As oito Rainhas

Gabriel Carneiro, Alexander Cristian

Dezembro 2020

1 Introdução

O problema das 8 rainhas consiste na ideia de colocar 8 rainhas num tabuleiro de xadrez de forma que nenhuma possa atacar a outra, e este projeto tem o objetivo de criar um algoritmo em grafos capaz de mostrar todas as possibilidades de posições em que podemos colocar as rainhas.

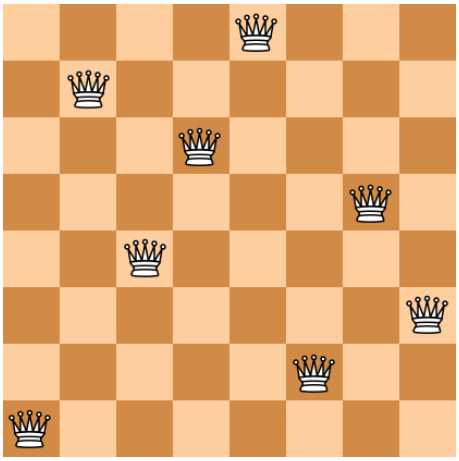

Figure 1: Uma possível solução do problema

2 O problema

Existem milhões de possíveis posições em que podemos colocar as 8 rainhas, e isso dificulta na solução do problema, pois torna inviável uma solução

utilizando força bruta. Uma das formas mais usuais de atacar esse problema é através de algoritmos de **backtracking**, mas o objetivo desse trabalho é resolver esse problema usando algoritmos de grafos, então optamos por fazer usando clique direto.

2.1 A implementação

O algoritmo foi feito usando a linguagem **python** pelo fato de existirem diversas bibliotecas que implementam funções de grafos, que facilitam parte do trabalho, e pela biblioteca chess que permite que possamos inserir peças num tabuleiro de xadrez, facilitando na parte visual da solução. O programa se inicia gerando um grafo com 64 vértices, um para cada posição do tabuleiro, e em seguida chama a função que percorre todo o grafo procurando por posições que se atacam, ou seja, posições que se invalidam caso tenha uma rainha, para fazer essa verificação ela olha a mesma linha, a mesma coluna, e na diagonal, caso não tenha problema, ele cria uma aresta, e faz isso até que não tenha mais espaço, quando acabado, chama o algoritmo de Bron-Kerbosch que retorna os cliques máximos do grafo, ou seja, as 92 soluções do problema.

3 O código

3.1 generate_grafo()

Cria um grafo de 64 vértices e chama a função de criação de arestas

3.2 set_arestas(grafo)

Percorre o grafo com uma relação de busca de todos para todos, e em cada caso chama a função pode_colocar() da seção 3.3, e caso retorne True, cria uma aresta entre os vértices.

3.3 pode_colocar(verticeA, verticeB)

Chama a função verifica(opcao, verticeA, verticeB), e caso retorne True, ela retorna False, e caso retorn False, ela retorna True

3.4 verifica(opcao, verticeA, verticeB)

Essa função tem 3 opções:

- **Linha**: retorna false quando não existe nenhuma rainha que pode atacar aquela linha
- **Coluna**: retorna false quando não existe nenhuma rainha que pode atacar aquela coluna
- **Diagonal**: retorna false quando não existe nenhuma rainha que pode atacar aquela diagonal

3.5 encontra_cliques(grafo)

Encontra e retorna todos os cliques máximos do grafo. Para cada vértice V, um clique máximo para V é o maior subgrafo completo contendo o vértice V.

3.6 main

Chama a função encontra_cliques(grafo) para encontrar os cliques máximos que são a solução do problema e usa a biblioteca chess para montar um tabuleiro com as 8 rainhas posicionadas, imprimindo o tabuleiro em unicode (figura 2)

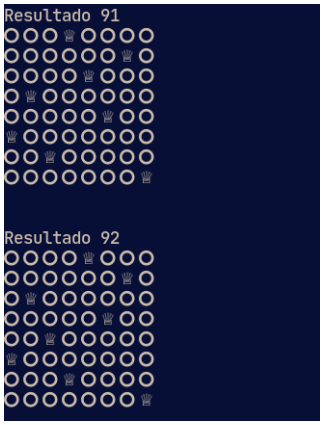

Figure 2: Duas possíveis soluções do problema

4 Análise dos resultados obtidos

O problema possui ao todo 92 possíveis soluções, e o algoritmo foi capaz de encontrar todas elas.

5 Conclusão

Este foi um problema extremamente trabalhoso de se resolver, especialmente o algoritmo de encontrar os cliques, mas no fim consegue mostrar todos os 92 possíveis resultados do problema.