

Cap. 2 - Russel & Norvig - Exercícios selecionados (respostas em azul)

1. Tanto a medida de desempenho quanto a função de utilidade medem o quanto um agente está desempenhando bem suas atividades. Explique a diferença entre as duas medidas.

Uma medida de desempenho é utilizada por um observador externo para avaliar o sucesso de um agente. Uma função de utilidade é utilizada por um agente para avaliar estados. A função de utilidade pode não ser igual a medida de desempenho; além disso, um agente pode não ter uma função de utilidade explícita, enquanto que sempre existe uma medida de desempenho.

2. Pode haver mais de um programa de agente que implemente uma dada função de agente? Dê um exemplo ou mostre porque não é possível.

Sim. Podemos criar um novo programa de agente por meio da modificação de um programa de agente existente, inserindo comandos inúteis que não alterem a saída do programa. Estes dois programas implementam a mesma função de agente.

3. Descreva uma função de agente racional para o mundo do aspirador de pó com a medida de desempenho modificada que deduz um ponto a cada movimento. O programa de agente correspondente exige estado interno?

O agente deve ter memória para guardar a informação de que um quadrado já foi limpo. Dessa forma ele só vai para o outro quadrado se ainda não verificou que o quadrado estava limpo. A memória que guarda essa informação é um estado interno.

4. Para cada um dos agentes a seguir, desenvolva uma descrição de PEAS do ambiente de tarefas.

- (a) Robô jogador de futebol.
- (b) Agente de compras na Internet.
- (c) Robô explorador de Marte.

<i>Agente</i>	<i>Medida de desempenho</i>	<i>Ambiente</i>	<i>Atuadores</i>	<i>Sensores</i>
<i>Futebol</i>	<i>Ganhar jogo, saldo de gols</i>	<i>Campo, bola, trave, próprio time, outro time, próprio corpo</i>	<i>Dispositivos para locomoção e chute</i>	<i>Câmera, sensores de toque, acelerômetros, sensores de orientação</i>
<i>Internet</i>	<i>Obter produtos requisitados, minimizar gastos</i>	<i>Internet</i>	<i>Seguir link, preencher dados em campos, mostrar para usuário</i>	<i>Páginas da Web, pedido dos usuários</i>
<i>Marte</i>	<i>Terrenos explorados, amostras recolhidas</i>	<i>Veículo lançador, Marte</i>	<i>Rodas, equipamento coletor de amostras, transmissor de rádio</i>	<i>Câmera, sensores de toque, acelerômetro, sensores de orientação, receptor de rádio</i>

5. Para cada um dos agentes a seguir, caracterize o ambiente de acordo com as propriedades dadas na Seção 2.3 e selecione um projeto de agente adequado.

- (a) Robô jogador de futebol.
- (b) Agente de compras na Internet.
- (c) Robô explorador de Marte.

<i>Ambiente</i>	<i>Observável</i>	<i>Determinístico</i>	<i>Episódico</i>	<i>Estático</i>	<i>Discreto</i>	<i>Agente</i>
<i>Futebol</i>	<i>Parcialmente</i>	<i>Estocástico</i>	<i>Sequencial</i>	<i>Dinâmico</i>	<i>Contínuo</i>	<i>Múltiplo</i>
<i>Internet</i>	<i>Parcialmente</i>	<i>Determinístico</i>	<i>Sequencial</i>	<i>Estático</i>	<i>Discreto</i>	<i>Único</i>
<i>Marte</i>	<i>Parcialmente</i>	<i>Estocástico</i>	<i>Sequencial</i>	<i>Dinâmico</i>	<i>Contínuo</i>	<i>Único</i>

Cap. 3 - Russel & Norvig - Exercícios selecionados

6. 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 . Defina SUCESSOR em termos de AÇÕES-VÁLIDAS e RESULTADO, e vice-versa.

$$SUCESSOR(s) = \{(a,s') \mid a \in \text{AÇÕES-VÁLIDAS}(s) \text{ E } s' = \text{RESULTADO}(a,s)\}$$

$$\text{AÇÕES-VÁLIDAS}(s) = \{a \mid (a,s') \in SUCESSOR(s)\}$$

$$\text{RESULTADO}(a,s) = \{s' \mid (a,s') \in SUCESSOR(s)\}$$

7. 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?

