



Anotações 3ª Aula

Métodos de Busca

Resolução do Problema como uma Busca no Espaço de Estados

- A maioria dos problemas interessantes de IA não dispõe de soluções algorítmicas.
- Historicamente, os primeiros problemas a serem estudados foram:
 - Prova automática de Teoremas;
 - Quebra-cabeças; e
 - Jogos.
- Apresentam características que os tornam bons candidatos para a pesquisa em IA.
 - São solucionáveis por seres-humanos e, neste caso, sua solução está associada à *inteligência*;
 - Formam classes de complexidade variável existindo desde instâncias triviais (por exemplo, o *jogo da velha*, no caso dos jogos) até instâncias extremamente complexas (*xadrez*).
 - São problemas de conhecimento *total*, isto é, tudo que é necessário para solucioná-los é conhecido, o que facilita sua formalização;
 - Suas soluções têm a forma de uma sequência de situações *legais* e as maneiras de passar de uma situação para outra são em número finito e conhecidas.

Diante da falta de solução algorítmica viável, o único método viável de solução possível é a **busca**.

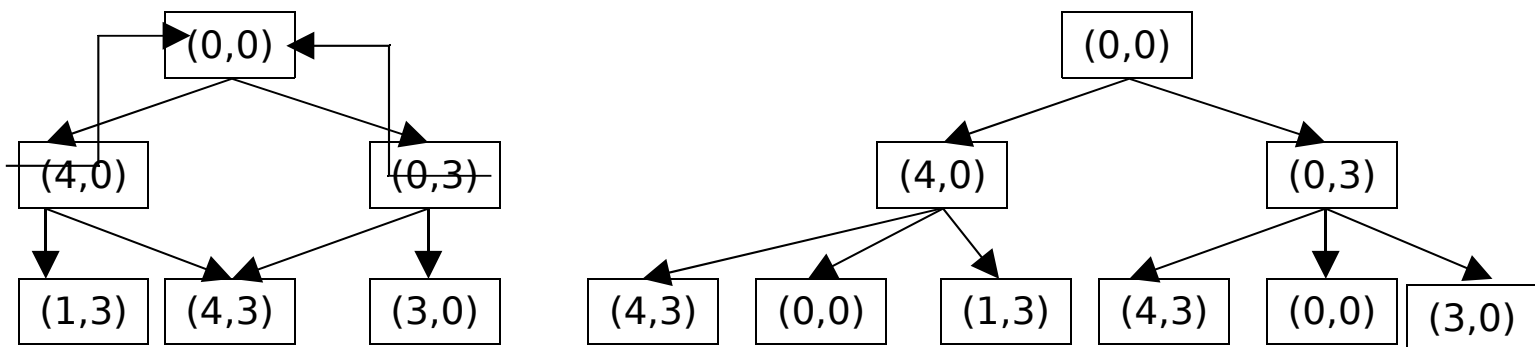
- De maneira geral, um problema de busca pode ser formalizado através da definição dos seguintes elementos:
 - Um conjunto de descrições chamado espaço de estados, onde cada elemento descreve uma situação possível do problema.
 - Um estado inicial que descreve a situação inicial do problema.
 - Um ou mais estados finais, isto é, a(s) situação(ões) que se deseja alcançar.

- Um conjunto de operadores, isto é, procedimentos que, dada a descrição de um estado, determinam todos os estados que podem ser alcançados a partir do estado dado.
 - Estes elementos constituem um Sistema de Produção.
- Sistemas de Produção podem ser utilizados para modelizar qualquer procedimento calculável.
- Exemplo: Jogo de Xadrez
 - Para construir um programa que possa jogar xadrez, teremos de especificar antes:
 - posição inicial;
 - regras que definem movimentos legais;
 - posições ou condições de vitória;
 - deixar implícito que a meta é ganhar e não apenas jogar.
 - A posição da partida pode ser descrita como uma matriz 8X8, em que cada posição contém um símbolo representando a peça correspondente em cada posição.
 - A posição inicial é representada pela matriz com as peças na posição oficial de abertura.
 - A meta é atingir qualquer posição de tabuleiro em que o oponente não tem um movimento legal e o rei está sob ataque.
 - Os movimentos legais fornecem o meio de ir do estado inicial para um estado meta.
 - Estes movimentos podem ser facilmente descritos como um conjunto de regras constituídas de duas partes:
 - *se* posição atual do tabuleiro, *então* posição seguinte do tabuleiro.
 - Teríamos de escrever uma regra distinta para cada uma das aproximadamente 10^{120} posições possíveis do tabuleiro.
 - nenhuma pessoa consegue fornecer um conjunto completo de tais posições;
 - nenhum programa pode manipular todas estas regras.

