

# Inteligência Artificial

---

Busca:  
Resolução de problemas por meio de busca

Prof. Dr<sup>a</sup>. Andreza Sartori  
[asartori@furb.br](mailto:asartori@furb.br)

# Documentos Consultados/Recomendados

- KLEIN, Dan; ABBEEL, Pieter. **Intro to AI**. UC Berkeley. Disponível em: <http://ai.berkeley.edu>
- LIMA, Edirlei Soares. **Inteligência Artificial**. PUC-Rio, 2015.
- RUSSELL, Stuart J. (Stuart Jonathan); NORVIG, Peter. **Inteligência artificial**. Rio de Janeiro: Campus, 2013. 1021 p, il.
- VIERIU, Radu-Laurențiu. **Artificial Intelligence**. Università degli Studi di Trento, 2016.

# Conteúdo Programático

**Unidade 1:** Fundamentos de Inteligência Artificial

**Unidade 2:** Aplicações de Inteligência Artificial

**Unidade 3:** Busca

**Unidade 4:** Sistemas Baseados em Conhecimento

**Unidade 5:** Redes Neurais Artificiais



# Conteúdo Programático

**Unidade 1:** Fundamentos de Inteligência Artificial

**Unidade 2:** Aplicações de Inteligência Artificial

**Unidade 3:** Busca

**Unidade 4:** Sistemas Baseados em Conhecimento

**Unidade 5:** Redes Neurais Artificiais



# Conteúdo Programático

**Unidade 1:** Fundamentos de Inteligência Artificial

**Unidade 2:** Aplicações de Inteligência Artificial

**Unidade 3: Busca**

3.1. Resolução de Problemas por meio de busca

3.2. Busca Cega ou Exaustiva

3.3. Busca Heurística

3.4. Busca Competitiva

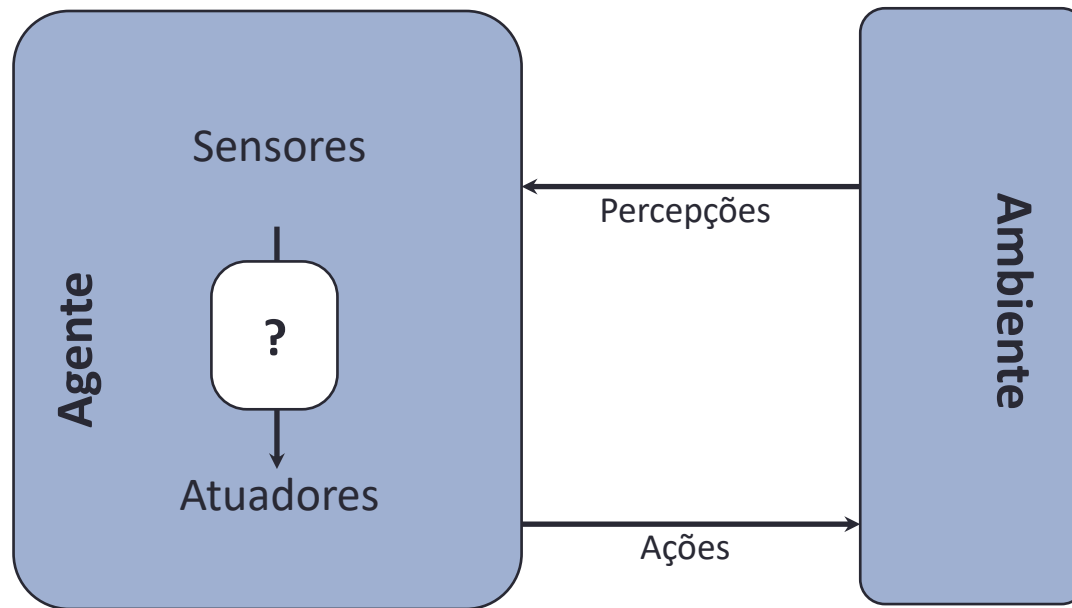
3.5. Busca Local

3.5.1 Algoritmos Genéticos (AG)



# Agentes Inteligentes

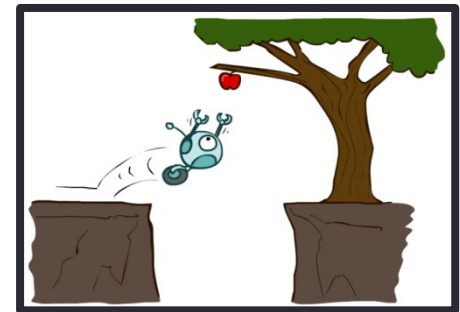
Observam o ambiente e agem de forma autônoma com o objetivo de maximizar sua medida de desempenho.