Não. Um espaço de estados finitos 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.

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

- (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.

Estado inicial: nenhuma região colorida.

Teste de objetivo: todas as regiões coloridas, e nenhuma região adjacente com a mesma cor.

Função sucessor: atribuir uma cor a uma região que esteja sem cor.

Função de custo: número total de atribuições (igual para todas as soluções).

- (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.

Estado inicial: como descrito no enunciado.

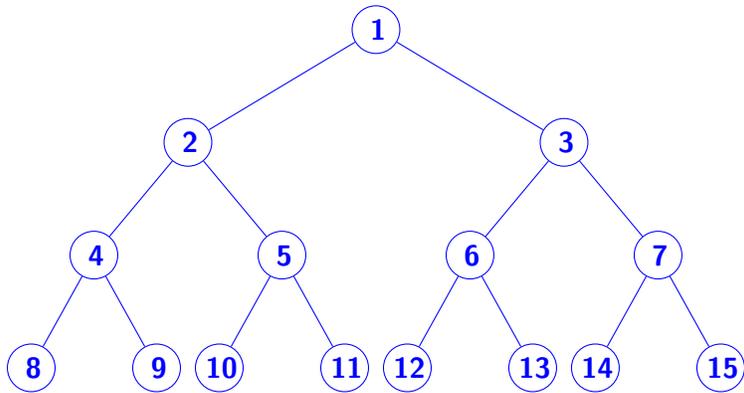
Teste de objetivo: macaco alcançou as bananas.

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.

9. 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.



(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.

Busca em extensão: 1 2 3 4 5 6 7 8 9 10 11

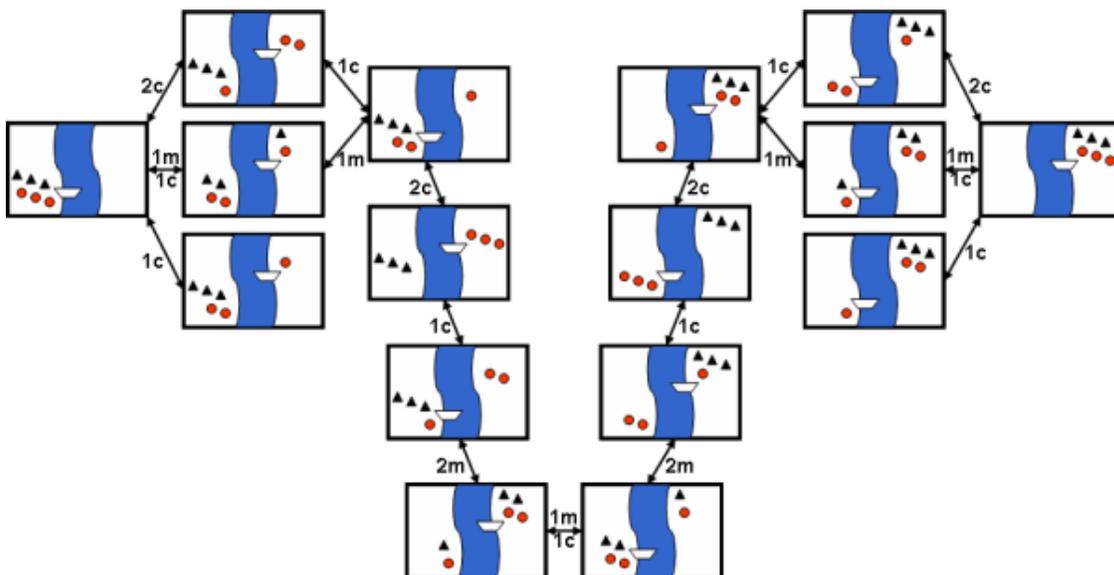
Busca em profundidade limitada: 1 2 4 8 9 5 10 11

Aprofundamento iterativo: 1; 1 2 3; 1 2 4 5 3 6 7; 1 2 4 8 9 5 10 11

10. 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.

