

Exame 15-16 Normal

- 1) a) O ambiente é determinístico porque o seu próximo estado pode ser determinado a partir do seu estado atual.
- É não episódico porque há uma dependência de episódios.
- É discreto porque origina uma série de percepções e ações perfeitamente distintas umas das outras.
- É um ambiente estático porque o mesmo não muda, está estático.
- b) Eventualmente o agente chega à célula G (objetivo), no entanto ele não sabe se chegou e passou pela célula B, que como diz no enunciado, é obrigatório.
- c) O ambiente mais complexo de todos é não acessível, estocástico, dinâmico, não episódico e contínuo.

- 2) a) Heurística Admissível se $h \leq g$ (custo real)

Olhando para H1

$$h(A) = 6 \leq \overbrace{A}^1 + \underbrace{B+D}_1 + \underbrace{E+C}_1 + F = 6 \quad \checkmark$$

$$h(B) = 5 \leq B+D+E+C+F = 5 \quad \checkmark$$

$$h(C) = 1 \leq C+F = 5 \quad \checkmark$$

$$h(D) = 4 \leq D+E+C+F = 4 \quad \checkmark$$

$$h(E) = 2 \leq E+C+F = 2 \quad \checkmark$$

$$h(F) = 0 \quad \checkmark$$

Em todos os estados é admissível, logo H1 é admissível

Olhando para H2

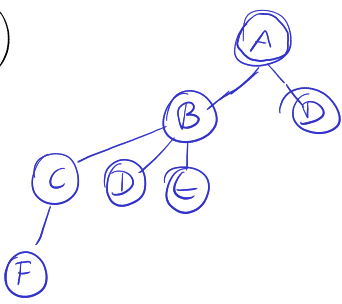
$$h(A) = 5 \leq 6 \quad \checkmark$$

$$h(B) = 7 \leq 5 \quad \times$$

Basta um estado não ser admissível para percebermos que

H2 não é admissível

b) i)

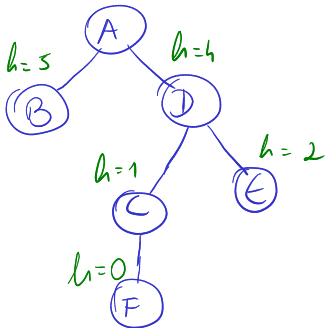


Usando $\epsilon = H/1$

Custo

$$A + B + C + F = 1 + 5 + 1 = 7$$

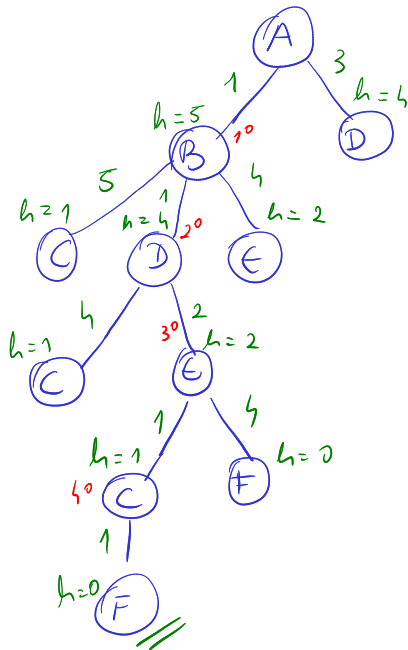
ii) $f = h$



Custo

$$A + D + C + F = 3 + 4 + 1 = 8$$

iii) $f = \epsilon_g + h$



Custo de solução

$$A + B + D + E + C + F = 6$$

3)

- a) $0 \rightarrow \text{Azul}$
 $1 \rightarrow \text{vermelho}$
 $2 \rightarrow \text{verde}$
- $\left. \begin{array}{l} [012012] \\ [021021] \end{array} \right\} \text{soluções próprias}$

Para lidar com soluções inválidas em que existem cores iguais nas cidades adjacentes, podemos reparam a solução olhando para as cores adjacentes e escolhendo uma cor diferente entre essas duas.

- b) A função de avaliação irá avaliar quantas cidades adjacentes têm a mesma cor. Neste caso o objetivo de otimização será minimizar o nº de cidades adjacentes com a mesma cor.

Se considerarmos o array $[A B C D E F]$, usando a função de avaliação proposta na Solução 1, irá retornar 1, uma vez que B e E têm a mesma cor.

Na Solução 2 iremos retornar 1 novamente porque B e E têm a mesma cor.

c) $P_1 = [012012]$
 $P_2 = [021021]$

$D_1 = [011022]$
 $D_2 = [022012]$

- d) $D_1 = [011022] \rightarrow$ Neste caso poderemos ter um operador de mutação de troca, em que iremos trocar duas cores de posição.

