

# Relatório de Inteligência Artificial

## Análise dos Algoritmos de Procura

De forma a obter uma melhor conclusão acerca da eficiência dos algoritmos para o problema apresentado, foram utilizados os testes disponibilizados pelos docentes sobre uma versão do código que exclui a propagação de restrições (uma vez que com a sua utilização a maior parte dos testes não usaria a procura).

As seguintes tabelas apresentam os resultados obtidos. Os testes cujo tempo de execução corresponde a um valor superior a dez segundos foram descartados.

	Greedy	BFS	DFS	A*
teste 1	0,781	0,67	0,765	0,717
teste 2	0,696	0,667	0,791	0,667
teste 3	0,906	-	0,775	-
teste 4	0,742	-	0,703	-
teste 5	1,238	-	0,991	-
teste 6	2,793	-	0,853	-
teste 7	2,9	-	0,79	-
teste 8	0,64	-	0,715	-
teste 9	-	-	0,688	-
teste 10	-	-	0,796	-
teste 11	-	-	0,745	-
teste 12	-	-	0,704	-
teste 13	-	-	0,837	-

Fig. 1 – Tabela com os tempos de execução dos testes por algoritmo.

	Greedy	BFS	DFS	A*
teste 1	24	254	14	254
teste 2	18	254	14	254
teste 3	752	-	84	-
teste 4	422	-	64	-
teste 5	1372	-	110	-
teste 6	2952	-	158	-
teste 7	2684	-	138	-
teste 8	56	-	38	-
teste 9	-	-	278	-
teste 10	-	-	368	-
teste 11	-	-	360	-
teste 12	-	-	332	-
teste 13	-	-	360	-

Fig. 2 – Tabela com os nós gerados por algoritmo.

	Greedy	BFS	DFS	A*
teste 1	12	127	7	127
teste 2	9	127	7	127
teste 3	376	-	42	-
teste 4	211	-	32	-
teste 5	686	-	55	-
teste 6	1476	-	79	-
teste 7	1342	-	79	-
teste 8	28	-	19	-
teste 9	-	-	139	-
teste 10	-	-	184	-
teste 11	-	-	180	-
teste 12	-	-	166	-
teste 13	-	-	180	-

Fig. 3 – Tabela com os nós expandidos por algoritmo.

Através das tabelas podemos perceber que a DFS apresenta melhores resultados nestes testes do que em qualquer outra procura. No entanto, não é possível para já concluir que este seja o melhor algoritmo uma vez que, sem a propagação das restrições, a procura A\* com a heurística escolhida perde eficiência e fica equivalente à BFS. Assim, foram criados 4 novos testes, todos de dimensão igual a 12 e já com propagação de restrições, como podemos ver em seguida.

### Tempos:

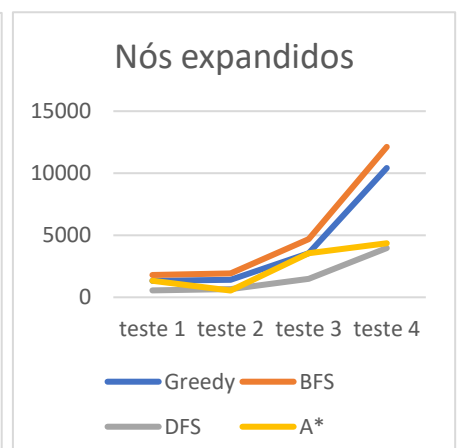
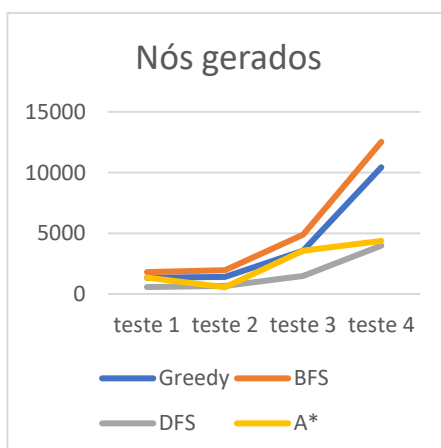
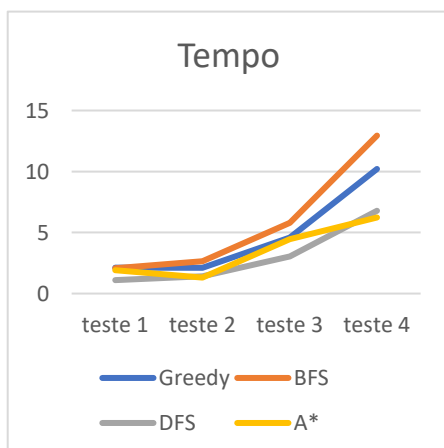
	Greedy	BFS	DFS	A*
teste 1	2,123	2,077	1,1	1,902
teste 2	2,111	2,639	1,424	1,308
teste 3	4,6	5,791	3,037	4,446
teste 4	10,211	12,946	6,785	6,234

