

Lista de Exercícios 1

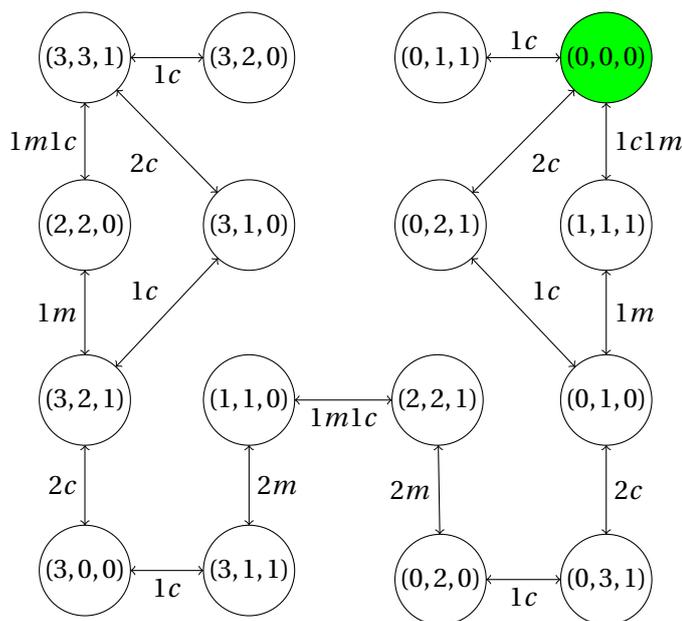
Resolução de problemas por meio de Busca

1 FORMULAÇÃO DE PROBLEMAS DE BUSCA

Forneça uma formulação completa do problema para cada um os seguintes itens. Escolha a formulação suficientemente precisa para ser implementada:

- Três missionários e três canibais estão em um lado de um rio, juntamente com um barco que pode conter uma ou duas pessoas. Descubra um meio de fazer todos atravessarem o rio, sem deixar que um grupo de missionários de um lado fique em número menor que o número de canibais.

Solução: Vamos representar cada estado como sendo uma tripla, onde o primeiro elemento refere ao número de missionários no lado inicial do rio, o segundo ao número de canibais também no lado inicial do rio e o terceiro ao lado do rio onde se encontra o barco (1 para o lado inicial e 0 ao outro lado). As ações possíveis são atravessar o rio com um missionário (1m), atravessar com um canibal (1c), atravessar com um missionário e um canibal (1m1c), atravessar com 2 canibais (2c) e atravessar com dois missionários (2m). Cada ação possui custo de passo unitário. O estado inicial é (3,3,1) e o estado meta é (0,0,0). O diagrama abaixo especifica a função de transição:



b. Sokoban é um jogo de movimentação de caixas ou engradados em um armazém. O objetivo é deslocar cada caixa de uma posição inicial até a área de estocagem, respeitando as seguintes regras:

- Apenas uma caixa pode ser movimentada por vez.
- Não pode haver sobreposição (pular outra caixa).
- O jogador não poderá ocupar o mesmo lugar da caixa ou andar nos muros.
- O jogo é finalizado quando todas as posições de armazenagem são ocupadas por caixas.

O puzzle foi criado em 1981 por Hiroyuki Imabayashi, e publicado em 1982 por Thinking Rabbit, uma empresa de software localizada em Takarazuka [Sokoban, 2018]. A figura abaixo contém um exemplo de configuração do jogo.



Solução: A formulação apresentada por Junghanns and Schaeffer [1997] assume tabuleiros de tamanho máximo 20-por-20 e visa minimizar o número de movimentações para levar cada caixa do sua posição inicial até uma possível posição desejada, ignorando movimentações do agente. Um estado é representado por uma matriz de tamanho 20-por-20 na qual o elemento na i -ésima linha e j -ésima coluna indica a presença de um objeto: C para caixa, A para agente, P para parede, M para área de armazenagem e espaço para ausência de objetos. A tabela abaixo representa o estado para a configuração na figura acima:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
a																				
b							P	P	P	P	P									
c							P				P									
d							P	C			P									
e				P	P	P			C	P	P									
f				P			C		C		P									
g		P	P	P		P		P	P		P				P	P	P	P	P	P
h		P				P		P	P		P	P	P	P	P			M	M	P
i		P		C			C											M	M	P
j		P	P	P	P	P		P	P	P		P	A	P	P			M	M	P
k						P						P	P	P	P	P	P	P	P	P
l						P	P	P	P	P	P	P								

Uma ação consiste em empurrar uma caixa de um célula para outra célula adjacente e livre. Por exemplo, se nomearmos as linhas na figura acima de a a t e as colunas de A a T,

com (a,A) denotando a célula superior esquerda, a ação Fh-Eh move a caixa em Fh uma célula à esquerda. Uma ação, é claro, só é aplicável se houver um caminho válido pelo qual o agente possa se deslocar para se colocar por “de trás” da caixa e empurrá-la para uma célula livre (por exemplo, para executar Fh-Eh o agente precisa se deslocar de Li para Gh através das células Lh, Kh, Jh, Ih e Hh). O custo de passo de cada ação é unitário. A função teste de meta verifica se existem elementos M na matriz de representação de estados. A função de transição verifica se uma ação é aplicável e modifica a matriz correspondente para gerar o estado resultante.