- É útil recorrer a estratégias apropriadas para reduzir o tamanho do espaço de busca sempre que isto for conveniente e possível.
- Para obter este efeito, costuma-se recorrer aos métodos heurísticos, os quais, para acelerar a busca da solução, valem-se de informações disponíveis que permitem explorar o espaço de busca com economia e eficiência.
- Resolver problemas difíceis exige com frequência o comprometimento da sistematicidade, passando ao largo da exigência de encontrar a melhor solução, embora, em geral, boas soluções sejam detectadas.

A introdução de idéias heurísticas melhora a eficiência de um processo de busca ao sacrificar idéias de perfeição.

Com estes elementos é possível construir uma **ÁRVORE DE BUSCA**, cujo nodo raiz está associado a um estado inicial e onde os sucessores de qualquer nodo são associados aos estados obtidos através da aplicação das regras (associadas ou não às heurísticas) sobre a descrição do estado associado ao nodo.



Árvores são mais simples para busca que os grafos. Primeiramente porque quando um novo nodo é gerado, podemos estar seguros que ele nunca foi visitado antes nem nunca será gerado depois.

● Estratégias de Busca

- **BUSCA ÀS CEGAS** (Blind Search ou Uniformed Search): Uma estratégia de busca é dita “CEGA” se ela não leva em conta informações específicas sobre o problema a ser resolvido.
 - Existem basicamente duas estratégias cegas para a construção e pesquisa em uma árvore de busca: Busca em Largura e Busca em Profundidade.

BUSCA EM LARGURA (ou Amplitude)

- Consiste em construir uma árvore de estados a partir do estado inicial, aplicando a cada momento, todas as regras possíveis aos estados do nível mais baixo, gerando todos os estados sucessores de cada um destes estados. Assim, cada nível da árvore é completamente construído antes que qualquer nodo do próximo nível seja adicionado à árvore.

BUSCA EM PROFUNDIDADE

- Procurar explorar completamente cada ramo da árvore antes de tentar o ramo vizinho.

- **A principal vantagem do algoritmo de busca em largura é que este encontra o MENOR caminho do nodo inicial até o nodo final mais próximo.**
- **O algoritmo de busca em profundidade não encontra necessariamente a solução mais próxima, mas pode ser MAIS EFICIENTE se o problema possui um grande número de soluções ou se a maioria dos caminhos pode levar a uma solução.**

— BUSCA HEURÍSTICA (Informed Search):

- Os métodos de busca vistos anteriormente fornecem uma solução para o problema de achar um caminho até um nó meta.
- Em muitos casos, a utilização destes métodos é impraticável devido ao número muito elevado de nós a expandir antes de achar uma solução.
- Como existem limites práticos de valores de tempo e espaço de armazenamento disponível para uso na busca, devemos procurar métodos alternativos mais eficientes.
- **Para muitos problemas, é possível estabelecer princípios ou regras práticas para ajudar a reduzir a busca.**
- Qualquer técnica usada para melhorar a busca depende de informações especiais acerca do problema em questão.
- Chamamos a este tipo de informação de INFORMAÇÃO HEURÍSTICA e os procedimentos de busca que a utilizam de MÉTODOS DE BUSCA HEURÍSTICA.

- A informação que pode compor uma informação heurística é o Custo do Caminho.
- O CUSTO DO CAMINHO pode ser composto pelo somatório de dois outros custos: O custo do caminho do estado inicial até o estado atual que está sendo expandido; e uma estimativa do custo do caminho do estado atual até o estado meta.

Uma boa heurística para a estratégia de busca no estado de espaços de um problema é a estratégia conhecida como SUBIDA DA ENCOSTA, também denominada “subida da montanha” ou “hill climbing”.

