MAC0425/5739 - Inteligência Artificial – Exercício Programa 1

Data de entrega: 27 de Setembro de 2013

O objetivo deste exercício-programa é implementar algumas técnicas de busca vistas em aula e ver sua aplicabilidade em casos mais gerais.

Mina de Ouro

Vamos simular uma mina de ouro. O ambiente dentro da mina não é muito seguro, então queremos ficar o mínimo possível embaixo da terra. Por outro lado, a perspectiva de sair carregado de ouro faz com que valha a pena correr algum risco. Vamos programar um agente baseado em busca que tem como objetivo principal maximizar os ganhos com o ouro, isto é, recolher o máximo possível de ouro com um mínimo de movimentos.

A mina será simulada por uma matriz n x n, contendo 1 em posições onde há um obstáculo, * nas posições onde há ouro e 0 nas posições livres. Nesta mina, há n/2 pepitas de ouro. Cada passo do agente custa 1 ponto. Cada pepita de ouro significa um ganho de 4n pontos. Ações possíveis para o agente são: Direita (D), Esquerda (E), Cima (C), Baixo (B) e Pega ouro (P).

O estado inicial do problema é dado pela matriz de ambiente e o agente no canto superior esquerdo. A meta do agente é percorrer a mina, coletar a quantidade mais lucrativa de ouros e retornar a sua posição original.

Vocês devem implementar:

1. Um ambiente de avaliação, que recebe as ações do agente e calcula sua pontuação. O ambiente é acessível e determinístico.
2. Um agente resolvedor de problemas.
3. Pelo menos três estratégias de busca: largura, profundidade (com limite) e A*.

Estrutura do programa:

O programa pode ser escrito em Java ou C/C++ e deve apresentar uma estrutura clara da separação entre ambiente e agente. Seu programa pode ser composto de quantas classes/módulos você julgar necessário, mas deve conter os seguintes arquivos básicos:

> `Main.java` (`main.c`): arquivo contento a implementação da função `main()` e de onde os demais módulos deverão ser chamados. Seu programa receberá como argumentos a localização do arquivo de teste.
>
> `Ambiente.java` (`ambiente.c`): implementação do ambiente mina. Sua descrição deve conter os elementos de uma descrição PEAS, ou seja, descrição do estado, dos sensores e atuadores, dos efeitos da atuação de uma agente no ambiente, cálculo da medida de desempenho, etc.
>
> `Agente.java` (`agente.c`): implementação do agente. Deve conter a implementação do programa agente, da percepção dos sensores, da interação com o ambiente (atuadores), etc.

Entrada:

O programa receberá dois argumentos como entrada. O primeiro é a localização do arquivo de teste que conterá a descrição de uma mina. O segundo argumento é a indicação da estratégia de busca que o agente deve usar.

```
java Main <arquivo_de_teste> <tipo_de_busca>
```

- Formato do arquivo de teste: primeira linha do arquivo contém um inteiro $1 \leq n \leq 50$ que indica o tamanho da mina; as n linhas seguintes descrevem uma matriz de caracteres $n \times n$ que representa: `0` posição livre, `1` posição com obstáculo e `*` posição com ouro.
- Estratégia de busca: indicada por uma única letra: `L` para busca em largura; `P` para busca em profundidade com limite; e `A` para A*.

Exemplo de chamada:

```
exemplo@ubuntu:~/EP1$ java Main mina.in L
```

Saída:

A saída do programa será o plano escolhido pelo agente para obtenção da meta. Você deve imprimir, na saída padrão, a pontuação total obtida e a sequência de ações escolhida pelo agente.

Relatório:

Você deve elaborar um pequeno relatório sobre o projeto. Dentre os itens abordados no relatório, você deve fornecer:

1. uma breve descrição das principais classes e funções de sua implementação;
2. descrição da heurística utilizada no A* e justificativa de sua escolha;
3. informações sobre testes e análise do desempenho das diferentes buscas.

Entregar

Um único arquivo zipado, contendo:

1. Código fonte da implementação;
2. Um relatório, em pdf.

Exemplo de teste:

```
8
0000100*
11101011
00001010
*0111000
00000000
11101110
00100*10
0000001*
```

Exemplo de saída:

```
70 pontos
D D D B B E E E B P D B D D D D C C C C D D P E E B B B D D B
B B B P C C C E E E E B B D D P E E C C E E C C D D C C E E E
```