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. Espaço de Estados (extraído de missionaries.html):

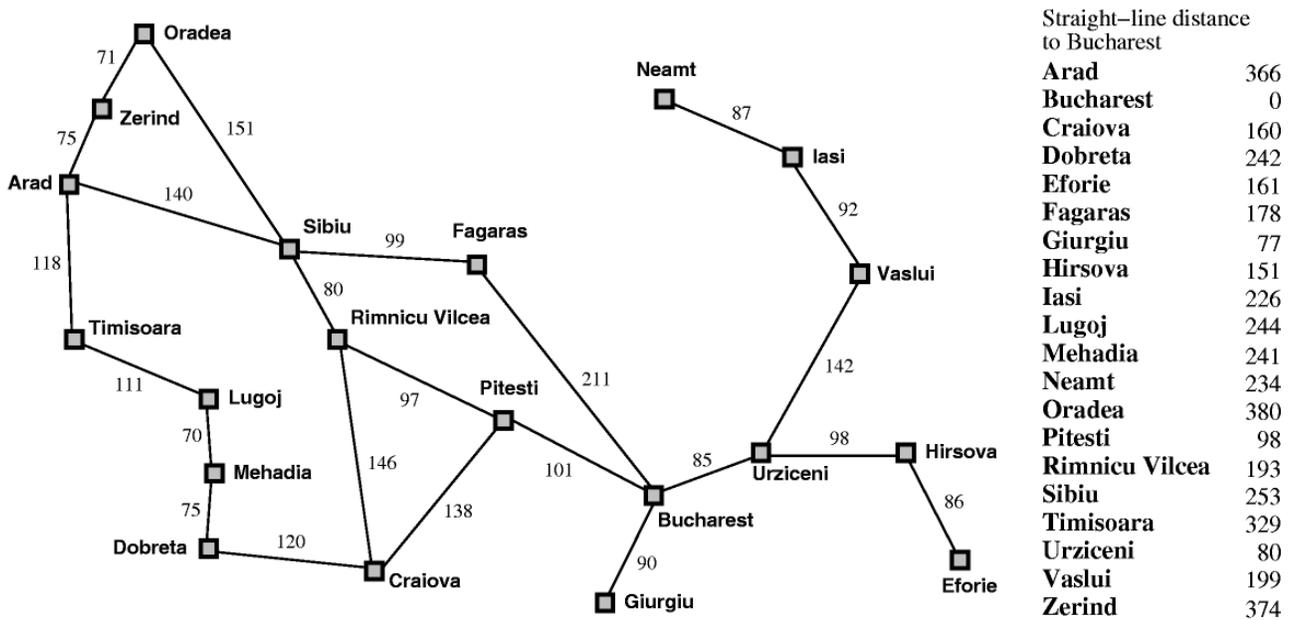


- (b) Resolva o problema de forma ótima, utilizando um algoritmo de busca apropriado. É boa ideia verificar a existência de estados repetidos?

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).

Cap. 4 - Russel & Norvig - Exercícios selecionados

11. Represente a operação da busca A* aplicada ao problema de ir até Bucareste a partir de Lugoj usando a heurística de distância em linha reta. Isto é, mostre a sequência de nós que o algoritmo irá considerar e a pontuação de f , g e h para cada nó.



Sequência de nós na borda:

L[0+244=244]
M[70+241=311], T[111+329=440]
L[140+244=384], D[145+242=387], T[111+329=440]
D[145+242=387], T[111+329=440], M[210+241=451], T[251+329=580]
C[265+160=425], T[111+329=440], M[210+241=451], M[220+241=461], T[251+329=580]
T[111+329=440], M[210+241=451], M[220+241=461], P[403+100=503], T[251+329=580],
R[411+193=604], D[385+242=627]
M[210+241=451], M[220+241=461], L[222+244=466], P[403+100=503], T[251+329=580],
A[229+366=595], R[411+193=604], D[385+242=627]
M[220+241=461], L[222+244=466], P[403+100=503], L[280+244=524], D[285+242=527],
T[251+329=580], A[229+366=595], R[411+193=604], D[385+242=627]
L[222+244=466], P[403+100=503], L[280+244=524], D[285+242=527], L[290+244=534],
D[295+242=537], T[251+329=580], A[229+366=595], R[411+193=604], D[385+242=627]
P[403+100=503], L[280+244=524], D[285+242=527], M[292+241=533], L[290+244=534],
D[295+242=537], T[251+329=580], A[229+366=595], R[411+193=604], D[385+242=627],
T[333+329=662]

