

Inteligência Artificial

Resolvendo Problemas através de Busca

1

Agente solucionador de problemas (guiado por objetivo – deliberativo)

- Busca uma **seqüência de ações** que o leve a estados desejáveis (**objetivos**).
- Propriedades do ambiente para este agente:
 - **Estático** (não muda enquanto o agente delibera)
 - **Discreto** (enumera seqüências alternativas de ações)
 - **Determinístico** (solução é seqüência de ações, i.e., não lida com eventos inesperados pois executa a seqüência definida sem considerar percepções → sistema de controle em malha aberta)
 - **Observável** (e sabe seu estado inicial)
 - Algumas flexibilizações serão feitas em relação às propriedades de determinismo e observabilidade.

2

Agentes solucionadores de problemas

- O que é um problema em I.A.?
- Como formulá-lo?
- Como buscar a solução do problema?
- Como avaliar a solução e o processo de encontrá-la?

3

Definição de Problema

Quatro componentes:

- Estado inicial do agente
- Descrição das possíveis ações do agente:
 - Pela **função sucessor**: dado um estado x , $suc(x)$ retorna um conjunto de pares ordenados (a, y) , onde a indica cada ação válida em x e y é o estado sucessor.
 - Pelo conjunto de **operadores** que podem ser aplicados em um estado para gerar os sucessores.
- Um teste de término:
 - Pode ser um conjunto de estados-objetivos ou
 - Propriedade mais abstrata (ex. cheque-mate em xadrez)
- Uma função de custo da solução
 - avalia numericamente cada solução (medida de desempenho)

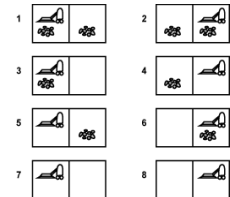
4

Definição de Solução

- O estado inicial e a função sucessor implicitamente definem o **espaço de estados** do problema
- O espaço de estados é descrito por um grafo onde os vértices representam estados e as arestas, ações.
- Um **caminho** no espaço de estados é uma seqüência de estados conectada por uma seqüência de ações.
- Uma **solução** para um problema é um caminho do estado inicial para um estado meta (objetivo).
- A **qualidade** da solução é medida pela função de custo da solução.

5

Exemplo 1: Agente Aspirador de Pó



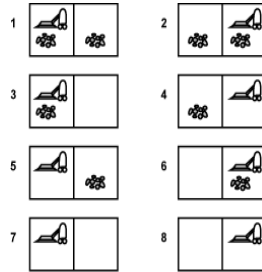
- Formulação do problema:
 - estado inicial = qualquer um dos 8 estados acima
 - função sucessor: operadores = mover direita (R), mover esquerda (L), aspirar (S); $suc(1) = \{(R,2), (L,1), (S,5)\}$, etc..
 - teste de término = os dois quartos limpos
 - custo do caminho = quantidade de ações realizadas (custo 1 para cada ação)

6

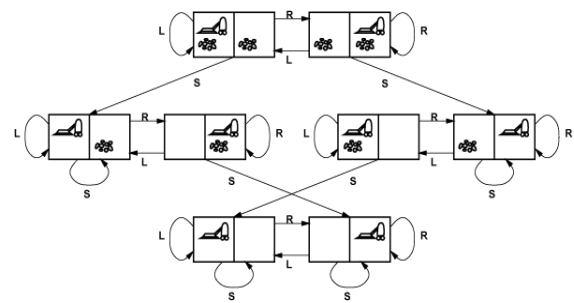
Exercício 1:

- Considerando os oito estados possíveis (ao lado), representar o espaço de estados como um grafo, com os estados como vértices e as ações como arestas.

- Ações: R, L e S

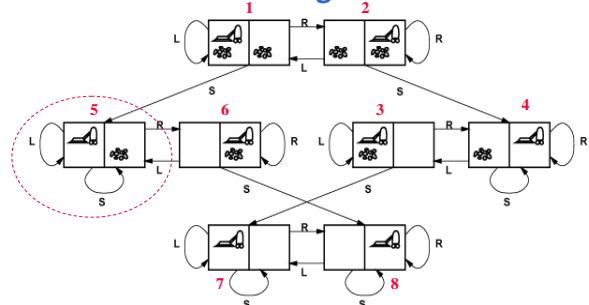


7

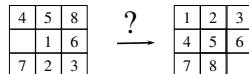
Espaço de estados do agente aspirador**Exercício 2**

- Considere que o agente inicie no estado 5.
- Utilizando o espaço de estados do problema, encontre pelo menos três soluções para o problema.

9

Problema: se agente em 5?

