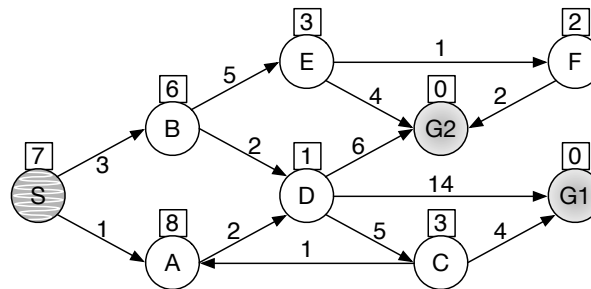


1º Teste

– Com consulta limitada –

I) [7val] Considere o seguinte grafo de estados de um problema de procura. Os valores apresentados nos arcos correspondem ao custo do operador (ação) respectivo, enquanto os valores nos rectângulos correspondem ao valor de uma heurística (estimativa do custo de chegar desse estado ao objectivo). Não se representam os nomes dos operadores, correspondendo cada arco a um operador distinto. Em todos os algoritmos cegos e em caso de empate nos algoritmos informados, quando um nó é expandido, assuma que os seus sucessores são sempre colocados na fronteira por ordem lexicográfica do nome do nó, de forma a que o nó mais próximo do início do alfabeto seja seleccionado antes dos seus irmãos. Pretende-se encontrar um caminho desde o estado S até G1 ou G2.



- Caracterize a heurística quanto à admissibilidade e consistência, justificando a sua resposta.
- Considere os algoritmos de procura em largura primeiro (otimizada), procura por aprofundamento progressivo, procura de custo uniforme (em grafos), procura sôfrega (em grafos) e A* (em grafos, na versão mais eficiente do algoritmo que garante a obtenção da solução óptima dadas as características da heurística). Para cada um dos algoritmos, indicar:
 - Qual o caminho encontrado pelo algoritmo
 - Quais os nós que são retirados da fronteira (para serem testados e/ou expandidos), pela ordem pela qual foram retirados da fronteira.
- Considere duas funções heurísticas admissíveis h_1 e h_2 . Considere um número real a e a função heurística $h_3(s) = ah_1(s) + (1 - a)h_2(s)$. Para que gama de valores a temos que h_3 é garantidamente admissível? Justifique.
- No algoritmo de Recristalização Simulada (Simulated Annealing), se o valor da função de avaliação no estado corrente for um valor positivo c , e o valor do vizinho escolhido também for c , como é que a probabilidade de nos movermos para o vizinho muda quando a temperatura desce? Assuma que estamos a maximizar a função de avaliação.

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

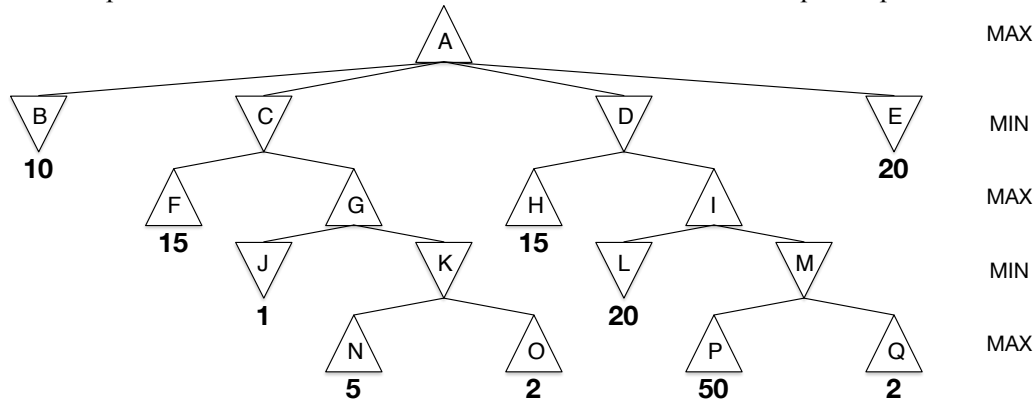
II) [4val] Considere um problema de escalonamento onde existem 5 atividades a serem escalonadas em 4 slots, que vamos representar como um problema de satisfação de restrições onde as atividades são representadas pelas variáveis A, B, C, D e E, com o domínio de cada variável igual a $\{1,2,3,4\}$, e sujeitas às seguintes 8 restrições:

$$A > D \quad D > E \quad C \neq A \quad C > E \quad C \neq D \quad B \geq A \quad B \neq C \quad C \neq D + 1$$

O objectivo é resolver o problema da forma mais eficiente possível, usando o algoritmo de consistência de arcos (AC-3) seguido do algoritmo para CSPs com estrutura arbórea, usando, se necessário, o condicionamento por conjunto de corte. Para tal, responda às seguintes perguntas. Sempre que houver alternativa, atribua os valores por ordem crescente.