— SUBIDA DA ENCOSTA

- É a estratégia mais simples e popular. Baseada na Busca em Profundidade.
- É um método de busca local que usa a idéia de que o objetivo deve ser atingido com o menor número de passos.
- A idéia heurística que lhe dá suporte é a de que o número de passos para atingir um objetivo é inversamente proporcional ao tamanho destes passos.
- Empregando uma ordenação total ou parcial do conjunto de estados, é possível dizer se um estado sucessor leva para mais perto ou para mais longe da solução. Assim o algoritmo de busca pode preferir explorar em primeiro lugar os estados que levam para mais perto da solução.
- Há duas variações do método: a Subida de Encosta SIMPLES e a Subida de Encosta PELA TRILHA MAIS ÍNGREME.
- SUBIDA DE ENCOSTA SIMPLES: Vai examinando os sucessores do estado atual e segue para o primeiro estado que for maior que o atual.
- SUBIDA DE ENCOSTA PELA TRILHA MAIS ÍNGREME: Examina TODOS os sucessores do estado atual e escolhe entre estes sucessores qual é o que está mais próximo da solução.

— BUSCA PELA MELHOR ESCOLHA

- Também conhecido como Algoritmo de Busca O-MELHOR-PRIMEIRO, “Best First”, ou A*.
- É um método de busca que procura otimizar a solução, considerando todas as informações disponíveis até aquele instante, não apenas as da última expansão.
- Todos os estados abertos até determinado instante são candidatos à expansão.
- Combina, de certa forma, as vantagens tanto da busca em largura como em profundidade
- Busca onde o nó de menor custo “aparente” na fronteira do espaço de estados é expandido primeiro.
- Função de avaliação:
 - $f(n) = g(n) + h(n)$
 - $g(n)$ = distância de n ao nó inicial
 - $h(n)$ = distância estimada de n ao nó final
- A* expande o nó de menor valor de f
- $g(n)$ normalmente recebe o valor 1, se o objetivo for encontrar o menor caminho, ou o valor 0, se o objetivo for encontrar uma solução qualquer.
- $h(n)$ não é conhecido uma vez que a meta ainda não foi atingida, logo ele é uma estimativa.

— Exemplo:

Considere o problema dos missionários e canibais e as seguintes funções:

$g(n)$ = distância do nodo inicial até o nodo n , ou seja, a cada movimentação do barco, este número é incrementado de 1.

$h(n)$ = número de pessoas que faltam para serem transportadas para a margem oposta.

- define que um estado está mais perto da solução se há mais pessoas na margem oposta.

- se há o mesmo número de pessoas nas duas margens, então se o barco está na margem original é melhor do que se o barco está na margem oposta.

$$h(n) = h'(6.5 - (m^2 + c^2 + b/2))$$

— Admissibilidade de A*

- Diz-se que um método de busca é ADMISSÍVEL se ele sempre encontra uma solução e se esta solução é a de menor custo.
- A busca em largura é admissível. O mesmo não ocorre com a busca em profundidade.
- Se um programa de busca heurística está em um estado n e que se conheça exatamente o custo mínimo de n até a meta. $h^*(n)$
- É possível provar que se $h(n)$ for menor ou igual a $h^*(n)$ para qualquer n , o programa que está realizando a busca é admissível, isto é, sempre encontra o caminho mais curto até a meta.

— TÊMPERA SIMULADA

- É adequado a problemas nos quais a subida de encosta encontra muitos platôs e máximos locais.
- Não utiliza backtracking e Não garante que a solução encontrada seja a melhor possível.
- Pode ser utilizado em problemas NP-completos.

É inspirado no processo de têmpera do aço. Temperaturas são gradativamente abaixadas, até que a estrutura molecular se torne suficientemente uniforme.

- O que o algoritmo de têmpera simulada faz é atribuir uma certa “energia” inicial ao processo de busca, permitindo que, além de subir encostas, o algoritmo seja capaz de descer encostas e percorrer platôs se a energia for suficiente.
- A energia é diminuída ao longo do tempo, o que faz com que o processo vá se estabilizando em algum máximo que tem maior chance de ser solução do problema.

