Trabalho Prático - Matemática Discreta

Data: 02/11/2022
Giovanna Naves Ribeiro
Matrícula: 2022043647

O problema do Passeio do Cavalo é um desafio que deve ser resolvido buscando um caminho que passa por todas as casas de um tabuleiro 8x8 sem repetição e respeitando os movimentos possíveis de uma peça de cavalo de xadrez.

Para iniciar a resolução do problema, consideramos os oito movimentos possíveis para um cavalo em um tabuleiro e os catalogamos em uma matriz 8x2 {{1,2},{1,-2},{2,1},{2,-1},{-1,2},{-1,-2},{-2,1},{-2,-1}}, sendo a coordenada x de cada par referente ao deslocamento horizontal e a coordenada y referente ao deslocamento vertical. Chamamos essa matriz de “MovsPossíveis”.

Sabemos que cada caminho imaginável para o cavalo é uma sequência de tomada de decisões cujas possibilidades são esses 8 movimentos, logo, é possível visualizar o problema como um enorme grafo com vários “caminhos”, em que a partir de cada nó (representação da casa do tabuleiro), saem no máximo 8 arestas (uma para cada movimento que resulta em uma casa dentro do limite do tabuleiro e que ainda não foi visitada pelo cavalo). O “caminho” que resolve o problema terá exatamente 64 nós, referente ao número de casas de um tabuleiro 8x8, e os caminhos que não apresentam soluções válidas para o problema serão interrompidos à medida que chegam em casas que já foram visitadas ou que estão fora dos limites do tabuleiro, não podendo, assim, continuar com o passeio.

A solução mais lógica e intuitiva para resolver esse problema seria escolher um dos movimentos catalogados e ir tomando decisões com os movimentos válidos até não ser mais possível prosseguir por não existirem mais movimentos que levam a casas ainda não visitadas e dentro dos limites do tabuleiro. Em seguida, o cavalo deveria voltar pelo mesmo caminho que veio até encontrar uma “bifurcação” em que ele pode escolher um outro movimento para prosseguir. Se os outros movimentos possíveis também não levarem a nenhuma solução, então ele volta ainda mais casas e procura outras possibilidades de deslocamentos. Esse modelo de tentativa e erro faz com que o cavalo tente e retroceda os caminhos que não chegam à solução até encontrar um que resolva o problema, e pode ser chamado de modelo de Backtracking.

Esse foi nosso primeiro modelo implementado para encontrar uma solução para o Passeio do Cavalo, porém, observamos que o algoritmo tinha um custo computacional muito alto, além de demorar muitos minutos para entregar uma resposta, avançar e retroceder centenas de casas até achar uma solução e dar erro em algumas casas escolhidas para serem a posição inicial. Esse algoritmo funciona melhor para tabuleiros menores e, como não estávamos satisfeitos com sua eficiência, foi necessária a pesquisa para a implementação de uma heurística que facilitasse a solução do problema.

Uma heurística é uma abordagem de resolução de problemas baseada em suposições e testes a fim de criar uma estratégia mais eficaz de resolver um problema. Após a pesquisa de heurísticas para resolver o problema do passeio do cavalo, encontramos a Regra de Warnsdorff, descrita em 1823.

A Regra de Warnsdorff diz que, ao escolhermos o próximo movimento do cavalo, devemos calcular quantos movimentos seguintes são possíveis a partir de cada uma das casas que o cavalo pode escolher. Por exemplo, suponha que começamos com o cavalo na posição (2,4). Se ele escolher o movimento que o leva para (1,6), ele terá, em seguida, 3 movimentos válidos possíveis para escolher. Se ele for para (3,6), ele terá 7 opções de movimento, e assim por diante para todos os movimentos válidos partindo da casa (2,4). Devemos, segundo Warnsdorff, escolher o movimento que oferece a menor quantidade de possibilidades para o cavalo caminhar a partir da posição em que parar (no nosso exemplo, devemos escolher o movimento que nos leva para (1,2), já que após essa movimentação teremos apenas 2 movimentos possíveis a partir dessa casa, e nenhuma outra movimentação nos retorna um número menor). Analogamente, se estamos visualizando o problema em um grafo, escolhemos avançar para o vértice do qual saem menos arestas para vértices posteriores.

Entretanto, na maioria dos casos de comparação, aparecerá mais de uma casa que oferece a mesma quantidade de casas para avançar em seguida. Nesse momento, é necessário aplicar um critério de desempate. Para isso, implementamos o critério de Pohl, que procura, dentre os movimentos possíveis das casas empatadas, qual continua nos oferecendo a menor quantidade de movimentações. Assim, estaremos sempre escolhendo o caminho cujas movimentações nos oferecem a menor quantidade de passos adiante, otimizando nosso algoritmo ao diminuir o custo e o tempo de execução. Além disso, eliminamos a necessidade de avançar e retroceder casas que não levarão a um caminho que solucione o problema.

Feita essa análise, implementamos a heurística em nosso código. As duas primeiras funções certificam se cada movimento é válido (se está dentro dos limites do tabuleiro e a casa ainda não foi visitada). Implementamos também duas funções de preferência e uma de desempate, que identificam quantos são os movimentos possíveis a partir de uma casa e escolhem a movimentação que nos leva à posição com menos movimentos adiante possíveis. O desempate é feito pelo método de Pohl, que utiliza novamente a heurística de Warnsdorff.

Para encontrar o caminho do cavalo, utilizamos uma função recursiva, que se repete para cada movimentação do cavalo e termina sua execução quando encontra o caso onde todas as casas do tabuleiro já foram visitadas (quando o contador de movimentos se iguala ao número de posições - 64 para um tabuleiro 8x8). A cada iteração, somamos à posição (x, y) do cavalo os valores catalogados no array “MovsPossíveis” que escolhemos através da heurística.

Essa função recursiva é chamada pela função “passeio”, que decresce em 1 unidade as posições recebidas para que uma casa (1,1) no tabuleiro se enquadre na posição (0,0) da matriz (a primeira posição) e assim por diante. Como utilizamos a heurística que melhor otimiza o algoritmo do passeio, o número de casas visitadas vai crescendo paralelamente ao contador da função recursiva e é exatamente o número total de posições do tabuleiro. Além disso, não há casas retrocedidas, já que só optamos por tomar o caminho após ter certeza que ele é a melhor opção e nos leva a uma solução para o problema. Sendo assim, atendemos a condição de que o número de casas avançadas decrescido do número de casas retrocedidas é o número total de casas do caminho (64).

Finalizamos gravando a matriz solução para o problema em um arquivo saida.txt, juntamente com o número de casas visitadas e retrocedidas, nessa ordem.

Podemos concluir, ao final da implementação, que a utilização da heurística de Warnsdorff otimiza consideravelmente o algoritmo que resolve o problema do Passeio do Cavalo e nos entrega a solução em milésimos de segundo, se mostrando, assim, muito mais eficaz do que a implementação do algoritmo intuitivo que testa todas as possibilidades de caminho até encontrar um que configura a solução.

Bibliografia:

A Knight’s Tour. https://bradfieldcs.com/algos/graphs/knights-tour/. Acessado 2 de novembro de 2022.

Frost, Daniel. “Implementing A Heuristic Solution To The Knight’s Tour Problem”. Medium, 23 de julho de 2020, https://medium.com/@danielfrost_3076/implementing-a-heuristic-solution-to-the-knights-tour-problem-513a73cc7e20.