

INE5430 - Inteligência Artificial

Métodos de Busca

Entrega: 17 de maio de 2019 (até 23:55h via Moodle)

O objetivo deste Exercício-Programa (EP) é implementar métodos de buscas vistos em sala de aula. Vamos simular um agente em uma mina de ouro. O agente possui uma bateria que limita a sua quantidade de movimentos. Por outro lado, a perspectiva de sair carregado de ouro faz com que valha a pena percorrer a mina e gastar uma certa quantidade de bateria.

1 Definição do problema

Vamos programar agentes baseados em buscas para capturar ouro em uma mina. Cada agente consegue se movimentar pela mina, mas isso gasta sua bateria que é muito cara e não é recarregável. Os movimentos do agente são **ir para a esquerda (E)**, **ir para a direita (D)**, **ir para cima (C)**, **ir para baixo (B)**. O agente pode também **pegar o ouro (PO)**. O objetivo principal de cada agente é maximizar a quantidade de ouro capturado e minimizar as perdas de energia de sua bateria.

A mina será considerada na entrada dos dados. Ela será representada por uma matriz M , com n linhas e n colunas. Cada posição (i, j) da mina pode significar:

- uma posição com obstáculo, se $M[i][j] = 1$;
- um posição com ouro, se $M[i][j] = *$;
- uma posição livre, se $M[i][j] = 0$.

Exemplo de entrada:

```
8
0 0 0 0 1 0 0 *
```

0	0	0	0	1	0	0	*
1	1	1	0	1	0	1	1
0	0	0	0	1	0	1	0
*	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0
1	1	1	0	1	1	1	0
0	0	1	0	0	*	1	0
0	0	0	0	0	0	1	*

Nesta mina, há uma expectativa de $n/2$ posições (não necessariamente alcançáveis) com ouro. A bateria do agente começa com $n^{1.5}$ pontos. Se um agente consegue se movimentar de uma posição para outra, então consideramos que houve uma perda de 1 ponto na sua bateria. Cada ouro capturado significa uma possível compra de cinco baterias novas (com $n^{1.5}$ pontos) para o agente. É claro que o agente fica impossibilitado de se movimentar quando a bateria acaba.

2 O que deve ser entregue?

É esperado a entrega de um *Makefile*, de programas que implementam *agentes* e de um relatório simples comparando **as pontuações obtidas** pelos diferentes métodos de busca e **o número de estados examinados** por cada agente.

Importante: Exemplos de entradas diferentes da descrita neste texto devem ser avaliadas.

O arquivo *Makefile* (<https://www.gnu.org/software/make/manual/make.html>) deve ser preparado para compilar e executar todos os agentes. Haverá desconto na nota se não houver um *Makefile* funcionando apropriadamente.

Os agentes resolvem problemas através de métodos de busca (um agente por método de busca). O prazo para a entrega é dia 17 de maio (até 23:55h via Moodle).

Os métodos que devem ser considerados são (pelo menos):

- Métodos de busca cega
 - Busca em profundidade limitada
 - Qualquer outro método de busca cega
- Métodos de busca informada (com heurísticas)
 - Busca A* (com heurística admissível)
 - Qualquer outro método de busca informada (com heurística livre)

3 Entrada dos dados para cada agente

Cada programa deve ler um arquivo que contém a mina como descrito anteriormente (a primeira linha do arquivo é um número natural $n \leq 50$ e as n próximas linhas contêm n caracteres 0, 1 ou *, separados por um espaço). O estado inicial é dado pela matriz e o agente no canto superior esquerdo que sempre terá o caracter 0.

4 Saída dos dados para cada agente

A saída deve conter uma sequência de ações escolhidas pelo agente, a pontuação obtida e o número de estados explorados. As ações possíveis são: E, D, B, C e PO (Pegar Ouro).

Obs.: A ação PO não consome bateria.

Observações importantes:

1. Os programas podem ser escritos em C (compatível com compilador gcc versão 7.3.0), C++ (compatível com compilador g++ versão 7.3.0), Java (compatível com compilador javac versão 11.0.2) ou Python (compatível com versão 3.6.7) e deve ser compatível com Linux/Unix.
2. Exercícios-Programas atrasados **não** serão aceitos.
3. Programas com *warning* na compilação terão diminuição da nota.
4. É importante que seu programa esteja escrito de maneira a destacar a estrutura do programa.
5. O relatório e o programa devem começar com um cabeçalho contendo pelo menos o seu nome.
6. Coloque comentários em pontos convenientes do programa, e faça uma saída clara.
7. A entrega do Exercício-Programa deverá ser feita no Moodle.
8. O Exercício-Programa é individual. Não copie o programa de outra pessoa, não empreste o seu programa para outra pessoa, e tome cuidado para que não copiem seu programa sem a sua permissão. Todos os programas envolvidos em cópias terão nota igual a ZERO.