

Inteligência Artificial

Relatório do Projeto - Grupo 49

Trabalho realizado pelos alunos:

Miguel Mano, 99286

Pedro Rodrigues, 99300

Descrição do Problema

O Takuzu é um jogo lógico que envolve a colocação de dois símbolos, 1s e 0s, em uma grade com o mesmo número de linhas e colunas. O objetivo é preencher a grade com 1s e 0s, onde há um número igual de 1s e 0s em cada linha e coluna e não mais que dois de qualquer número adjacente um ao outro. Além disso, não pode haver linhas ou colunas idênticas. Existem vários quadrados por preencher. Neste projeto, propus um algoritmo para o resolver.

Descrição da Solução

Este projeto pode ser visto como um problema CSP em vez de um algoritmo de procura, uma vez que não importa o caminho, mas apenas a solução.

Deste modo, a propagação das ações é feita escolhendo uma das variáveis (uma célula do tabuleiro) e expandindo com os dois valores possíveis, 0 e 1. A escolha desta célula é feita através da heurística de maior grau, ou seja, escolhemos a célula vazia que cumpre o requisito de participar em mais tripletos (3 células consecutivas) não completos. Apenas são contabilizados tripletos com uma das casas preenchidas porque a atribuição de outro número pode forçar a célula restante. Em caso de empate, vai ser escolhida a célula vazia cuja coluna e a linha estão mais preenchidas.

Ao gerar um tabuleiro, todas as células triviais são preenchidas, ou seja, são preenchidas células forçadas pela restrição dos tripletos ou pela restrição de cada linha

ou coluna ter o mesmo número de 0's e 1's. Quando um estado contém um tabuleiro impossível não propaga nenhuma ação.

A DFS e a Greedy Search, que tendo em conta o modo de propagação das ações funciona como uma procura em profundidade, distinguem-se pelo fato de que nas bifurcações das DFS serem sempre testados os valores 0, enquanto nas da Greedy Search é sempre escolhido o estado que tem um tabuleiro com menos quadrados por preencher. A heurística do algoritmo de procura é o número de células por preencher, de modo a chegar à solução mais depressa ou a falhar mais depressa.

A Greedy Search comporta-se como uma procura em profundidade porque, segundo a heurística escolhida, se um estado s' é gerado por s , então o valor da heurística de s' é sempre menor do que o valor da heurística de s .

Reforçando a ideia, como este problema pode ser visto como um problema CSP, apenas interessa a solução final, sendo indiferente os estados passados para lá chegar. Isto torna a procura A^* escusada porque o fator do peso do caminho na heurística não interessa.

Análise de Resultados

Para a obtenção de uma análise mais profunda, utilizamos 5 testes adicionais, em anexo, juntamente com os testes já fornecidos.

Quanto à completude, podemos afirmar que todos os algoritmos de procura são completos porque todas as pesquisas efetuadas encontram uma solução, no caso de ela existir. A DFS e a Greedy Search dividem o pódio dos tempos para solucionar os problemas, existindo ocasiões em que um é mais eficaz que o outro, dependendo do estado que é escolhido pela heurística de procura.

