

Inteligência Artificial

Métodos de resolução de problemas

Técnicas de busca

Prof. Paulo Martins Engel

Resolução de problemas por busca

Durante o *Turing Award Lecture* (1976), Newell e Simon sustentam que a atividade inteligente, quer seja humana ou de uma máquina, é alcançada pelo uso de:

- *Padrões simbólicos* para representar aspectos significativos de um domínio de problema.
- *Operações* sobre estes padrões para gerar soluções potenciais dos problemas.
- *Busca* para selecionar uma solução entre estas possibilidades.

As questões de representação do conhecimento e busca são o núcleo da pesquisa da IA clássica.

2

Resolução de problemas por busca

- Representação do estado (uma configuração do problema)
- Representação das ações: operadores
- Busca: processo de examinar as diversas opções de seqüências de ações possíveis que podem levar ao estado objetivo, escolhendo a melhor seqüência.
- Um algoritmo de busca recebe como entrada um *problema* e retorna uma *solução* na forma de uma seqüência de ações.

3

Busca Cega

- A busca cega é a estratégia menos inteligente de todas.
- A idéia tem origem no que é conhecido como *Algoritmo do Museu Britânico*: se você puser um chimpanzé ou o que quer que seja na frente de um teclado, então algum dia ele será capaz de gerar todos os livros do Museu Britânico!
- A busca cega é obviamente um procedimento não sistemático e ineficiente.
- Para aumentar a eficiência deve-se acrescentar algum tipo de *estrutura de controle* à geração de alternativas.

4

Busca em Grafo

- Se representarmos as várias soluções candidatas como nós num grafo, nós podemos visualizar facilmente diferentes tipos de controle.
- Num grafo de espaço de estados, cada nó representa um estado legal.
- Um elo de um nó N para um nó M denota o fato que a aplicação de um certo gerador (operador) ao estado N mapeia este estado para o estado M .
- Neste caso, diz-se que M é *alcançável diretamente* do estado N .

5

Grafo de Estados

- Podem existir outros estados, além de M , que são diretamente alcançáveis de N .
- O número destes estados é chamado de *fator de ramificação*.
- Graficamente, estas alternativas são representadas como um conjunto de elos indo de N para o conjunto de estados diretamente alcançáveis, $\{M_1, M_2, \dots, M_i, \dots, M_j\}$.
- Tipicamente, muitos estados diferentes podem ser diretamente alcançados de cada um dos estados M_i .
- Por ex., L_1 pode ser diretamente alcançável de M_1 .
- Diz-se que L_1 é alcançável de N (*caminho* de N a L_1)

6

Busca Sistemática

- Um método de busca sistemático é aquele que organiza eficientemente a geração e a busca dos diversos caminhos representados num grafo de estados.
- Os nós que são diretamente alcançáveis de um outro nó N são chamados de nós *filhos* de N e o nó N é o nó *pai*.
- Se dois nós têm o mesmo pai eles são nós *irmãos*.
- Nós que estão no caminho até um nó M são chamados de *ancestrais* de M (M é *descendente* de um destes nós).
- Um grafo *radicado* tem um único nó (a raiz) do qual se originam todos os caminhos do grafo. (não tem pai)
- Um nó *folha* é um nó terminal, que não tem filhos.

7

Exemplo de representação por espaço de estados

Exemplo: jogo dos 8

$9!/2 = 181.440$
estados

2	8	3
1	6	4
7		5

8

Jogo dos 8



9

Estados e Operadores

- **Estado:** uma configuração particular das peças
- **Operador:** transforma um estado em outro

A configuração inicial e o objetivo do jogo são os estados inicial e final.

