

EA 072 – Inteligência Artificial em Aplicações Industriais
Solução Lista 4 (Capítulo 4 Livro Texto)

4.2.

$$f(n) = (2 - w)g(n) + wh(n)$$

$w = 0 \Rightarrow f(n) = 2 g(n)$. Corresponde ao algoritmo de custo uniforme.

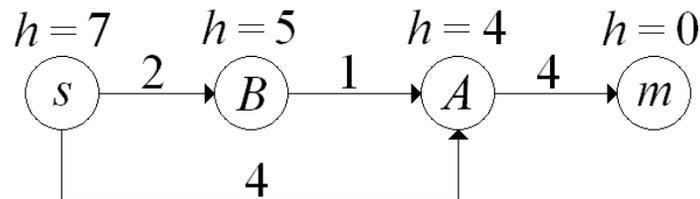
$w = 1 \Rightarrow f(n) = g(n) + h(n)$. Corresponde ao algoritmo A*.

$w = 2 \Rightarrow f(n) = 2 h(n)$. Corresponde ao algoritmo Greedy best-first.

Observar que $f(n) = (2 - w) [g(n) + w/(2 - w) h(n)]$. Logo, o algoritmo se comporta como o A* com heurística $w/(2 - w) h(n)$. Para $w \leq 1$, $w/(2 - w) h(n) \leq h(n)$. Portanto, a heurística é admissível se $h(n)$ é admissível.

4.4.

Sugestão:



h é admissível, ou seja, não superestima h^* . Porém, não é consistente.

$$h(s) \leq c(s, a, B) + h(B) = 2 + 5 = 7.$$

$$h(s) \leq c(s, a', A) + h(A) = 4 + 4 = 8.$$

$f(A) < f(B) \Rightarrow A$ expandido antes de B vai para a lista fechada.

4.5.

Sim. Rimnicu Vilcea e Lugoj é um exemplo.

Caminho ótimo: Rimnicu Vilcea, Craiova, Dobreta, Mehadia, Lugoj.

Busca gulosa começando de Rimnicu Vilcea vai para o lado errado: vai para Sibiu ao invés de Craiova.

Busca gulosa começando de Lugoj vai para o lado certo: Lugoj, Mehadia, ...

4.7.

Heurística consistente \Rightarrow heurística admissível

h consistente: $\forall n, n'$; n' sucessor de n , então $h(n) \leq c(n, a, n') + h(n')$.

Indução: número de nós k no mesmo caminho para chegar a meta a partir do nó n .

$k = 1$, n' é meta: $h(n) \leq c(n, a, n')$. Assim n' está a k passos no caminho ótimo para a meta e $h(n')$ é admissível para a hipótese. Então, $h(n) \leq c(n, a, n') + h(n') \leq c(n, a, n') + h^*(n') = h^*(n)$. Logo, $h(n)$ a $(k+1)$ passos da meta também é admissível.

4.13.

A complexidade espacial do LRTA* é dominada pela tabela $\text{result}[a,s]$, isto é, o produto do número de estados visitados n e o número de ações por estado m . Logo, $O(mn)$.

Complexidade temporal é no mínimo $O(nm^2)$ porque para cada ação é computada m vezes para $H[s]$ (implementação ingênua, desfiguração), mas pode ser reduzida para $O(mn)$, isto é, quando cada par estado-ação é verificado no máximo uma vez, mas na realidade ele pode ser verificado várias vezes.