- Desenhe o grafo de restrições.
- Indique o domínio das variáveis após a aplicação do algoritmo de consistência de arcos AC-3.
- Encontre uma solução através do algoritmo para CSPs com estrutura arbórea, indicando:
 - O conjunto de corte.
 - A linearização da árvore resultante e o domínio das suas variáveis após a fase de remoção de inconsistências.
 - A solução encontrada.

III) [5val] Considere a árvore de jogo de dois jogadores (MAX e MIN), onde os valores nas folhas são estimativas do ganho para MAX a partir desse estado e os filhos de um nó são visitados da esquerda para a direita.



- Calcule o valor MINIMAX de cada nó não terminal e indique qual o movimento que MAX deverá escolher.
- Assumindo que a árvore é percorrida da esquerda para a direita, indique quais os nós que nunca chegariam a ser visitados pelo algoritmo de procura α - β .
- Ignore os valores nas folhas. Existe algum conjunto de valores a atribuir às folhas de forma a que o algoritmo de busca α - β efetuasse os seguintes cortes (responda SIM/NÃO em cada caso):
 - No arco de A para C.
 - No arco de G para K.
 - No arco de D para I.
- Considere agora que os nós K e M da árvore da figura são nós estocásticos (CHANCE) onde cada sucessor é equiprovável. Calcule o valor EXPECTEDMINIMAX dos nós A, C e D.
- Considere a seguinte afirmação: “Em geral, se o jogador MAX efetuar sempre uma jogada que maximize o valor de EXPECTEDMINIMAX, irá ganhar pelo menos o valor de EXPECTEDMINIMAX, independentemente do da forma de jogar do jogador MIN”. A afirmação é verdadeira? Justifique.

XX

IV) [4val] Verifique, usando o algoritmo DPLL, se o seguinte conjunto de frases em lógica proposicional são satisfazíveis (sempre que encontrar uma ramificação, deve escolher as variáveis proposicionais alfabeticamente, e atribuir VERDADE antes de FALSO):

$$\begin{aligned} p &\Rightarrow (q \Leftrightarrow r) \\ q &\Leftrightarrow \neg(r \wedge w) \\ q &\Rightarrow (p \wedge w) \end{aligned}$$

XX

V) [Bónus: até 2val] Desenhe um grafo de estados onde o algoritmo de procura sôfrega encontre uma resposta diferente da encontrada pelo algoritmo A*. Não use mais do que 4 estados, e explique a sua resposta.

FIM!

Nome:

Número:

I.a) Admissível: SIM**Consistente: NÃO**

Justificação:

É admissível por o valor da heurística para cada estado (nó do grafo) não ser superior ao custo menor para chegar ao objectivo, que pode ser verificado diretamente no grafo, dado o seu tamanho reduzido: $h(S)=7 < h^*(S)=c(S \rightarrow A \rightarrow D \rightarrow G2)=9$, $h(A)=8 < h^*(A)=c(A \rightarrow D \rightarrow G2)=8$, etc.

Não é consistente pois existem estados n_1 , n_2 , ligados por um arco com custo $c(n_1, n_2)$ tal que $h(n_1) > c(n_1, n_2) + h(n_2)$. Por exemplo, se $n_1=A$ e $n_2=D$, temos que $h(A)=8$, $c(A, D)=2$ e $h(D)=1$, logo, $h(A) > c(A, D) + h(D)$, pelo que a heurística não é consistente.

I.b) Algoritmo	Solução	Nós Retirados da Fronteira
Largura primeiro	$S \rightarrow A \rightarrow D \rightarrow G1$	S, A, B, D
Aprofundamento progressivo	$S \rightarrow A \rightarrow D \rightarrow G1$	S, S, A, B, S, A, D, B, D, E, S, A, D, C, G1
Custo uniforme	$S \rightarrow A \rightarrow D \rightarrow G2$	S(0), A(1), B(3), D(3), C(8), E(8), F(9), G2(9)
Sôfrega	$S \rightarrow B \rightarrow D \rightarrow G1$	S(7), B(6), D(1), G1(0)
A*	$S \rightarrow A \rightarrow D \rightarrow G2$	S(7), A(9), D(4), B(9), G2(9)

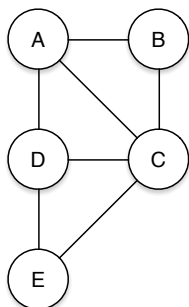
I.c) Gama de valores: $0 \leq a \leq 1$

Justificação:

Para $0 \leq a \leq 1$, dado que tanto a como $(1-a)$ são valores positivos, e $h_1(s) \leq c(s)$ e $h_2(s) \leq c(s)$, temos que $ah_1(s) \leq ac(s)$ e $(1-a)h_2(s) \leq (1-a)c(s)$. Logo, $h_3(s) = ah_1(s) + (1-a)h_2(s) \leq ac(s) + (1-a)c(s) = c(s)$, ou seja, $h_3(s) \leq c(s)$ i.e. $h_3(s)$ é admissível

Para $a > 1$, construímos o seguinte contra-exemplo: seja $h_1(s) = c(s)$ e $h_2(s) = 0$. Nesse caso, $h_3(s) = ah_1(s) > c(s)$ ou seja, $h_3(s)$ não é admissível.

