

CEFET/RJ
Disciplina: Inteligência Artificial
Professor: Eduardo Bezerra
Lista de exercícios 01

Créditos: alguns itens desta lista são adaptados do material da disciplina *CS188 - Artificial Intelligence*¹, da University of Berkeley. Outros são adaptações de exercícios propostos no livro-texto da disciplina, *AIMA*².

1. Suponha que AÇÕES-VÁLIDAS(s) denote o conjunto de ações válidas no estado s , e que RESULTADO(a, s) denote o estado que resulta da execução de uma ação válida a no estado s . Considere a função SUCESSOR(s) que produz o conjunto de pares ordenados (a, s') em que cada par define o estado s' resultante de o agente tomar a ação a no estado s . Defina a função SUCESSOR em termos das funções AÇÕES-VÁLIDAS e RESULTADO, e vice-versa.

Solução

$$\begin{aligned} \text{SUCESSOR}(s) &= \{(a, s') \mid a \in \text{AÇÕES-VÁLIDAS}(s) \wedge s' = \text{RESULTADO}(a, s)\} \\ \text{AÇÕES-VÁLIDAS}(s) &= \{a \mid (a, s') \in \text{SUCESSOR}(s)\} \\ \text{RESULTADO}(a, s) &= \{s' \mid (a, s') \in \text{SUCESSOR}(s)\} \end{aligned}$$

2. Um espaço de estados finito conduz a uma árvore de busca finita? E no caso de um espaço de estados finito que é uma árvore? Você poderia ser mais preciso em definir que tipos de espaços de estados sempre levam a árvores de busca finitas?

Solução

Não. Um espaço de estados finito nem sempre leva a uma árvore de busca finita. Considere um espaço de estados com dois estados, cada um deles com uma ação que leva ao outro. Isso gera uma árvore de busca infinita, porque podemos ir e voltar infinitas vezes. Porém, se o espaço de estados for uma árvore finita, ou, em geral, um DAG (grafo acíclico direcionado), não haverá loops e a árvore de busca será finita.

3. Forneça o estado inicial, o teste de objetivo, a função sucessor e a função de custo para cada um dos itens a seguir:

¹<http://ai.berkeley.edu/home.html>

²<http://aima.cs.berkeley.edu/>

- (a) Você tem de colorir um mapa plano usando apenas quatro cores, de tal modo que não haja duas regiões adjacentes com a mesma cor.

Solução

- Estado inicial: mapa com nenhuma região colorida.
- Teste de objetivo: verifica se uma mapa tem todas as regiões coloridas, e que não há nenhuma região adjacente com a mesma cor.
- Função sucessor: atribuir uma cor a uma região do mapa que esteja sem cor.
- Função de custo: número total de atribuições (essa função de custo assim definida é igual para todas as soluções deste problema).

- (b) Um macaco com um metro de altura está em uma sala em que algumas bananas estão presas no teto, a 2,5 metros de altura. Ele gostaria de alcançar as bananas. A sala contém dois engradados empilháveis, móveis e escaláveis, com um metro de altura cada.

