Exame 13-14 Normal

1)

a) É determinístico porque o seu próximo estado é determinado a partir do seu estado atual e da ação a executar

É não episódico porque há uma dependência de episódios.

É discreto porque existe séries de perceções e ações perfeitamente distintas umas das outras.

É estático porque o ambiente não muda a sua natureza.

b) Como o objetivo é recolher as fontes de energia, um agente reativo sem memória não vai saber quando é que já terminou porque não sabe se já visitou em todas as células de energia.

c) Informação útil a armazenar na memória:

- Sempre que passe por uma célula de Energia guardar as suas coordenadas

Regras:

- Antes de se movimentar verificar a existência de obstáculos
- Não avançar para obstáculos

2)

a) Heurística admissível $h( ) <= g$ (custo real)

$h(S) = 10 \leq 10$ ✓ $h(B) = 6 \leq 8$ ✓

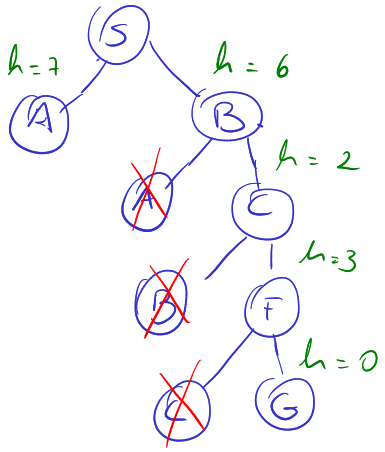
$h(A) = 7 \leq 7$ ✓ $h(C) = 2 \leq 18$ ✓

$h(D) = 5 \leq 5$ ✓ $h(E) = 4 \leq 5$ ✓

$h(F) = 3 \leq 20$ ✓ $h(G) = 0$ ✓

A heurística é admissível, porque a heurística de cada nó é $\leq g$

b) Sôfrega = h

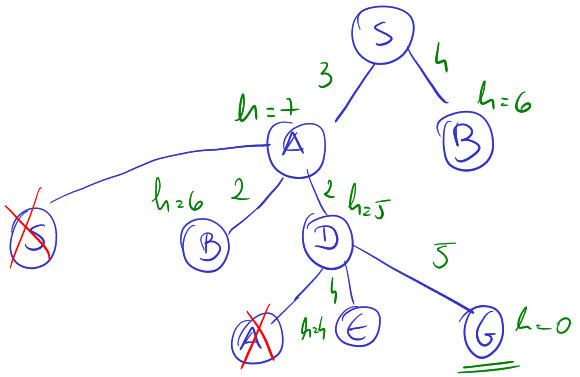


Custo de solução

S + B + C + F + G

4 + 10 + 5 + 20 = 39

c) A* = h + Eg



Custo de solução

S + A + D + G

3 + 2 + 5 = 10

d) Como o A* encontra sempre a solução ótima, o custo computacional é maior comparativamente à Sôfrega uma vez que a Sôfrega expande os nós com uma heurística menor enquanto que o A* soma o custo real à heurística do nó atual, fazendo assim mais algumas iterações, no entanto a sôfrega também pode chegar à solução ótima, mas não é certo que isso aconteça

3)

a) Representação da solução: [1 0 0 0 0 0 0 0 0] → Representa o -7
(-7 -5 -3 -2 1 5 6 8 12)

Função avaliação: Ir buscar o array binário e ver a que valor corresponde no conjunto inicial, enquanto faz a soma desses valores; tendo em atenção que o resultado final tenic de ser ∅

Operador vizinhança: Alterar um bit para gerar outra solução (vizinho)

b) 1ª Iteração

S_1 [100000000] resultado = -7 → solução inválida

gerar vizinho...

2ª Iteração

S_2 = [100010000] resultado = -7 + 1 = -6 → solução inválida

gerar vizinho...

