



GSIO50 - INTELIGENCIA ARTIFICIAL

Profa. Márcia Aparecida

Dhiogo Pereira Santos – 12021BSI262

Ellen Christina Amaral Santana – 12011BSI208

Uberlândia, 02 de junho de 2022



- Descrição do problema

O quebra-cabeça de 8 é um quebra-cabeça deslizante que é jogado em uma grade de 3 por 3 com 8 peças quadradas marcadas de 1 a 8, mais um quadrado em branco. O objetivo é reorganizar as peças para que fiquem na ordem da linha maior, usando o menor número de movimentos possível. Você tem permissão para deslizar ladrilhos horizontal ou verticalmente no quadrado em branco.

- Principais elementos para a definição do algoritmo

Os principais elementos para a definição do algoritmo foram as seguintes classes:

- No: contém o nó pai e propriamente os atributos de um nó, tais como f , que representa o custo final para cada estado, que é dado pela soma dos atributos $g(n)$ e $h(n)$ e um atributo para indicar qual o tipo de heurística que será utilizada na execução do algoritmo, "m" ou "e", que é atribuído ao nó inicial e, cada vez que um novo nó é criado, este segue a heurística do pai.
- Sucessores: tem-se uma lista com os nós sucessores e funções para expandir e testá-los frente ao objetivo.



-Objetivo: classe do modelo Singleton, que permite apenas uma instanciação, representando o estado objetivo, visto que ele é usado por outras classes.

- Borda: usada para a lista com os nós a serem expandidos e para a lista com os nós que já foram expandidos. Possuem, também, um atributo que armazena a quantidade de elementos.

- Main: executa o programa.

- Heurísticas utilizadas

$h1$ = o número de blocos em posições erradas.

$h1$ é admissível, pois um bloco fora do lugar deve ser movido pelo menos uma vez.

$h2$ = a soma das distâncias dos blocos as suas posições objetivo. É a distância horizontal e vertical, que é conhecida como distância de Manhattan.

$h2$ é admissível, pois qualquer movimento desloca algum bloco para uma posição mais próxima do objetivo.



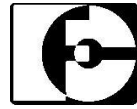
- Conclusão relativa aos resultados

Após vários testes, notamos que a implementação que usa a heurística da distância de Manhattan, apresenta resultados expressivamente mais eficientes, como por exemplo quanto ao custo computacional e ao tempo de execução.

- Imagem em que não se encontra solução para o problema

```
Inserir como no modelo -> {{x,x,x},{x,x,x},{x,x,x}}

Insira o estado final:
{{1,2,3},{4,5,6},{7,8,0}}
Insira o estado inicial:
{{8,1,2},{0,4,3},{7,6,5}}
Insira o tipo de heuristica:
(m = Manhattan | e = Posições erradas)
m
Borda:
No@1d81eb93
Solução não encontrada!
```



- Referências

https://ufubr.sharepoint.com/sites/IA_BSI/Material%20de%20Aula/Semana3/Aula%206%20-%20Heur%C3%ADsticas.pdf?CT=1653997002799&OR=ItemsView

http://www.dsc.ufcg.edu.br/~joseana/IAPos_NA04.pdf

<https://github.com/search?l=Java&q=quebra+cabe%C3%A7as+a%2A&type=Repositories>

<http://www.decom.ufop.br/menotti/paa111/files/PCC104-111-ars-11.1-NelsonFlorencioJunior.pdf>