B[504+0=504], L[280+244=524], D[285+242=527], M[292+241=533], L[290+244=534],
D[295+242=537], T[251+329=580], A[229+366=595], R[411+193=604], D[385+242=627],
T[333+329=662], R[500+193=693], C[541+160=701]

12. O algoritmo de caminho heurístico é uma busca pela melhor escolha na qual a função objetivo é $f(n) = (2 - w)g(n) + wh(n)$. Para que valores de w esse algoritmo oferece a garantia de ser ótimo? Que espécie de busca ele executa quando $w = 0$? E quando $w = 1$? E quando $w = 2$?

w=0 faz com que $f(n) = 2g(n)$, que equivale à busca de custo uniforme (a multiplicação por 2 não modifica a ordem em que os nós são expandidos). $w = 1$ faz com que $f(n) = g(n) + h(n)$, que equivale à busca A. $w = 2$ faz com que $f(n) = 2h(n)$, que equivale à busca gulosa pela melhor escolha. Este algoritmo é ótimo quando $h(n)$ é admissível e $w \leq 1$.*

13. Prove cada uma das afirmações a seguir:

- (a) A busca em extensão é um caso especial de busca de custo uniforme.

Quando todos os custos são iguais, temos que $g(n)$ é proporcional a profundidade (n), logo a busca de custo uniforme reproduz a busca em extensão, já que os nós são expandidos em ordem de menor profundidade (menor custo) para maior profundidade.

- (b) A busca em extensão, a busca em profundidade e a busca de custo uniforme são casos especiais da busca pela melhor escolha.

Busca em extensão equivale à busca pela melhor escolha com $f(n) = \text{profundidade}(n)$; busca em profundidade e busca pela melhor escolha com $f(n) = -\text{profundidade}(n)$; busca de custo uniforme e busca pela melhor escolha com $f(n) = g(n)$.

- (c) A busca de custo uniforme é um caso especial da busca A*.

Busca de custo uniforme equivale a A com $h(n) = 0$.*

14. Na página 111, definimos o relaxamento do quebra-cabeça de 8 peças em que um bloco pode se mover do quadrado A para o quadrado B , se B estiver vazio. A solução exata desse problema define a heurística de Gaschnig. Explique por que a heurística de Gaschnig é pelo menos tão precisa quanto h_1 (blocos mal posicionados) e mostre casos em que ela é mais precisa que h_1 e h_2 (distância Manhattan). Você poderia sugerir um modo de calcular a heurística de Gaschnig com eficiência?

A heurística dos blocos mal posicionados é exata para o problema em que um bloco pode ser movido de um quadrado A para qualquer quadrado B . Como esse problema é um relaxamento da condição que um quadrado pode ser movido do quadrado A para um quadrado B se B estiver vazio, o valor da heurística de Gaschnig não pode ser menor que o valor da heurística dos blocos mal posicionados. Como ela também é admissível (por ser um relaxamento do problema original), ela é mais precisa. Se permutarmos dois blocos adjacentes no estado objetivo, teremos um estado em que a heurística dos blocos mal posicionados e a heurística da distância Manhattan terão valor 2, e a heurística de Gaschnig terá valor 3. Para calcular a heurística de Gaschnig, repita o seguinte até que o estado objetivo seja atingido: seja B a posição atual do espaço vazio; se B for ocupado pelo quadrado X (não vazio) no estado objetivo, mova X para B ; senão, mova qualquer bloco mal posicionado para B .

15. Forneça o nome do algoritmo que resulta de cada um dos seguintes casos especiais:

- (a) Busca de feixe local com $k = 1$.

Busca de subida de encosta

- (b) Busca em feixe local com $k = \infty$.

(Não faz sentido)

-
- (c) Têmpera simulada com $T = 0$ em todos os momentos.
Busca de subida de encosta
- (d) Algoritmo genético com tamanho da população $N = 1$.
Busca aleatória