Prova 1: Artigo

Breno Otavio Santana da Silva

10/10/2020

Sumário

1 Introdução **1**

2 Fundamentação Teórica **1**
2.1 Busca sem informação 1
2.2 Busca com informação 2
2.3 Busca local 2

3 Análise dos Problemas **2**
3.1 Cubo de Rubik 2
3.2 Missionários e Canibais 3
3.3 Problema das 8 Rainhas 3
3.4 Sudoku 3

4 Considerações Finais **4**

1 Introdução

A inteligência artificial é um avanço tecnológico que permite que sistemas simulem uma inteligência similar à humana. É uma das ciências mais recentes, teve início após a Segunda Guerra Mundial e, atualmente, abrange uma enorme variedade de subcampos, desde áreas de uso geral, como aprendizado e percepção, até tarefas específicas como jogos de xadrez, demonstração de teoremas matemáticos, criação de poesia e diagnóstico de doenças. A inteligência artificial sistematiza e automatiza tarefas intelectuais e, portanto, é potencialmente relevante para qualquer esfera da atividade intelectual humana. Nesse sentido, ela é um campo universal, GOMES (2010).

A resolução de problemas por meio de algoritmos de busca é o processo de examinar as diversas opções de sequencias de ações possíveis que podem levar ao estado objetivo, escolhendo a melhor sequenciá. É uma importante ramificação da Inteligência Artificial, sendo responsável por várias aplicações práticas utilizadas em nosso dia a dia, tal como o mecanismo para encontrar a menor rota em um aparelho GPS.

2 Fundamentação Teórica

Um algoritmo de busca recebe como entrada um problema e retorna uma solução na forma de uma sequencia de ações. A seguir os tipo de algoritmos de busca que podem ser utilizados.

2.1 Busca sem informação

Os algoritmos de busca sem informação utilização apenas a informação disponível no definição do problema. A seguir os algoritmos que são distinguídos pela ordem que os nós são explorados.

Busca em largura: Nesse algoritmo são expandidos todos os nós mais perto da raiz.

Ele é implementado com uma FIFO onde os novos itens entram no final.

Busca em profundidade: Os nós mais profundos são expandidos. Ele é implementado com uma LIFO numa estrutura de pilha.

Busca de custo uniforme: Os nós com custo de caminho mais baixo são expandidos. Implementado utilizando uma *heap*.

Busca em profundidade limitada: Realizar a busca em profundidade até uma altura definida.

Busca de aprofundamento iterativo: Realiza busca em profundidade em níves de altura pré definidos

2.2 Busca com informação

Utiliza conhecimento sobre o problema para encontrar soluções mais eficientes do que no caso de busca cega. É utilizada uma função para avaliar cada nó. Os algoritmos são:

Busca gulosa: Expande o nó que parece mais próximo ao objetivo de acordo com a função heurística. Segue o melhor passo considerando somente o estado atual.

Busca A*: Evita expandir caminhos que são caros.

2.3 Busca local

Apenas encontrar o estado objetivo não importando a sequencia de ações. Não tem necessidade de manter a árvore de busca.

***Hill Climbing*:** O algoritmo consiste em uma repetição que percorre o espaço de estados no sentido do valor crescente (ou decrescente). Termina quando encontra um pico (ou vale) em que nenhum vizinho tem valor mais alto.

Busca Tabu: Utiliza uma memória auxiliar com estados já visitados e eles não são visitados novamente.

***Simulated Annealing*:** Sistema de temperatura similar ao de arrefecimento de metais, a temperatura alta indica maior chance de selecionar soluções candidatas piores, enquanto temperatura baixa indica menor chance de escolher soluções piores.

3 Análise dos Problemas

Cada problema possuí uma modelagem e algoritmos que se adéquam mais à sua resolução. A seguir veremos uma descrição dos problemas, sua modelagem e os algoritmos para serem utilizados.

