

Grupo 004: João Antunes nº 87668, Maria Inês Moraes nº 83609  
Relatório Projeto – 1ª Entrega  
Inteligência Artificial

Resultados dos problemas:

Para cada um dos quatro problemas fornecidos foi efetuada uma procura em profundidade primeiro, uma procura gananciosa e uma procura A\*. Os resultados obtidos contêm o tempo de execução, que consiste numa média de 5 testes realizados para cada problema, o número de nós expandidos e o número de nós gerados.

**Problema 1** - Tabuleiro de 5x5 (linhas x colunas)

	Nós expandidos	Nós gerados	Tempo de execução (s)
Profundidade Primeiro	13	20	0,0065
Procura gananciosa	12	21	0,0109
Procura A*	13	23	0,0126

**Problema 2** - Tabuleiro de 4x4 (linhas x colunas)

	Nós expandidos	Nós gerados	Tempo de execução (s)
Profundidade primeiro	5985	6002	1,6407
Procura gananciosa	332	365	0,1870
Procura A*	490	892	0,9922

**Problema 3** - Tabuleiro de 4x5 (linhas x colunas)

	Nós expandidos	Nós gerados	Tempo de execução (s)
Profundidade primeiro	53637	53664	19,9063
Procura gananciosa	5452	5509	3,4116
Procura A*	2039	2695	3,1961

**Problema 4** - Tabuleiro de 4x6 (linhas x colunas)

	Nós expandidos	Nós gerados	Tempo de execução (s)
Profundidade primeiro	14760525	14760576	9213,0969
Procura gananciosa	1091	1261	0,8600
Procura A*	53	304	0,2068

### Análise crítica dos resultados:

- **Compleitude:** Através dos resultados obtidos podemos observar que para cada um dos quatro problemas anteriores foi encontrada uma solução e nenhum dos algoritmos de procura entrou em ciclo infinito ao resolver os vários tabuleiros. Assim, podemos concluir que todos os três algoritmos de procura utilizados são completos.
- **Eficiência:**
  - *Procura em profundidade primeiro:* Para os testes efetuados este algoritmo de procura foi o mais eficiente a resolver o 1º problema. Isto deve-se ao facto de o tabuleiro deste problema ser o que contém o menor número de peças, o maior número de espaços livres e consequentemente, o maior número de movimentos válidos (i.e., o deslocamento de uma peça sobre outra peça para uma posição vazia). O algoritmo pode simplesmente percorrer a árvore "inconscientemente", gerando e expandindo todos os nós até encontrar a solução. Pelo contrário, nos tabuleiros do 2º, 3º e 4º problema que apresentam uma maior quantidade de peças e poucos espaços livres, o algoritmo torna-se muito ineficiente, exibindo um tempo de execução muito elevado, bem como um grande número de nós gerados e expandidos.
  - *Procura gananciosa:* Trata-se do algoritmo mais eficiente para o 2º problema. Dado que se refere a um algoritmo de procura informada, quando o tabuleiro apresenta mais peças e menos espaços livres, este resolve o problema mais rapidamente que uma procura em profundidade primeiro, pois a procura gananciosa compreende conhecimento específico do problema utilizado para determinar a ordem de expansão dos nós.
  - *Procura A\*:* Corresponde a um algoritmo de procura informada que tem em conta o custo de uma transição de um nó para o outro. Assim, apresenta-se como a procura mais eficiente para o 3º e 4º problema. Uma vez que os tabuleiros destes problemas compreendem maiores dimensões, a procura A\* possui melhor desempenho que a procura gananciosa, pois tem em consideração os custos de transições entre peças.
- **Heurística:** A heurística implementada é a soma do número de peças do tabuleiro com o número de peças cuja posição não é possível de alterar devido ao facto de não possuírem movimentos válidos.

Apesar de não haver noção de solução ótima, ao correr os testes, as procuras informadas dão soluções com menores tempos de execução e menores números de nós gerados e expandidos do que os da procura em profundidade primeiro. Assim, podemos concluir que esta heurística é adequada para resolução dos problemas e permite poupar recursos face à pesquisa não informada.