Relatório IA P4 2021/2022

Neste projeto são utilizados os quatro algoritmos de procura pedidos. No desenvolvimento da heurística encontrámos duas soluções, acabando por escolher a mais eficiente.

Análise da Heurística

Inicialmente pensámos em utilizar como heurística o número de posições por preencher no tabuleiro. Num tabuleiro em que faltem preencher X posições, a heurística terá o valor de X nesse estado. O custo real do melhor caminho desde esse estado até ao estado objectivo será X. Logo, a condição para ser admissível - **h(n) ≤ h*(n)** - verifica-se. Para ser consistente teria de verificar **h(A) > c(A, B) + h(B)**, no entanto, visto que a heurística de A seria X e a do próximo estado (B) seria X-1, com custo de ambas as ações iguais a 1 a condição não se verifica. Assim, fomos procurar outra heurística.

Encontrámos uma melhor solução em que a heurística devolve o número de ações que é possível explorar em cada posição vazia, sendo que verificámos uma melhoria no tempo de execução.

Para cada posição vazia, analisamos o número de ações válidas que se pode executar. Os valores para cada posição podem ser: 1, se apenas uma ação retornar um estado válido ou 2, se ambas as ações forem válidas. Caso nenhuma ação for válida, tal significa que alguma das ações executadas anteriormente não era a certa, assim, o estado é inválido e o algoritmo deve voltar atrás, para que tal aconteça, retornamos um valor elevado, nomeadamente o triplo do tamanho do tabuleiro em questão. Para os casos válidos, a soma dos valores obtidos para cada posição será o valor retornado pela heurística para esse estado.

No caso desta heurística, num tabuleiro em que faltem preencher X posições, a heurística téra no mínimo o valor de X e no máximo o valor de 2X nesse estado. O custo real do melhor caminho desde esse estado até ao estado objectivo será no mínimo o valor de X e no máximo o valor de 2X. Logo, a condição para ser admissível **h(n) ≤ h*(n)** verifica-se.

Análise de Resultados

Na tabela de resultados é possível observar os resultados obtidos para cada tabuleiro de teste. Os resultados apresentam o tabuleiro a ser testado, a sua dimensão e para cada tipo de procura avaliada, o tempo de execução em segundos e o número de nós expandidos e gerados.

Podemos agora analisar os 4 diferentes tipos de procura avaliados. Todos os tipos de procura são completos, visto que todos os casos de teste chegam a uma solução válida e o tempo de execução é menor que os estipulados 3 segundos. Podemos então passar à análise em termos de eficiência.

A Procura em Largura Primeiro demonstrou ser um dos melhores algoritmos de procura para o problema em questão. Em alguns casos, verificou ter um tempo de execução ligeiramente menor que o da Procura em Profundidade Primeiro, mas na maioria dos casos encontra-se

Grupo tg005: Carolina Ramos 93694; Maria Beatriz Venceslau 93734

ligeiramente acima. Esta procura teve nos casos de teste disponibilizados um máximo de 0.109340 segundos. Este algoritmo vai explorando todas as ações em cada posição vazia do tabuleiro, sendo que deixa de explorar um ramo se encontrar uma lista de ações vazia, e termina o algoritmo quando encontrar o estado objetivo. Assim, apesar de explorar mais opções nunca tem de voltar atrás.

A Procura em Profundidade Primeiro demonstrou ser também um dos melhores algoritmos de procura para o problema em questão. Em alguns casos, verificou ter um tempo de execução ligeiramente maior que o da Procura em Largura Primeiro, mas na maioria dos casos consegue um tempo ligeiramente abaixo. Esta procura teve nos casos de teste disponibilizados um máximo de 0.093725 segundos. Este algoritmo vai explorar a primeira posição vazia do tabuleiro com a primeira ação que lhe for dada, e irá continuar a explorar esse ramo até encontrar o estado objetivo, ou até que lhe seja dada uma lista de ações vazia. Se lhe é dado uma lista vazia, quer dizer que houve um erro numa ação anterior e nesse caso o algoritmo irá para o próximo nó que estiver na lista de nós de fronteira.