Anexos

Algorithm	T00	T01	T02	T03	T04	T05	T06	T07	T08	T09	T10	T11	T12	T13	T14	T15	T16	T17	T18	Total
breadth_first_tree_search	0.000	0.001	0.000	0.001	0.002	0.002	0.003	0.000	0.000	0.001	0.000	0.000	0.001	0.000	0.196	0.949	0.174	1.387	1.967	4.683
depth_first_tree_search	0.000	0.000	0.000	0.001	0.001	0.001	0.002	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.059	0.779	0.139	0.500	1.310	2.792
greedy_search	0.001	0.000	0.000	0.000	0.001	0.002	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.181	0.616	0.059	0.973	1.630	3.465
astar_search	0.000	0.000	0.000	0.001	0.001	0.001	0.002	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.184	0.640	0.059	0.978	1.627	3.493
Algorithm	T00	T01	T02	T03	T04	T05	T06	T07	T08	T09	T10	T11	T12	T13	T14	T15	T16	T17	T18	
breadth_first_tree_search	<0/0>	<0/0>	<0/0>	<2/2>	<2/1>	<2/1>	<2/2>	<0/0>	<0/0>	<0/0>	<0/0>	<0/0>	<0/0>	<0/0>	<210/209>	<1106/1046>	<186/186>	<1376/1376>	<1988/1960>	
depth_first_tree_search	<0/0>	<0/0>	<0/0>	<2/1>	<2/2>	<2/2>	<2/1>	<0/0>	<0/0>	<0/0>	<0/0>	<0/0>	<0/0>	<0/0>	<68/60>	<900/891>	<150/137>	<492/477>	<1374/1363>	
greedy_search	<0/0>	<0/0>	<0/0>	<2/2>	<2/1>	<2/1>	<2/1>	<0/0>	<0/0>	<0/0>	<0/0>	<0/0>	<0/0>	<0/0>	<208/205>	<688/684>	<70/68>	<918/911>	<1590/1586>	
astar_search	<0/0>	<0/0>	<0/0>	<2/2>	<2/1>	<2/1>	<2/1>	<0/0>	<0/0>	<0/0>	<0/0>	<0/0>	<0/0>	<0/0>	<208/205>	<688/685>	<70/68>	<918/911>	<1590/1586>	
Caption: <generated/expanded>																				

Teste 14

14														
2	2	2	2	2	2	0	2	2	2	2	0	2	2	
2	2	0	2	0	2	2	0	2	2	1	0	2	0	
2	0	2	2	0	2	2	2	2	2	2	2	2	2	
1	2	2	2	2	2	1	0	2	2	2	2	2	2	
2	2	0	2	2	2	0	2	2	2	1	1	2	2	
2	1	2	2	2	2	2	1	2	2	2	2	2	1	
2	2	2	2	2	1	2	2	0	0	2	2	1	2	
2	2	2	1	2	2	2	2	2	2	2	0	2	2	
1	2	2	1	2	2	0	2	2	1	2	0	2	1	
2	1	2	2	2	0	2	2	2	2	2	0	2	1	
2	2	0	2	1	2	1	2	2	1	2	2	2	2	
2	2	2	2	2	2	2	0	2	2	2	2	1	2	
1	0	2	0	2	0	2	2	2	1	2	2	2	1	
1	2	2	0	2	2	2	2	2	0	2	2	0	2	

Teste 15

16	2	2	1	2	2	2	2	2	1	2	1	2	2	2	2	0
2	0	2	2	0	0	2	2	2	2	2	2	2	2	2	2	0
2	2	2	2	2	2	0	2	2	1	1	2	2	1	2	2	
0	1	2	1	2	1	2	2	0	2	2	2	2	1	2	1	
2	2	2	2	2	2	2	2	2	2	2	2	1	2	2	2	
2	2	2	2	0	2	2	2	2	0	2	2	2	1	2	2	
2	2	0	2	2	2	0	2	2	2	2	1	2	2	2	2	
0	2	2	0	2	1	2	2	0	2	2	1	2	2	0	2	
2	2	2	2	2	2	0	2	2	2	0	2	2	2	2	2	
2	2	2	1	2	1	2	2	2	2	2	2	2	1	2	1	
2	2	2	2	2	2	2	2	2	2	2	2	0	2	2	2	
2	2	0	0	2	2	2	2	2	2	2	2	2	2	2	1	
1	2	2	2	2	2	2	2	2	2	2	0	2	2	2	1	
1	2	0	2	2	1	2	2	1	2	2	2	1	2	2	2	
2	2	2	2	2	2	0	2	2	2	2	0	2	1	1	2	
2	1	2	1	2	1	2	1	2	2	2	2	2	2	2	2	

Teste 16

16	2	2	2	1	2	2	2	2	1	1	2	0	2	2	2	1
0	2	2	1	1	2	0	2	2	0	2	0	2	0	1	2	
2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	2	
0	2	2	0	2	2	0	2	0	2	1	1	2	2	2	2	
2	2	2	2	2	1	2	2	2	2	2	2	2	2	2	0	
2	2	1	2	0	0	2	2	0	0	2	2	0	1	2	2	
0	2	2	2	2	2	2	2	2	2	2	2	2	1	2	0	
2	2	2	2	2	2	0	2	2	0	2	2	2	2	2	2	
2	2	2	2	2	1	2	2	1	2	2	2	2	2	2	1	
2	2	2	2	0	2	0	2	2	2	2	1	2	0	2	2	
2	2	1	2	2	2	2	2	2	2	2	2	0	2	2	0	
2	0	2	2	1	1	2	2	2	2	2	2	2	2	2	2	
2	2	2	2	2	1	2	2	2	2	2	2	2	1	2	2	
0	2	2	2	2	2	2	2	1	2	2	1	2	2	0	0	
2	2	2	2	2	1	0	2	2	2	2	2	2	2	2	0	
2	0	0	2	2	2	0	2	2	2	1	1	2	2	2	2	

