

Resolução de problemas por meio de estratégias de busca

Sara Ferreira
sara.ferreira@utp.edu.br

09/10/2020

1 Introdução

O presente trabalho é sobre as estratégias de busca utilizadas na inteligência artificial, e alguns problemas que podem ser solucionados com essas estratégias.

Está organizado em 4 capítulos principais. O primeiro aborda dois tipos de busca, busca sem informação e com informação. Já o segundo capítulo, aborda os seguintes problemas: Cubo de Rubik, Missionários e Canibais, Problema das N Rainhas e Sudoku.

nível mais baixo, gerando todos os estados sucessores de cada um destes estados. Assim, cada nível da árvore é completamente construído antes que qualquer nodo do próximo nível seja adicionado à árvore. [1]

A estratégia da pesquisa em largura é expandir primeiro todos os nós de menor profundidade. É uma pesquisa muito sistemática, normalmente demora muito tempo e ocupa muito espaço. A principal vantagem do algoritmo de busca em largura é que este encontra o menor caminho do nodo inicial até o nodo final mais próximo.

2 Fundamentação teórica

2.1 Busca às cegas

Uma estratégia de busca é dita cega se ela não leva em conta informações específicas sobre o problema a ser resolvido. Essas estratégias usam somente a informação disponível na definição do problema, geram sucessores e verificam se o estado objetivo foi atingido. Existem basicamente duas estratégias cegas para a construção e pesquisa em uma árvore de busca: Busca em Largura e Busca em Profundidade. [3]

2.1.1 Pesquisa em largura

Consiste em construir uma árvore de estados a partir do estado inicial, aplicando a cada momento, todas as regras possíveis aos estados do

2.1.2 Pesquisa em profundidade

A estratégia da pesquisa em profundidade, ao contrário da pesquisa em largura, é expandir sempre o nó mais profundo da árvore. O algoritmo de busca em profundidade não encontra necessariamente a solução mais próxima, mas pode ser mais eficiente se o problema possui um grande número de soluções ou se a maioria dos caminhos pode levar a uma solução.

2.1.3 Pesquisa em profundidade limitada

É muito semelhante à busca em profundidade, mas é definido um limite de profundidade, ou seja, o algoritmo faz a pesquisa somente até certo ponto.

2.2 Busca com informação

Utiliza conhecimento específico além da definição do problema para encontrar soluções de forma mais eficiente do que a busca cega. Utiliza-se uma função de avaliação, para definir o quão desejável é cada nó, e então escolher o nó mais desejável, que ainda não foi expandido. [2]

2.2.1 Busca Gulosa

A busca gulosa expande sempre o nó que, naquele momento, parece mais próximo do objetivo. Para realizar essa medição, usa-se a heurística como função de avaliação.

Como essa estratégia leva em consideração somente o nó atual, não é uma solução ótima.

2.2.2 A Estrela (A*)

Faz uso de mais uma função no momento de avaliar o nó. Leva em conta o custo até o momento, mais o custo estimado do nó atual até o objetivo.

3 Análise dos problemas

3.1 Cubo de Rubik

O famoso Cubo de Rubik, ou Cubo Mágico, foi inventado por Erno Rubik em 1974, e trata-se de um cubo, com um total de 26 peças móveis. Cada face do cubo possui uma cor, e ao mover as peças, as cores ficam embaralhadas. O objetivo então, é organizar novamente as cores.

3.1.1 Estado Inicial

Cubo com todas as cores embaralhadas.

3.1.2 Ações

- Mover topo sentido horário
- Mover topo sentido anti-horário

- Mover base sentido horário
- Mover base sentido anti-horário
- Mover face sentido horário
- Mover face sentido anti-horário
- Mover trás sentido horário
- Mover trás sentido anti-horário
- Mover direita sentido horário
- Mover direita sentido anti-horário
- Mover esquerda sentido horário
- Mover esquerda sentido anti-horário

3.1.3 Modelo de transição

Realizar uma ação, com o objetivo de juntar cores iguais.

3.1.4 Custo do caminho

Cada ação terá custo igual a 1(um).

3.1.5 Teste do Objetivo

Cubo com as cores organizadas.

3.1.6 Estratégia escolhida

Para resolver o cubo mágico, pode-se usar a estratégia A*, pois, em alguns casos, será necessário optar por um caminho pior, bagunçando algumas peças que já haviam sido organizadas, para organizar as outras, e o algoritmo A* se sai melhor nesse caso.