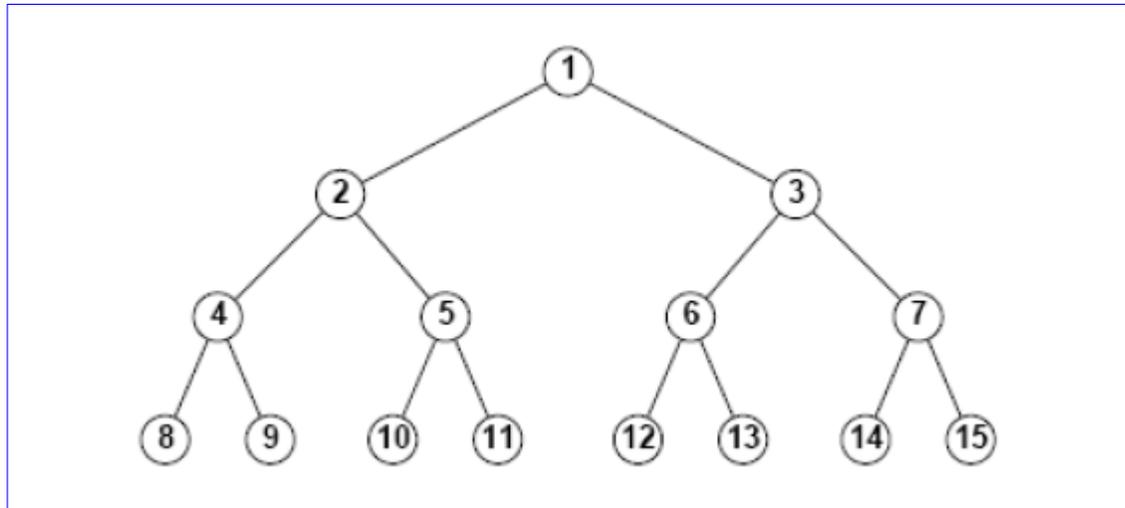
Solução

- Estado inicial: como descrito no enunciado.
- Teste de objetivo: macaco alcançou as bananas.
- Função sucessor: subir no engradado; descer do engradado; mudar engradado de lugar; andar de um lugar a outro; agarrar bananas
- Função de custo: número total de ações.

4. Considere um espaço de estados onde o estado inicial é o número 1 e a função sucessor para o estado n retorna dois estados, com os números $2n$ e $2n+1$.

- (a) Desenhe a porção do espaço de estados correspondente aos estados 1 a 15.

Solução



- (b) Suponha que o estado objetivo seja 11. Liste a ordem em que os nós serão visitados no caso da busca em extensão, da busca em profundidade limitada com limite 3 e da busca por aprofundamento iterativo.

Solução

- Busca em extensão: 1 2 3 4 5 6 7 8 9 10 11
- Busca em profundidade limitada: 1 2 4 8 9 5 10 11
- Aprofundamento iterativo: 1; 1 2 3; 1 2 4 5 3 6 7; 1 2 4 8 9 5 10 11

5. Problema de missionários e canibais: 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.
- (a) Formule o problema precisamente. Trace um diagrama do espaço de estados completo.
- (b) Resolva o problema de forma ótima, utilizando um algoritmo de busca apropriado. É boa ideia verificar a existência de estados repetidos?

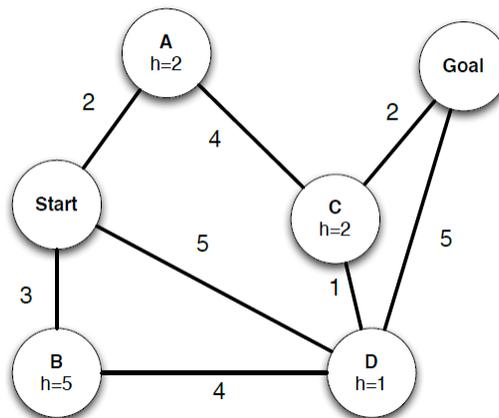
Solução

- Uma representação possível: Um estado é um vetor com três inteiros listando o número de missionários, canibais e barcos na margem inicial do rio. Sendo assim, o estado inicial é $(3,3,1)$ e o estado objetivo é $(0,0,0)$. O teste de objetivo verifica se o estado objetivo $(0,0,0)$ foi alcançado. A função de custo tem valor um para cada ação. Os sucessores de um estado são todos os estados que

movem uma ou duas pessoas e um barco de um lado para o outro, sem que o número de missionários de um lado fique menor do que o número de canibais.

- Qualquer algoritmo de busca funciona bem, porque o espaço de estados é muito pequeno. Basta eliminar estados repetidos e estados inválidos (com maior número de canibais do que missionários do mesmo lado).

6. Considere o grafo apresentado na figura abaixo. Considere também que empates (com relação à escolha do vértice selecionado para expandir) são resolvidos usando a ordem lexicográfica aplicada aos rótulos dos vértices. Para cada uma das estratégias de busca a seguir, forneça (1) a ordem dos vértices expandidos e (2) o caminho retornado pela busca em grafo. Lembre-se de que na busca em grafos, um estado é expandido apenas um vez.



