

UNIVERSIDADE FEDERAL DE SÃO CARLOS
DEPARTAMENTO DE COMPUTAÇÃO

INTELIGÊNCIA ARTIFICIAL
TURMA A

Prof Lúcia Rino

Trabalho 1

Nomes:

Matheus Fernando Finatti 379603
Rafael Eduardo Wolf de Goes 374441

Problema - 8 Puzzle

Para este trabalho, foi escolhido o problema do Racha-Cuca(8-puzzle), o qual tem como entrada uma matriz 3 por 3, sendo que 8 células são preenchidas aleatoriamente com dados que possuem alguma ordem, como por exemplo, uma imagem de Quebra-Cabeça, números, letras, entre outros. Veja exemplos:



No caso deste trabalho, o tabuleiro terá apenas números de 1 até 8 e o espaço vazio (também referenciado como 0). O intuito deste trabalho é, dado uma configuração inicial, o robô resolva o problema ordenando os número em ordem circular, deixando o espeço vazio no meio, como o exemplo de entrada e saída esperado abaixo:

Entrada		Saída
+-----+		+-----+
2 1 3	Trajetória de solução ----->	1 2 3
8 4 0		8 4
7 5 6		7 6 5
+-----+		+-----+

Para se jogar esse puzzle, apenas é possível mover o espeço vazio para vertical ou horizontal, um de cada vez. E assim, cada movimento realizado é contabilizado (+1). Assim, ganha quem resolver o puzzle com menos movimentos. (melhor solução)

Heurísticas adotadas

Para a resolução deste problema, foram adotadas duas heurísticas diferentes h1 e h2. Para ambas, foi escolhido como função g, no qual é a distância do nó até a raiz, ou seja, o nível

que se encontra o nó. Neste caso, há variação em h_1 e h_2 como descrito abaixo:

Heurística h_1 : Quantidade de células fora da posição correta

- Para cada célula posicionada em lugar errado, ou seja, que não está na posição do resultado final, a heurística tem seu valor aumentado em 1. Sendo assim, o valor da heurística para o exemplo de entrada anterior, o valor será 5. (Não inclui o 0 (vazio))

Heurística h_2 : Distância de Manhattan

- Para cada célula é contabilizado a distância vertical e horizontal da peça até sua posição correta. No mesmo caso de entrada adotado acima, o resultado desta heurística será 5.

Para ambos os casos, o valor das heurísticas são menores ou no máximo igual ao valor h^* , o qual representa o valor real de quantidade de movimentos necessários para por um peça em sua posição. Isto pois para se mover um peça para sua posição correta, as vezes é necessário mover outras peças, além disso, no jogo real não é possível mover na diagonal.

Propriedades das heurísticas

Admissibilidade

Dado a definição de admissibilidade:

Se $h(n) \leq h^*(n)$, $\forall n \in N$, h é admissível (Fonte: Slides)

Como h_1 considera a soma de elementos fora de posição, e h_2 a soma das distância de Manhattan, podemos supor que embora simples a solução do problema, $h_1(n)$ e $h_2(n)$ sempre serão menor ou igual à $h(n)$, para todo n :

```
+-----+
| 1 | 2 | 3 |  $h_1(n): 1$ 
| 0 | 8 | 4 |  $h_2(n): 1$ 
| 7 | 6 | 5 |  $h(n) : 1$ 
+-----+

+-----+
| 1 | 2 | 3 |  $h_1(n): 4$ 
| 4 | 0 | 8 |  $h_2(n): 6$ 
| 7 | 5 | 6 |  $h(n) : 14$ 
+-----+
```

Desta forma, para problemas mais complexos, $h(n)$ será sempre maior ou igual as heurísticas adotadas.

Monotonicidade

Dado a definição de monotonicidade:

$$h(N) \leq \text{custo}(N,P) + h(P), \text{ para todo } N \text{ e } P \text{ (Fonte: Wikipedia)}$$

Como o custo real sempre considera um desvio, enquanto as heurísticas h_1 e h_2 não, também é possível concluir que ambas são monótonas.

Informatividade

Visto que para cada elemento em h_2 (distância de Manhattan) que estiver fora de sua posição, o valor atribuído a esta célula será no mínimo 1, e que para h_1 (# elementos fora), o valor para cada célula fora de sua posição correta o valor será exatamente 1, podemos concluir que, dado a definição de informatividade:

$$h_x(N) \leq h_y(N) \leq h^*(N), \text{ qd } n \in N, \text{ } h_y \text{ é mais informada que } h_x \text{ (Fonte: Slides)}$$

Podemos concluir que h_2 é mais informada que h_1 , pois esta será sempre no mínimo igual à $h_1(n)$, e assim, nunca será menor. Portanto h_2 é mais informada.