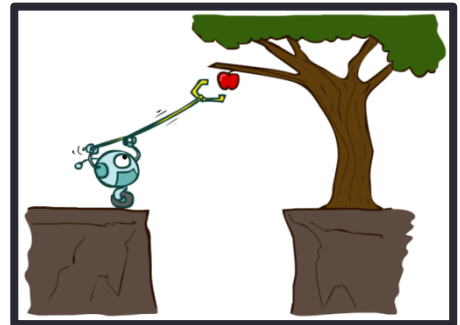
# Tipos básicos de agentes

- Cinco tipos básicos, do mais simples ao mais geral

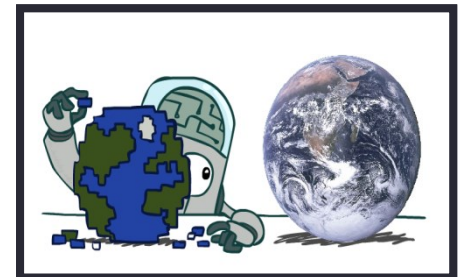
1. Agentes reativos simples
2. Agentes reativos baseados em modelos



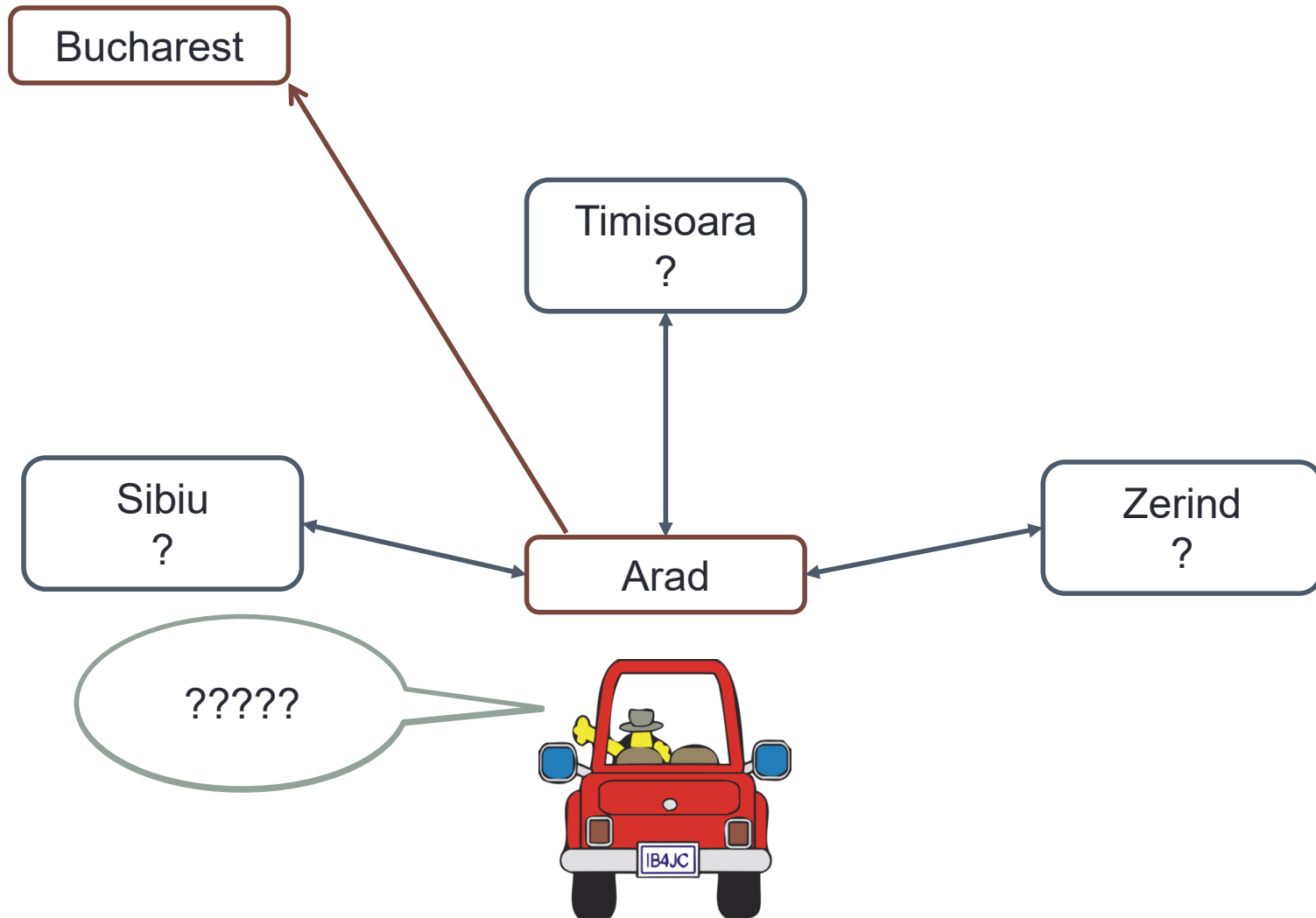
3. Agentes baseados em objetivos
4. Agentes baseados na utilidade