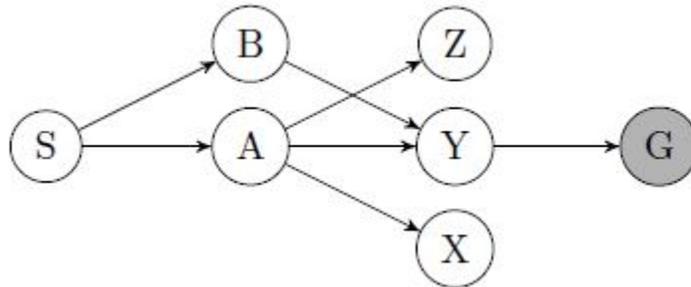
- Busca em Profundidade
- Busca em Largura
- Busca com Custo Uniforme
- Busca A^* com heurística apresentada na figura

Solução

- DFS
Estados expandidos: Start, A, C, D, B, Goal
Caminho retornado: Start-A-C-D-Goal
- BFS
Estados expandidos: Start, A, B, D, C, Goal
Caminho retornado: Start-D-Goal

- UCS
Estados expandidos: Start, A, B, D, C, Goal
Caminho retornado: Start-A-C-Goal
- Busca A*
Estados expandidos: Start, D, Goal
Caminho retornado: Start-D-Goal

7. Forneça as sequências de estados (NÃO a fronteira) consideradas pelos algoritmos DFS e BFS no seguinte grafo dirigido, a partir do estado S , e presumindo ordem alfabética de estados (isto é, quando há uma escolha arbitrária de qual estado expandir, escolha o que ocorre primeiro no alfabeto) e que a busca pára quando o algoritmo atinge o alvo, que é o estado G .



8. Por definição, dois algoritmos de busca são *equivalentes* se e somente se eles expandem os mesmos nós na mesma ordem e retornam o mesmo caminho. Nos exercícios abaixo, estudamos o que acontece se executamos uma busca de custo uniforme (UCS) com custos de ação d_{ij} que são potencialmente diferentes dos custos de ação reais do problema de busca c_{ij} . Concretamente, estudaremos como isso pode, ou não, resultar na execução do algoritmo *Uniform Cost Search* (com essas novas opções de custos de ação) equivalente a outro algoritmo de busca.

- (a) Marque todas as opções para os custos d_{ij} que fazem com que a execução do algoritmo **Uniform Cost Search** com estes custos d_{ij} seja *equivalente* à execução do algoritmo **Breadth-First Search**.

- $d_{ij} = 0$
- $d_{ij} = \alpha, \alpha > 0$
- $d_{ij} = \alpha, \alpha < 0$
- $d_{ij} = 1$
- $d_{ij} = -1$
- Nenhuma das alternativas acima

Solução

Alternativas 2 e 4. O BFS expande o nó na profundidade mais rasa primeiro. Atribuir uma constante positiva a todas as arestas permite ponderar os nós pelas suas respectivas profundidades na árvore de busca.

- (b) Marque todas as opções para os custos d_{ij} que fazem com que a execução do algoritmo **Uniform Cost Search** com estes custos d_{ij} seja *equivalente* à execução do algoritmo **Depth-First Search**.
- $d_{ij} = 0$
 - $d_{ij} = \alpha, \alpha > 0$
 - $d_{ij} = \alpha, \alpha < 0$
 - $d_{ij} = 1$
 - $d_{ij} = -1$
 - Nenhuma das alternativas acima

Solução

Alternativas 3 e 5. O DFS expande primeiramente o nó que foi mais recentemente adicionado à fronteira. Atribuir uma constante negativa a todas as arestas reduz o valor dos nós mais recentemente adicionados por essa constante, transformando-os nos nós com o mínimo valor na fronteira quando usamos o UCS.

