

Inteligência Artificial

Busca

Profa. Dra. Jaqueline Brigladori Pugliesi

1

Problemas x IA

- ▶ A maioria dos problemas interessantes de IA dispõe de difíceis soluções algorítmicas.
- ▶ Historicamente, os primeiros problemas a serem estudados foram:
 - Prova automática de Teoremas;
 - Quebra-cabeças;
 - Jogos.
- ▶ Diante da falta de solução algorítmica viável, o único método viável de solução possível é a busca.

Profa. Jaqueline Brigladori Pugliesi

2

Problemas x IA (cont.)

- ▶ 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 seqüência de situações legais e as maneiras de passar de uma situação para outra são em número finito e conhecidas.

Busca em Espaço de Estados

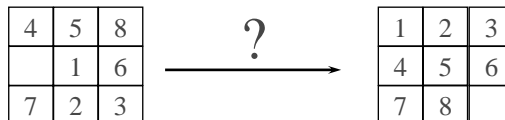
- ▶ De maneira geral, um problema de busca pode ser formalizado através da definição dos seguintes elementos:
 - Um conjunto de descrições chamado espaço de estados, onde cada elemento descreve uma situação possível do problema.
 - Um estado inicial que descreve a situação inicial do problema.
 - Um 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.

Árvore de busca

- ▶ Com os elementos que compõem a descrição do problema da busca é possível construir uma ÁRVORE DE BUSCA, cujo no raiz está associado a um estado inicial e onde os sucessores de qualquer no são associados aos estados obtidos através da aplicação das regras (associadas ou não às heurísticas) sobre a descrição do estado associado ao nó.

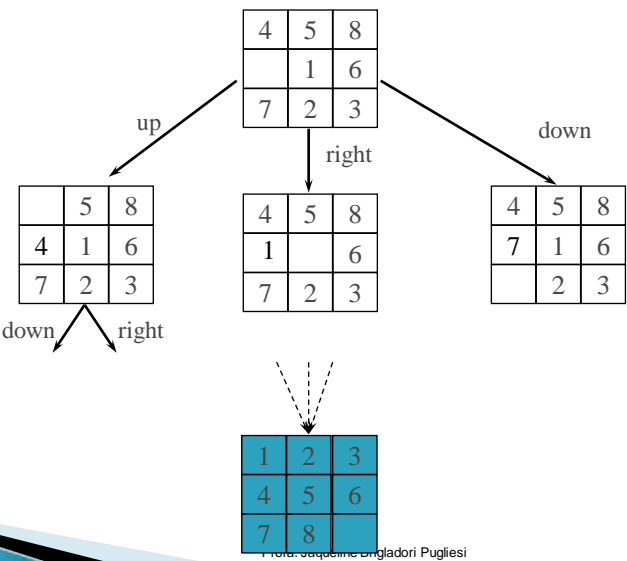
Exemplo

- ▶ Jogo do 8



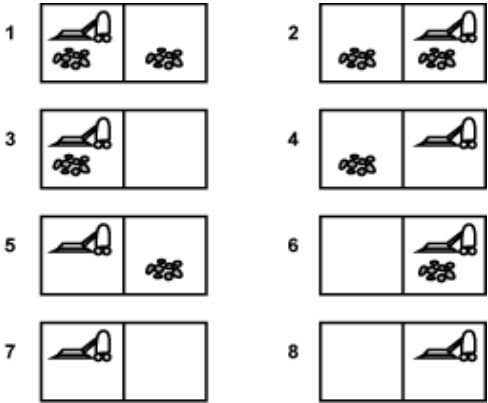
- estados = cada possível configuração do tabuleiro
- estado inicial = qualquer um dos estados possíveis
- estado meta = ordenado, com branco na posição [3,3]
- operadores = mover branco (esquerda, direita, para cima e para baixo)

Árvore de busca para o jogo do 8



Profa. Jaqueline Brigidori Pugliesi

O mundo do aspirador de pó



Profa. Jaqueline Brigidori Pugliesi

O mundo do aspirador de pó

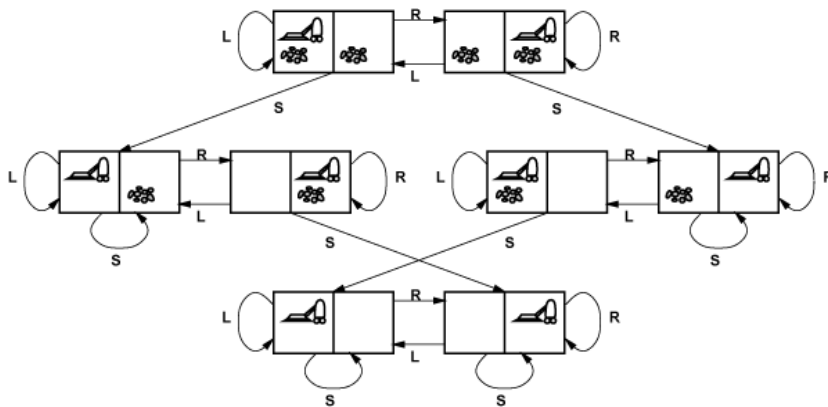
(cont.)

- Formulação do problema:
 - estados = mostrados na figura
 - estado inicial = qualquer um dos estados possíveis
 - teste de término = os dois quartos limpos
 - operadores = mover direita, mover esquerda, aspirar
 - custo do caminho = quantidade de ações realizadas

Profa. Jaqueline Brigidori Pugliesi

9

Árvore de busca



Profa. Jaqueline Brigidori Pugliesi

10

Classificação dos problemas de busca

- ▶ Solução de problemas
- ▶ Jogos
- ▶ Se diferenciam pela capacidade de controle.
 - Na solução de problemas, o mecanismo de busca é livre para escolher qualquer caminho da árvore, e a solução é o caminho percorrido desde o estado inicial até o estado meta.
 - No jogo, a busca deve levar em consideração a jogada do oponente de forma interativa.

