

Análise Teórica

A heurística implementada para a procura informada consiste no cálculo do número de conjuntos de peças presentes num tabuleiro. Este cálculo não é otimista, portanto a heurística implementada não é admissível. Isto faz com que não haja garantia de optimalidade na solução, no entanto, tal não é relevante neste caso porque queremos apenas encontrar uma solução possível para o problema. Esta heurística usa apenas a função *board_find_groups*, pelo que as suas complexidades são as mesmas ($O(L \times C^2)$).

Relativamente à completude, uma vez que cada ação no jogo é irreversível, não é possível repetir sequências de estados no tabuleiro e, portanto, se existir uma solução esta é sempre encontrada por todos os métodos.

Para encontrar conjuntos de peças no tabuleiro, este é percorrido linha a linha, coluna a coluna e quando são encontradas peças adjacentes com a mesma cor, o id do conjunto da peça é propagado para os adjacentes, agrupando-os. A complexidade temporal deste procedimento é de $O(L \times C^2)$, sendo que na maior parte dos casos é $L \times C$.

Relativamente aos métodos de procura, teoricamente a procura em profundidade primeiro tem complexidade $O(L \times C^2 \times r^m)$, a procura gananciosa tem complexidade $O(L \times C^2 \times b^m)$ e a procura A^* tem complexidade $O(L \times C^2 \times b^m)$, sendo que a presença de uma boa heurística, permita complexidades mais baixas na procura gananciosa e A^* .

L - Número de linhas, *C* - Número de colunas
b - Fator de ramificação, *r* - fator de expansão, *m* - profundidade.

Resultados

Apresentamos de seguida os testes apresentados no enunciado e aplicados ao nosso programa. Os testes foram corridos num computador com um processador i5 de 2.3GHz. Todos os tempos obtidos são a média de cinco repetições do respetivo teste.

Teste 1: Tabuleiro de 4x5 (linhas x colunas) com 2 cores sem solução

Input: *[[1,2,1,2,1], [2,1,2,1,2], [1,2,1,2,1], [2,1,2,1,2]]*

	Procura em Profundidade Primeiro	Procura Gananciosa	A^*
<i>Tempo de execução (s)</i>	0.0726s	0.0744s	0.0766s
<i>Nós expandidos</i>	1	1	1
<i>Nós gerados</i>	0	0	0

Teste 2: Tabuleiro de 4x5 (linhas x colunas) com 3 cores

Input: *[[1,2,2,3,3], [2,2,2,1,3], [1,2,2,2,2], [1,1,1,1,1]]*

	Procura em Profundidade Primeiro	Procura Gananciosa	A^*
<i>Tempo de execução (s)</i>	0.0754s	0.0780s	0.0766s
<i>Nós expandidos</i>	4	3	4
<i>Nós gerados</i>	7	6	7

Teste 3: Tabuleiro de 10x4 (linhas x colunas) com 3 cores sem solução

Input: [[3,1,3,2], [1,1,1,3], [1,3,2,1], [1,1,3,3], [3,3,1,2], [2,2,2,2], [3,1,2,3], [2,3,2,3], [5,1,1,3], [4,5,1,2]]

	Procura em Profundidade Primeiro	Procura Gananciosa	A*
<i>Tempo de execução (s)</i>	12.4554s	25.9684s	26.6036s
<i>Nós expandidos</i>	74702	74702	74702
<i>Nós gerados</i>	74701	74701	74701

Teste 4: Tabuleiro de 10x4 (linhas x colunas) com 3 cores

Input: [[3,1,3,2], [1,1,1,3], [1,3,2,1], [1,1,3,3], [3,3,1,2], [2,2,2,2], [3,1,2,3], [2,3,2,3], [2,1,1,3], [2,3,1,2]]

	Procura em Profundidade Primeiro	Procura Gananciosa	A*
<i>Tempo de execução (s)</i>	0.0916s	0.0970s	0.0880s
<i>Nós expandidos</i>	54	42	24
<i>Nós gerados</i>	85	59	43

Teste 5: Tabuleiro de 10x4 (linhas x colunas) com 5 cores

Input: [[1,1,5,3], [5,3,5,3], [1,2,5,4], [5,2,1,4], [5,3,5,1], [5,3,4,4], [5,5,2,5], [1,1,3,1], [1,2,1,3], [3,3,5,5]]

	Procura em Profundidade Primeiro	Procura Gananciosa	A*
<i>Tempo de execução (s)</i>	8m12s	0.1924s	0.0968s
<i>Nós expandidos</i>	3123308	256	16
<i>Nós gerados</i>	3123363	319	91

Análise de Resultados

Comparando os resultados, é possível verificar que as complexidades mencionadas na análise teórica se verificam. Nomeadamente, quando é aumentado o fator de ramificação, neste caso, o número de cores, o número de nós expandidos e gerados aumenta exponencialmente na procura em profundidade primeira, sendo que nas procuras informadas se mantém linear.

Relativamente ao tamanho do tabuleiro, quando este aumenta, o número de nós expandidos e gerados não aumenta tão significativamente na procura não informada, sendo que as procuras informadas continuam a ser as mais eficientes, nomeadamente a A*.

Finalmente, nos casos sem solução a procura não informada mostra-se ser tão ou mais eficiente do que as procuras informadas.