PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Instituto de Ciências Exatas e de Informática

Marcos Antonio Lommez Candido Ribeiro[1]

Lista #8

Inteligência Artificial

[1]Aluno de Graduação em Ciência da Computação – tonilommez@hotmail.com

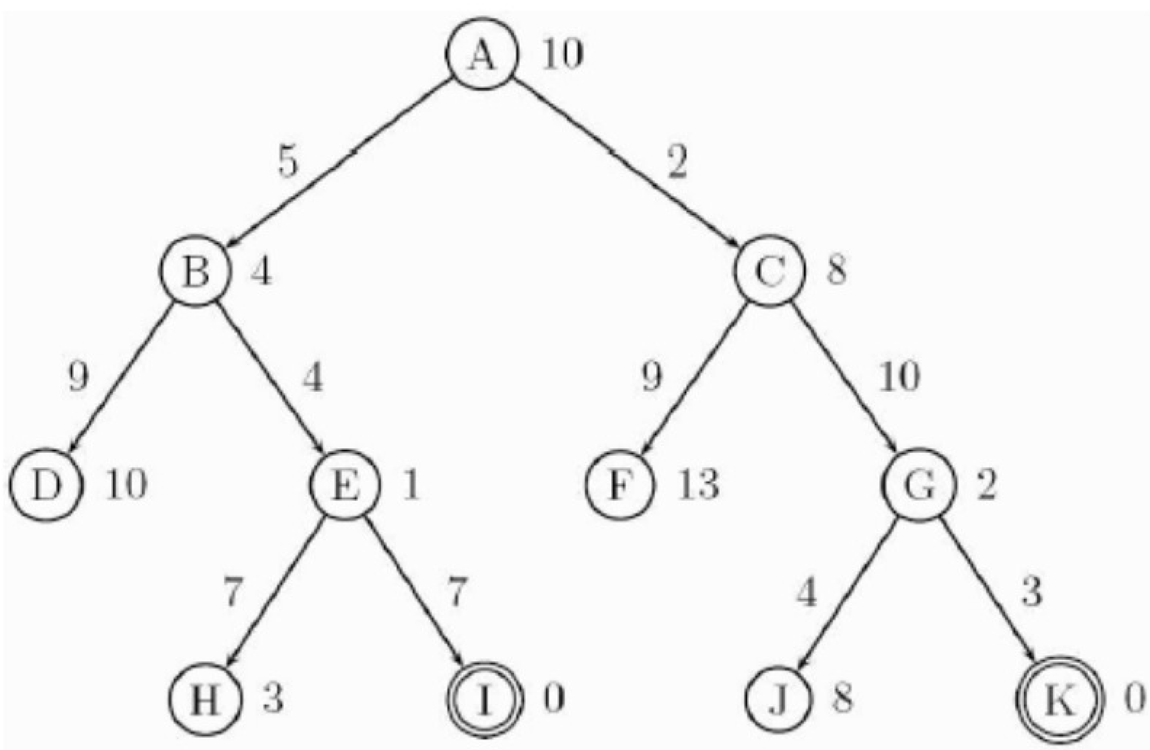
1) Considere o espaço de busca a seguir. Cada nó é rotulado por uma letra. Cada nó objetivo é representado por um círculo duplo. Existe uma heurística estimada para cada dado nó (indicada por um valor ao lado do nó). Arcos representam os operadores e seus custos associados. Para cada um dos algoritmos a seguir, pede-se:

1) Os nós visitados na ordem em que eles são examinados, começando pelo nó A

2) Forneça também a solução obtida por cada método

3) Pergunta-se: a heurística é admissível? Justifique.

No caso de escolhas equivalentes entre diferentes nodos, prefira o nodo mais próximo da raiz, seguido pelo nodo mais à esquerda na árvore. O algoritmo pára a busca quando encontra o I ou o K. Ou seja, não é necessário encontrar os dois objetivos.