Aplicações: Problemas Reais

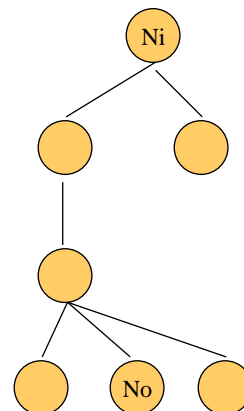
- ▶ Cálculo de rotas
 - rotas em redes de computadores
 - sistemas de planejamento de viagens
 - planejamento de rotas de aviões
 - Caixeiro viajante
- ▶ Alocação (Scheduling)
 - Salas de aula
 - Máquinas industriais (job shop)
- ▶ Navegação de robôs
 - generalização do problema da navegação
 - robôs movem-se em espaços contínuos, com um conjunto (infinito) de possíveis ações e estados
- ▶ Montagem de objetos complexos por robôs
 - ordenar a montagem das diversas partes do objeto
- ▶ etc...

Busca em Espaço de Estados

- ▶ Uma vez o problema bem formulado... o estado final deve ser “buscado”
- ▶ Em outras palavras, deve-se usar um método de busca para saber a ordem correta de aplicação dos operadores que levará do estado inicial ao final
- ▶ Uma vez a busca terminada com sucesso, é só executar a solução (= conjunto ordenado de operadores a aplicar)

Busca

- ▶ Problemas de busca são frequentemente descritos utilizando diagramas de árvores
 - Nó inicial = onde a busca começa
 - Nó objetivo = onde ela termina
- ▶ Objetivo: Encontrar um caminho que ligue o nó inicial a um nó objetivo

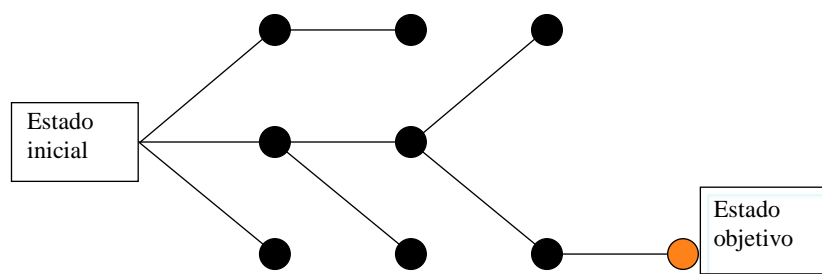


Busca (cont.)

- ▶ Entrada:
 - Descrição dos nós inicial e objetivo
 - Procedimento que produz os sucessores de um nó arbitrário
- ▶ Saída:
 - Sequência legal de nós iniciando com o nó inicial e terminando com o nó objetivo
 - Exemplo: palavras cruzadas

Uma árvore de busca

- ▶ Uma busca pode ser definida graficamente:



- ▶ O objetivo é atravessar a árvore partindo do estado inicial até o estado objetivo

Problemas da busca

- ▶ Com o aumento da árvore de decisão e do número de possíveis caminhos, o tempo de busca aumenta
- ▶ Existem várias formas de reduzir o tempo de busca, alguns dos quais serão discutidos mais adiante

Possíveis situações

- ▶ Mais de um nó objetivo
- ▶ Mais de um nó inicial
- ▶ Nestas situações
 - Encontrar qualquer caminho de um nó inicial para um nó objetivo
 - Encontrar melhor caminho

Definições importantes

- ▶ Profundidade: número de ligações entre um dado nó e o nó inicial
- ▶ Largura: número de sucessores (filhos) de um nó

Algoritmo básico de busca

- 1 Definir um conjunto L de nós iniciais;
- 2 Se L é vazio
Então Busca não foi bem sucedida
Senão Escolher um nó n de L;
- 3 Se n é um nó objetivo
Então Retornar caminho do nó inicial até n;
Parar
Senão Remover n de L;
Adicionar a L todos os filhos de n,
rotulando cada um com o seu
caminho até o nó inicial;
Voltar ao passo 2

Algoritmos de Busca

- ▶ Existem vários algoritmos de busca diferentes, o que os distingue é a maneira como o nó n é escolhido no passo 2
- ▶ Métodos de busca
 - Busca cega: a escolha depende da posição do nó na árvore de busca
 - Busca heurística: A escolha utiliza informações específicas do domínio para ajudar na decisão

Busca cega

- ▶ Busca em Profundidade (BP)
 - A árvore é examinada de cima para baixo
 - Aconselhável nos casos onde os caminhos improdutivos não são muito longos
- ▶ Busca em Largura (BL)
 - A árvore é examinada da esquerda para a direita
 - Aconselhável quando o número de ramos não é muito grande