10

Jogo dos 8: operadores

- mover a peça 1 para cima, baixo, direita, esquerda
- mover a peça 2 para cima, baixo, direita, esquerda
- mover a peça 3 para cima, baixo, direita, esquerda
-
- mover a peça 8 para cima, baixo, direita, esquerda

total de 32 operadores

11

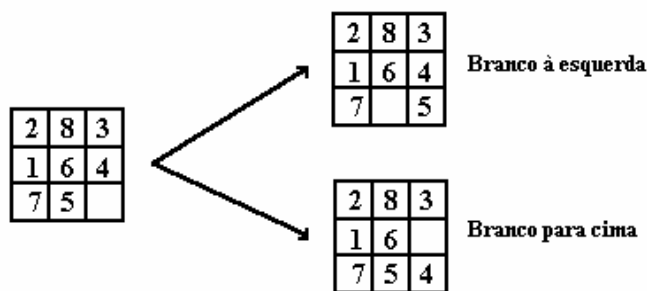
Jogo dos 8: operadores

- branco para cima
- branco para baixo
- branco para a direita
- branco para a esquerda

total de 4 operadores

12

Jogo dos 8: operadores



13

Representação do problema

A representação de um problema deve conter:

- forma de representar os estados
- descrição dos estados inicial e objetivo
- descrição dos operadores

14

Exemplo de representação: listas

- Estado inicial: $[2,8,3,1,6,4,7,0,5]$
- Estado objetivo: $[1,2,3,8,0,4,7,6,5]$
- exemplos de operadores

$[a,b,c,d,e,f,g,h,0] \rightarrow [a,b,c,d,e,f,g,0,h]$ p/ esquerda

$[a,b,c,d,e,f,g,h,0] \rightarrow [a,b,c,d,e,0,g,h,f]$ p/ cima

total de 24 casos possíveis

15

$[a,b,c,d,e,f,g,h,0] \rightarrow [a,b,c,d,e,f,g,0,h]$ E
 $[a,b,c,d,e,f,g,h,0] \rightarrow [a,b,c,d,e,0,g,h,f]$ C
 $[a,b,c,d,e,f,g,0,h] \rightarrow [a,b,c,d,e,f,0,g,h]$ E
 $[a,b,c,d,e,f,g,0,h] \rightarrow [a,b,c,d,0,f,g,e,h]$ C
 $[a,b,c,d,e,f,g,0,h] \rightarrow [a,b,c,d,e,f,g,0,h]$ D
 $[a,b,c,d,e,f,0,g,h] \rightarrow [a,b,c,0,e,f,d,g,h]$ C
 $[a,b,c,d,e,f,0,g,h] \rightarrow [a,b,c,d,e,f,g,0,h]$ D
 $[a,b,c,d,e,0,f,g,h] \rightarrow [a,b,c,d,0,e,f,g,h]$ E
 $[a,b,c,d,e,0,f,g,h] \rightarrow [a,b,0,d,e,c,f,g,h]$ C
 $[a,b,c,d,e,0,f,g,h] \rightarrow [a,b,c,d,e,h,f,g,0]$ B
 $[a,b,c,d,0,e,f,g,h] \rightarrow [a,b,c,0,d,e,f,g,h]$ E
 $[a,b,c,d,0,e,f,g,h] \rightarrow [a,0,c,d,b,e,f,g,h]$ C
 $[a,b,c,d,0,e,f,g,h] \rightarrow [a,b,c,d,e,0,f,g,h]$ D
 $[a,b,c,d,0,e,f,g,h] \rightarrow [a,b,c,d,g,e,f,0,h]$ B
 $[a,b,c,0,d,e,f,g,h] \rightarrow [0,b,c,a,d,e,f,g,h]$ C
 $[a,b,c,0,d,e,f,g,h] \rightarrow [a,b,c,d,0,e,f,g,h]$ D
 $[a,b,c,0,d,e,f,g,h] \rightarrow [a,b,c,f,d,e,0,g,h]$ B
 $[a,b,0,c,d,e,f,g,h] \rightarrow [a,0,b,c,d,e,f,g,h]$ E
 $[a,b,0,c,d,e,f,g,h] \rightarrow [a,b,e,c,d,0,f,g,h]$ B
 $[a,0,b,c,d,e,f,g,h] \rightarrow [0,a,b,c,d,e,f,g,h]$ E
 $[a,0,b,c,d,e,f,g,h] \rightarrow [a,b,0,c,d,e,f,g,h]$ D
 $[a,0,b,c,d,e,f,g,h] \rightarrow [a,d,b,c,0,e,f,g,h]$ B
 $[0,a,b,c,d,e,f,g,h] \rightarrow [a,0,b,c,d,e,f,g,h]$ D
 $[0,a,b,c,d,e,f,g,h] \rightarrow [c,a,b,0,d,e,f,g,h]$ B

