

Universidade Federal de Ouro Preto
BCC 325 - Inteligência Artificial
Prova 1

Prof. Rodrigo Silva

1 Observações

1. É obrigatória a entrega do código fonte dos algoritmos de busca em espaços de estados da última lista de exercícios.
2. As linhas de código devem estar numeradas.
3. A falha em cumprir as exigências acima acarretará em nota 0.

2 Questões

1. Quais critérios devemos utilizar para comparar a inteligência de agentes? Como estes critérios podem ser medidos?
2. Dado um problema de busca em espaços de estados P , responda:
 - (a) Quais características P deve ter para que seja recomendado usar um algoritmo de busca em profundidade?
 - (b) Quais características P deve ter para que seja recomendado usar um algoritmo de busca em largura?
 - (c) Quais características P deve ter para que seja recomendado usar o algoritmo de menor custo primeiro?
 - (d) Quais características P deve ter para que seja recomendado usar o algoritmo A^* ?
3. Considere a sua implementação dos algoritmos de busca em largura e busca em profundidade:
 - (a) Em relação à implementação, qual a diferença entre os dois? Indique em quais linhas de código está esta diferença.
 - (b) Como funciona o algoritmo de poda de ciclos? Caso você tenha implementado, indique em quais linhas de código a poda está implementada. Caso contrário, indique como o seu código deve ser alterado para a inclusão da poda.
4. Considere a sua implementação do algoritmo A^* .
 - (a) Indique as linhas de código que implementam a função heurística.
 - (b) A função heurística que você implementou é admissível? Explique.
 - (c) Discuta vantagens e desvantagens do A^* em relação ao Branch and Bound.
5. Considere o problema das 3-rainhas (problema das n rainhas com $n = 3$):
 - (a) Qual o tamanho do espaço de busca? Ou seja, no pior caso, quantas soluções candidatas podem ser geradas?
 - (b) Represente este problema como uma rede de restrições?

6. Considere o problema a seguir:

$$\begin{aligned}X &= \{K, Y, Z\} \\D &= \{\{1, 2, 3, 4\}, \{1, 2, 3, 4\}, \{1, 2, 3, 4\}\} \\C &= \{K < Y, Y > Z\}\end{aligned}$$

- (a) Demonstre a execução do algoritmo de consistência de arcos, GAC (Generalized Arc Consistency Algorithm) para o problema da questão anterior.
- (b) Quais conclusões pode ser tiradas após a execução do GAC, no geral? O que podemos concluir após a execução do GAC para este problema?
- (c) Após a execução do GAC, qual o tamanho do espaço de busca?

```
1 GAC(<X,D,C>, {<X,c> | c in C and X in scope(c)}):  
2  
3 def GAC(<X,D,C>, to_do):  
4     while to_do:  
5         select and remove <X, c> from to_do  
6         let {Y1,...,Yk} = scope(c) \ {X}  
7         new_domain = {x|x in D(X) and exists y1 in D(Y1),...,yk in D(Yk)  
8                     such that c(X=x,Y1=y1,...Yk=yk)==True}  
9         if new_domain not equal D(X):  
10             to_do = to_do union {<Z,_c>| {X,Z} in scope(_c), _c!=c, Z!=X}  
11             D(X) = new_domain  
12     return D
```

Figure 1: Algoritmo GAC