2 BUSCA

Um agente precisa percorrer o ambiente formado por células (salas) interligadas mostrado abaixo, partindo da célula número 6, e chegando ou a célula número 18 ou à célula número 14. A cada etapa, o agente pode acessar as células numeradas acima, abaixo, à esquerda e à direita da célula atual. Mover-se para a célula 18 custa 4 unidades. Mover-se para qualquer outra célula custa 2 unidades.

	14	15	16	17	18	
				13		
	9	10		11	12	
	6	7		8		
	1	2	3	4	5	

- a. Desenhe a árvore de busca de uma busca em grafo em profundidade. Expanda nós de mesma prioridade em ordem crescente do número da célula correspondente. Qual o custo da solução encontrada? Qual a profundidade e o fator de ramificação médio da busca?

Solução:

Profundidade da solução: 9

Custo da solução: 20

Fator de ramificação médio: $\frac{25}{26} = 0.96$

A Figura 2.1 mostra a árvore de busca para o algoritmo busca em profundidade em grafo.

- b. Formule uma heurística admissível para o problema e desenhe a árvore de busca do algoritmo A^* usando essa heurística (expanda os nós de mesma prioridade em ordem crescente do número da célula correspondente).

Solução:

Heurística: Distância de Manhattan ($|x_1 - x_2| + |y_1 - y_2|$)

A Figura 2.2 mostra a árvore de busca para o algoritmo A^* .

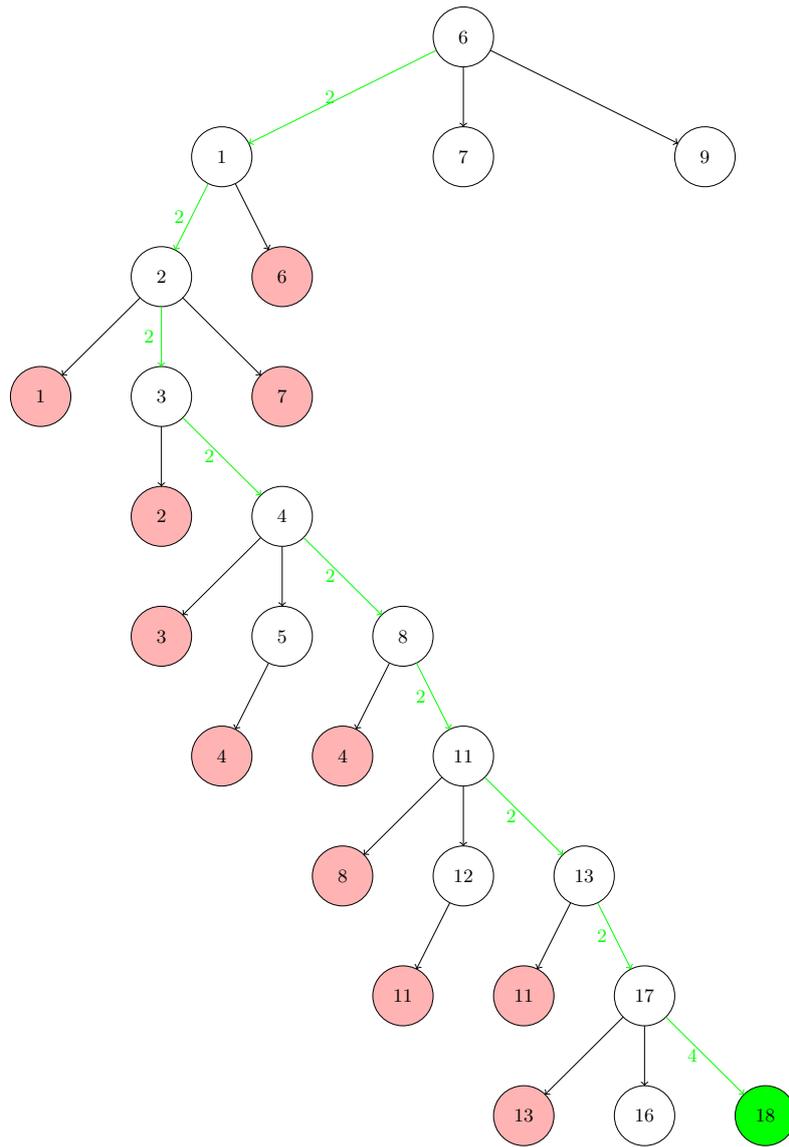


Figure 2.1: Busca em profundidade para o exercício 2

REFERENCES

A. Junghanns and J. Schaeffer. Sokoban: A challenging single-agent search problem. In *In IJCAI Workshop on Using Games as an Experimental Testbed for AI Research*, 1997.