1) Algoritmo de Busca em Largura (amplitude): Ordem de visitação:

Ordem de Visitação do algoritmo:
$A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F \rightarrow G \rightarrow H \rightarrow I$

Resultado:
$A \rightarrow B \rightarrow E \rightarrow I$

No caso da busca em largura a heurística de cada nó não é admissível porque o algoritmo apenas é capaz de computar o custo associado a cada arco, sendo assim poderia ser usado para decidir quem esta mais proximo relacionado ao peso, seja I ou K

Devido a restrição de parar no primeiro encontrado o valor resultante foi I, mas no caso de uma busca pelo mais próximo resultaria em K

2) Algoritmo de Busca em Profundidade

Ordem de Visitação:
$A \rightarrow B \rightarrow D \rightarrow E \rightarrow H \rightarrow I$

Resultado:
$A \rightarrow B \rightarrow E \rightarrow I$

Na busca em profundidade uma heurística também não seria admissível devido a natureza do algoritmo, que foi feito para medir o custo associado aos arcos, da mesma maneira que a busca em largura, ela seria capaz de encontrar o caminho mais curto para I ou K, mas decididmos parar em K.

3) Custo uniforme

Ordem de Visitação:
$A \rightarrow C \rightarrow B \rightarrow E \rightarrow F \rightarrow G \rightarrow D \rightarrow K$

Resultado:
$A \rightarrow C \rightarrow G \rightarrow K$

O algoritmo de Custo Uniforme por ser um Dijkstra like não se utiliza de heurística adicionada aos nós, então é baseado somente no ponderamento das arestas.

4) Algoritmo de Busca Gulosa

Ordem de Visitação:
$A \rightarrow B \rightarrow E \rightarrow I$

Resultado:
$A \rightarrow B \rightarrow E \rightarrow I$

O algoritmo de busca gulosa se preocupa única e exclusivamente com a heurística relacionada a cada nó, portando ignora o peso das arestas e segue diretamente pela heurística mais curta.

5) Algoritmo A*

Ordem de Visitação:
$A \rightarrow C \rightarrow G \rightarrow K$

Resultado:
$A \rightarrow B \rightarrow C \rightarrow E \rightarrow G \rightarrow K$

O algoritmo de A* faz uma união entre o Custo uniforme e o Busca Gulosa, utilizando heurística e ponderamento das arestas juntamente.

2)Para o problema do Puzzle de 8, pede-se:

1. A heurística de Manhattan é admissível? Justifique

A heurística de Manhattan é admissível. Porque uma heurística é considerada admissível se ela nunca superestima o custo para alcançar o objetivo a partir de qualquer estado no espaço de estados. Ou seja, o valor da heurística é sempre menor ou igual ao número real de movimentos necessários para atingir o estado final a partir do estado atual. Portanto Manhattan é admissível porque ela reflete, no mínimo, o número de movimentos que cada peça teria que fazer para chegar em sua posição correta.

2. Proponha uma outra heurística para este problema. Ela é admissível? Justifique

Outra heurística possivel sera a Hamming, que calcula a partir do número total de peças que estão na posição errada, sem considerar a distância entre a posição atual e a posição correta. Ela é admissível porque, para cada peça fora do lugar, é necessario pelo menos um movimento para colocá-la na posição correta, portanto, o custo estimado para o estado final nunca será maior do que o número real de movimentos necessários para resolver o quebra-cabeça.

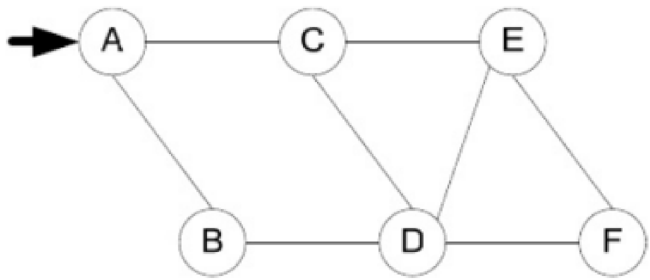
3) Julgue os itens a seguir, relativos a métodos de busca com informação (busca heurística) e sem informação (busca cega), aplicados a problemas em que todas as ações têm o mesmo custo, o grafo de busca tem fator de ramificação finito e as ações não retornam a estados já visitados.

I. A primeira solução encontrada pela estratégia de busca em largura é a solução ótima.
II. A primeira solução encontrada pela estratégia de busca em profundidade é a solução ótima.
III. As estratégias de busca com informação usam funções heurísticas que, quando bem definidas, permitem melhorar a eficiência da busca.
IV. A estratégia de busca gulosa é eficiente porque expande apenas os nós que estão no caminho da solução.

Estão certos apenas os itens
a) I e II.
b) I e III.
c) I e IV.
d) II e IV.
e) III e IV.