5. Agentes com aprendizagem



# Problema de Busca

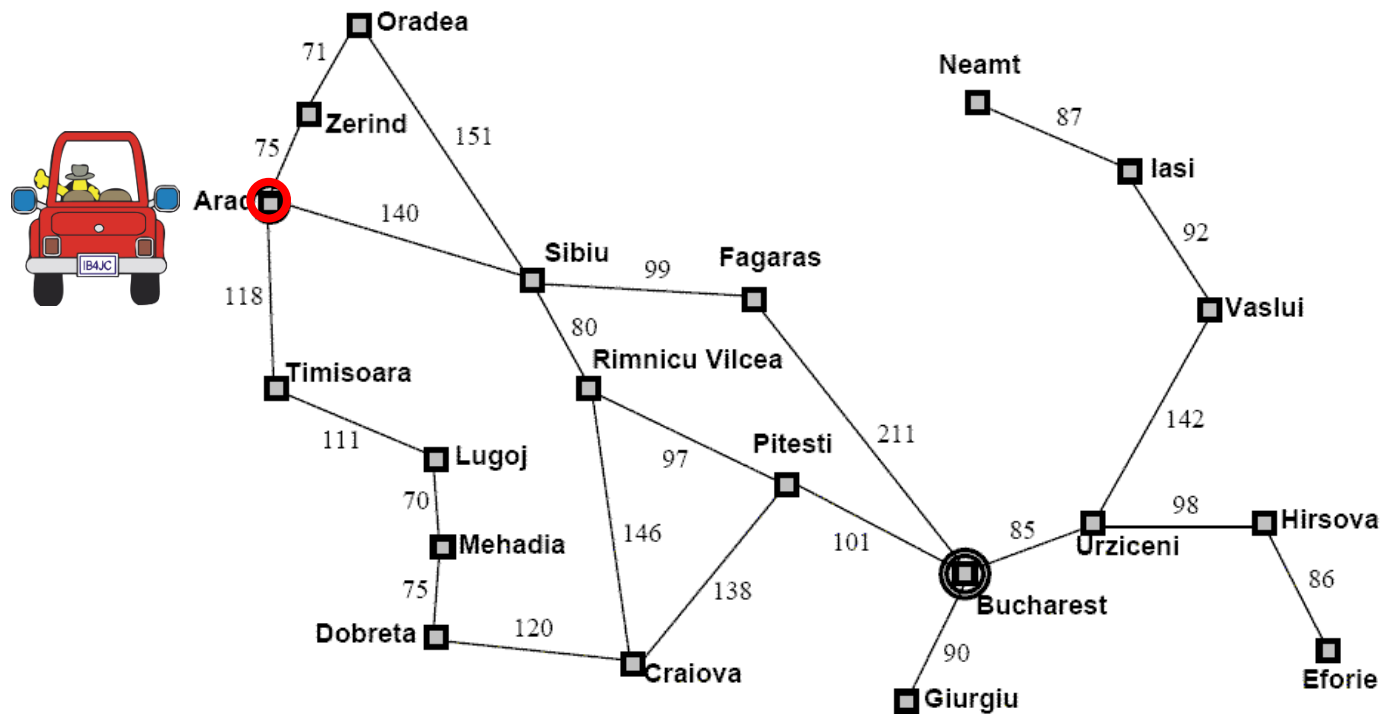




# Problema de Busca

- **Objetivo:** conjunto de estados do mundo em que o objetivo é satisfeito.
- **Tarefa do agente:** descobrir a sequência de ações o levará do estado atual até um estado objetivo.
  - Quais ações devem ser consideradas?
  - Quais estados devem ser considerados?

# Problema de Busca



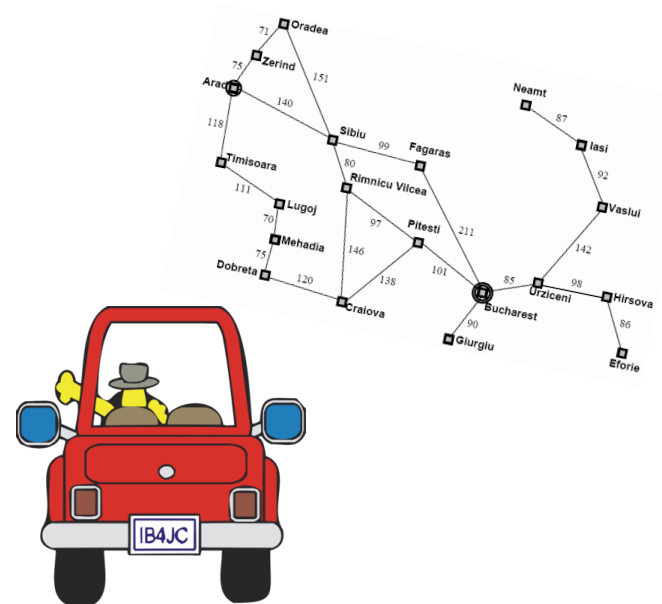
# Busca

- Um agente com várias opções imediatas pode decidir o que fazer examinando diferentes sequências de ações possíveis e então escolher a melhor sequência.
- O processo de procurar pela melhor sequência de ações é chamado de **busca**.
- Encontrada a solução, o agente pode **executar** a sequência de ações para chegar no objetivo.

Formular objetivo → Buscar → Executar sequência de ações

# Exemplo: Agente de Férias na Romênia

- Atualmente em Arad.
- Vão sair amanhã de Bucareste.
- **Formular objetivo:**
  - Estar em Bucareste
- **Formular problema:**
  - Estados: cidades
  - Ações: dirigir de uma cidade para outra
- **Encontrar solução:**
  - Sequência de cidades
    - Exemplo: Arad → Sibiu → Fagaras → Bucareste.

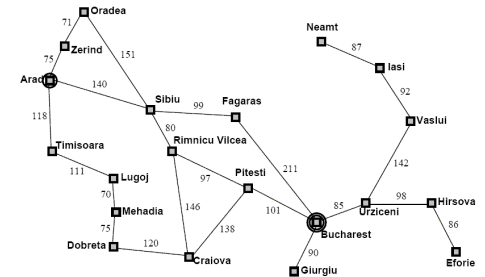


Formular objetivo → Buscar → Executar sequência de ações

# Definição do Problema

# Definição do Problema

- É a **primeira e mais importante** etapa do processo de resolução de problemas de IA por meio de Buscas.
- Analisa o **espaço de possibilidades** de resolução do problema e encontra sequências de ações que levam a um objetivo desejado.



