

***Peg Solitaire* - Relatório**

Descrição do problema

O projeto teve como objetivo desenvolver um programa que resolve diferentes puzzles de uma variante do jogo *Peg Solitaire*. Foram disponibilizados três algoritmos de procura: *Greedy*, *A** e procura em profundidade primeiro (PPP).

Implementação da solução

Teoria:

Os fatores limitantes da solução são o tempo e espaço. As partes com maior efeito na execução do algoritmo são a heurística e a função de custo, utilizadas nos algoritmos de procura informada.

A heurística a utilizar não tem de refletir o custo do caminho e não faz sentido avaliar se é aceitável ou consistente, pois trata-se de um problema de satisfação em que todas as soluções estarão no nível $n-1$ da árvore de procura (sendo o nível 0 o estado inicial) em que n é o número de berlindes do estado inicial.

Sendo assim, devemos implementar uma heurística que guie a procura para casos favoráveis. A ideia é desencorajar o algoritmo a seguir caminhos pouco prováveis de levarem a uma solução, promovendo caminhos com certas características promissoras.

Função de custo implementada:

A função de custo vai ser o fator diferenciador entre as procuras *Greedy* e *A**. Deve ajudar a heurística a não ser induzida em erro com caminhos que parecem promissores a curto prazo a custo de alguma penetração. A nossa implementação foi a adição de uma constante (1) sempre que se desce um nível na árvore.

Função heurística implementada:

A nossa implementação da heurística avalia o tabuleiro consoante o número de berlindes encostados a uma parede (+3), cantos (+6) e posições inválidas (+1) – peças de risco.

A heurística avalia um estado com um valor tanto maior quanto maior for o número de peças de risco no tabuleiro. As peças de risco têm menos opções para serem removidas do tabuleiro, como tal, esta heurística dá prioridade a estados que tenham conseguido retirar peças de posições de risco e desencoraja a colocação de peças nessas posições, visto que, de maneira geral, um estado sem peças de risco é um estado mais promissor.

Avaliação teórica

Todas as procuras têm complexidade temporal exponencial $O(b^{n-1})$. As procuras informadas, no entanto, podem ter menos dependendo da heurística pelo que deve ser medida experimentalmente.

A PPP tem complexidade espacial linear $O(nb)$. As procuras informadas têm complexidade espacial exponencial $O(b^{n-1})$, mas quanto maior for a penetração de heurística, menos complexidade experimental terá.

É importante referir que a procura A^* é completa, bem como, neste caso, a *Greedy* e a PPP, visto que neste jogo o tabuleiro chegará sempre a um estado no qual já não existem jogadas possíveis e porque o desenvolvimento do problema é feito através de uma árvore, pelo que não tem ciclos.

Comparação de procuras

Correu-se o programa com os 4 tabuleiros disponibilizados no enunciado, estando os resultados disponíveis nas tabelas que se seguem:

A^*	Nós expandidos	Nós gerados	Tempo (s)
Tabuleiro 1	20	29	25×10^{-3}
Tabuleiro 2	1348	1399	0.10
Tabuleiro 3	15060	15104	1.25
Tabuleiro 4	5564	5707	0.66

Tabela 1: Resultados da procura A^*

<i>Greedy</i>	Nós expandidos	Nós gerados	Tempo (s)
Tabuleiro 1	20	29	27×10^{-3}
Tabuleiro 2	1649	1688	0.11
Tabuleiro 3	17848	17887	1.10
Tabuleiro 4	2526	3068	0.64

Tabela 2: Resultados da procura *Greedy*

PPP	Nós expandidos	Nós gerados	Tempo (s)
Tabuleiro 1	14	21	8.0×10^{-3}
Tabuleiro 2	741	750	0.02
Tabuleiro 3	93008	93009	2.70
Tabuleiro 4	-----	-----	-----

Tabela 3: Resultados da PPP. O Tabuleiro 4 não foi resolvido em tempo útil.

Discussão dos resultados

O objetivo foi cumprido, visto que todos os tabuleiros são resolvidos por procuras informadas em menos de 1 minuto. Os algoritmos A^* e *Greedy* têm tempos, número de nós expandidos e gerados semelhantes, o que era esperado pois a função custo implementadas pouco contribui para o $f(n)$ da procura A^* .

Os tempos da PPP foram melhores do que ambas as procuras informadas nos tabuleiros 1 e 2 devido ao acaso, sendo que o tabuleiro 4 não foi resolvido por esta em tempo útil. Quanto aos tabuleiros mais complexos, é notório que as procuras informadas são mais eficientes, resolvendo o problema em menos tempo e gerando/expandindo menos nós.