Letra (b)
Quando todas as acoes tem o mesmo custo o primeiro valor encontrado na busca em largura sera o valor ótimo automaticamente, por outro lado na busca em profundidade isso nem sempre ocorrera porque ela privilegia exaurir um caminho antes de explorar outras oportunidades.
A introdução da heurística na busca vem exatamente para poder melhorar a eficiência dos algoritmos, dando a eles mais informação sobre o contexto, para que não seja necessário buscar exaustivamente dentro de uma quantidade muito alta de caminhos possíveis, embora não garantem encontrar a melhor solução possível, apenas que ela é ótima e sera encontrada (se houver).
A busca gulosa procura seguir primeiro pelo caminho que parece melhor naquele momento, no entanto, ela não é eficiente em todos os casos, pois pode ficar presa em máximos locais ou seguir caminhos que parecem bons a curto prazo.

4) Considere o algoritmo de busca em largura em grafos. Dado o grafo a seguir e o vértice A como ponto de partida, a ordem em que os vértices são descobertos é dada por:



A) A B C D E F
B) A B D C E F
C) A C D B F E
D) A B C E D F
E) A B D F E C

Correta: letra (A)

5) Analise as seguintes as seguintes afirmativas:

I. A estratégia de busca em largura encontra a solução ótima quando todos os operadores de mudança de estado têm o mesmo custo.
II. A estratégia de busca em profundidade sempre expande um menor número de nós que a estratégia de busca em largura, quando aplicadas ao mesmo problema.
III. A estratégia de busca heurística encontra sempre a solução de menor custo.
IV. A estratégia de busca heurística expande um número de nós em geral menor que o algoritmo de busca em largura, mas não garante encontrar a solução ótima.
V. O algoritmo de busca heurística que utiliza uma função heurística admissível encontra a solução ótima.

A respeito, pode-se concluir que:
(a) apenas a afirmativa V é correta
(b) todas as afirmativas são corretas.
(c) todas as afirmativas são falsas.
(d) apenas as afirmativas II e V são corretas.
(e) apenas as afirmativas I, IV, e V são corretas.

Afirmativa correta: letra (e)

6) Considerando que h(n) é o custo estimado do nó n até o objetivo, em relação à busca informada, pode-se afirmar que:

(a) a busca gulosa minimiza h(n).
(b) a busca A* minimiza h(n).
(c) a busca de custo uniforme minimiza h(n).
(d) a busca gulosa minimiza h(n) somente se a heurística for admissível.
(e) a busca A* minimiza h(n) somente se a heurística for admissível.

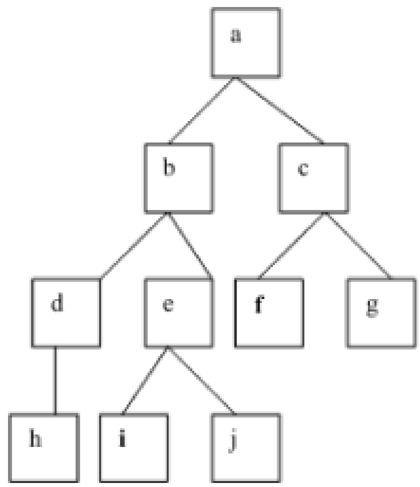
Afirmativa correta: letra (a)

7) Considere h(x) como uma função heurística que define a distância de x até a meta; Considere ainda hº(x) como a distancia real de x até a meta. h(x) é dita admissível se e somente se:

(a) $\exists n\ h(n) \leq h*(n)$
(b) $\forall n\ h(n) \leq h*(n)$
(c) $\forall n\ h(n) \leq h*(n)$
(d) $\exists n\ h(n) > h*(n)$
(e) $\exists n\ h(n) < h*(n)$

Afirmativa correta: letra (b)

8) Seja a árvore binária abaixo a representação de um espaço de estados para um problema p, em que o estado inicial é a, e i e f são estados finais



Um algoritmo de busca em largura-primeiro forneceria a seguinte sequência de estados como primeira alternativa a um caminho-solução para o problema p:

a) a b d h e i

b) a b c d e f

c) a b e i

d) a c f

e) a b d e f

Afirmativa correta: letra (b)

9) Suponha um algoritmo de busca pelo melhor primeiro (best-first ou busca gulosa) em que a função objetivo é f(n) = (2 − w).g(n) + w.h(n). Que tipo de busca ele realiza quando w = 0? Quando w = 1? E quando w = 2?