# Solução para um Problema

- A **solução** para um problema é um caminho desde o estado inicial até o estado objetivo (estado final).
- A qualidade da solução é medida pela função de custo de caminho, isto é, a **solução que tiver menor custo** de caminho entre todas as soluções.



# Considerações em Relação ao Ambiente

- **Observável:**

- É necessário que o estado inicial do ambiente seja conhecido previamente.

- **Estático:**

- O Ambiente não pode mudar enquanto o agente está realizando a resolução do problema.

- **Discreto:**

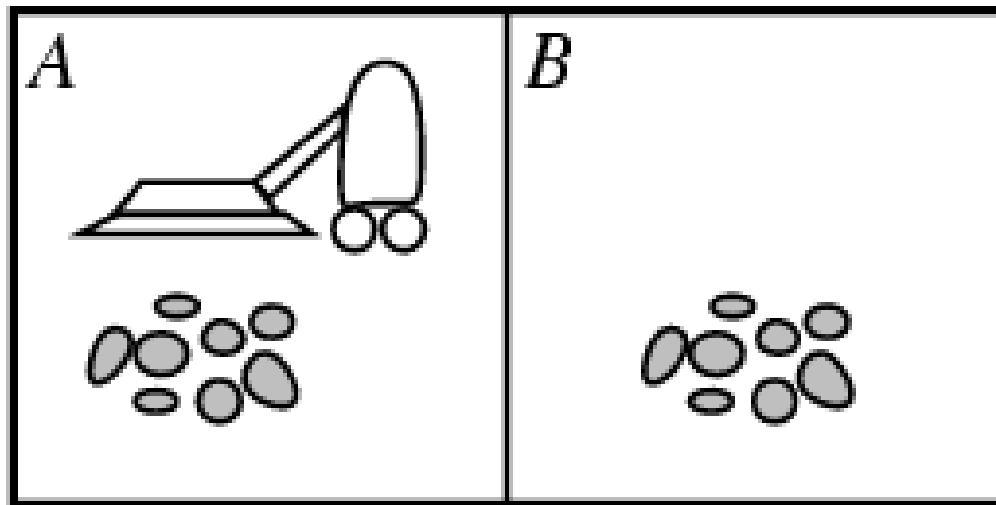
- Existe um conjunto finito de estados que ambiente pode assumir.

- **Determinístico:**

- O próximo estado do agente deve ser determinado pelo estado atual + ação. A execução da ação não pode falhar.