Teste 17

18	1	2	2	1	1	2	2	2	2	2	2	2	1	2	2	2	2	2	2
2	2	1	2	2	2	2	2	2	2	2	2	0	2	2	2	2	0	0	
2	2	2	0	2	2	2	2	0	2	2	2	2	2	2	2	0	2	2	
2	2	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	2	
2	1	2	2	0	2	2	1	0	2	2	2	2	2	2	2	2	2	2	
2	2	2	0	2	2	2	2	1	2	1	2	1	2	1	2	2	2	2	
2	2	1	2	2	0	2	0	2	0	2	2	0	0	2	2	1	2	2	
2	2	1	2	2	2	2	2	2	2	2	2	2	2	2	0	2	2	2	
2	2	2	2	2	2	1	1	2	1	1	2	2	2	2	1	2	1	2	
2	1	2	2	2	2	2	2	0	2	2	0	2	2	2	2	0	2	2	
2	1	2	2	2	2	1	2	2	2	2	2	2	2	2	2	2	2	2	
0	2	2	2	2	2	2	2	2	2	1	2	2	2	2	2	2	1	1	
2	2	2	0	0	2	2	1	2	2	2	0	2	2	2	2	2	2	0	
2	2	2	2	2	2	2	2	2	2	2	1	2	2	2	2	2	2	2	
1	2	1	2	2	2	2	2	2	2	2	2	1	2	2	2	1	2	2	
2	0	2	2	2	2	2	2	1	2	2	1	2	2	2	2	2	2	0	
2	2	2	2	2	2	0	0	2	2	2	2	0	2	2	1	2	1	2	
2	0	0	2	2	2	2	2	2	2	2	2	2	2	0	2	0	2	1	

Teste 18

20	0	2	2	1	2	2	2	2	2	2	2	2	2	2	0	2	2	2	1
2	2	2	2	2	2	1	2	0	2	0	2	2	2	2	2	2	1	2	1
2	0	2	0	2	0	1	2	0	2	2	2	1	1	2	2	2	1	2	2
2	2	2	2	2	2	2	2	2	2	1	2	2	1	2	2	2	2	2	
2	2	2	2	1	2	2	2	2	2	2	2	2	2	2	0	2	2	0	
2	2	0	2	1	2	0	0	2	0	0	2	2	2	2	2	2	2	2	
2	2	0	2	2	2	2	2	2	2	2	1	2	2	2	1	1	2	0	
2	2	2	2	2	2	2	2	2	0	2	2	1	1	2	2	2	2	2	
2	0	0	2	0	2	2	2	0	2	2	1	2	2	2	1	2	2	2	
2	2	2	1	2	1	2	2	2	2	2	2	2	2	2	2	2	0	2	
2	2	2	2	2	1	2	2	2	0	2	2	2	2	2	2	0	2	1	
2	1	2	2	2	2	2	2	0	2	2	1	2	2	0	2	2	2	2	
1	1	2	2	2	2	2	2	0	2	2	2	2	2	0	2	2	1	2	
2	2	0	2	2	0	2	2	2	2	2	2	2	2	2	2	2	2	0	
2	2	2	2	2	0	2	0	2	1	2	2	0	2	2	2	2	1	2	
1	2	2	2	2	2	2	2	2	2	2	1	2	2	0	2	0	2	2	
2	0	2	1	2	2	0	0	2	2	0	2	2	2	2	2	0	2	2	
2	2	2	2	1	2	2	2	2	2	2	2	2	2	2	2	1	2	2	
2	1	2	2	2	2	2	2	2	2	2	2	2	2	2	0	2	2	1	
2	2	2	2	2	2	0	1	2	1	0	2	0	2	2	2	1	2	2	