Quando **w = 0** a formula anula a segunda parte que se refere a heurística dos nós e se importa apenas com o valor real do grafo, onde inclusive se usa uma multiplicação de peso multiplicado por 2, resultando na formula **2g(n)**, o que equivaleria ao algoritmo de **custo uniforme**

Quando **w = 1** a formula se torna **g(n) + h(n)**, o que equivale ao algoritmo de **A***

Por fim quando **w = 2** o valor da primeira parte se torna uma multiplicação por 0, ficando **2h(n)**, assim ignorando o valor real e se importando exclusivamente com a heurística, o que equivale ao algoritmo de **busca gulosa**

10)Para cada uma das frases abaixo, verifique se a afirmação é verdadeira ou falsa. Justifique.

a) A busca em largura sempre encontra a solução ótima.

Verdadeiro. A busca em largura sempre ira encontrar a solução ótima do problema, mas isso não garante que a primeira solução encontrada sera a melhor, mas garante que ao fim do algoritmo ela sera encontrada.

b) A estratégia de busca heurística sempre encontra a solução de menor custo

Falso. A busca heurística encontra a solução de menor custo garantidamente quando a heurística escolhida for admissível, ou seja, para todo n o resultado de h(n) é menor ou igual ao h(n) real.

c) A busca de custo uniforme minimiza do g(n)

Verdadeiro. O objetivo da busca de custo uniforme é exatamente minimizar g(n), porque ela ignora qualquer heurística existente.

11) Considere um jogo do tipo 8-puzzle, cujo objetivo é conduzir o tabuleiro esquematizado na figura abaixo para o seguinte estado final.

1	2	3
8		4
7	6	5

Considere, ainda, que, em determinado instante do jogo, se tenha o estado E0 a seguir:

3	4	6
5	8	
2	1	7

Pelas regras desse jogo, sabe-se que os próximos estados possíveis são os estados E1, E2 e E3 mostrados abaixo.