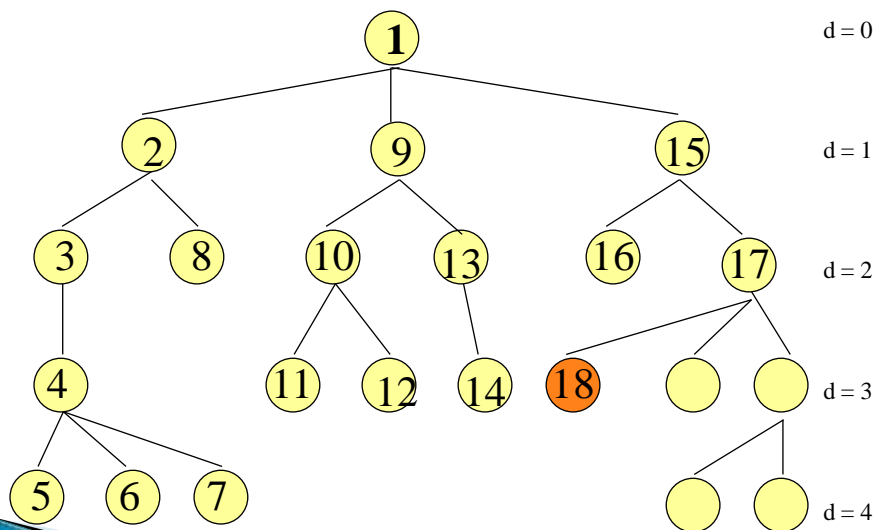
Algoritmo BP

- 1 Definir um conjunto L de nós iniciais
- 2 Se L é vazio
Então Busca não foi bem sucedida
Senão Seja n o primeiro nó de L;
- 3 Se n é um nó objetivo
Então Retornar caminho do nó inicial até n;
Parar
Senão Remover n de L;
Adicionar ao início de L todos os filhos de n,
rotulando cada um com o seu caminho até
o nó inicial;
Voltar ao passo 2;

Profa. Jaqueline Brigladori Pugliesi

23

Busca em profundidade



Profa. Jaqueline Brigladori Pugliesi

24

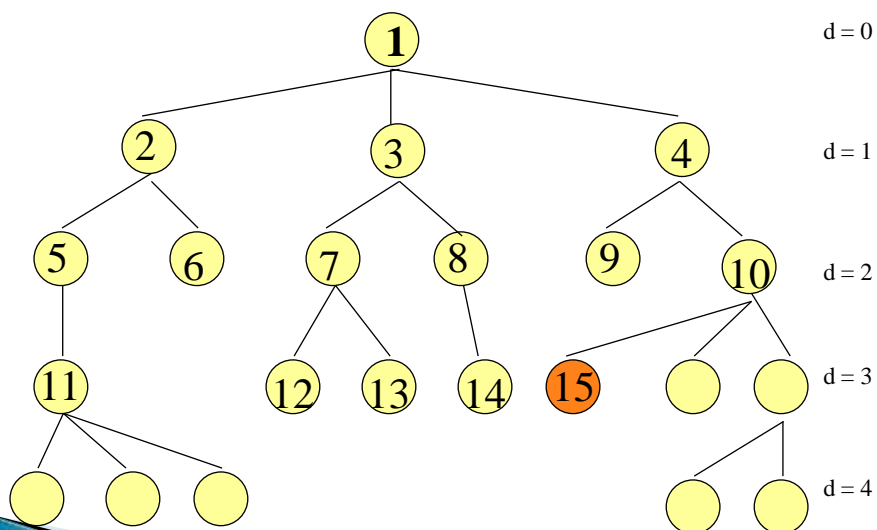
Algoritmo BL

- 1 Definir um conjunto L de nós iniciais
- 2 Se L é vazio
Então Busca não foi bem sucedida
Senão Seja n o primeiro nó de L;
- 3 Se n é um nó objetivo
Então Retornar caminho do nó inicial até n;
Parar
Senão Remover n de L;
Adicionar ao final de L todos os filhos de n,
rotulando cada um com o seu caminho até o
nó inicial;
Voltar ao passo 2;

Profa. Jaqueline Brigladori Pugliesi

25

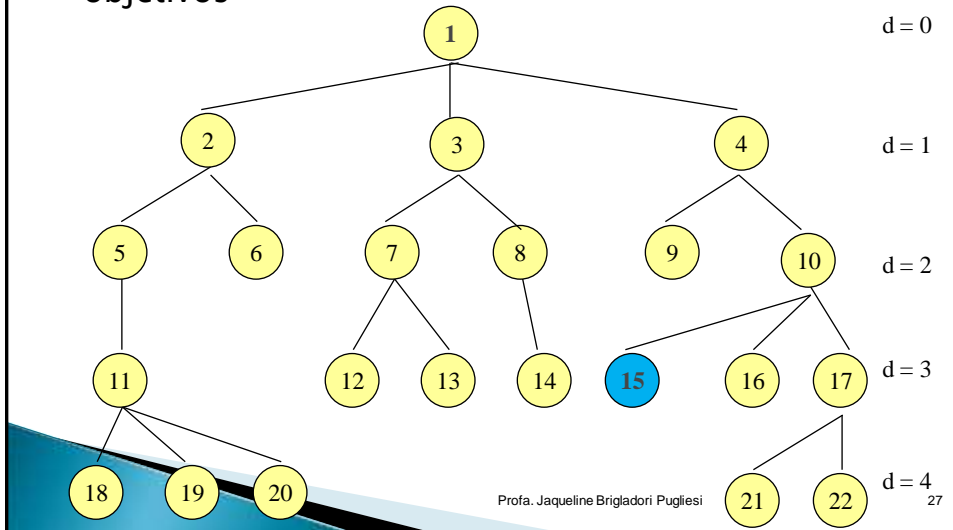
Busca em largura



Profa. Jaqueline Brigladori Pugliesi

26

Exemplo 1: Dada a árvore abaixo, utilizando BP, indique: a) Memória máxima e b) Número mínimo de passos necessários para atingir um dos nós objetivos



Resposta ao exemplo 1

- 1 L = {1}
- 2 L = {21, 31, 41}
- 3 L = {521, 621, 31, 41}
- 4 L = {11521, 621, 31, 41}
- 5 L = {1811521, 1911521, 2011521, 621, 31, 41}
- 6 L = {1911521, 2011521, 621, 31, 41}
- 7 L = {2011521, 621, 31, 41}
- 8 L = {621, 31, 41}
- 9 L = {31, 41}
- 10 L = {731, 831, 41}
- ...