Solução 1: LLSRRS → CUSTO=6
 Solução 2: SRRS → CUSTO=4
 Solução 3: RS → CUSTO=2

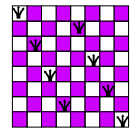
Exemplo 2: Jogo dos 8 Números

- Estado inicial:**
 - cada estado especifica a posição de cada uma das 8 peças e do branco no tabuleiro de 9 posições
 - O estado inicial pode ser qualquer estado
- Função sucessor:**
 - Gera os estados possíveis que resultam ao aplicar cada uma de 4 ações: branco para esquerda (L), para a direita (R), para cima (U), para baixo (D)
- Teste de término**
 - Números ordenados, branco em [3,3].
- Custo do caminho**
 - quantidade de ações realizadas (custo 1 para cada ação)

11

Importância da formulação (1)**Jogo das 8 Rainhas**

- Teste de término: dispor 8 rainhas de forma que não possam se "atacar"
- Custo da solução: ignorado aqui.

**Formulações:**

- Incremental:** envolve operadores que "crescem" a descrição de estado, iniciando com um estado vazio
- Estado completo:**
 - Estado inicial:** qualquer estado com disposição das 8 rainhas, uma em cada coluna
 - Função sucessor:** retorna todos os possíveis estados ao mover uma rainha para outra casa na mesma coluna (cada estado tem $8 \times 7 = 56$ sucessores).

12

Importância da formulação (2)

■ Formulação incremental (1)

- **Estado inicial:** tabuleiro sem rainhas; estado: qualquer disposição de 0 a 8 rainhas no tabuleiro
- **Função sucessor:** adicionar uma rainha a qualquer casa vazia
- **Teste de término:** 8 rainhas sem ataque mútuo

Nesta formulação o espaço de estados = $64 \times 63 \times \dots \times 57 \approx 3 \times 10^{14}$

■ Formulação incremental (2)

- **Estado inicial:** tabuleiro sem rainhas; estado: tabuleiro com n . ($0 \leq n \leq 8$) rainhas dispostas na n -ésima coluna mais à esquerda sem ataque mútuo
- **Função sucessor:** adicionar uma rainha em qualquer casa na coluna vazia mais à esquerda de forma que não possa ser atacada (**teste gradual**)

Formulação melhor (espaço de estados bem menor!)

13

Como encontrar a solução?

- Uma vez o problema bem formulado, o estado meta deve ser “buscado” no espaço de estados

- A busca é representada em uma **árvore de busca**:

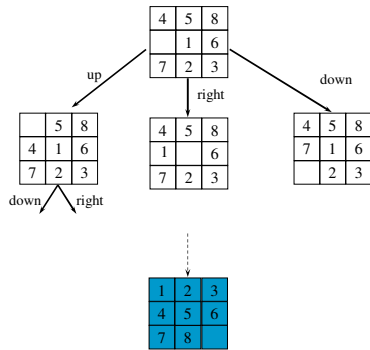
1. Raiz: corresponde ao estado inicial
2. Expande-se o estado corrente: aplica-se a função sucessor ao estado corrente, gerando um novo conjunto de sucessores
3. Escolhe-se o próximo estado a expandir seguindo uma estratégia de busca
4. Prossegue-se até sucesso (atingir estado meta – retorna solução) ou falha

- Espaço de estados \neq árvore de busca

- TSP com 20 cidades: espaço de estados = 20 estados (=cidades), árvore de busca com infinitos vértices (na prática, a busca evita repetir estados).

14

Árvore de busca para o “jogo dos 8 números”



15

Medida de Desempenho na Busca (1)

- Desempenho de um algoritmo de busca:

- **Completezude:** se existir uma solução, ela certamente é encontrada?
- **Otimidade:** a busca encontra a solução de menor custo?
- **Complexidade temporal:** quanto tempo demora para encontrar a solução?
- **Complexidade espacial:** quanto de memória é usado para realizar a busca?

- Em IA a árvore de busca é tipicamente infinita → complexidade é expressa por:

- **b** – fator de ramificação (*branching*) ou número máximo de sucessores de um nó;
- **d** – profundidade (*depth*) do nó-meta mais próximo da raiz;
- **m** – comprimento máximo de um caminho no espaço de estados.

16

Medida de Desempenho na Busca (2)

■ **Custo total = custo da solução + custo da busca**

- custo da solução (ex. TSP: caminho a percorrer, em km)
- custo da busca (tipicamente depende da complexidade em tempo)

Problema: relacionar custo da solução (km) com o da busca (seg)

- Espaço de estados grande:

- compromisso (conflito) entre a melhor solução (menor custo da solução) e a solução mais barata (menor custo da busca)

17

Métodos de Busca

- Busca cega (busca não informada)

- Não tem informação sobre qual sucessor é mais promissor para atingir a meta.
- Estratégias de Busca (ordem de expansão dos nós):
 - *busca em largura*
 - *busca de custo uniforme*
 - *busca em profundidade*
 - *busca em profundidade limitada*
 - *busca em profundidade com aprofundamento iterativo*
 - *busca bidirecional*

- **Busca heurística (busca informada)**

- Possui informação (estimativa) de qual sucessor é mais promissor para atingir a meta.

18