INTELIGÊNCIA ARTIFICIAL

Relatório 1º Projeto (2018/2019)

Grupo 22 (Alameda) | Francisco Sena - 86420 | João Almeida – 86447

1. Introdução

O problema que foi proposto consiste em resolver diferentes puzzles de uma variante do jogo *Solitaire.* Este jogo é para um único agente e tem como objetivo retirar peças do tabuleiro até sobrar apenas uma peça.

2. Avaliação Experimental

Os critérios usados para comparar os algoritmos de procura foram o **tempo de execução**, os **nós gerados** e os **nós expandidos**.

Procedendo à análise do programa realizaram-se vários testes, dos quais foram extraídos os critérios pedidos. Estes valores resultam de uma média ponderada de 5 medições e apresentam-se na seguinte tabela:

Tabuleiro 5x5:

	Tempo (s)	Nós gerados	Nós expandidos
DFS	0.0010	59	34
Greedy	0.0031	19	10
A*	0.0015	38	20

Tabuleiro 4x4:

	Tempo (s)	Nós gerados	Nós expandidos
DFS	0.2684	8189	8095
Greedy	0.0095	123	88
A*	0.0110	246	176

Tabuleiro 4x5:

	Tempo (s)	Nós gerados	Nós expandidos
DFS	4.2943	149548	149430
Greedy	1.1624	15394	15349
A*	1.1548	30788	30698

Tabuleiro 4x6:

	Tempo (s)	Nós gerados	Nós expandidos
DFS	-	-	-
Greedy	0.1104	935	826
A*	0.1071	1870	1652

3. Análise Experimental – considerações sobre eficiência, penetrância

No tabuleiro 5x5 a procura usando DFS comporta-se tão bem ou melhor que as Greedy e A* apesar de ser uma procura cega. Isto deve-se ao facto de o tabuleiro ser esparso, ou seja, em cada estado o número de movimentos possíveis é reduzido, o que leva a um fator de ramificação baixo.

Nos restantes tabuleiros, dado que a profundidade das árvores de procura é constante em qualquer algoritmo, as procuras informadas, dado que utilizam uma heurística, têm uma penetrância bastante superior pois examinam muito menos nós que as procuras não informadas. Isto resulta numa diferença de tempo considerável para encontrar a solução, como é observável no caso do tabuleiro 4x6, onde ao ser utilizada uma procura em profundidade primeiro (DFS), esta não termina sequer em tempo útil, ao contrário das procuras informadas.

4. Conclusões - completude, otimalidade e função heurística

As três procuras garantem a **completude**, pois em cada ação tomada é sempre retirada uma peça do tabuleiro, logo é impossível a procurar entrar em loop. A **otimalidade**, que se traduz em encontrar a solução de maior qualidade quando há várias distintas, não é relevante para este projeto visto que todas as soluções possíveis estarão ao mesmo nível de profundidade. Sendo assim, a heurística escolhida apenas ajuda a resolver o problema de forma mais rápida. Tem-se que para qualquer tabuleiro, o custo ótimo do problema (C*) é sempre igual a *(nº_de_peças – 1)*.

Observando o problema, a primeira verificação de quão bom é um estado são os possíveis movimentos que podem ser feitos a partir dele. Quanto maior o número de movimentos, maior a possibilidade de continuar o jogo. Portanto, temos que um estado *A* é melhor que outro *B* quando *nº_moves_possíveis_A* > *nº_moves_possíveis_B*. É sabido que peças isoladas (sem peças adjacentes) podem impedir que se complete o jogo. Portanto, outro fator a ter em conta é que um estado é tanto melhor quanto menor o número de peças isoladas no tabuleiro. Por fim, quanto menor o número de peças no tabuleiro, melhor é o estado do tabuleiro. A **função heurística** utilizada nas procuras informadas resulta de uma combinação destes 3 fatores apresentados. Para h(*nó.estado_solução*) = 0, basta subtrair 2 dado que num estado solução há apenas uma peça no tabuleiro, uma peça isolada e zero movimentos possíveis (1 + 1 - 0 - 2 = 0). A **função heurística** é, então:

- $h(n) = (nº\_de\_peças)^2 + nº\_peças\_isoladas - nº\_de\_movimentos\_possíveis - 2$

Em relação à admissibilidade e consistência da heurística, é facilmente observável que esta não é admissível, logo não é consistente. Para mostrar a não admissibilidade da heurísitica, basta considerar um tabuleiro inicial de *Solitaite* tradicional com tamanho *N*M* (com *N*M - 1* peças) onde no nó inicial a função heurística terá o valor de $(N*M - 1)^2 + 0 - 4 - 2 = (N*M - 1)^2 - 6 \quad > \quad N*M - 2$ (**h***).