- a) Memória máxima → 6
- b) Número de passos → 18

Resposta ao exemplo 1 (cont.)

...

11 L = {12731, 13731, 831, 41}

12 L = {13731, 831, 41}

13 L = {831, 41}

14 L = {14831, 41}

15 L = {41}

16 L = {941, 1041}

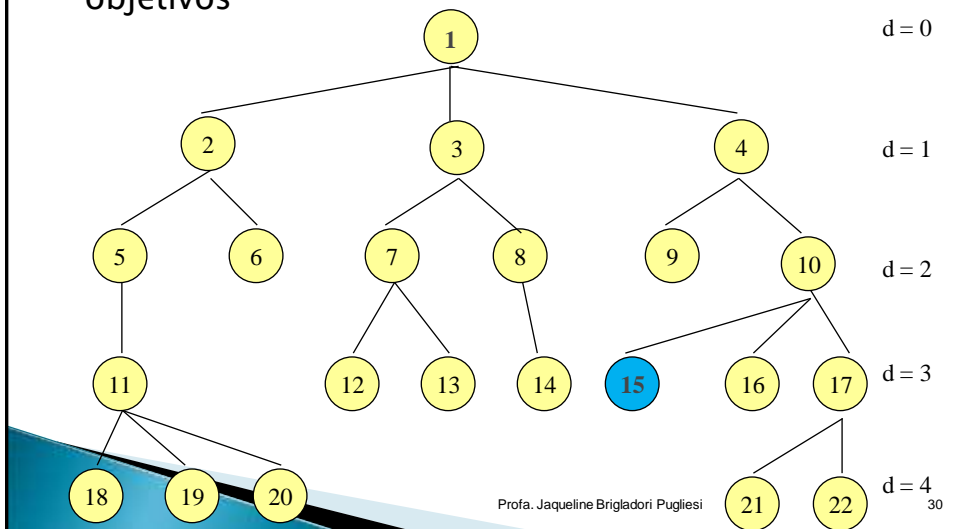
17 L = {1041}

18 L = {151041, 161041, 171041}

Profa. Jaqueline Brigidori Pugliesi

29

Exemplo 2: Dada a árvore abaixo, utilizando BL, indique: a) Memória máxima e b) Número mínimo de passos necessários para atingir um dos nós objetivos



Resposta ao exemplo 2

1 $L = \{1\}$
2 $L = \{21, 31, 41\}$
3 $L = \{31, 41, 521, 621\}$
4 $L = \{41, 521, 621, 731, 831\}$
5 $L = \{521, 621, 731, 831, 941, 1041\}$
6 $L = \{621, 731, 831, 941, 1041, 11521\}$
7 $L = \{731, 831, 941, 1041, 11521\}$
8 $L = \{831, 941, 1041, 11521, 12731, 13731\}$
9 $L = \{941, 1041, 11521, 12731, 13731, 14831\}$
10 $L = \{1041, 11521, 12731, 13731, 14831\}$
...

a) Memória máxima $\rightarrow 9$
b) Número de passos $\rightarrow 15$

Resposta ao exemplo 2 (cont.)

...

11 $L = \{11521, 12731, 13731, 14831, 151041, 161041, 171041\}$
12 $L = \{12731, 13731, 14831, 151041, 161041, 171041, 1811521, 1911521, 2011521\}$
13 $L = \{13731, 14831, 151041, 161041, 171041, 1811521, 1911521, 2011521\}$
14 $L = \{14831, 151041, 161041, 171041, 1811521, 1911521, 2011521\}$
15 $L = \{151041, 161041, 171041, 1811521, 1911521, 2011521\}$

Observações

- ▶ BP e BL não precisam ser realizadas em uma ordem específica
- ▶ Memória utilizada pelas duas técnicas
 - BP: precisa armazenar todos os filhos não visitados de cada nó entre nó atual e nó inicial
 - BL: antes de examinar nó a uma profundidade d , é necessário examinar e armazenar todos os nós a uma profundidade $d - 1$
 - BP utiliza menos memória

Observações (cont.)

- ▶ Quanto ao tempo
 - BP é geralmente mais rápida
 - Métodos de busca cega não examinam a árvore de forma ótima, o que poderia minimizar o tempo gasto para resolver o problema

Busca Heurística

Profa. Dra. Jaqueline Brigladori Pugliesi

35

Tópicos

- ▶ Busca: paradigma de resolução de problemas
- ▶ Métodos de busca
 - Busca cega
 - Busca em profundidade
 - Busca em largura
 - Busca heurística

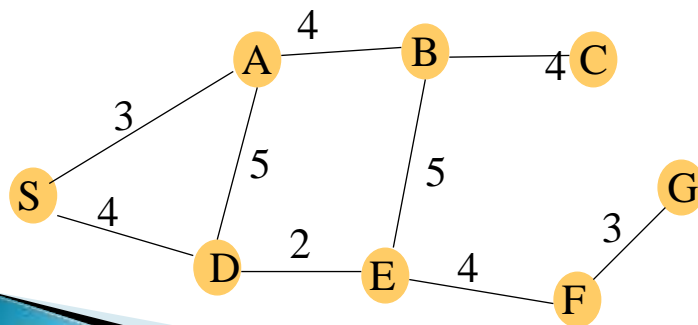
Profa. Jaqueline Brigladori Pugliesi

36

Introdução

► Problema:

- Suponha que você quer descobrir o caminho de uma cidade (S) para outra (G) usando um mapa



Profa. Jaqueline Brigidori Pugliesi

37

Introdução (cont.)