Implementação

O programa foi desenvolvido em Java, e foi dividido em dois arquivos Solver.java e Tabuleiro.java.

Tabuleiro: representa o objeto tabuleiro, o qual contém a matriz do tabuleiro, o valor de cada célula, e também executa funções como mover o espavo vazio para cima, baixo, esquerda e direita, contabilizar a distância de Manhattan, e número de peças fora de posição e também imprimir o tabuleiro na tela com informações.

Solver: responsável por iniciar a execução do programa, e inicializar todos o algoritmo. Feito isso, este utiliza objetos da class Tabuleiro para executar o algoritmo e assim encontrar a solução desejada. Como número máximo de tentativas, o programa foi limitado até 10000 tentativas. Ao final da execução, é gerado 4 arquivos com todas as trajetórias descartadas, estes arquivos são:

h1-trajetorias-open.txt -> Trajetórias parciais ainda em aberto pela heurística 1

h1-trajetorias-closed.txt -> Trajetórias parciais descartadas pela heurística 1

h2-trajetorias-open.txt -> Trajetórias parciais ainda em aberto pela heurística 2

h2-trajetorias-closed.txt -> Trajetórias parciais descartadas pela heurística 2

Execução do Programa

Para a execução do programa, é necessário ter java instalado em sua máquina, o mesmo pode ser encontrado em (Pode ser qualquer versão JRE ou JDK):

http://www.java.com/pt_BR/

Após instalado, é necessário fazer a configuração das variáveis de ambientes para execução do Java por linha de comando. Veja como proceder aqui:

http://www.java.com/pt_BR/download/help/path.xml

Feito isso, abra o prompt de comandos (Windows) ou terminal (Linux, Mac), navegue até a pasta onde foi extraído o arquivo referente ao trabalho e execute o seguinte comando:

```
java -jar 8Puzzle.jar
```

Será exibido algumas informações, e será pedido para que o usuário digite os valores, que devem variar de 0 até 8 para que o programa funcione corretamente.

Exemplo de execução

Segue exemplo dos resultados gerado pelo programa após a execução do seguinte tabuleiro:

Entrada (Ilustrativa - Tabuleiro deve ser digitado na forma que o programa pede):

```
+-----+
| 2 | 1 | 3 |
| 8 | 4 | 0 |
| 7 | 5 | 6 |
+-----+
```

Saída:

<pre>Heuristica 1 - Quantidade de pecas fora do lugar Sucesso +-----+ 2 1 3 +-----+ 8 4 Soma do numero de pecas fora do lugar: 5 +-----+ 7 5 6 +-----+ +-----+ 2 1 3 +-----+ 8 4 Soma do numero de pecas fora do lugar: 4 +-----+ 7 5 6 +-----+</pre>	<pre>Heuristica 2 - Distancia de Manhattan Sucesso +-----+ 2 1 3 +-----+ 8 4 Soma distancia das pecas fora do lugar: 5 +-----+ 7 5 6 +-----+ +-----+ 2 1 3 +-----+ 8 4 Soma distancia das pecas fora do lugar: 4 +-----+ 7 5 6 +-----+</pre>
--	---