- (c) Marque todas as opções para os custos d_{ij} que fazem com que a execução do algoritmo **Uniform Cost Search** com estes custos d_{ij} seja *equivalente* à execução do algoritmo **Uniform Cost Search** com os custos originais c_{ij} .
- $d_{ij} = c_{ij}^2$
 - $d_{ij} = 1/c_{ij}$
 - $d_{ij} = \alpha c_{ij}, \quad \alpha > 0$
 - $d_{ij} = c_{ij} + \alpha, \quad \alpha > 0$
 - $d_{ij} = \alpha c_{ij} + \beta, \quad \alpha > 0, \beta > 0$
 - Nenhuma das alternativas acima

Solução

Alternativa 3. O UCS expande primeiramente o nó com o menor custo até o momento = $\sum_{ij} c_{ij}$ na fronteira. Sendo assim, a ordem relativa entre dois nós é determinada pelo valor de $\sum_{ij} c_{ij}$ para um dado nó. Entre as alternativas acima, apenas para $d_{ij} = \alpha c_{ij}, \alpha > 0$, é que podemos concluir que, para dois nós n e m :

$$\sum_{ij \in \text{path}(n)} d_{ij} \geq \sum_{ij \in \text{path}(m)} d_{ij} \iff \sum_{ij \in \text{path}(n)} c_{ij} \geq \sum_{ij \in \text{path}(m)} c_{ij}$$

- (d) Seja $h(n)$ o valor da função heurística no nó n . Marque *todas* as opções para custos d_{ij} que tornam o algoritmo **Uniform Cost Search** com esses custos d_{ij} *equivalente* à execução do algoritmo **Greedy Search** com os custos originais c_{ij} e função heurística h .

- $d_{ij} = h(i) - h(j)$
 $d_{ij} = h(j) - h(i)$
 $d_{ij} = \alpha h(i), \quad \alpha > 0$
 $d_{ij} = \alpha h(j), \quad \alpha > 0$
 $d_{ij} = c_{ij} + h(j) + h(i)$
 Nenhuma das alternativas acima

Solução

Alternativa 2. A busca gulosa expande primeiramente o nó com o menor valor para a função heurística $h(n)$. Se $d_{ij} = h(j) - h(i)$, então o custo de um nó n na fronteira na execução do UCS será $\sum_{ij} d_{ij} = h(n) - h(start)$. Posto que $h(start)$ é uma constante subtraída do custo de todos os nós na fronteira, a ordem relativa dos nós na fronteira é ainda determinada por $h(n)$, i.e. o valor da função heurística.

- (e) Seja $h(n)$ o valor da função heurística no nó n . Marque *todas* as opções para custos d_{ij} que tornam o algoritmo **Uniform Cost Search** com esses custos d_{ij} *equivalente* à execução do algoritmo **A* Search** com os custos originais c_{ij} e função heurística h .

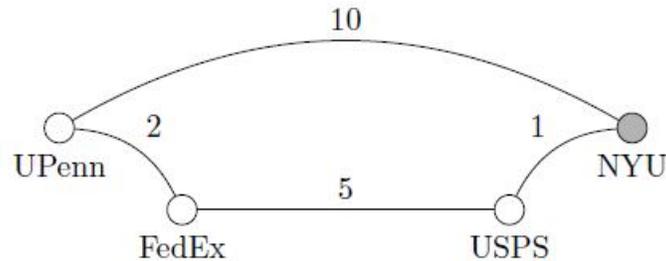
- $d_{ij} = \alpha h(i), \quad \alpha > 0$
 $d_{ij} = \alpha h(j), \quad \alpha > 0$
 $d_{ij} = c_{ij} + h(i)$
 $d_{ij} = c_{ij} + h(i) - h(j)$
 $d_{ij} = c_{ij} + h(j) - h(i)$
 Nenhuma das alternativas acima

Solução

Alternativa 5. A busca A^* expande primeiramente o nó com o menor valor para $f(n) + h(n)$, em que $f(n) = \sum_{ij} c_{ij}$ é o custo até o momento, e h o valor produzido pela heurística. Se $d_{ij} = c_{ij} + h(j) - h(i)$, então o custo de um nó n na fronteira durante a execução do UCS será $\sum_{ij} d_{ij} = \sum_{ij} c_{ij} + h(n) - h(start) = f(n) + h(n) - h(start)$. Posto que $h(start)$ é uma constante subtraída do custo de todos os nós na fronteira, a ordem relativa dos nós na fronteira é ainda determinada por $f(n) + h(n)$.

