Matéria: Inteligência Artificial

Professora: Márcia

Data: 04/09/2018

Busca em árvores:

* A raiz da árvore corresponde ao estado inicial;
* Reservar as opções não selecionadas porque a selecionada pode não conduzir à solução;
* O espaço de estados contém todos os estados válidos. A busca em árvore pode ocorrer repetições de estados.
* O nó da árvore pode conter:
  + Estado;
  + Nó-pai;
  + Custo Caminho;
  + Ações;
  + Profundidade;
* A borda é uma coleção de nós gerados, mas não expandidos.
* A forma como os nós são retirados da borda depende da estratégia de busca.
* OBS: Colocar o nó raiz na borda, se não é o estado objetivo, fazer uma expansão, colocando seu filhos na borda se não alcançar o estado objetivo e repetir até achar vasculhar todos nós. Não trata repetição.

OBS: Solução não é o estado final, e sim, o conjunto de estado até alcançar o objetivo.

Algoritmo de busca podem ser avaliados de acordo com alguns critérios:

1. Completeza: Encontra solução, se existe?
2. Otimização: Encontra solução ótima?
3. Complexidade de tempo: Quanto tempo para encontrar a solução ótima.
4. Complexidade de Espaço: Quanta memória é necessária para armazenar os nós.

Para analisar tempo e espaço, são utilizados três parâmetros:

* b : fator de ramificação e expressa o número que podem ser gerados a partir de qualquer nó.
* d : profundidade do nó objetivo mais raso.
* m : profundidade máxima da busca.

Buscas não informadas:

* Buscas deste tipo não têm informação adicional, como por exemplo, caminhos mais promissores.

Exemplos:

Busca em Largura

* Gera todos os filhos de um determinado nó. Além disso, só gera filhos do nível seguinte, se todos os nós do nível atual tiverem sido gerados.
* Avaliação:
  + Completa: Se b é finito, é completa.
  + Ótima: Se todos os custos das ações forem idênticos é ótima.
  + Tempo: Como no nível d tem um estado objetivo, no pior caso, todos os nós até este nível são gerados além de nós do nível d+1. Daí, o tempo é proporcional.

* + Espaço: Número máximo de nós sendo gerados. Logo, também é

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| b = 10 | d = 6 | 10⁶ | 1,1 seg | 1 GB |
| b = 10 | d = 12 | 10¹² | 13 dias | 1 Peta |

Busca em profundidade:

* Pode ser implementada recursivamente ou com pilha.
* Expande o nó mais recentemente gerado.
* Completa: Não é completa, pois pode entrar em “loop”em caminho que não contém solução.
* Ótima: Pode encontrar soluções não tão próximas à raiz, por isso é não ótima.
* Tempo: Pode ser O(bm), se considerar que b nós de cada dos m níveis sejam “gerados”. No pior caso, gera nós.
* Espaço em relação ao tempo, pode ocupar espaço proporcional a tiverem sido gerados

TAREFA: Implementar busca em largura e profundidade, permitindo repetições, em árvores.