

Universidade de São Paulo
Instituto de Ciências Matemáticas e Computação de São Carlos
SCC-218 - Algoritmos Avançados

Trabalho 1 - Métodos de busca

Docente:
Prof Dr. João Batista do E.S. Neto

PAE:
Nicolas Oe

Monitor:
Samuel Ferreira

Alunos:	
Wesley Tiozzo	8077925
Werik Fachim	7656512

Sumário

1. O problema	2
2. Modelagem de estados	3
3. Operações	3
4. O método de busca informada	3
5. Justificativa para escolha da função de avaliação	4
6. O método de busca cega	4
7. Como compilar os programas	5
8. Formatos de entrada e saída	5
9. Experimentos realizados	6
10. Complexidade	9
11. Conclusões	9
12. Fontes utilizadas no projeto	10

1 O problema

O problema a ser modelado consiste no jogo dos 8. Esse jogo é um quebra-cabeça (puzzle) que desafia o jogador a obter uma sequência de um a oito deslocando um espaço vazio em uma matriz 3x3.

Por exemplo, dado um estado inicial em que as peças do jogo estão fora de ordem (embaralhadas):

5	4	
6	1	8
7	3	2

Estado Inicial

O objetivo do jogo consiste em o jogador chegar ao seguinte estado final:

1	2	3
4	5	6
7	8	

Estado final

Portanto, esse é um problema que pode ser modelado em um espaço de estados e utilizando métodos de busca informada e busca cega para ser resolvido.

2 Modelagem de estados

Para este problema o espaço de estados foi modelado da seguinte maneira: Cada matriz 3x3 gerada do jogo, corresponde a um estado possível.

3 Operações

Tanto para busca informada como para a busca cega são estabelecidos os movimentos possíveis do espaço vazio em cada estado possível. Os movimentos são quando possíveis: para cima, para baixo, para direita e para esquerda.

4 O método de busca informada

O método de busca informada escolhido para resolver o problema do jogo dos 8 foi o A* (A estrela), porque pode fornecer o menor número de movimentos na resolução de cada estado inicial (problema).

Para cada estado temos o cálculo da função de avaliação f.

$$f = g * \alpha + h * \beta$$

onde g é a profundidade em que se encontra o estado e h é o número de blocos fora do lugar em relação ao estado final (em ordem). Alfa tem peso 1 e beta tem peso 2.

5 Justificativa para escolha da função de avaliação

A função de avaliação foi elaborada desta maneira para que cada resultado de cada estado tenha um valor diferente baseado na profundidade (g) e no número de blocos fora do lugar (h) com relação ao estado final ordenado. A função h é admissível, pois, seu valor é único para cada estado gerado, portanto, é possível descobrir qual estado é melhor que os outros.

6 O método de busca cega

A busca cega utilizada foi a BFS devido aos seguintes motivos:

- Há várias maneiras de resolver com um estado inicial do problema e o algoritmo sempre resolve com o menor caminho, facilitando a comparação com a busca informada que também resolve o problema encontrando a mesma solução;
- A BFS apresenta melhor estabilidade nas buscas em relação ao DFS, pois o algoritmo não “se perde” em alguns casos simples e não resolve rapidamente casos “muito embaralhados”.

7 Como compilar os programas

Temos dois arquivos de programa neste trabalho. Um é o `astar8puzzle.c` e o outro é o `bfs8puzzle.c`. Para compilá-los recomendase utilizar o terminal no linux.

Os comandos para compilação são:

```
gcc o astar8puzzle Wall astar8puzzle.c
```

e

```
gcc o bfs8puzzle Wall bfs8puzzle.c
```

Obs: os comandos devem ser executados na mesma pasta onde estão os arquivos *.c.

8 Formatos de entradas e saída

Para as duas buscas a entrada é feita durante a execução do programa seguindo as instruções que aparecem na linha de comando. Exemplo:

5	4	
6	1	8
7	3	2

Para o tabuleiro 3x3 acima, tanto no `astar8puzzle.c` quanto no `bfs8puzzle`:

540618732

Lembrando que o número zero (0) representa o espaço vazio.

