Implementação do 8 puzzle utilizando Heurísticas A*

Guilherme Augusto Defalque[1]

[1]Faculdade de Computação– Universidade Federal de Mato Grosso do Sul (UFMS)
Caixa Postal 549 – 79070-900 – Campo Grande – MS – Brazil

guilhermedefalque@gmail.com

Abstract. *This scientific article describes the main features of the implementation of the game 8 puzzle using A* heuristics. This work is motived to be develop with the intention of learning search algorithms studied in discipline of Artificial Intelligence.*

Resumo. *Este artigo científico descreve as principais características da implementação do jogo 8 puzzle utilizando heurísticas A*. Este trabalho está motivado a ser desenvolvido com intuito de aprendizado em algoritmos de busca estudado na disciplina de Inteligência Artificial.*

1. Introdução

O desenvolvimento tecnológico nas mais diversas áreas, nos dias atuais, deve-se a sistemas e mecanismos automatizados que suprem as necessidades humanas, utilizando componentes que proporcionam eficácia nos resultados de maneira célere através de recursos computacionais bem aprimorados.

A utilização de algoritmos eficientes é um dos principais fatores ligados a evolução computacional. Por exemplo, a utilização de céleres em uma consulta de um banco de dados é o fator principal que determina a velocidade geral da resposta.

O foco desse trabalho é mostrar a eficiência de métodos ligados a buscas, utilizando heurísticas que fazem com que essas buscas sejam voltadas para encontrar a solução ótima, reduzindo assim o tempo de resposta.

2. 8 puzzle

O objetivo do jogo é mover as peças a partir de um estado inicial até encontrar seu estado final, quando o Puzzle está ordenado de forma crescente dizemos que ele chegou ao seu estado final. As regras do jogo são bastante simples, a peça vazia é a única que pode movimentar-se, dependendo da situação pode haver de dois a quatro movimentos possíveis (cima, baixo, direita e esquerda). Estes movimentos geram novos estados até encontrar o estado final. O Puzzle possui um espaço de estados no valor de 9.

3. Heurísticas utilizadas

Utilizou-se para a implementação do trabalho duas Heurísticas h(n):

H1(n) = número de quadrados em locais errados;

H2(n) = distância de Manhattan total: número de espaços que devem ser movimentados para chegar ao tabuleiro objetivo. Pode ser calculada da seguinte forma:

$$D = |x - x'| + |y - y'|$$

Onde x e y são as coordenadas da posição atual de um determinado número no tabuleiro e x´e y´, as posições que a peça no tabuleiro deveria estar. Deve se lembrar que deve-se calcular a soma de todo D correspondente aos 8 valores do puzzle.

4. O algoritmo A*

É um algoritmo de busca em grafos que utiliza uma função de avaliação para cada nó e garante uma solução ótima:

$$F(n) = G(n) + H(n).$$

Nesse trabalho considera-se o peso de cada G(n) como 1, tendo em vista que a distância de um nó para outro nó filho ou pai, é de uma unidade. Já H(n) é baseado nas Heurísticas H1 e H2 utilizadas.

5. A ideia

Utilizou-se a ideia baseada em buscar sempre o nó de menor caminho até o nó de destino, sendo que a distância H1(n) e H2(n) serviam como base para esse cálculo. Buscava-se, a partir de um nó inicial, percorrer a árvore pelo filho de possuía a menor heurística H(n). A cada nó visitado, marcava-se o mesmo como percorrido, a fim de não percorre-lo novamente. A cada filho não visitado, marcava-se o mesmo como não percorrido, a fim de gerar a possibilidade de, caso se necessita-se, percorre-lo novamente. No nó corrente, encontrava-se o filho de menor distância e comprava-se com a do nó da lista de não visitados que possuía o menor valor. Caso a distância do filho do nó corrente fosse menor, ele seria o próximo nó corrente. Caso contrário, o nó da lista de não visitados com a menor distância seria retirado da mesma e se tornado o nó corrente. Assim o algoritmo trabalharia até encontrar o nó final correspondente ao objetivo do jogo.

6. Problemas encontrados e possíveis soluções

Apesar da ideia, algum problema na implementação gerou uma impossibilidade de comparar os algoritmos, sendo que para alguns casos, um erro ocorre, sendo o mesmo causado por algum tipo de inconsistência na função que faz a comparação dos filhos do nó corrente com os demais nós não visitados. Apesar do entendimento do problema, não foi possível resolvê-lo em tempo hábil. As demais funções estão executando de maneira correta.

Contudo, para alguns casos de testes, ambos os programas funcionam e conseguem retornar em um tempo ótimo as soluções.

7.Conclusão

Baseando-se na teoria e nos estudos envolvendo heurísticas interligadas ao algoritmo A*, absorveu-se de maneira satisfatória e compreensível a ideia do algoritmo, apesar dos problemas encontrados na implementação do código.

Referencias

Baranauskas, José Augusto. "Estratégias de busca (2014)". http://ead.facom.ufms.br/pluginfile.php/8889/mod_resource/content/1/IA-Estrategias-Busca.pdf

Tavares, Wladimir. (2014) "Busca Heurística e Local", http:// http://lia.ufc.br/~wladimir/ia/aula4.pdfLieberknecht, Eduardo. (2013) "

Gárcia, Alberto Avedo. (2014) "8 puzzle". http://www.lcc.uma.es/~blas/apuntes/PDAv/T2011-2012/G1AlbertoAcevedoGarcia8Puzzle.pdf

Dantas, Marciel. (2014) "Jogo dos 8 com heurística h2", https://www.youtube.com/watch?v=Vd256vcm05k

Barrios, Luiz Fernando. "Comparación de Heurísticas para la solución del Problema 8-puzzle mediante el Algoritmo A* em Common Lisp", http://www.luisespino.com/pub/comparacion_heuristicas_8puzzle_astar_common_lisp-luis_espino.pdf