Para $a < 0$, construímos o seguinte contra-exemplo: seja $h_1(s) = 0$ e $h_2(s) = c(s)$. Nesse caso, $h_3(s) = (1-a)h_2(s) > c(s)$, dado que $(1-a) > 1$ ou seja, $h_3(s)$ não é admissível.

I.d) A probabilidade mantém-se inalterada, e igual a 1.**II a) Grafo de restrições****b)**

Variável	A	B	C	D	E
Domínio	3 4	3 4	2 4	2 3	1 2

c)i.

Conjunto de Corte: C

c)ii.

Variável	E	D	A	B	
Domínio	1	3	3 4	3 4	

c)iii.

Variável	A	B	C	D	E
Solução	4	4	2	3	1

III.a)			A: 20 C: 2 D: 15 G: 2 I: 20 K: 2 M: 2 Movim. de MAX: E		
b)					
Nós não visitados: O M P Q					
c)			d)		
i: NÃO ii: SIM iii: SIM			A: 20 C: 3,5 D: 15		
e) Resposta (SIM/NÃO): NÃO			No caso da árvore de jogo à esquerda, o valor EXPECTEDMINIMAX é igual a 1. No entanto, ao jogar esse valor (aliás, neste caso nem tem escolha), MAX pode acabar por receber 0, i.e. um valor menor do que o EXPECTEDMINIMAX. Neste caso, o MIN também não tem escolha.		
Justificação:					
</					

IV) Satisfazível (SIM/NÃO)? SIM Se SIM, indique o modelo encontrado: p: F q: F r: V w: V Forma clausal: $(\neg p \vee \neg q \vee r) \wedge (\neg p \vee q \vee \neg r) \wedge (\neg q \vee \neg r \vee \neg w) \wedge (q \vee r) \wedge (q \vee w) \wedge (\neg q \vee p) \wedge (\neg q \vee w)$ Aplicação do DPLL:		
$ \begin{array}{l} (\neg p \vee \neg q \vee r) \\ (\neg p \vee q \vee \neg r) \\ (\neg q \vee \neg r \vee \neg w) \\ (q \vee r) \\ (q \vee w) \\ (p \vee \neg q) \\ (\neg q \vee w) \end{array} \begin{array}{l} \text{Split} \\ (p) \end{array} \Rightarrow \begin{array}{l} (\neg q \vee r) \\ (q \vee \neg r) \\ (\neg q \vee \neg r \vee \neg w) \\ (q \vee r) \\ (q \vee w) \\ (\neg q \vee w) \end{array} \begin{array}{l} \text{Split} \\ (q) \end{array} \Rightarrow \begin{array}{l} (r) \\ (\neg r \vee \neg w) \\ (w) \end{array} \begin{array}{l} \text{Unit} \\ (r) \end{array} \Rightarrow \begin{array}{l} (\neg w) \\ (w) \end{array} \begin{array}{l} \text{Unit} \\ (w) \end{array} \Rightarrow (\perp) $		
$ \begin{array}{l} \text{Backtrack} \\ (\neg q) \end{array} \Rightarrow \begin{array}{l} (\neg r) \\ (r) \\ (w) \end{array} \begin{array}{l} \text{Unit} \\ (r) \end{array} \Rightarrow (\perp) $		
$ \begin{array}{l} \text{Backtrack} \\ (\neg p) \end{array} \Rightarrow \begin{array}{l} (\neg q \vee \neg r \vee \neg w) \\ (q \vee r) \\ (q \vee w) \\ (\neg q) \\ (\neg q \vee w) \end{array} \begin{array}{l} \text{Unit} \\ (\neg q) \end{array} \Rightarrow \begin{array}{l} (r) \\ (w) \end{array} \begin{array}{l} \text{Unit} \\ (r) \end{array} \Rightarrow \begin{array}{l} (w) \\ (w) \end{array} \begin{array}{l} \text{Unit} \\ (w) \end{array} \Rightarrow \top $		

V) Grafo de Estados:	Justificação: No grafo de estados à esquerda, onde o estado inicial é S e o estado objectivo é G, o algoritmo de pesquisa sôfrega vai encontrar o caminho $S \rightarrow B \rightarrow G$, com um custo de 11, enquanto o algoritmo A* vai encontrar o caminho $S \rightarrow A \rightarrow G$, com um custo de 3. O algoritmo de pesquisa sôfrega vai retirar da fronteira, e expandir, o nó B(1) antes do nó A(2), pois selecciona sempre o nó que aparenta estar mais perto do objectivo, ignorando o custo incorrido para lá chegar. O algoritmo A* retira da fronteira o nó que tem menor soma do valor da heurística com o custo para o atingir, ou seja, vai expandir o nó A(3) antes do nó B(11).
--	--