3.2 Missionários e canibais

Três canibais e três missionários estão viajando juntos e chegam à margem de um rio. Eles desejam atravessar para a outra margem para, desta forma, continuar a viagem. O único meio

de transporte disponível é um barco que comporta no máximo duas pessoas. Há uma outra dificuldade: em nenhum momento o número de canibais pode ser superior ao número de missionários pois desta forma os missionários estariam em grande perigo de vida.

3.2.1 Estado Inicial

Três canibais e três missionários, do mesmo lado do rio.

3.2.2 Ações

Colocar uma ou duas pessoas no barco, e atravessar o rio.

3.2.3 Modelo de transição

Colocar uma ou duas pessoas no barco, sem deixar mais canibais do que missionários em uma das margens.

3.2.4 Custo do caminho

Cada ação terá custo igual a 1(um).

3.2.5 Teste do Objetivo

Três canibais e três missionários, do outro lado do rio.

3.2.6 Estratégia escolhida

Pode ser resolvido com a busca em largura, o que garantirá o menor caminho até a solução.

3.3 Problema das n rainhas

O problema das oito rainhas é o problema matemático de dispor oito rainhas em um tabuleiro de xadrez de dimensão 8x8, de forma que nenhuma delas seja atacada por outra. Para tanto, é necessário que duas damas quaisquer não estejam numa mesma linha, coluna, ou diagonal. Este é um caso específico do Problema

das n rainhas, no qual temos n rainhas e um tabuleiro com n x n casas(para qualquer $n \geq 4$).

3.3.1 Estado Inicial

Tabuleiro vazio.

3.3.2 Ações

Colocar peça na posição (x,y).

3.3.3 Modelo de transição

Colocar peça na posição (x,y). Essa posição deve estar vazia, e não deve estar na linha de ataque de outra peça.

3.3.4 Custo do caminho

Cada ação terá custo igual a 1(um).

3.3.5 Teste do Objetivo

Tabuleiro com n rainhas, nenhuma atacando outra.

3.3.6 Estratégia escolhida

Pode ser resolvido com pesquisa em profundidade, pois esse problema pode apresentar muitas soluções, e a busca em profundidade pode ser mais eficiente com problemas assim.

3.4 Sudoku

O Sudoku é um quebra-cabeça baseado na colocação lógica de números. O objetivo do jogo é a colocação de números de 1 a 9 em cada uma das células vazias numa grade de 9x9, constituída por 3x3 sub-grades chamadas regiões. O quebra-cabeça contém algumas pistas iniciais, que são números inseridos em algumas células, de maneira a permitir uma indução ou dedução dos números em células que estejam vazias.

Cada coluna, linha e região só pode ter um número de cada um dos 1 a 9.

3.4.1 Estado Inicial

Grade com células vazias.

3.4.2 Ações

Colocar um número de 1 a 9 em uma posição (x,y) da grade.

3.4.3 Modelo de transição

Colocar um número de 1 a 9 em uma posição (x,y) da grade, obedecendo as seguintes regras:

- O número a ser colocado não pode já ter aparecido em outra célula, na mesma linha.
- O número a ser colocado não pode já ter aparecido em outra célula, na mesma coluna.
- O número a ser colocado não pode já ter aparecido em outra célula, na mesma sub-grade.

3.4.4 Custo do caminho

Cada ação terá custo igual a 1(um).

3.4.5 Teste do Objetivo

Grade com todas as células preenchidas corretamente, com nenhum número repetido na linha, coluna ou sub-grade.

3.4.6 Estratégia escolhida

Pode ser utilizada a busca gulosa. Apesar de não garantir a solução ótima, o sudoku será resolvido sem maiores complicações, principalmente se for um sudoku de nível mais fácil. Para um sudoku de nível difícil, a solução poderá demorar mais para ser encontrada, visto que existirá mais "tentativas e erros".

4 Considerações finais

Este trabalho foi muito importante para meu conhecimento e aprofundamento dos temas abordados, além de ter-me permitido aperfeiçoar competências de investigação, seleção e organização da informação.

Referências

- [1] DOS SANTOS, F. M. Buscas em largura e profundidade aplicadas à inteligência artificial.
- [2] RUSSELL, S., AND NORVIG, P. *Artificial Intelligence: A Modern Approach*. 2010.
- [3] ZAMBIASI, S. P. Métodos de busca.