16

Exemplo de representação: matrizes

2 8 3	1 2 3
1 0 4	8 0 4
7 6 5	7 6 5

Estado Inicial Estado Objetivo

Exemplo de operador:

a b 0	→	a 0 b	
c d e		c d e	esquerda
f g h		f g h	

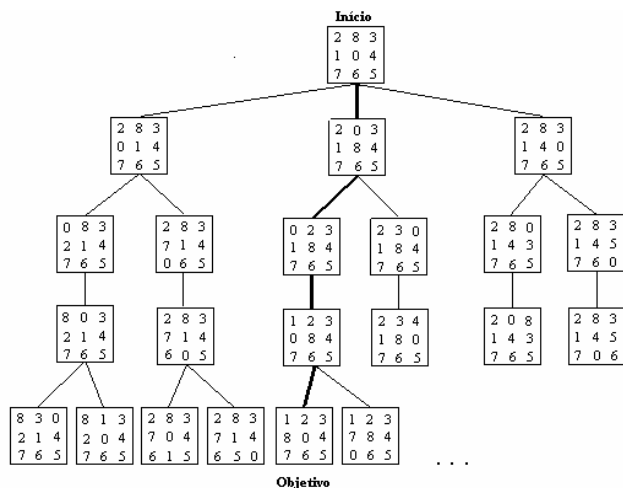
17

Grafo de estados

- nó: representa um estado
- arco: representa um operador

18

Grafo de estados: exemplo



19

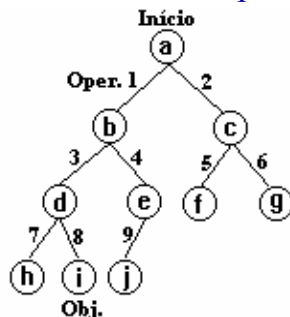
Métodos de busca em grafos de estado

- Estratégias quanto à direção de busca:
 - Percorre-se o grafo até encontrar o estado objetivo (*busca guiada por dados* ou *encadeamento progressivo*).
 - Começar pelo objetivo em direção aos fatos (*busca guiada por objetivo* ou *encadeamento regressivo*).
- Tipos de busca:
 - busca sistemática
 - busca heurística

20

Busca em largura ou amplitude

- Para cada estado são aplicados todos os operadores possíveis - busca por nível



- Ordem: operadores 1, 2, 3, 4, 5, 6, 7, 8, 9

21

Admissibilidade

- Como a busca em amplitude examina todos os nós de um nível antes de passar para o próximo nível, ela sempre encontra o *caminho mais curto* para um nó objetivo.
- Um algoritmo de busca é *admissível* se houver a garantia de encontrar um caminho mínimo até uma solução sempre que tal solução exista.
- A busca em amplitude é um algoritmo admissível.
- Entretanto, se houver um fator de ramificação desfavorável (média alta de estados descendentes, **B**), a explosão combinatória pode impedir que o algoritmo encontre uma solução usando o espaço disponível.
- A utilização do espaço da busca em amplitude é uma função exponencial da profundidade **n**: $B^n \Rightarrow$ problema em soluções profundas

22

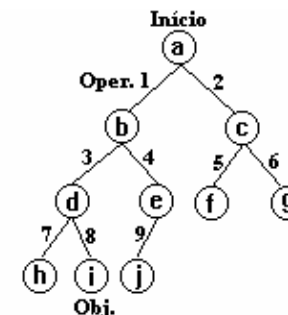
Busca em profundidade

- A busca em profundidade avança rapidamente num espaço de busca *profundo*.
- Se soubermos que o caminho-solução será longo, a busca em profundidade não perderá tempo examinando um grande número de estados “superficiais”.
- Por outro lado a busca em profundidade pode se “perder” nas profundezas de um grafo, **não encontrando o caminho mais curto** até um objetivo, ou mesmo ficando presa num caminho infinitamente longo que não leva a um objetivo.
- A busca em profundidade é muito mais eficiente para busca com muitos ramos, porque em cada nível ela retém apenas os filhos de um único estado.
- A utilização de espaço é linear com a profundidade: $B \times n$

23

Busca em Profundidade

Examina-se os nós sempre em direção às folhas, afastando-se da raiz.



