

**Introdução:**

Este relatório expõe uma análise sobre a performance de 3 algoritmos de procura em várias instâncias duma adaptação do jogo Solitário. Os algoritmos são:

- Procura em Profundidade Primeiro(DFS)
- Procura Gananciosa (Greedy Search)
- Procura A\* (A\*)

A DFS é uma procura cega. Ela procura todo o espaço de estados (em profundidade) até encontrar o primeiro estado objetivo, sendo que não é aplicado qualquer critério na escolha dos ramos. A largura e profundidade da árvore de procura serão factores críticos no desempenho deste algoritmo (pois ele não considera nenhuma heurística).

A segunda (Greedy Search) é uma procura informada. Ela utiliza uma heurística de estimativa de distância até ao estado objectivo, impondo uma preferência a estados que sugerem um caminho mais curto até ao dito objetivo.

A terceira procura (A\*), também informada, utiliza uma heurística semelhante à anterior só que tem também em conta o custo das acções passadas, a optimalidade do algoritmo é apenas garantida se a heurística for **admissível**.

O desempenho temporal e de memória dos algoritmos é influenciado pelas dimensões do tabuleiro, pelo número e tipo de peças em campo e pela distância entre as mesmas peças (que por sua vez irá afectar as novas configurações que cada tabuleiro irá gerar).

A heurística considerada para as procuras informadas é uma ponderação entre 3 factores:

- x - O número de jogadas disponíveis a partir do tabuleiro actual.
- y- O número de peças “O” no tabuleiro.
- z- As configurações de tabuleiros que tivessem peças “O” com “X” como vizinhos.

**Análise de Resultados:****Tabuleiro 1:**

Este tabuleiro é um (5x5), sem peças bloqueadas (Não tem “X’s) com solução. Quando comparado com os outros 3 tabuleiros ele é o mais esparsa e como tal a árvore de procura vai ser muito menor em todos os tipos de procura. De notar que apesar da dimensão reduzida da árvore é já possível notar, algo que vem a ser confirmado pelos resultados dos outros tabuleiros, que a eficiência do Greedy > A\* > DFS.

**Tabuleiro 2:**

Este tabuleiro é um (4x4) com 1 peça bloqueada e com solução. É de notar que este tabuleiro é mais denso que o anterior apesar de possuir menor dimensões, no entanto como se observa nos resultados a densidade do tabuleiro contribuiu mais para o tempo da procura e o tamanho da árvore ( A DFS é 13.7 vezes mais lenta, o Greedy 2.5 vezes e o A\* 3.25 vezes que o tabuleiro 1).

**Tabuleiro 3:**

Este tabuleiro é um (4x5) com peças bloqueadas e com solução. Apesar de a densidade ser mais semelhante ao tabuleiro 2 do que com o tabuleiro 1 e de apenas ter mais uma coluna que o tabuleiro 2 observa-se um aumento de 131.7 vezes no tempo de execução da DFS, 93.318 no Greedy e 103.7 no A\* face ao Tabuleiro 2. Este fenómeno (do A\* ser pior que o Greedy ) é explorado na conclusão.

**Tabuleiro 4:**

Este tabuleiro é um (4x6) com peças bloqueadas e com solução. Ele partilha com o Tabuleiro 2 e 3 o facto de ter uma densidade elevada. Ele é também o com mais colunas. Com este tabuleiro consegue-se verificar empiricamente a importância de utilizar heurísticas e como estas tornam as procuras informadas mais eficientes. A DFS neste tabuleiro quando passados 5 minutos de tempo de execução ainda não tinha terminado e como tal não foi esperada a sua conclusão para retirar resultados. Os tempos de execução do Greedy e do A\* quando comparados com o tabuleiro 3 foram respectivamente 10.56 e 8.02 vezes superiores.

#### DFS:

Tabuleiros	Tempo de Execução	Nós Expandidos	Nós Gerados
Tabuleiro 1	0.00566 segundos	37	64
Tabuleiro 2	0.07753 segundos	809	913
Tabuleiro 3	9.22894 segundos	90312	90574
Tabuleiro 4	Demorou > 5 minutos	21422	2581

#### Greedy Search:

Tabuleiros	Tempo de Execução	Nós Expandidos	Nós Gerados
Tabuleiro 1	0.00347 segundos	23	43
Tabuleiro 2	0.00868 segundos	68	163
Tabuleiro 3	0.81353 segundos	9787	10017
Tabuleiro 4	8.59864 segundos	21422	2581

#### A\* Search:

Tabuleiros	Tempo de Execução	Nós Expandidos	Nós Gerados
Tabuleiro 1	0.00362 segundos	12	22
Tabuleiro 2	0.01175 segundos	33	88
Tabuleiro 3	1.14127 segundos	4877	5067
Tabuleiro 4	9.15176 segundos	4080	7722

#### Conclusão:

Apesar de terem sido apenas considerados 4 tabuleiros para estas análises é possível concluir que a DFS não é um algoritmo indicado para problemas de grandes dimensões devido à rápida evolução exponencial do seu tempo de execução.

A **incompletude** do algoritmo Greedy não foi verificada nesta análise pois foi encontrada uma solução para todos os tabuleiros. Por outro lado é de destacar que em todos os tabuleiros a performance do Greedy foi superior à DFS e ao A\* Star. **Isto (Greedy ser superior ao A\*) deve-se ao facto do path cost ser irrelevante para o problema em questão, o que favorece o Greedy pois o A\* Star vai gastar mais tempo a analisar os nós nos níveis superiores da árvore (Tendo a sua procura uma componente mais em largura).**

A **completude** do A\* foi confirmada assim como a sua **optimalidade** pois a heurística programada é **admissível**. Verificou-se empiricamente que os valores devolvidos pela heurística (apesar de negativos e floats) eram menores que o número de peças “O” -1 no tabuleiro, logo ela nunca faz uma sobre-estima do custo.