► Para encontrar o melhor caminho, dois custos diferentes devem ser considerados:

- Custo computacional gasto para encontrar um caminho
- Custo de “viagem” decorrente da utilização deste caminho

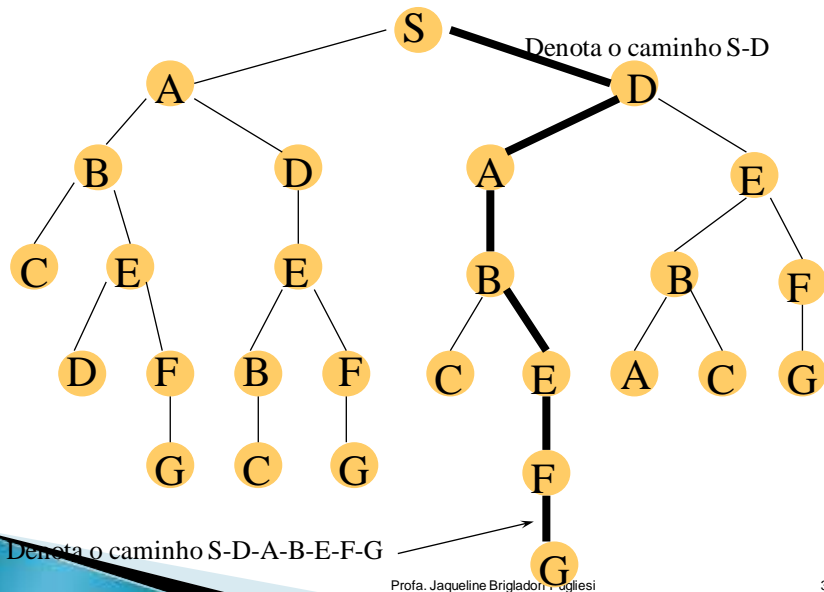
► Possíveis situações:

- Viagem frequente: vale a pena gastar algum tempo para encontrar um bom caminho
- Viagem rara e difícil de achar um caminho: basta encontrar um caminho

Profa. Jaqueline Brigidori Pugliesi

38

Exemplo



Prof. Jaqueline Brigladori Pugliese

39

Problemas da busca

- ▶ Com o aumento da árvore de decisão e do número de possíveis caminhos, o tempo de busca aumenta
- ▶ Existem várias formas de reduzir o tempo de busca, alguns dos quais serão discutidos mais adiante

Profa. Jaqueline Brigladori Pugliesi

40

Busca Heurística

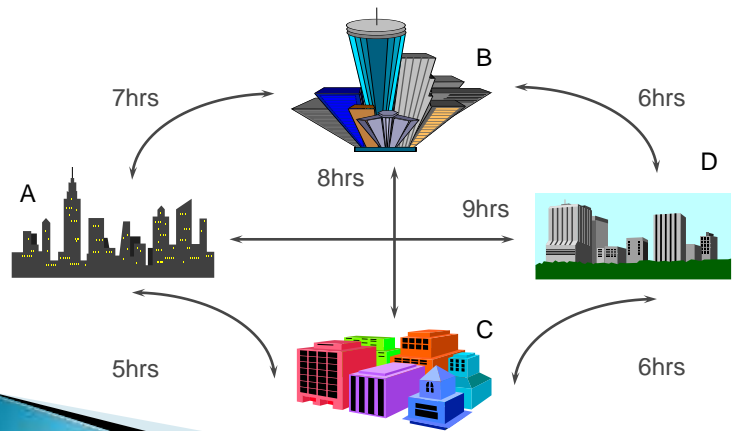
- ▶ Digamos que você está numa Cidade, e quer pegar um trem para casa, mas não sabe qual deve pegar.
- ▶ Se você morasse na zona Norte, naturalmente ignoraria todos os trens que fossem para o sul.
- ▶ Se você morasse na zona Sul, naturalmente ignoraria todos os trens que fossem para o Norte.

Exemplo: problema do caixeiro viajante (TSP)

- ▶ Um caixeiro viajante deve visitar N cidades em sua área de vendas
- ▶ O caixeiro começa de uma base, visita cada cidade uma única vez e retorna à sua cidade no final
- ▶ A cada viagem esta associado um custo
 - O caixeiro deve percorrer a rota mais curta

O problema TSP (cont.)

Considere as rotas definidas entre estas quatro cidades:

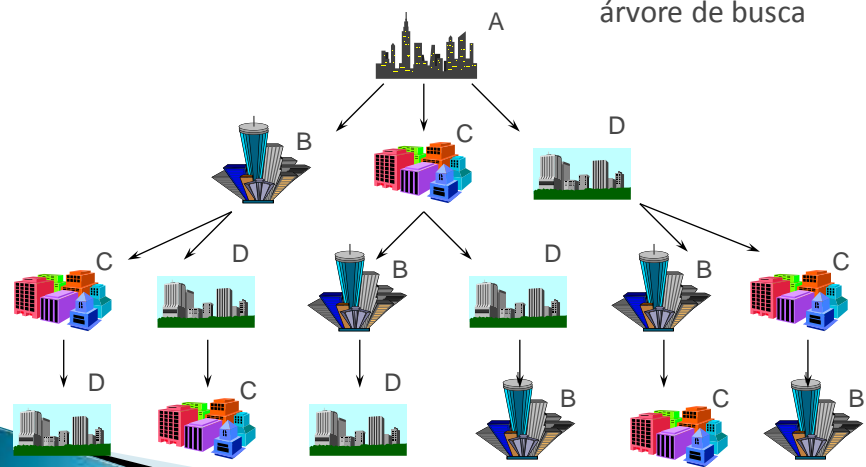


Profa. Jaqueline Brigidori Pugliesi

43

O problema TSP (cont.)