Ordem: operadores 1, 3, 7, 8, 4, 9, 2, 5, 6

24

Busca heurística

- O processo de busca é dirigido através de *informações* que auxiliam a seleção dos operadores
- Heurísticas são formalizadas como regras para escolher aqueles ramos que tem a maior chance de levarem a uma solução aceitável para o problema.
- Função heurística, $h(n)$: estima a distância entre n e o objetivo.
- Para favorecer soluções em caminhos mais curtos, introduz-se um termo $g(n)$ que mede o comprimento real do caminho de um estado n qualquer até o estado inicial.
- Função de avaliação: $f(n) = g(n) + h(n)$
- Na busca pela melhor escolha, cada estado é rotulado com o seu peso heurístico $f(n)$.

25

Busca heurística - exemplo

- exemplo: jogo dos 8
Estado inicial: [2,8,3,1,0,4,7,6,5]
Estado objetivo: [1,2,3,8,0,4,7,6,5]
Soma das diferenças: $1+6+0+7+0+0+0+0+0 = 14$
- é escolhido o operador que gerar a menor diferença depois de aplicado
- O objetivo é alcançado quando a soma das diferenças for igual a zero.

26

Busca heurística - exemplo

Estado inicial: [2,8,3,1,0,4,7,6,5]

Estado objetivo: [1,2,3,8,0,4,7,6,5]

sucessores possíveis soma das diferenças

a) [2,0,3,1,8,4,7,6,5] $1+2+0+7+8+0+0+0+0=18$

b) [2,8,3,0,1,4,7,6,5] $1+6+0+8+1+0+0+0+0=16$

c) [2,8,3,1,4,0,7,6,5] $1+6+0+7+4+4+0+0+0=22$

d) [2,8,3,1,6,4,7,0,5] $1+6+0+7+6+0+0+6+0=26$

Seria escolhida a jogada b)

27

Busca pela melhor escolha

- Busca heurística
- em cada etapa escolhemos o nó mais promissor gerado até o momento
- utiliza uma função de avaliação que retorna o **custo** de se chegar a uma solução (quanto menor melhor)
- o método A^* é derivado deste

28

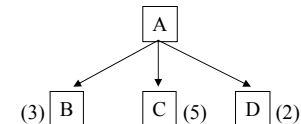
Buscas Admissíveis

- Definindo a função de avaliação $f^*(n) = g^*(n) + h^*(n)$
 $g^*(n)$ custo do caminho mais curto do nó inicial até n
 $h^*(n)$ custo real do menor caminho até o objetivo.
- Pode-se provar que a busca pela melhor escolha utilizando f^* é admissível.
- Além disso, se utilizarmos estimativas $g(n)$ e $h(n)$ para $g^*(n)$ e $h^*(n)$, desde que $g(n) \geq g^*(n)$ e $h(n) \leq h^*(n)$, então a estratégia de busca resultante também é admissível (algoritmo A^*).

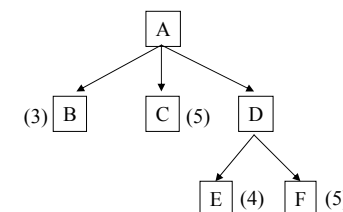
29

Exemplo

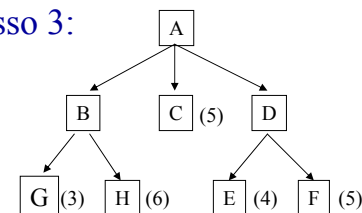
- Passo 1:



- Passo 2:



- Passo 3:



30