Sokoban. Sokoban — Wikipedia, the free encyclopedia, 2018. URL <https://pt.wikipedia.org/wiki/Sokoban>. [Online; accessed 19-March-2019].

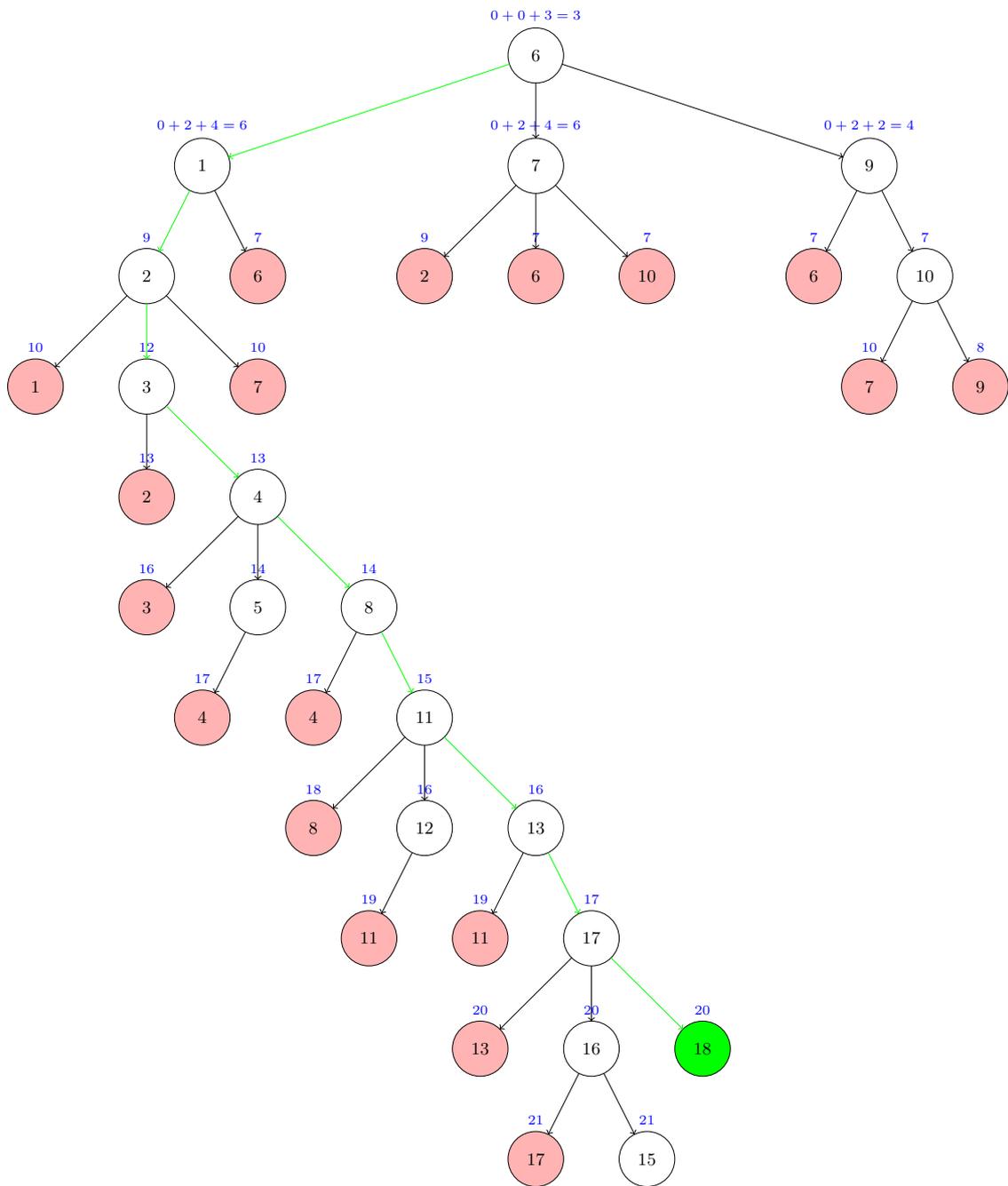


Figure 2.2: Busca A* para o exercício 2