Também pode ser gerado um tabuleiro aleatório respondendo com 's' quando perguntado inicialmente pelo programa.

A saída também é dada pela linha de comando após a execução da busca e contém os passos para chegar na resolução no caso do código `astar8puzzle.c`

9 Experimentos realizados

Abaixo estão os experimentos realizados com vários tabuleiros gerados

aleatoriamente, cada um com diferentes quantidades de jogadas necessárias para chegar no estado final, comparando o tempo (em segundos) entre as buscas.

Tabuleiros com poucos movimentos dão resultados muito próximos de 0 e por isso não são mostrados. Em 20 movimentos necessários a busca cega tem um tempo indeterminado. O tabuleiro mais difícil contém 31 jogadas mínimas necessárias, dessa forma nenhuma das buscas resolve com mais de 31 movimentos.

Estado inicial (caso de teste)	Movimentos	BFS	A*
0 1 2 8 5 3 4 7 6	10	~0	0.015
1 2 6 0 3 5 4 7 8	11	0.016	0.015

4 3 6 7 2 1 0 5 8	12	0.031	0.015
4 3 6 7 2 1 0 5 8	13	0.175	0.015
6 3 4 1 0 2 7 5 8	14	0.329	0.016
2 1 4 7 6 3 8 0 5	15	0.634	0.015
3 5 2 7 1 4 8 6 0	16	2.767	0.015
7 1 5 8 2 0 4 3 6	17	6.950	0.015
5 4 6 2 0 1 7 3 8	18	20.385	0.046
6 5 7 2 1 0 4 8 3	19	31.589	0.156
1 5 3 7 6 2 0 8 4	20	???	0.062
1 3 7 5 4 2 6 0 8	21	???	0.031
1 3 7 5 4 2 6 8 0	22	???	0.054
4 3 8 0 5 6 1 2 7	23	???	0.109
4 6 8 2 5 7 3 1 0	24	???	0.485
2 1 6 7 8 0 4 5 3	25	???	2.538

8 6 2 5 4 1 0 3 7	26	???	2.020
8 6 2 5 4 1 3 0 7	27	???	4.464

8 6 2 5 4 1 3 7 0	28	???	7.778
8 6 2 5 4 0 3 7 1	29	???	18.080
8 6 0 5 4 2 3 7 1	30	???	33.080
8 6 7 2 5 4 3 0 1	31	???	134.066

10 Complexidade

- Busca em largura (BFS): A complexidade da busca em largura pode ser expressa por $O(V + E)$, sendo que cada vértice e cada aresta será explorada no pior caso, onde 'V' é o número de vértices e 'E' o número de arestas na representação do problema como um grafo.
- Busca A*: A complexidade do algoritmo A* depende da heurística utilizada. No pior caso, em um espaço de busca ilimitado, o número de nodos expandidos é exponencial para o menor caminho o qual seria 'd', ou seja, $O(b^d)$ onde b é o fator de ramificação (número médio de sucessores por estados). Uma boa heurística para o algoritmo A* permitiria a poda dos nodos o qual resultaria em tempo de execução polinomial.

11 Conclusões

Concluimos com este trabalho que a busca informada é muito melhor que a busca cega, pois, faz uso de informações (heurística) inerentes ao problema proposto.

Foi interessante realizar esta comparação e pudemos perceber a importância dos métodos de busca para nossa futura profissão.

12 Fontes utilizadas no projeto

[1] Material de aula encontrado no site da disciplina. Disponível em
<<http://www.lcad.icmc.usp.br/~jbatista/scc218/>>

[2] Wikipedia, 15 puzzle. Disponível em
<https://en.wikipedia.org/wiki/15_puzzle>.

[3] GeeksforGeeks, How to check if an instance of 8 puzzle is solvable?.
Disponível em
<<http://www.geeksforgeeks.org/checkinstance8puzzlesolvable/>>.

[4] The hardest eight-puzzle instances takes 31 moves to solve. Disponível
em
<<http://w01fe.com/blog/2009/01/thehardesteightpuzzleinstancestake31movesto solve/>>.