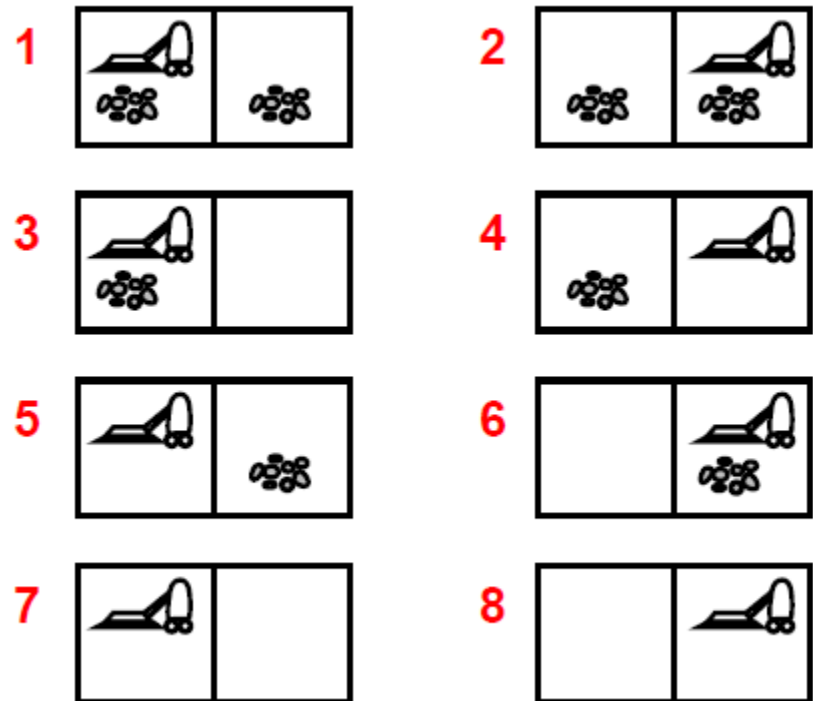
# Exemplo: O mundo do aspirador de pó

- **Percepções:** local (A ou B) e conteúdo (limpo ou sujo)
  - Exemplo: [A, sujo]
- **Ações:** Esquerda, Direita, Aspirar, NoOp

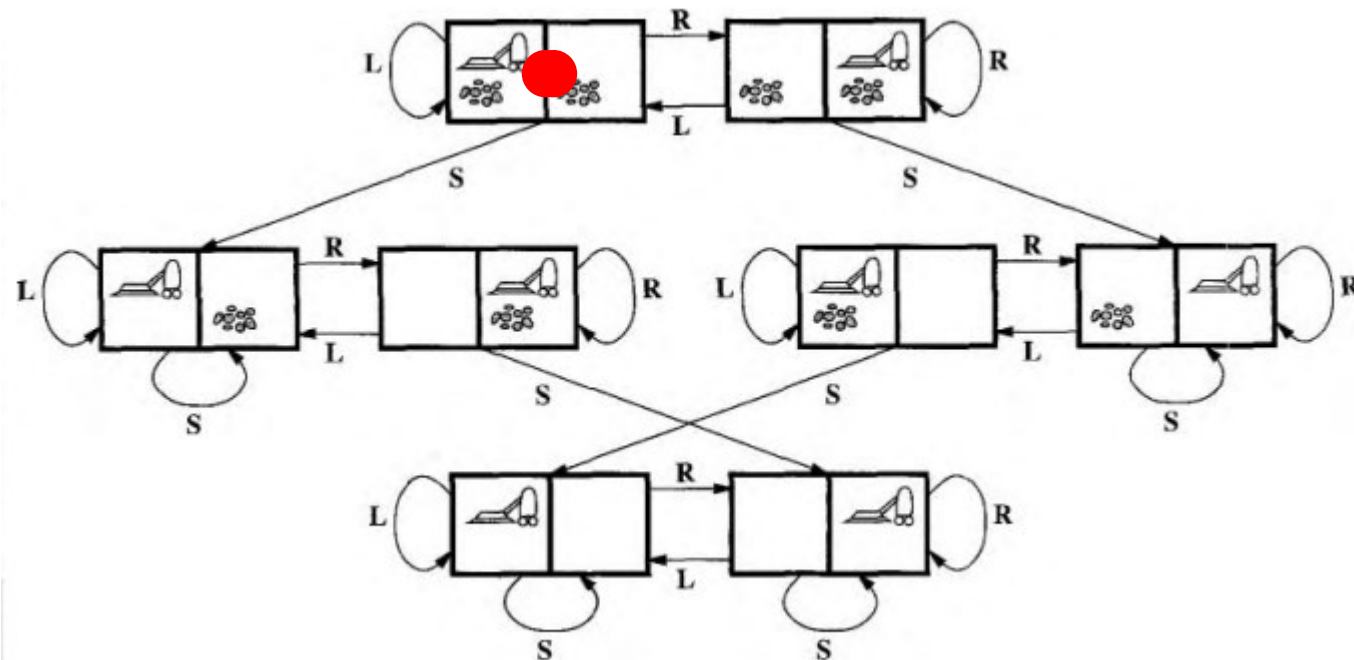


# Exemplo: Aspirador de Pó

- **Espaço de Estados:** 8 estados possíveis (figura ao lado);
- **Estado Inicial:** Qualquer estado;
- **Estado Objetivo (Estado Final):** Estado 7 ou 8 (ambos quadrados limpos);
- **Função Sucessor (Ações Possíveis):** Mover para direita, mover para esquerda e limpar;
- **Custo de caminho:** Cada passo custa 1, assim o custo do caminho é definido pelo número de passos;