### Nós Gerados:

	Greedy	BFS	DFS	A*
teste 1	1342	1802	568	1342
teste 2	1412	1964	656	556
teste 3	3568	4886	1490	3568
teste 4	10432	12526	3978	4368

### Nós Expandidos:

	Greedy	BFS	DFS	A*
teste 1	1332	1799	557	1332
teste 2	1402	1918	649	544
teste 3	3559	4681	1480	3559
teste 4	10422	12124	3967	4355



Tendo em conta que cada nível da árvore corresponde à atribuição de um valor a uma variável, a árvore de procura nunca terá profundidade infinita/ciclos, pelo que a DFS e a procura Greedy serão completas. BFS e procura A\* são também completas. Podemos então concluir que não teremos problemas de completude em qualquer uma das procuras.

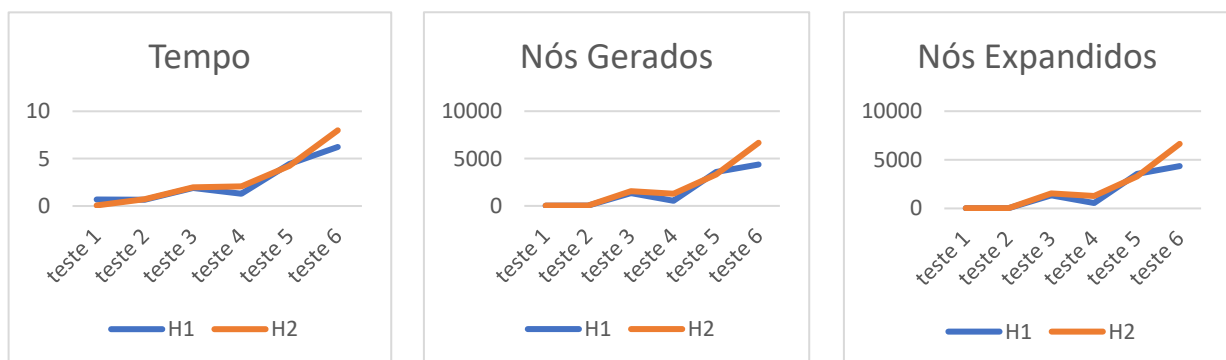
Podemos reparar pelos gráficos que a procura mais adequada seria a DFS, que, apesar de se aproximar muito da procura A\*, é a que aparenta ter melhores tempos de execução e que aparenta precisar de menos memória. Assim, a nossa solução final possui a DFS como algoritmo de procura, utilizando propagação de restrições de forma a obter um resultado eficiente para o tabuleiro de Takuzu.

### Heurística implementada:

Na escolha da heurística pensámos em duas opções diferentes:

- H1 - Retornar apenas o número de espaços que faltam preencher (os que se encontram com o número 2) – esta não será igual para todos os nós de um nível devido à propagação de restrições;
- H2 - Retornar o número de espaços que faltam preencher exceto os da linha e coluna da ação – equivalente à Heurística de maior grau.

Após alguns testes, obteve-se os seguintes gráficos (para procura A\*):



Apesar das duas retas aparentarem estar muito próximas, em alguns testes ainda se nota uma diferença bastante significativa (principalmente no número de nós gerados e expandidos). Concluimos assim que a melhor heurística a utilizar é a H1. Apesar de esta ter maior probabilidade de ser não ser admissível, aparenta ter um erro significativamente menor da realidade. Neste caso podemos admitir heurísticas não admissíveis, pois não está em questão encontrar a solução ótima (por só existir uma solução).