O problema do TSP
representado como
árvore de busca



Profa. Jaqueline Brigidori Pugliesi

44

Explosão Combinatória

- ▶ Com quatro cidades, temos 6 caminhos possíveis.
- ▶ Com dez cidades, temos 362.880 caminhos possíveis.
- ▶ Quanto mais cidades adicionarmos ao TSP, mais caminhos possíveis há.
- ▶ O que nos leva a uma explosão combinatória.
- ▶ Como prevenir ou pelo menos limitar isto?

Heurística

- ▶ O TSP e outros problemas de IA são basicamente problemas de busca.
- ▶ Precisamos limitar de alguma forma o espaço de busca, e assim tornar o processo de busca mais rápido e eficiente.
- ▶ Humanos utilizam “macetes”
- ▶ Em IA são chamados de heurísticas, que ajudam a limitar as buscas.

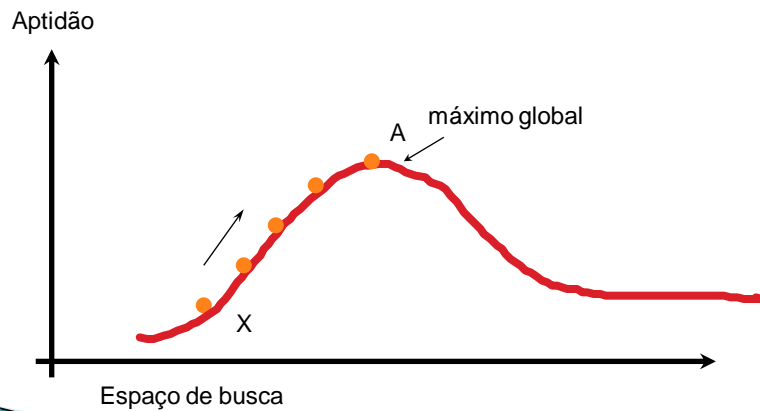
Algoritmos heurísticos

- ▶ Escolher primeiro as opções mais promissoras
 - Em algumas situações é possível colher medidas que determinam uma ordenação razoável
- ▶ Alguns dos Principais métodos
 - *Hill Climbing*
 - *Best-first*
 - Busca em Feixe
 - *Branch and Bound*
 - A^*

Hill Climbing

- ▶ Filosofia: Procurar o ponto mais alto em um terreno durante uma caminhada
 - Objetivo: subir o máximo possível
 - Semelhante ao algoritmo de busca em profundidade
- ▶ Procurar entre os nós próximos, aquele mais perto do objetivo
- ▶ Funciona como BP, mas escolhe o filho de acordo com sua “distância” ao nó objetivo

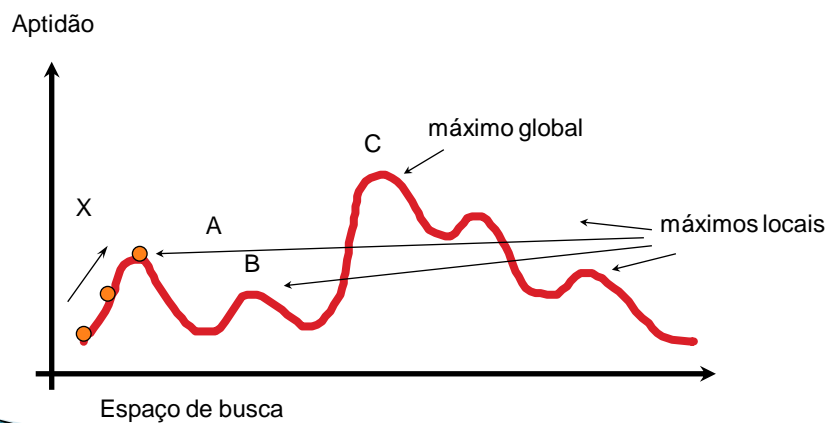
Hill Climbing (cont.)



Profa. Jaqueline Brigidori Pugliesi

49

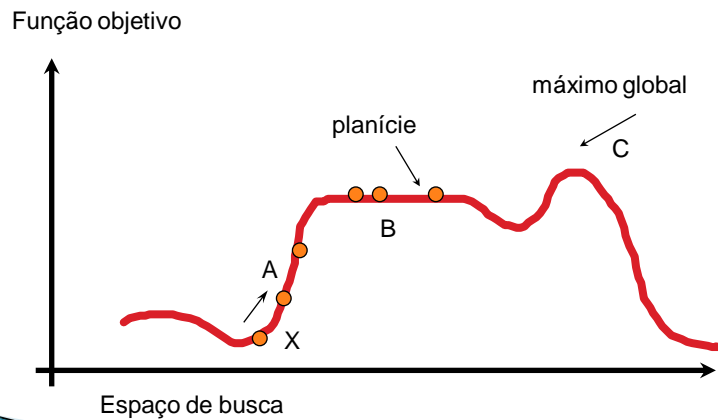
Hill Climbing (cont.)