$D_2 = [022012] \rightarrow$ Neste caso poderemos ter um operador de mutação de troca de cor uma vez que há cores que só aparecem uma vez.

h)



Peso valor
↓ ↓
(3, 4)

- a) Podemos ter um array de 4 posições em que os objetos são identificados de 1 a 4, se tivermos o número 0 significa que não há um objeto naquela posição.
Ex: [1 2 0 0] → Neste caso só temos o objeto 1 e 2

A função de avaliação irá percorrer o array e irá calcular o valor de todos os objetos lá inseridos enquanto o peso ≤ 5 .

Para gerar vizinhos, podemos tirar ou colocar um objeto no saco respeitando sempre a restrição do peso.

- b) Sim, podemos usar com 2 movimentos TABU

Solução inicial → [1 0 0 0] $p=3, q=4$

1ª Iteração

[0 0 0 0] $p=0, q=0$

[1 0 0 0] $p=3, q=4$

[1 2 0 0] $p=5, q=7$

[1 0 3 0] $p=5, q=9$ → movimento TABU

[1 0 0 4] $p=8, q=11$ X

MOVIMENTO TABU = 3

2ª Iteração

Pegamos no movimento TABU

[0 0 3 0] $p=2, q=5$

[1 2 3 0] $p=7, q=12$ X

[1 0 0 0] → TABU

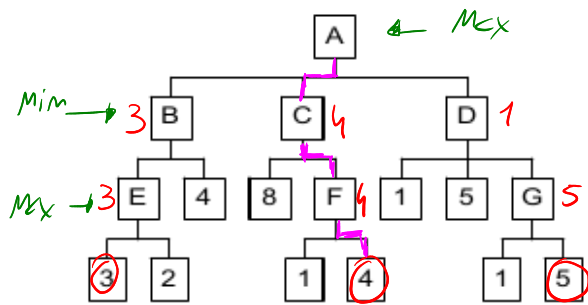
[1 0 3 4] $p=10$ X

c) Poderão considerar um array de 8 posições, em que as 4 primeiras representam o Saco 1 e as restantes o Saco 2. No entanto não podemos ter o mesmo objeto em 2 sacos, logo no array só podemos repetir o n.º do objeto uma vez.

Pene é função de avaliação, bastava somar os valores de todas as posições respeitando a restrição do peso.

O operador de vizinhança é igual, respeitando o peso e a restrição de não haver objetos duplicados no mesmo Saco.

4) a)

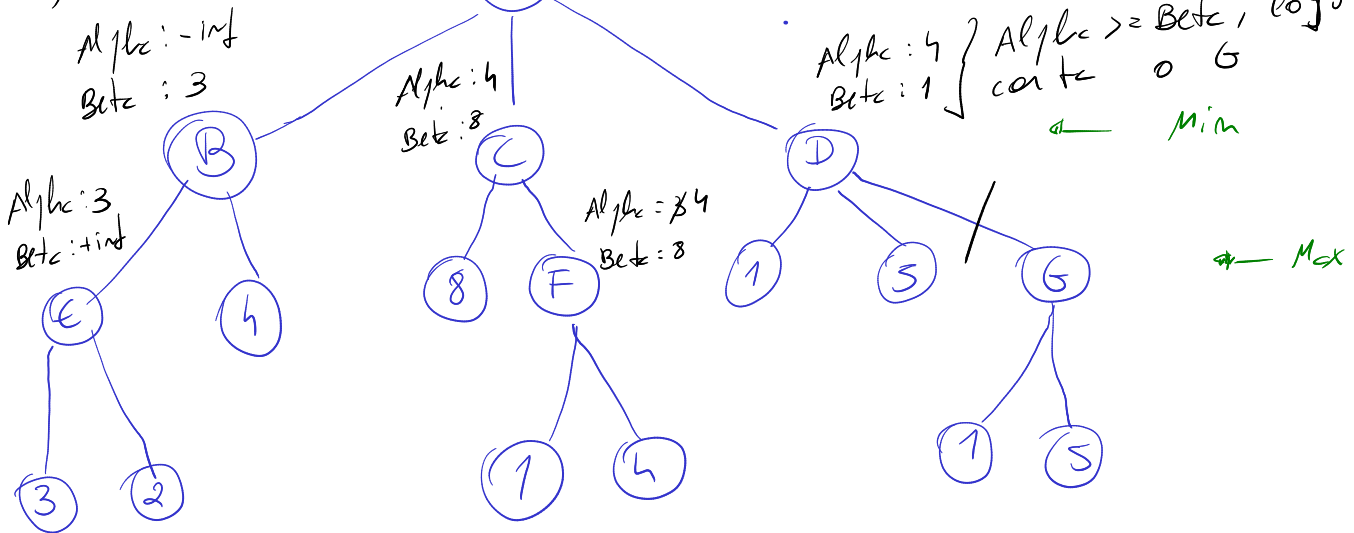


Escolhe o C

Alpha: $-\infty$
Beta: $+\infty$

Max

b)



c) Não porque só conta os ramos com valores que são menores que o valor já gerado.

d) Sim, pode-se adaptar o algoritmo minimax aos jogos com elemento sorte, mas a envoltura deve incluir também nós que reduzem o fator sorte.

