

[IA]

1º Relatório

Resultados

• Tabuleiros:

A → [[" ", "O", "O", "O", " "],
["O", " ", "O", " ", "O"],
[" ", "O", " ", "O", " "],
["O", " ", "O", " ", " "],
[" ", "O", " ", " ", " "]]

B → [["O", "O", "O", "X"],
["O", "O", "O", "O"],
["O", " ", "O", "O"],
["O", "O", "O", "O"]]

C → [["O", "O", "O", "X", "X"],
["O", "O", "O", "O", "O"],
["O", " ", "O", " ", "O"],
["O", "O", "O", "O", "O"]]

D → [["O", "O", "O", "X", "X", "X"],
["O", " ", "O", "O", "O", "O"],
["O", "O", "O", "O", "O", "O"],
["O", "O", "O", "O", "O", "O"]]

• Tabelas:

Depth First Search

Tabuleiros \ Parâmetros	Nós gerados	Nós expandidos	Tempo de execução (s)
A	21	13	0.003
B	6002	5984	0.759
C	80524	80497	12.121
D	91	26	0.011

Greedy Search

Tabuleiros \ Parâmetros	Nós gerados	Nós expandidos	Tempo de execução (s)
A	19	12	0.005
B	64	25	0.012
C	7589	7557	1.920
D	124	20	0.034

A Search*

Tabuleiros \ Parâmetros	Nós gerados	Nós expandidos	Tempo de execução (s)
A	24	12	0.009
B	1323	1043	0.560
C	6658	6553	2.082
D	138	24	0.041

Análise Crítica

A heurística do programa desenvolvido favorece os estados com menor número de peças e maior número de movimentos permitidos em direção ao centro do tabuleiro. Sabendo isto, apesar da procura A* ser completa (heurística é admissível), a procura Greedy tem melhor performance em termos de tempo de execução.

Apesar de a heurística ser partilhada por ambas as procuras, a Greedy tem uma execução mais simples, sendo ligeiramente mais rápida do que na procura A*. Esta ligeira diferença na velocidade de execução das duas procuras também se pode dever a propriedades inerentes à heurística desenvolvida.

Isto deve-se ao facto do problema dado ser um problema de satisfação apenas e não ser requisitado um caminho ótimo, por exemplo, o caminho com menor número de movimentos. Caso isso fosse verdade, a procura Greedy poderia não encontrar a solução ótima pois esta procura não é completa.

A procura em Profundidade Primeiro é mais rápida em problemas mais simples, tendo um tempo de execução semelhante às outras duas procuras. No entanto, devido a não ser uma procura informada, nos casos em que os problemas são mais complexos (como no tabuleiro C), o tempo de execução da procura é muito superior ao das outras duas.

Em termos de nós, considerando os problemas mais complexos (neste caso tabuleiros B e C), os algoritmos que fazem uma procura informada são preferíveis visto que o número de nós gerados e expandidos é dramaticamente menor nestas procuras.

Já entre as duas procuras informadas, verifica-se que o A* gera e expande menos nós no problema do tabuleiro C, no entanto, gera e expande mais nós no problema B. Assim, teremos de concluir que não há uma procura específica que gere e expanda menos nós, sendo que isto depende da complexidade e natureza do problema em questão.