3	4	6
5		8
2	1	7

E1

3	4	6
5	8	7
2	1	

E2

3	4	
5	8	6
2	1	7

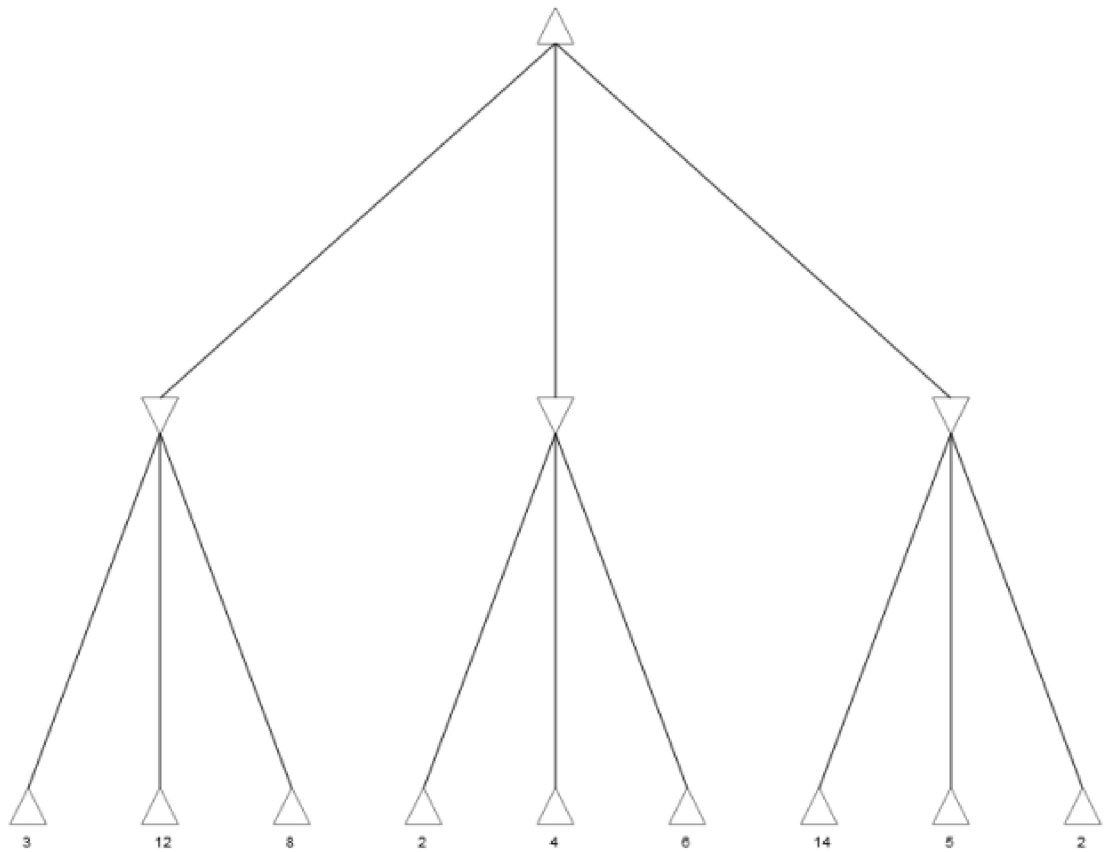
E3

Considere uma função heurística h embasada na soma das distâncias das peças em relação ao estado final desejado, em que a distância d a que uma peça p está da posição final é dada pela soma do número de linhas com o número de colunas que a separam da posição final desejada.
Por exemplo, em E1, $d(1) = 2 + 1 = 3$. A partir dessas informações analise as asserções a seguir.
Utilizando-se um algoritmo de busca gulosa pela melhor escolha que utiliza a função h, o próximo estado no desenvolvimento do jogo a partir do estado E0 tem de ser E3
porque,
dos três estados E1, E2 e E3 possíveis, o estado com menor soma das distâncias entre a posição atual das peças e a posição final é o estado E3.
Assinale a opção correta a respeito dessas asserções.

a) As duas asserções são proposições verdadeiras, e a segunda é uma justificativa correta da primeira.
b) As duas asserções são proposições verdadeiras, e a segunda não é uma justificativa correta da primeira.
c) A primeira asserção é uma proposição verdadeira, e a segunda é uma proposição falsa.
d) A primeira asserção é uma proposição falsa, e a segunda é uma proposição verdadeira.
e) As duas asserções são proposições falsas.

Afirmativa correta: letra (a)

12) Rode o MINIMAX na árvore abaixo. Qual será a decisão do MAX?



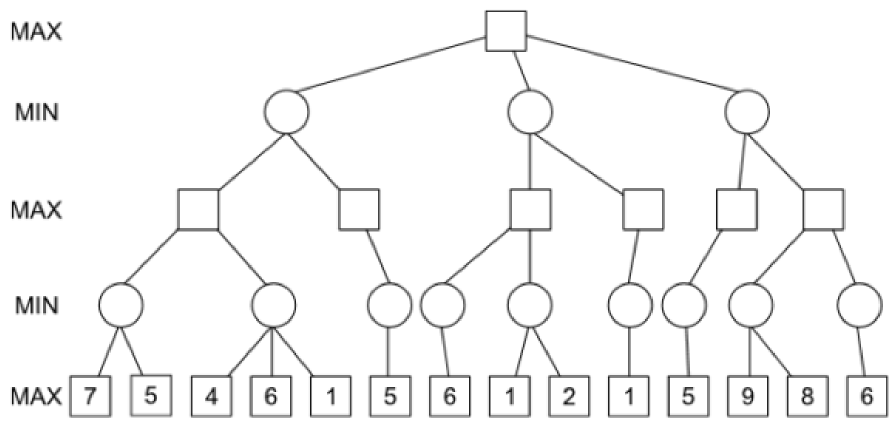
A decisão do MAX nesta arvore sera de 3

13) Rode o MINIMAX com corte alfa-beta para a árvore acima. Haverá corte? Onde? Quais?

Haverá corte ao analisar o segundo nó de MIN, porque ao avaliar o seu primeiro filho contendo o numero 2, podemos afirmar que seu valor automaticamente será menor ou igual a dois, logo não precisamos avaliar seus demais filhos, afinal a recursão de MAX já havia encontrado 3 no primeiro nó de MIN. Diferentemente do segundo nó de MIN, o terceiro só tera essa conclusão ao avaliar seu terceiro filho.

Portanto com o corte alfa-beta na árvore foram realizados 2 cortes, no nó central nos valores 4 e 9.

14) Considere a árvore minimax abaixo representando um jogo. Simule a execução do algoritmo minimax com podas alfa-beta preenchendo os valores na árvore e indicando quais nodos serão podados.



Resultado: