

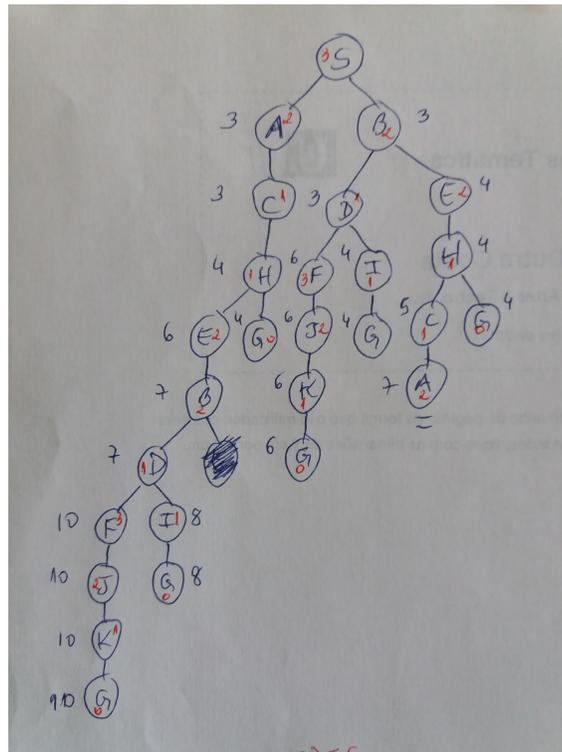
**Departamento de Ciência de Computadores - FCUP**  
**Primeiro Teste de Inteligência Artificial / Sistemas Inteligentes**  
**(Duração: 1 hora)**

Data: 05 de Abril de 2018

Considere o problema de mover um cavalo no jogo de xadrez num sub-retângulo do tabuleiro de dimensão 3x4, como mostrado na Figura 1. A posição inicial do cavalo é S e a final é G. Cada letra corresponde a uma célula no tabuleiro 3x4 e seus índices correspondem à heurística. Todas as transições têm custo 1. Tomando como base a Figura 1 responda às perguntas 1 a 4 abaixo, assumindo o seguinte:

- os algoritmos não geram caminhos com ciclos
- os nós são selecionados em ordem alfabética quando o algoritmo pode explorar qualquer dos sucessores

A árvore gerada para o grafo da Figura 1 é mostrada na figura abaixo, onde os números a vermelho dentro de cada nó são as heurísticas (estimativas de distância de cada nó ao objetivo G) e os números ao lado de cada nó, em azul, correspondem à avaliação usada pelo A\*:  $f(n)=g(n)+h(N)$ . Exemplos de buscas com a ordem de nós visitados podem ser encontrados nos slides das aulas teóricas: [http://www.dcc.fc.up.pt/~ines/aulas/1718/IA/buscas\\_ao\\_informadas.pdf](http://www.dcc.fc.up.pt/~ines/aulas/1718/IA/buscas_ao_informadas.pdf), slide 28.



1) Qual é a ordem de visita dos nós se estivesse utilizando:

- (A) IDFS    (B) BCU (Busca de Custo Uniforme)    (C) Greedy    (D) A\*  
(E) IDA\*

**IDFS:**

S  
 S-A-B  
 S-A-C-B-D-E  
 S-A-C-H-B-D-F-I-E-H  
 S-A-C-H-E-G

**BCU:**

S-A-B-C-D-E-H-F-I-H-E-G

**Greedy:**

S-A-C-H-G

**A\*:**

S-A-B-C-D-E-H-G

**IDA\*:**

f=3: S-A-C-H-B-D-F-I-E  
 f=4: S-A-C-H-E-G

**Após revisão:** no caso de **BCU** também aceitei as respostas S-A-B-C-D-E-H-F-I-G (não seguindo ordenação total alfabética e usando o algoritmo que está no livro, página 84, que não adiciona um nó repetido se o seu custo for maior ou igual ao que já existe na fila de prioridades), S-A-B-C-D-E-F-H-I-G (obdecendo ordem total alfabética). No caso do **A\***, também aceitei: S-A-B-C-D-E-H-I-G

2) Quais são os valores de  $f(E)$  e  $f(F)$  ao utilizar a estratégia de busca  $A^*$ ?

$f(E)=4$ ,  $f(F)=6$

3) Qual é o custo real da solução ótima?

$f^*=4$

4) Escolha um dos nós do grafo e modifique sua heurística de forma que esta passe a ser não admissível. Qualquer valor de heurística que ultrapasse o custo real ótimo daquele nó ao objetivo poderia ser usado. Por exemplo  $h(K)=2$ . A distância **real** mais curta do nó K até o nó objetivo G é 1, portanto ao atribuir o valor de heurística 2 a este nó, esta heurística passa a ser não admissível (estimativa de distância passa a ser maior do que o custo real para chegar de K à G).

5) Na configuração de jogo 4-em-linha da Figura 2, qual é o valor da função utilidade, assumindo que o X é o símbolo do computador e que existem as seguintes regras para segmentos de 4 peças: (a) qualquer sequência de 3 peças do computador e nenhuma do adversário vale 50 pontos, (b) qualquer sequência de 2 peças do computador e nenhuma do adversário vale 10 pontos, (c) uma sequência com uma peça do computador e nenhuma do adversário vale 1 ponto, (d) uma sequência com peças misturadas vale zero.

X linhas: 10

O linhas: 1

X colunas: 10 + 1 + 1

O colunas: 1+1+1+10

X diagonais: 1+1+1+1+1

O diagonais: 1+1+10+1+1+1

Total X: 27

Total O: 29

Utilidade = -2

6) No algoritmo de *simulated annealing* uma probabilidade é usada para continuar a busca pela solução quando o custo do nó sucessor é pior do que o custo do nó pai. Se a probabilidade de escolher um nó com custo pior for muito baixa ou próxima de zero, o *simulated annealing* reduz-se a outra estratégia. Qual delas?

Se a probabilidade de escolher um nó com função de avaliação pior do que o pai for próxima de zero, o *simulated annealing* reduz-se ao algoritmo **hill climbing**, ou seja, a busca termina.

```
function SA(problem,schedule) return a solution state
  current <- MAKE_NODE(INITIAL_STATE[problem])
```

```

for t <- 1 to infinito do
  T <- schedule(t)
  if T = 0 then return current
  next <- sucessor de current selecionado
             aleatoriamente
  deltaE <- value[next] - value[current]
  if deltaE > 0 then current <- next
  else RETURN CURRENT % INTERRUPTS THE SEARCH WITH BEST CURRENT
             SO FAR
endif
endfor

```

7) Seja a árvore de jogo da Figura 3. Responda às perguntas abaixo:

(A) Por que num jogo com adversários não é recomendável utilizar um algoritmo “greedy” que sempre maximiza as jogadas do jogador da vez a cada nível?

Por que um algoritmo “greedy” pode fazer o computador seguir caminhos que não serão seguidos pelo adversário. Ou seja, o algoritmo vai sempre tentar maximizar a sua chance de vitória esperando o pior cenário possível de jogada, assumindo que o adversário vai querer fazer a melhor jogada. Portanto, muitas vezes, no momento da jogada do adversário, o computador vai escolher jogadas que não são as melhores para ele, tentando minimizar as suas chances de derrota.

(B) Quais são os nós não explorados pelo corte alpha-beta na árvore da Figura 3?

A subárvore J e, conseqüentemente, seus filhos, O e P, nunca serão criados. Também o nó L nunca será criado.

8) Um problema de satisfação de restrições é representado pelo grafo da Figura 4, onde as variáveis são representadas nos vértices do grafo e as restrições pelas arestas. Assuma que o algoritmo escolhe a variável  $V_4$  e um determinado valor para esta variável. Qual seria o próximo passo do algoritmo, imediatamente após estas duas ações, para resolver este problema?

As arestas do grafo representam as restrições que envolvem as variáveis representadas nos nós. Portanto, se alguma modificação for feita no domínio da variável  $V_4$ , o algoritmo precisa verificar se o domínio da variável  $V_6$  também precisa ser modificado. A escolha da próxima variável para escolher valor somente acontece após o domínio de  $V_6$  ter sido alterado. E a próxima variável a ser escolhida não necessariamente será a variável  $V_6$  porque depende da heurística utilizada pelo algoritmo para a escolha da próxima variável.

S <sub>3</sub>	H <sub>1</sub>	D <sub>1</sub>	K <sub>1</sub>
I <sub>1</sub>	J <sub>2</sub>	A <sub>2</sub>	E <sub>2</sub>
C <sub>1</sub>	B <sub>2</sub>	G <sub>0</sub>	F <sub>3</sub>

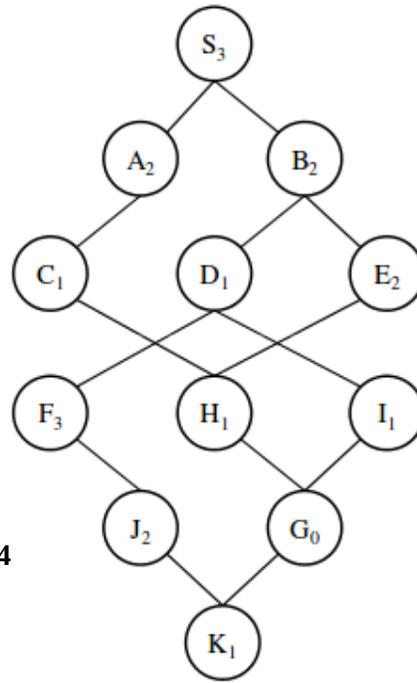


Figura 1: Grafo para as perguntas 1 a 4

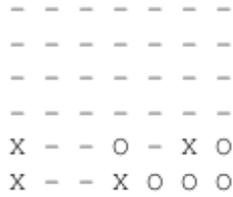


Figura 2: Configuração 4-em-linha para a pergunta 5

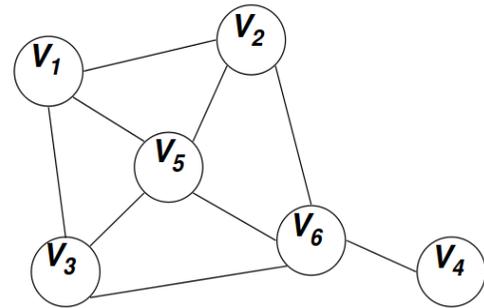


Figura 4: Grafo CSP para a pergunta 8

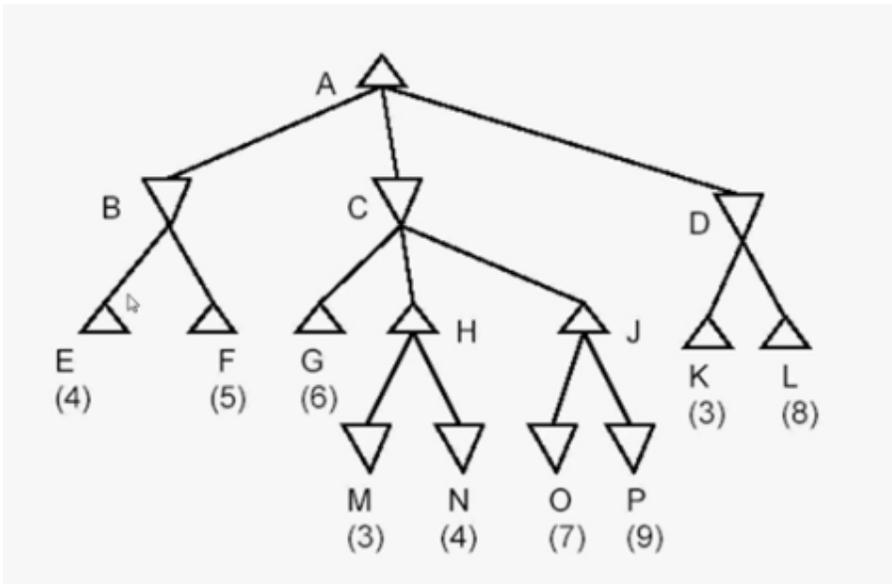


Figura 3: Árvore de jogo para a pergunta 7