3ª Iteração

S_3 = [100010100] resultado = -7 + 1 - 6 = 12 → solução inválida

c) O algoritmo de recristalização simulada não impede a largar em ótimos locais porque aceita soluções piores com alguma probabilidade à mistura.

4)

a) $k = 3$ (3 cores)

$N = 6$ (6 mosaicos)

Representação: [1 2 3 1 2 3]

Caso houvesse uma solução inválida, por exemplo [1 1 2 3 2 3], iria percorrer o array e verificar que cores estavam em adjacência, se fossem cores repetidas, trocaria com uma cor aleatória do array e voltava a verificar se essa troca não originou outra solução inválida.

b) Função de avaliação: Verifica se há cores iguais adjacentes, caso fosse verdade, a sua qualidade seria o nº de cores diferentes adjacentes.

O objetivo da otimização seria minimizar o nº de cores iguais adjacentes.

c) Operador de recombinação: 1 ponto de corte aleatório

$S_1 = [1\ 2\ 3 \mid 1\ 2\ 3]$

$S_2 = [1\ 1\ 3 \mid 2\ 2\ 3]$

$D_1 = [1\ 2\ 3\ 2\ 2\ 3]$

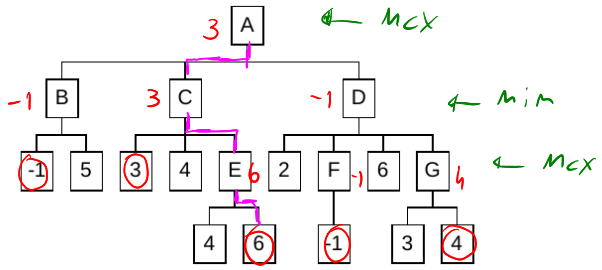
$D_2 = [1\ 2\ 3\ 1\ 1\ 3]$

Mutação: troca entre cores

$D_1 = [1\ 2\ 3\ \underline{2}\ 2\ 3]$ → troca entre 2 e 1 → $D_1 = [1\ 2\ 3\ \underline{1}\ 2\ 3]$

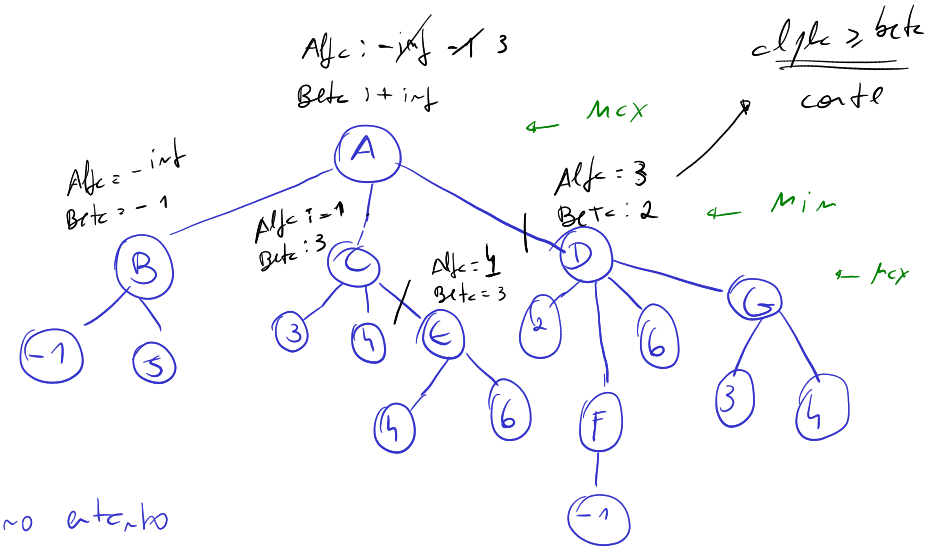
5)

a)



Escolhe o C

b)



c) Sim, no entanto a arvore deve incluir também nó's que reduzem o factor sorte.