TÉCNICO LISBOA

Relatório do Primeiro Projeto de IA

Programa em Python que resolva uma variante do Same Game

Grupo: 038

Para efeitos de medida de uma implementação do agente para o jogo Same Game, segue-se um conjunto de resultados de 5 problemas com tabuleiros de atributos variados. Com base nos dados obtidos iremos refletir sobre a eficiência, completude, comparação entre métodos e heurística.

Resultados das Procuras:

Os seguintes resultados de Tempo de Execução baseiam-se numa média de 5 testes efetuados em cada problema (com a exceção do 5º Problema por demorar muito tempo).

Os números dispostos de nós em tabuleiros sem solução correspondem ao número de nós expandidos/gerados até o programa concluir que não tinha solução.

- 1º Problema - Tabuleiro de 4x5 (linhas x colunas) com 2 cores sem solução

	Profundidade Primeiro	Procura Gananciosa	Procura A*
Tempo de execução	0.157s	0.097s	0.100s
Nº de nós expandidos	1	1	1
Nº de nós gerados	0	0	0

- 2º Problema - Tabuleiro de 4x5 (linhas x colunas) com 3 cores

	Profundidade Primeiro	Procura Gananciosa	Procura A*
Tempo de execução	0.137s	0.102s	0.106s
Nº de nós expandidos	4	3	3
Nº de nós gerados	7	6	6

- 3º Problema - Tabuleiro de 10x4 (linhas x colunas) com 3 cores sem solução

	Profundidade Primeiro	Procura Gananciosa	Procura A*
Tempo de execução	19.087s	34.097s	35.634s
Nº de nós expandidos	74 702	74 702	74 702
Nº de nós gerados	74 701	74 701	74 701

- 4º Problema - Tabuleiro de 10x4 (linhas x colunas) com 3 cores

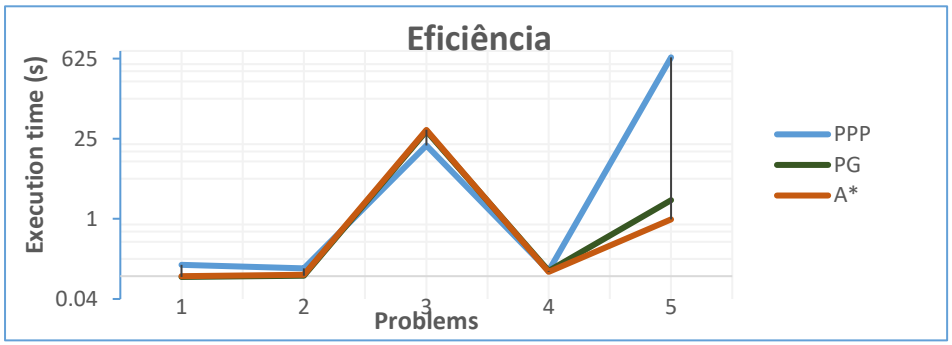
	Profundidade Primeiro	Procura Gananciosa	Procura A*
Tempo de execução	0.124s	0.123s	0.119s
Nº de nós expandidos	54	42	42
Nº de nós gerados	85	59	59

- 5º Problema - Tabuleiro de 10x4 (linhas x colunas) com 5 cores

	Profundidade Primeiro	Procura Gananciosa	Procura A*
Tempo de execução	10m 57.387s	2.128s	0.973s
Nº de nós expandidos	3 123 308	4 637	1 500
Nº de nós gerados	3 123 363	4 692	1 645

Análise Crítica dos resultados obtidos:

- **Completude:**
  - A Procura Profundidade Primeiro demonstrou ser completa nos problemas anteriores pois em cada problema não houve ciclos infinitos (profundidade infinita).
  - A Procura Gananciosa é semelhante à Procura em Profundidade pois também é completa quando não há ciclos infinitos apesar de ser mais exigente em memória.
  - O A* é uma procura completa porque não houve nenhum caso em que o número de nós com f ≤ f(G) fosse infinito, por outras palavras, infinitos estados em que o custo é inferior ao do objetivo.
- **Eficiência**:
  - Segundo os resultados obtidos, o A* foi a procura mais eficiente no caso de o tabuleiro ter uma grande dimensão e não dependendo do número de cores (4º e 5º problemas). No entanto, verificou-se a sua ineficiência no caso de o tabuleiro ser pequeno e ter poucas cores (2º Problema) devido ao facto de ter de calcular a heurística, mas também no caso do tabuleiro ser grande.
  - A Procura Gananciosa (PG) foi a mais eficiente no 1º e 2º problema, casos em que os tabuleiros são pequenos e contêm poucas peças de cores diferentes. Nos restantes 3 problemas também não foi a mais ineficiente, como tal o seu comportamento foi o mais equilibrado de entre os 3 métodos.
  - Por último, a Procura Profundidade Primeiro (PPP) foi a mais ineficiente no 1º, 2º, 4º e 5º tabuleiros, ou seja, na maioria dos tabuleiros com dimensões suficientemente grandes e de um número consideravelmente alto de cores, porque percorre a árvore "cegamente", gerando e expandindo todos os nós até encontrar a solução (apesar de um dos tabuleiros ser pequeno e ter poucas cores). Por outro lado, foi a mais eficiente no 3º problema em que o tabuleiro não tem solução, e portanto todos os estados acabam por ser expandidos, por não ter de calcular custos para esses mesmos estados.



Heurística

Como heurística optámos por escolher o número de grupos de peças uma vez que é uma boa aproximação do número de ações necessárias para terminar o jogo. A Procura Gananciosa em teoria tem uma complexidade temporal $O(b^m)$, mas a nossa heurística conseguiu reduzir este valor consideravelmente de maneira a que este método se tornou o mais eficiente possível na maioria dos problemas. A Procura A* tem uma complexidade temporal exponencial $O(b^d)$ no caso geral, mas conseguimos fazer com que expandisse o menor número de nós no caminho da solução ótima (em comparação com os outros métodos).