# Exemplo: Aspirador de Pó

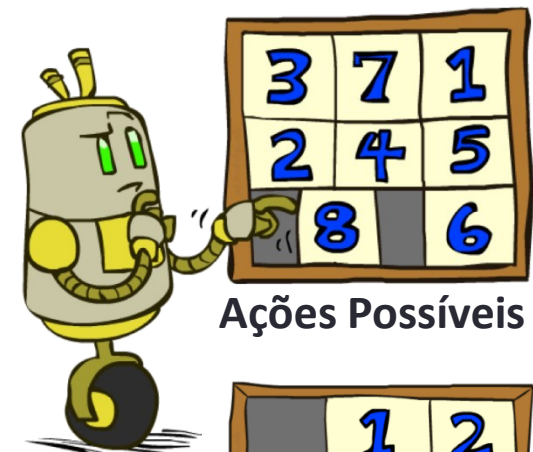


# Exemplo: Quebra-Cabeça de 8 Peças

- **Espaço de Estados:** 181.440 possíveis estados
- **Estado Inicial:** Qualquer estado
- **Estado Objetivo (Estado Final):** Figura ao lado
- **Ações Possíveis:** Mover o quadrado vazio para direita, para esquerda, para cima ou para baixo
- **Custo de Caminho:** Cada passo tem o custo 1, assim o custo do caminho é definido pelo número de passos
- **15-puzzle (4x4)** – 1.3 trilhões estados possíveis.
- **24-puzzle (5x5)** –  $10^{25}$  estados possíveis.

|   |   |   |
|---|---|---|
| 7 | 2 | 4 |
| 5 |   | 6 |
| 8 | 3 | 1 |

Estado Inicial



Ações Possíveis

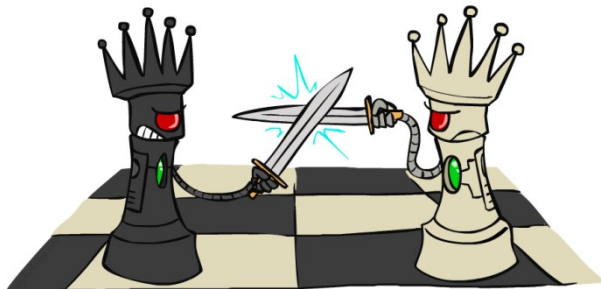
|   |   |   |
|---|---|---|
|   | 1 | 2 |
| 3 | 4 | 5 |
| 6 | 7 | 8 |

Estado Objetivo

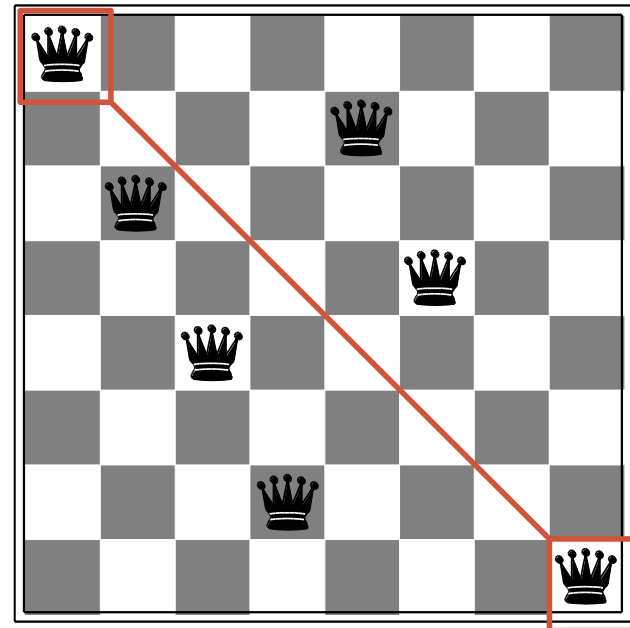
Difícil de resolver de forma ótima com máquinas e algoritmos atuais.

# Exemplo: 8 Rainhas

- Posicionar 8 rainhas em um tabuleiro de xadrez de tal forma que nenhuma rainha ataque a outra.
- Uma rainha ataca qualquer peça situada:
  - Na mesma linha,
  - Na mesma coluna,
  - Na diagonal.