Profa. Jaqueline Brigidori Pugliesi

50

Hill Climbing (cont.)



Profa. Jaqueline Brigidori Pugliesi

51

Busca Best-first

- ▶ Busca segue pelo melhor nó atual
- ▶ *Hill Climbing* sem a restrição da busca em profundidade
 - Geralmente encontra caminhos mais curtos que o *Hill Climbing*
 - Sempre move em direção ao nó mais próximo do objetivo, não importa onde ele esteja na árvore
- ▶ “Pula” bastante no espaço de busca

Profa. Jaqueline Brigidori Pugliesi

52

Busca em Feixe

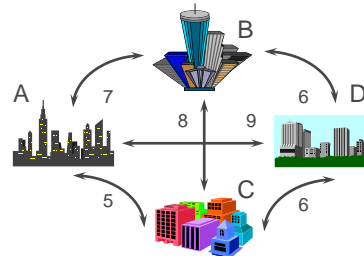
- ▶ Assim como busca em largura, progride nível a nível
 - Move para baixo apenas através dos melhores M nós de cada nível
 - Outros nós do mesmo nível são ignorados
 - M é constante para todos os níveis
- ▶ Vantagens
 - Reduz número de nós visitados
 - Escapa do problema de ramificação infinita

Branch and Bound

- ▶ O nome vem da idéia de que a cada passo estabelecemos um *bound* (limite) de quais *branches* (ramos) serão investigados
- ▶ Aplicando o “*branch and bound*” ao TSP nos levaria a considerar somente aqueles caminhos que levem a uma nova cidade, e que tenham o tempo mais curto entre cidades

Branch and Bound (cont.)

De A nós poderíamos viajar para a próxima cidade escolhendo o menor caminho



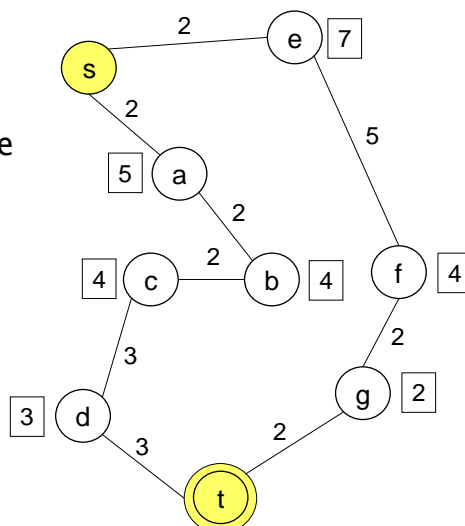
- ▶ Se pegássemos o menor caminho de cada cidade, teríamos: A → C → D → B → A, que é o menor caminho (total 24 horas).
- ▶ Esta heurística nem sempre vai funcionar.

A*

- ▶ Utiliza tanto função de avaliação quanto a de custo para selecionar o estado sucessor mais promissor
 - Função de avaliação se refere ao futuro: “adivinha” quão perto está um estado do estado meta
 - Função de custo se refere ao passado: “sabe” quão longe está um estado do estado inicial
- ▶ Frequentemente tem um desempenho melhor que os outros métodos

A* (cont.)

- ▶ Dado um mapa, o objetivo é encontrar o caminho mais curto entre a cidade inicial **s** e a cidade destino **t**
- ▶ Para estimar o custo do caminho restante da cidade **X** até a cidade **t** utilizaremos a distância em linha reta denotada por **dist(X,t)**
- ▶ $f(X) = g(X) + h(X)$
 $= g(X) + \text{dist}(X,t)$

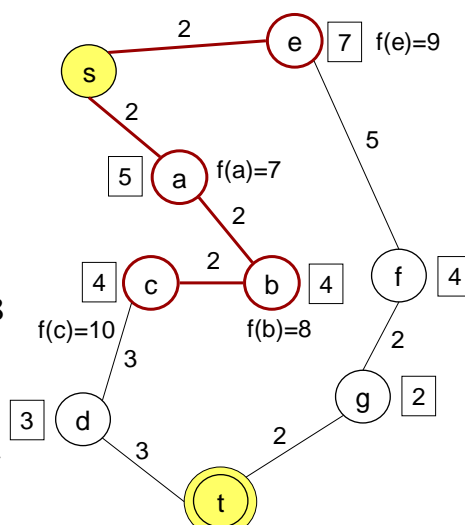


Profa. Jaqueline Brigidori Pugliesi

57

A* (cont.)

- ▶ $f(a) = g(a) + \text{dist}(a,t) = 2 + 5 = 7$
- ▶ $f(e) = g(e) + \text{dist}(e,t) = 2 + 7 = 9$
- ▶ Como o valor-**f** de **a** é menor do que de **e**, o processo 1 (busca via **a**) permanece ativo enquanto o processo 2 (busca via **e**) fica em estado de espera
- ▶ $f(b) = g(b) + \text{dist}(b,t) = 4 + 4 = 8$
- ▶ $f(c) = g(c) + \text{dist}(c,t) = 6 + 4 = 10$
- ▶ Como $f(e) < f(c)$ agora o processo 2 prossegue para a cidade **f**

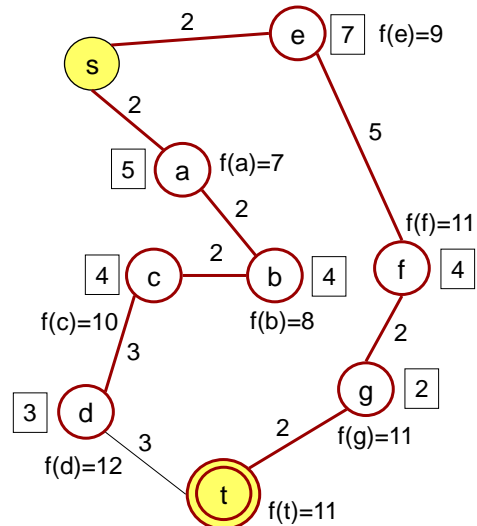


Profa. Jaqueline Brigidori Pugliesi

58

A* (cont.)

