

Inteligência Artificial

1º Projeto

Por: Carlos Carvalho (81395) e Mariana Loureiro(83520)

Introdução

Este projecto teve como objectivo desenvolver um programa em Python que resolvesse diferentes puzzles de uma variante do jogo Same Game, recorrendo a três algoritmos de procura: procura em profundidade primeiro, a procura gananciosa e o A*. Neste relatório iremos usar os valores experimentais tempo de execução, nós gerados e nós expandidos para poder-mos comparar os três algoritmos de procura referidos em cima.

Avaliação Experimental dos Resultados

1- Tabuleiro 4:5 com 2 cores sem solução

	Profundidade Primeiro	Gananciosa	A*
Tempo de Execução (s)	0,000178	0,000231	0,000205
Nós Expandidos	1	1	1
Nós Gerados	0	0	0

2- Tabuleiro 4:5 com 3 cores

	Profundidade Primeiro	Gananciosa	A*
Tempo de Execução (s)	0,003038	0,002384	0,001994
Nós Expandidos	4	3	3
Nós Gerados	7	6	6

3- Tabuleiro 10:4 com 3 cores sem solução

	Profundidade Primeiro	Gananciosa	A*
Tempo de Execução (s)	62,406430	6,814347	6,827963
Nós Expandidos	74702	5573	5573
Nós Gerados	74701	10638	10638

4- Tabuleiro 10:4 com 3 cores

	Profundidade Primeiro	Gananciosa	A*
Tempo de Execução (s)	0,064134	0,017706	0,018163
Nós Expandidos	54	7	7
Nós Gerados	85	25	24

5- Tabuleiro 10:4 com 5 cores

	Profundidade Primeiro	Gananciosa	A*
Tempo de Execução (s)	2540,237234	0,127217	0,131227
Nós Expandidos	3123308	59	71
Nós Gerados	3123363	170	167

Análise Experimental

Na primeira avaliação experimental os três algoritmos expandiram um nó e finalizaram logo pois o problema em si não é solúvel.

Na segunda avaliação experimental podemos concluir que o A* foi o mais rápido a chegar à solução com 3 nós expandidos e 6 gerados. A procura em profundidade primeiro demorou mais tempo com a expansão de 4 nós e 7 gerados.

Na terceira avaliação experimental o A* e a procura gananciosa demoraram o mesmo tempo e ambas expandiram e geraram o mesmo número de nós, 5573 e 10638 respetivamente. O tempo da gananciosa e da A* foi aproximadamente o mesmo. A procura em profundidade primeiro demorou 62,41 segundos, expandiu todos os nós da árvore de procura e gerou todos os nós menos 1.

Na quarta avaliação experimental a profundidade primeiro foi a que demorou mais tempo em comparação com as outras duas e foi a que gerou mais nós (54) e expandiu mais nós (85). A A* gerou menos um nó do que a gananciosa. Os tempos de execução foram quase idênticos, no entanto a gananciosa foi mais rápida a executar.

Na quinta avaliação experimental a A* gerou 167 nós enquanto que a gananciosa gerou 170. Em relação à expansão de nós a A* expandiu mais do que a gananciosa. A gananciosa foi um pouco mais rápida do que a A*. A procura em profundidade primeiro demorou 42 minutos aproximadamente. Expandiu 3123308 nós e gerou 3123363 nós.

Conclusões

Podemos concluir que para dimensões grandes de tabuleiro e com muitas cores a procura gananciosa e a procura A* (procuras informadas que utilizam heurísticas) são mais eficientes do que a procura em profundidade primeiro.

No caso em que os tabuleiros não tem solução a três procuras demoram mais tempo, pois estas são obrigadas a gerar e expandir mais nós para descobrirem que não existe solução, no entanto a gananciosa e a A* continuam a ser mais rápidas.

Finalmente, no caso em que o tabuleiro tem dimensões pequenas a A* é mais rápida do que a gananciosa. No entanto, quando as dimensões do tabuleiro aumentam (avaliações experimentais 3, 4 e 5) a Gananciosa passa a ser um pouco mais rápida do que a A* devido ao facto da função $g(n)$ da A* custar apenas uma unidade para chegar a outro nó o que não é muito relevante quando os números da função $h(n)$ se tornam muito grandes. No entanto a A* gera menos nós.

Em suma, quanto maior as dimensões do tabuleiro e quanto maior o número de cores, maior vão ser o número de nós gerados e expandidos nas três procuras, sendo a gananciosa e a A* mais eficientes. A heurística utilizada por estas duas funções de procura têm em conta os dois parâmetros referidos na primeira frase, o que ajuda numa procura ótima e eficaz.