9. Considere as seguintes heurísticas, h_1 , h_2 e h_3 , que representam estimativas do custo do nó especificado até o nó objetivo, NYU.

	UPenn	FedEx	USPS	NYU
h_1	10	1	1	0
h_2	5	1	1	0
h_3	5	4	1	0



- (a) Para cada uma das heurísticas, indique se ela é admissível e/ou consistente.

Solução

h_2 admissível, h_3 admissível e consistente. h_1 não é admissível e portanto também não é consistente, porque $h_1(\text{UPenn})$ é maior do que o custo do caminho ótimo. h_2 subestima e é admissível, mas não é consistente, porque $h_2(\text{UPenn}) > h_2(\text{FedEx}) + c(\text{UPenn} \rightarrow \text{FedEx})$; i.e., $f(\text{UPenn}) = 5$, mas $f(\text{FedEx}) = 3$, portanto f não é não-decrescente. Para h_3 , f ao longo do caminho ótimo é $\{0 + 5 = 5; 2 + 4 = 6; 7 + 1 = 8; 8 + 0 = 8\}$, de modo que h_3 é consistente e portanto admissível.

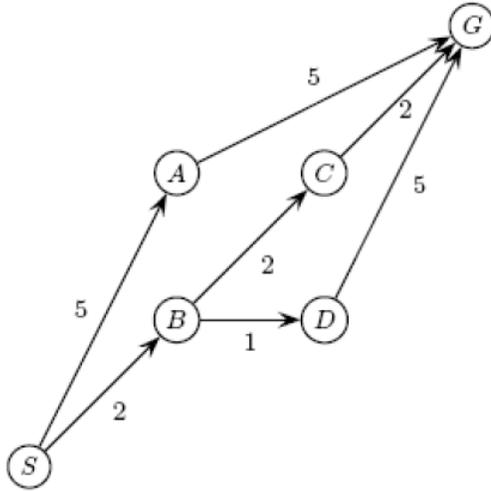
- (b) Quantas vezes A^* irá expandir o nó FedEx se h_2 é usada como heurística, supondo que A^* NÃO mantenha registro dos estados visitados?

Solução

Apenas uma vez.

Um erro comum nessa questão é o seguinte. $f(\text{UPenn} \rightarrow \text{FedEx}) = 2 + 1 = 3$, e $f(\text{UPenn} \rightarrow \text{FedEx} \rightarrow \text{UPenn} \rightarrow \text{FedEx}) = 2 + 1 + 2 + 2 = 7$, que são ambas menores do que 8 e seriam visitadas se h_2 fosse consistente. Contudo, h_2 não é consistente: $f(\text{UPenn} \rightarrow \text{FedEx} \rightarrow \text{UPenn}) = 2 + 2 + 5 = 9$, o que significa que o caminho ótimo será descoberto antes de UPenn ser expandido uma segunda vez. Portanto FedEx será visitada apenas uma vez.

10. (Busca A^*) Considere o espaço de busca abaixo, onde S é o estado inicial e G é o único estado que satisfaz o teste de objetivo. Os rótulos nas arestas indicam o custo de percorrê-las. A tabela ao lado apresenta três heurísticas: h_0 , h_1 e h_2 .



Node	h_0	h_1	h_2
<i>S</i>	0	5	6
<i>A</i>	0	3	5
<i>B</i>	0	4	2
<i>C</i>	0	2	5
<i>D</i>	0	5	3
<i>G</i>	0	0	0

- Quais são os nós expandidos pela busca A* usando cada uma das heurísticas (h_0 , h_1 e h_2)?
- Qual é a solução (caminho) encontrada por cada uma delas?
- Quais das heurísticas são admissíveis? Justifique sua resposta.

Solução

Ver solução fornecida para o exercício anterior.