

Projeto 1 - Inteligência Artificial – Grupo 23

Introdução

Este projeto consiste num conjunto de funções que ajudam um agente a resolver diferentes puzzles (tabuleiros) de uma variante do jogo Solitaire, independentemente da sua dimensão e do seu conteúdo.

O tabuleiro usado para este jogo tem dimensões linhas x colunas e pode nem sempre ser quadrado ou retangular. As peças do jogo são representadas pelo carater “O”, os espaços livres são representados por “_” e os espaços que estão bloqueados são representados por “X”.

Existem **quatro condições** para fazer uma **jogada válida**:

- 1) A peça só se pode mover ortogonalmente (cima, baixo, esquerda, direita);
- 2) A peça só pode ser movida, se passar por cima de outra peça e eliminá-la do jogo;
- 3) Uma peça não ficar em cima de outra que já estava nessa posição;
- 4) Não é possível mover uma peça para um espaço bloqueado (“X”).

O **objetivo** do jogo é chegar ao fim com apenas uma peça no tabuleiro, independentemente da posição onde está.

O relatório está organizado em quatro partes, sendo que na primeira é explicado a escolha da **heurística**, na segunda os **testes e resultados** usados para fazer a avaliação da procura, e por fim uma **discussão dos resultados**.

1. Heurística

No caso das procuras informadas (**A*** e **Greedy Search**), foi escolhida uma composição linear de heurísticas, sendo estas:

- 1) Dar maior peso, e como tal menor preferência, aos tabuleiros cuja posição final da ação leva a peça para algum canto do mesmo;
- 2) Quanto menos peças tiver um tabuleiro, menor será o peso do nó, dando preferência a esse nó.

A solução do tabuleiro é encontrada, quando a função *goal_test* retorna *True*, ou seja, o número de peças existentes no tabuleiro é 1.

2. Exemplos de Teste

- Tabuleiro de 5x5 (linhas x colunas):
[["_", "O", "O", "O", "_"], ["O", "_", "O", "_", "O"], ["_", "O", "_", "O", "_"], ["O", "_", "O", "_", "_"], ["_", "O", "_", "_", "_"]]
- Tabuleiro de 4x4 (linhas x colunas):
[["O", "O", "O", "X"], ["O", "O", "O", "O"], ["O", "_", "O", "O"], ["O", "O", "O", "O"]]
- Tabuleiro de 4x5 (linhas x colunas):
[["O", "O", "O", "X", "X"], ["O", "O", "O", "O", "O"], ["O", "_", "O", "_", "O"], ["O", "O", "O", "O", "O"]]

- Tabuleiro de 4x6 (linhas x colunas):

```
[[ "O", "O", "O", "X", "X", "X"], [ "O", " ", "O", "O", "O", "O"], [ "O", "O", "O", "O", "O", "O"], [ "O", "O", "O", "O", "O", "O"]]
```

3. Resultados dos Testes

Procura-Gananciosa

Tabuleiro	Tempo de execução (s)	Nós gerados	Nós expandidos
1	0.000000	11	19
2	0.015625	53	78
3	4.593750	21 872	21 912
4	0.593750	1 729	1 810

Procura em Profundidade Primeiro

Tabuleiro	Tempo de execução (s)	Nós gerados	Nós expandidos
1	0.015625	12	20
2	0.640625	5 984	6 002
3	7.375000	53 636	53 664
4	3 065.750000 (51 min)	14 760 524	14 760 576

Procura A*

Tabuleiro	Tempo de execução (s)	Nós gerados	Nós expandidos
1	0.000000	11	19
2	0.015625	53	78
3	4.500000	21 872	21 912
4	0.562500	1 729	1 810

4. Discussão dos Resultados

Neste relatório utilizamos apenas as procuras **A***, **Greedy**, **Depth First Graph Search** para testarmos a complexidade das heurísticas utilizadas e também do programa em si.

Tal como se pode verificar pelos tempos obtidos, vemos que o tempo utilizado pelos dois algoritmos de procura informada (**A*** e **Greedy**) é muito menor que aquele utilizado pela **DFS**. Isto deve-se ao facto de a **DFS** expandir todos os nós em profundidade até encontrar o objetivo, enquanto os algoritmos de procura informada apenas expandem os nós que têm menor *peso*, recorrendo às heurísticas para diferenciar o *peso* de cada nó.

Como era de esperar, o aumento dos nós gerados na **DFS** é exponencial, como podemos verificar adicionando apenas mais uma linha ou coluna no tabuleiro, o **número de nós** aumenta cerca de **8 vezes** do tabuleiro 2 para o tabuleiro 3, e cerca de **275 vezes** do tabuleiro 3 para o tabuleiro 4.

Tal como na **DFS** em que o aumento de nós deve-se ao aumento das dimensões do tabuleiro ou peças existentes, também na **A*** e no **Greedy** o aumento de nós deve-se à mesma coisa. No entanto, notamos que o aumento dos nós não é exponencial como na **DFS**.

Devido a problemas nas heurísticas definidas pelo nosso grupo, verificamos que apesar das mesmas otimizarem o tempo de procura nos tabuleiros 1, 2 e 4, o mesmo não acontece no tabuleiro 3, no qual vemos que há um crescimento incompreensível de nós gerados.

Verificando sempre os resultados obtidos, foi sempre retornado um último nó cujo tabuleiro correspondia ao estado objetivo.