- ▶ $f(f) = g(f) + \text{dist}(f, t) = 7 + 4 = 11$
- ▶ Como $f(f) > f(c)$ agora o processo 2 espera e o processo 1 prossegue
- ▶ $f(d) = g(d) + \text{dist}(d, t) = 9 + 3 = 12$
- ▶ Como $f(d) > f(f)$ o processo 2 reinicia chegando até o destino t
- ▶ $f(g) = g(g) + \text{dist}(g, t) = 9 + 2 = 11$
- ▶ $f(t) = g(t) + \text{dist}(t, t) = 11 + 0 = 11$



Profa. Jaqueline Brigidori Pugliesi

59

A* (cont.)

- 1 L = {S}
- 2 L = {A⁷S, E⁹S}
- 3 L = {B⁸AS, E⁹S}
- 4 L = {C¹⁰BAS, E⁹S}
- 5 L = {C¹⁰BAS, F¹¹ES}
- 6 L = {D¹²CBAS, F¹¹ES}
- 7 L = {D¹²CBAS, G¹¹FES}
- 8 L = {D¹²CBAS, T¹¹GFES}

Profa. Jaqueline Brigidori Pugliesi

60

Problemas com a Busca Heurística

- ▶ E o caso onde o menor caminho da primeira cidade leva a uma cidade com caminhos muito longos?
- ▶ Nesse caso deveríamos voltar atrás e tomar o segundo menor caminho, etc.
- ▶ Este processo de “olhar para a frente e voltar atrás” certamente irá levar tempo.

Problemas com Busca Heurística

(cont.)

- ▶ “O tempo gasto na avaliação da função heurística para selecionar um nó para expansão deve ser recuperado por uma redução correspondente, no tamanho do espaço de busca explorado.” Ginsberg, Essentials of Artificial Intelligence.
- ▶ Por que escolher e usar regras heurísticas quando é mais rápido executar uma busca cega?

Heurísticas e Aprendizado

- ▶ No curto prazo pode haver pouca ou nenhuma vantagem de se usar heurísticas.
- ▶ Mas e se o sistema fosse capaz de aprender?
- ▶ Com o tempo a economia obtida ao usar heurísticas certamente iria aumentar, à medida que o sistema aprendesse quais as melhores heurísticas para cada circunstância.

Busca heurística

- ▶ Procure utilizar uma ordem mais próxima de ótima
- ▶ Procure selecionar no passo 2 o nó n mais próximo do objetivo
- ▶ Exemplo: palavras cruzadas
 - Procurar o nó mais profundo
 - Algoritmo BP utiliza heurística simples (fraca)
- ▶ Intenção básica: selecionar de L o nó n mais próximo possível do nó objetivo

Busca heurística (cont.)

- ▶ O tempo gasto avaliando uma função heurística deve ser recuperado por uma redução correspondente no espaço de pesquisa
 - Atividade nível base: esforço gasto tentando resolver o problema
 - Atividade nível meta: trabalho gasto decidindo como resolver o problema

Busca heurística (cont.)

- ▶ *Trade-off* atividade no nível base versus atividade no nível meta
- ▶ Busca eficiente: tempo gasto no nível meta é recuperado com reduções no tempo necessário para o nível base

Observações

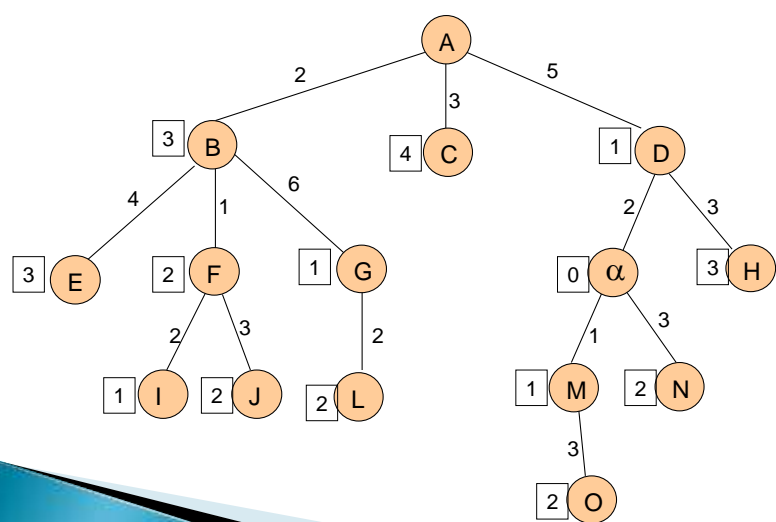
- ▶ Perguntas a serem feitas antes de utilizar métodos de busca:
 - Busca é a melhor maneira para resolver o problema?
 - Quais métodos de busca resolvem o problema?
 - Qual deles é o mais eficiente para este problema?

Exercício

Dada a árvore abaixo, utilizando busca heurística A* e busca em profundidade e largura:

- a) Mostre os passos necessários para se atingir o nó objetivo α .
- b) Compare os resultados obtidos em cada tipo de busca.

Exercício



69

FIM

70