<pre> +-----+ 2 3 +-----+ 8 1 4 Soma do numero de pecas fora do lugar: 4 +-----+ 7 5 6 +-----+ </pre>	
<pre> +-----+ 2 3 +-----+ 8 1 4 Soma do numero de pecas fora do lugar: 5 +-----+ 7 5 6 +-----+ </pre>	
<pre> +-----+ 2 3 4 +-----+ 8 1 Soma do numero de pecas fora do lugar: 6 +-----+ 7 5 6 +-----+ </pre>	
<pre> +-----+ 2 3 4 +-----+ 8 1 6 Soma do numero de pecas fora do lugar: 6 +-----+ 7 5 +-----+ </pre>	
<pre> +-----+ 2 3 4 +-----+ 8 1 6 Soma do numero de pecas fora do lugar: 5 +-----+ 7 5 +-----+ </pre>	
<pre> +-----+ 2 3 4 +-----+ 8 1 6 Soma do numero de pecas fora do lugar: 6 +-----+ 7 5 +-----+ </pre>	
<pre> +-----+ 2 3 4 +-----+ 1 6 Soma do numero de pecas fora do lugar: 7 +-----+ 8 7 5 +-----+ </pre>	
<pre> +-----+ 2 3 4 +-----+ 1 6 Soma do numero de pecas fora do lugar: 7 +-----+ 8 7 5 +-----+ </pre>	
<pre> +-----+ 2 3 4 +-----+ 1 6 Soma do numero de pecas fora do lugar: 7 +-----+ 8 7 5 +-----+ </pre>	
<pre> +-----+ 2 3 +-----+ 1 6 4 Soma do numero de pecas fora do lugar: 6 +-----+ 8 7 5 +-----+ </pre>	
	<pre> +-----+ 2 1 3 +-----+ 8 5 4 Soma distancia das pecas fora do lugar: 5 +-----+ 7 6 +-----+ </pre>
	<pre> +-----+ 2 1 3 +-----+ 8 5 4 Soma distancia das pecas fora do lugar: 6 +-----+ 7 6 +-----+ </pre>
	<pre> +-----+ 2 1 3 +-----+ 5 4 Soma distancia das pecas fora do lugar: 7 +-----+ 8 7 6 +-----+ </pre>
	<pre> +-----+ 1 3 +-----+ 2 5 4 Soma distancia das pecas fora do lugar: 8 +-----+ 8 7 6 +-----+ </pre>
	<pre> +-----+ 1 3 +-----+ 2 5 4 Soma distancia das pecas fora do lugar: 7 +-----+ 8 7 6 +-----+ </pre>
	<pre> +-----+ 1 3 +-----+ 2 5 4 Soma distancia das pecas fora do lugar: 8 +-----+ 8 7 6 +-----+ </pre>
	<pre> +-----+ 1 3 4 +-----+ 2 5 Soma distancia das pecas fora do lugar: 9 +-----+ 8 7 6 +-----+ </pre>
	<pre> +-----+ 1 3 4 +-----+ 2 5 Soma distancia das pecas fora do lugar: 8 +-----+ 8 7 6 +-----+ </pre>
	<pre> +-----+ 1 3 4 +-----+ 2 5 Soma distancia das pecas fora do lugar: 7 +-----+ 8 7 6 +-----+ </pre>
	<pre> +-----+ 1 3 4 +-----+ 8 2 5 Soma distancia das pecas fora do lugar: 6 +-----+ 7 6 +-----+ </pre>

```
+-----+
| 2 | 3 |
+-----+
| 1 | 6 | 4 |
+-----+
| 8 | 7 | 5 |
+-----+
```

Soma do numero de pecas fora do lugar: 5

```
+-----+
| 2 | 3 |
+-----+
| 1 | 6 | 4 |
+-----+
| 8 | 7 | 5 |
+-----+
```

Soma do numero de pecas fora do lugar: 4

```
+-----+
| 1 | 2 | 3 |
+-----+
| 6 | 4 |
+-----+
| 8 | 7 | 5 |
+-----+
```

Soma do numero de pecas fora do lugar: 3

```
+-----+
| 1 | 2 | 3 |
+-----+
| 8 | 6 | 4 |
+-----+
| 7 | 5 |
+-----+
```

Soma do numero de pecas fora do lugar: 2

```
+-----+
| 1 | 2 | 3 |
+-----+
| 8 | 6 | 4 |
+-----+
| 7 | 5 |
+-----+
```

Soma do numero de pecas fora do lugar: 1

```
+-----+
| 1 | 2 | 3 |
+-----+
| 8 | 4 |
+-----+
| 7 | 6 | 5 |
+-----+
```

Soma do numero de pecas fora do lugar: 0

Tabuleiro foi resolvido em 1267 tentativas (eficiencia)
Eficacia: 17 movimentos

Gerando arquivos para trajetorias descartadas
Arquivos gerados com sucesso
Esta soluçao tomou: 0.929108258 segundos

```
+-----+
| 1 | 3 | 4 |
+-----+
| 8 | 2 | 5 |
+-----+
| 7 | 6 |
+-----+
```

Soma distancia das pecas fora do lugar: 5

```
+-----+
| 1 | 3 | 4 |
+-----+
| 8 | 2 | 5 |
+-----+
| 7 | 6 |
+-----+
```

Soma distancia das pecas fora do lugar: 4

```
+-----+
| 1 | 3 | 4 |
+-----+
| 8 | 2 |
+-----+
| 7 | 6 | 5 |
+-----+
```

Soma distancia das pecas fora do lugar: 3

```
+-----+
| 1 | 3 |
+-----+
| 8 | 2 | 4 |
+-----+
| 7 | 6 | 5 |
+-----+
```

Soma distancia das pecas fora do lugar: 2

```
+-----+
| 1 | 3 |
+-----+
| 8 | 2 | 4 |
+-----+
| 7 | 6 | 5 |
+-----+
```

Soma distancia das pecas fora do lugar: 1

```
+-----+
| 1 | 2 | 3 |
+-----+
| 8 | 4 |
+-----+
| 7 | 6 | 5 |
+-----+
```

Soma distancia das pecas fora do lugar: 0

Tabuleiro foi resolvido em 417 tentativas (eficiencia)
Eficacia: 17 movimentos

Gerando arquivos para trajetorias descartadas
Arquivos gerados com sucesso
Esta soluçao tomou: 0.087361324 segundos