A procura com o algoritmo de procura informada Greedy foi um dos que demonstrou piores resultados. Este algoritmo verificou ter tempos de execução bastantes superiores aos dois algoritmos de procura anteriores e também se encontra ligeiramente acima dos tempos de execução do algoritmo A*. Esta procura teve nos casos de teste disponibilizados um máximo de 1.140388 segundos.

Uma das propriedades da procura A* é ser ótimamante eficiente, no entanto esta procura depende da heurística utilizada. A heurística que encontrámos não é ótima, como se verifica pelo facto de os tempos de execução que este tipo de procura obteve estarem significativamente a cima dos dois primeiros tipos de procura analisados. Esta procura teve nos casos de teste disponibilizados um máximo de 1.140356 segundos. Assim, verificamos que apesar de ter obtido uns dos piores resultados dos quatro tipos de procura, com uma heurística adequada este tipo de procura seria o mais eficiente.

Conclusão

De todos os tipos de procura analisados, podemos concluir que os algoritmos que estão dependentes da heurística utilizada - Greedy e A*, visto não termos encontrado uma melhor solução para a heurística, apresentam tempos de execução muito elevados em comparação com os outros dois tipos de procura analisados - BFS e DFS. Assim, os algoritmos Greedy e A* com a heurística desenvolvida, podem ser descartados como os mais eficientes. De entre os outros dois tipos de procura desenvolvidos, apesar de terem obtido tempos de execução muito semelhantes nos casos de teste disponibilizados, o de Procura em Profundidade Primeiro foi o que obteve melhores tempos de execução, na maioria dos casos de teste, é portanto a melhor opção para a resolução do problema em questão, sendo esse o algoritmo utilizado no projeto submetido.

Tabela de Resultados:

Teste	Dimensão	Procura	Tempo Execução (s)	Nós Gerados	Nós Expandidos
1	4x4	BFS	0.0	7	7
		DFS	0.0	7	7
		Greedy	0.0	7	7
		A*	0.0	7	7
2	6x6	BFS	0.0	7	7
		DFS	0.0	7	7
		Greedy	0.0	7	7
		A*	0.0	7	7
3	8x8	BFS	0.015620	45	45
		DFS	0.0	43	42
		Greedy	0.025865	43	42
		A*	0.031235	43	42
4	9x9	BFS	0.0	32	32
		DFS	0.015618	32	32
		Greedy	0.015621	32	32
		A*	0.0	32	32
5	10x10	BFS	0.0	64	64
		DFS	0.0	62	61
		Greedy	0.031241	57	55
		A*	0.046861	57	55
6	12x12	BFS	0.031241	137	137
		DFS	0.031241	118	117
		Greedy	0.124968	85	83
		A*	0.140593	85	83
7	14x14	BFS	0.031244	69	69
		DFS	0.015621	69	69
		Greedy	0.078103	69	69
		A*	0.078103	69	69
8	15x15	BFS	0.0	19	19
		DFS	0.0	19	19
		Greedy	0.0	19	19
		A*	0.015617	19	19
9	18x18	BFS	0.062485	139	139
		DFS	0.062485	139	139
		Greedy	0.406150	139	139
		A*	0.406183	139	139
10	20x20	BFS	0.109340	184	184
		DFS	0.093725	184	184
		Greedy	0.843515	184	184
		A*	0.906005	184	184
11	21x21	BFS	0.062487	180	180
		DFS	0.062487	180	180

Grupo tg005: Carolina Ramos 93694; Maria Beatriz Venceslau 93734

11	21x21	Greedy	0.781066	180	180
		A*	0.823440	180	180
12	25x25	BFS	0.031242	166	166
		DFS	0.031243	166	166
		Greedy	0.734204	166	166
		A*	0.765441	166	166
13	31x31	BFS	0.046828	180	180
		DFS	0.046864	180	180
		Greedy	1.140388	180	180
		A*	1.140356	180	180