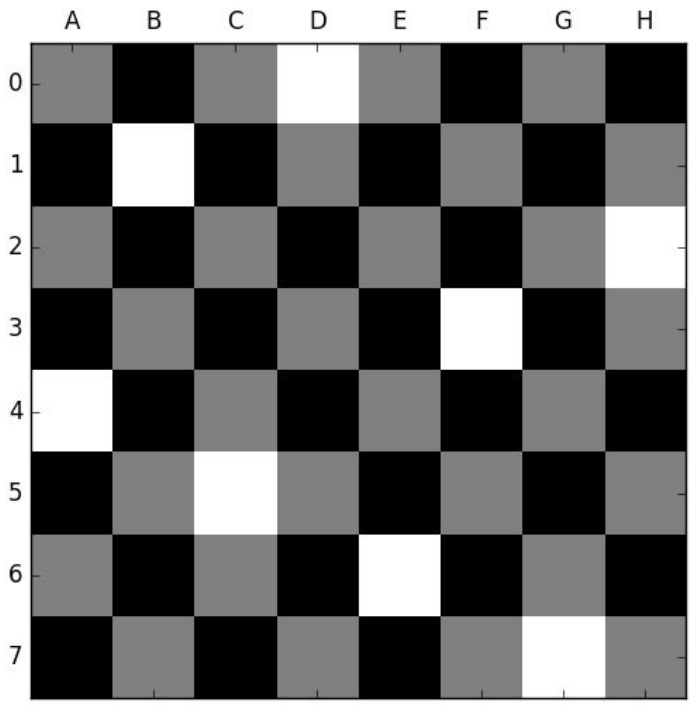

Figura 7: Posição das Rainhas no Tabuleiro no Primeiro Caso

3. Segundo caso de teste

Nesse caso o algoritmo foi executado com os seguintes parâmetros: População = 10, Taxa de Crossover = 0.7 e Taxa de Mutação = 0.5.

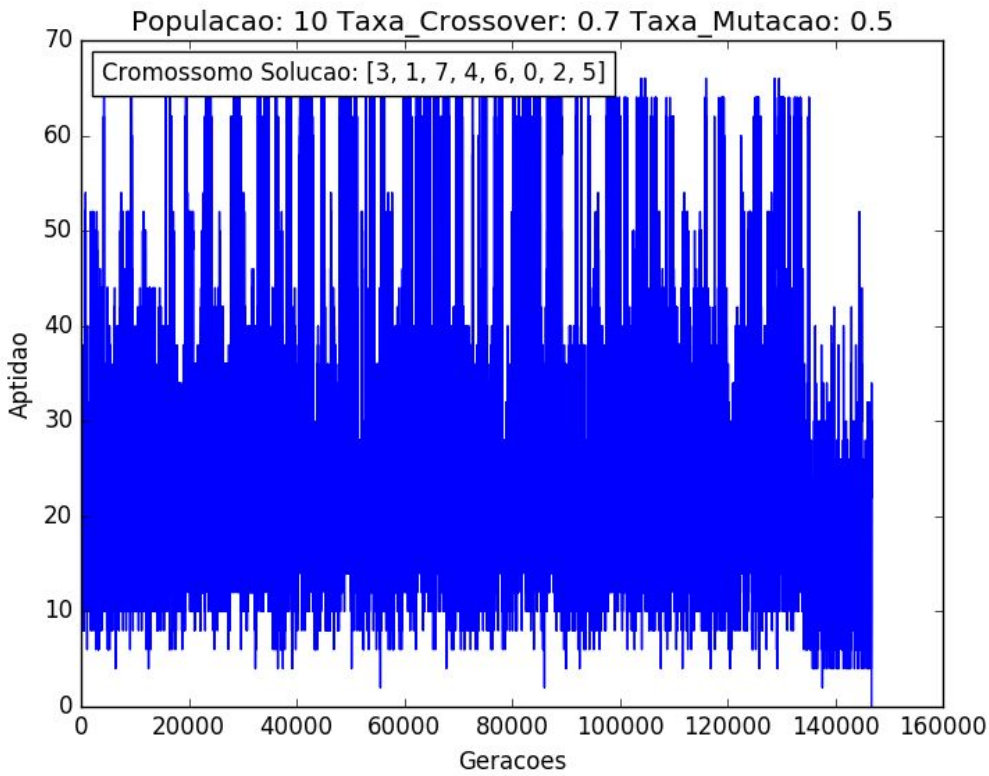

Figura 8: Gráfico Gerações por Aptidão no Segundo Caso

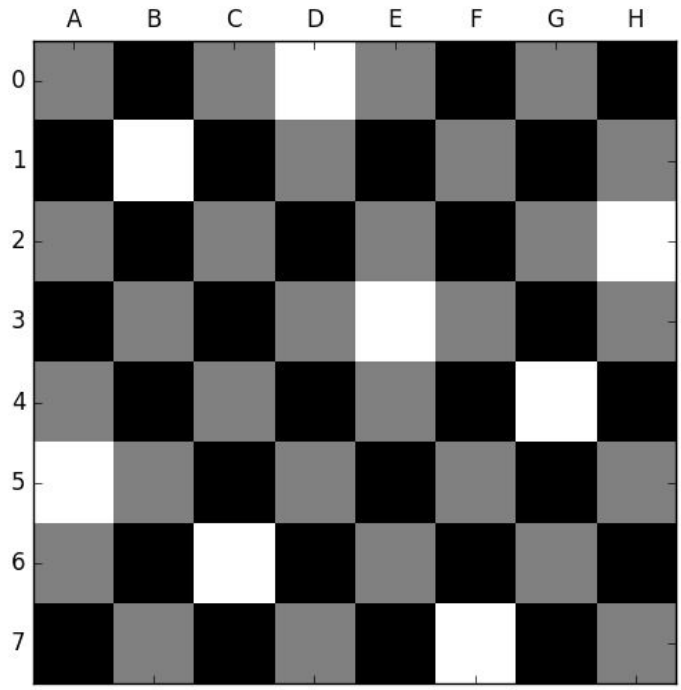

Figura 9: Posição das Rainhas no Tabuleiro no Segundo Caso

4. Terceiro caso de teste

Nesse caso o algoritmo foi executado com os seguintes parâmetros: População = 15, Taxa de Crossover = 0.9 e Taxa de Mutação = 0.15.

Não foi atingido nenhuma resposta ao executar o algoritmo neste caso de teste. O processo ficou em execução por mais de 15 horas, e não resolveu o problema, dessa forma, o processo foi encerrado.

5. Conclusão

O experimento foi bem sucedido, e mostrou que a alteração dos parâmetros de entrada no algoritmo genético impactam significativamente no seu aproveitamento, sendo o terceiro caso de teste o único que não conseguiu chegar ao resultado esperado. Assim, os parâmetros do segundo caso de teste, foi o que atingiu os melhores resultados, conseguindo resolver o problema com menos de 160.000 gerações, em comparação ao primeiro caso, que necessitou de quase 350.000 gerações para resolver o problema.

A maior dificuldade encontrada ao implementar os testes, foi em relação ao tempo de execução do algoritmo, demorando um tempo significativo para cada resultado.

Link para o Código Fonte: https://github.com/hugaofernandes/8_queens_algorithm_genetic.git