3.1 Cubo de Rubik

O Cudo de Rubik, mais popularmente conhecido como "Cubo Mágico", se tratada de um cubo com cada face dividida em 9 quadrados podendo ser de 6 cores diferentes. O objetivo é rotacionar os eixos do cubo para que haja somente uma cor em cada face.

Estado Inicial: Cubo embaralhado.

Ações: Girar 12 possíveis faces.

Teste de objetivo: Todas as faces com a mesma cor.

Custo de caminho: 1.

Modelo de transição: Rotacionar os eixos.

Nesse problema pode ser utilizado busca com informação com o algoritmo A* que tem um função heurística – h(n) – é número de peças fora do lugar e não a distância delas à sua posição original. Uma busca em largura resolveria esse problema, mas o número de estados gerados pela tal busca é exagerado para os atuais computadores.

3.2 Missionários e Canibais

No problema dos canibais e missionários, três missionários e três canibais devem atravessar um rio com um barco que pode transportar no máximo duas pessoas, sob a restrição de que, para ambas as margens, se há missionários presentes naquela margem, eles não podem ser ultrapassados pelo número de canibais na mesma margem (se fossem, os canibais comeriam os missionários). O barco não pode atravessar o rio por si só, sem pessoas a bordo.

Estado Inicial: Todos do mesmo lado do rio junto ao barco.

Ações: Colocar missionários e/ou canibais no barco, mover o barco de uma margem a outra.

Teste de objetivo: Todos o missionários e canibais do outro lado do rio.

Custo de caminho: 1.

Modelo de transição: Mover o barco com os passageiros.

Qualquer algoritmo de busca funciona bem, porque o espaço de estados é muito pequeno, basta eliminar estados repetidos e estados inválidos.

3.3 Problema das 8 Rainhas

O objetivo do problema é colocar 8 rainhas em um tabuleiro de forma que nenhuma rainha ataque outra. (Uma rainha ataca outra se esta estiver na diagonal e/ou horizontal e/ou vertical).

Estado Inicial: Tabuleiro vazio.

Ações: Posicionar peça(x,y).

Teste de objetivo: Qualquer estado permitido que possui a característica de não haver ataque entre as rainhas.

Custo de caminho: Zero, uma vez que o tipo de solução é o estado meta.

Modelo de transição: Posicionar peça(x,y) em uma posição não ocupada e não atacada.

Para o problema das 8 rainhas, podemos utilizar a função heurística da quantidade de quadrados não atacados por nenhuma rainha. Pode ser utilizar o algoritmo Hill climbing, porem pode ser que ele não encontre a solução que se espera. Utilizar busca sem informação também é aceita.

3.4 Sudoku

O objetivo do Sudoku é preencher uma matriz de 9x9 com dígitos para que cada coluna, fila e secção de 3x3 contenha números de 1 a 9. No início do jogo, a matriz de 9x9 terá alguns quadrados preenchidos. O seu trabalho é usar a lógica para preencher os dígitos que faltam e completar a matriz.

Estado Inicial: Algumas posições preenchidas.

Ações: Colocar um numero.

Teste de objetivo: Quando todas as posições estão preenchidas.

Custo de caminho: 1.

Modelo de transição: Colocar um número em uma posição que seja exclusivo dessa coluna, fileira do setor 3x3.

Nesse problema pode ser o algoritmo A* com uma função heurística que calcula a solução a partir da distribuição dos números.

4 Considerações Finais

Alguns algoritmos podem ser utilizados em todos os problemas, mas cada problema tem um algoritmo que acha a melhor solução com um melhor desempenho. Algoritmos sem informação precisam ser bem analisados para não entrar em loop ou percorrer muitos nós desnecessários, enquanto os algoritmos com informação precisam de um boa função heurística para ter um melhor desempenho.

Referências

[GOMES 2010] GOMES, D dos S.: Inteligência Artificial: conceitos e aplicações. In: *Olhar Científico. v1* (2010), Nr. 2, S. 234–246