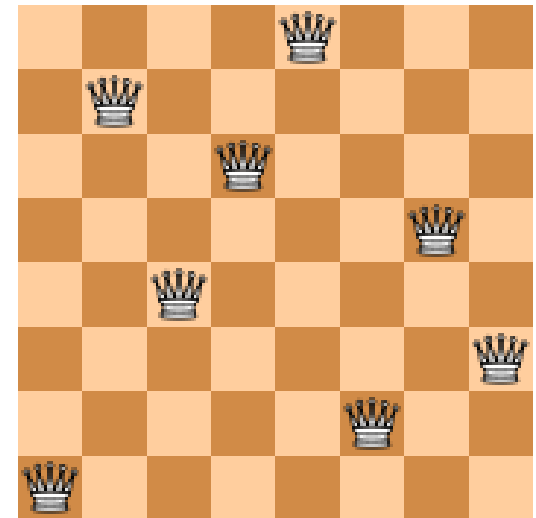
Dan Klein and Pieter Abbeel  
ai.berkeley.edu



falha

# Exemplo: 8 Rainhas (Formulação Incremental)

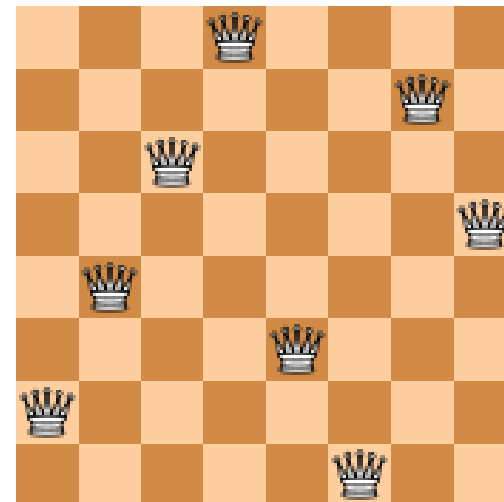
- **Espaço de Estados:** Qualquer disposição de 0 a 8 rainhas no tabuleiro
  - $(64 \times 63 \times \dots \times 57 = 3 \times 10^{14}$  possíveis estados);
- **Estado Inicial:** Nenhuma rainha no tabuleiro;
- **Estado Objetivo (Estado Final):** Qualquer estado onde as 8 rainhas estão no tabuleiro e nenhuma está sendo atacada;
- **Ações Possíveis:** Colocar uma rainha em um quadrado vazio do tabuleiro;
- **Custo de Caminho:** Não importa nesse caso. Apenas o estado final é importante.



LIMA, Edirlei S. **Inteligência Artificial**. PUC-Rio, 2015.

# Exemplo: 8 Rainhas (Estados Completos)

- **Espaço de Estados:** Tabuleiro com  $n$  rainhas, uma por coluna, nas  $n$  colunas mais a esquerda sem que nenhuma rainha ataque outra.
  - 2057 possíveis estados.
- **Estado Inicial:** Todas as rainhas no tabuleiro;
- **Estado Objetivo (Estado Final):** Qualquer estado onde as 8 rainhas estão no tabuleiro e nenhuma esta sendo atacada;
- **Ações Possíveis:** Adicionar uma rainha em qualquer casa na coluna vazia mais à esquerda de forma que não possa ser atacada;
- **Custo de Caminho:** Não importa nesse caso. Apenas o estado final é importante.



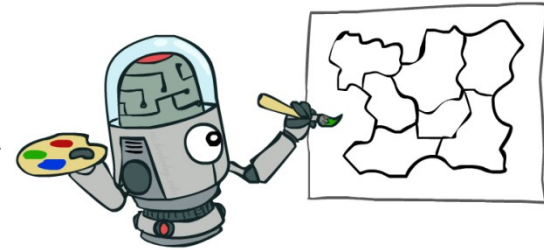
LIMA, Edirlei S. *Inteligência Artificial*. PUC-Rio, 2015.



# Exercício:

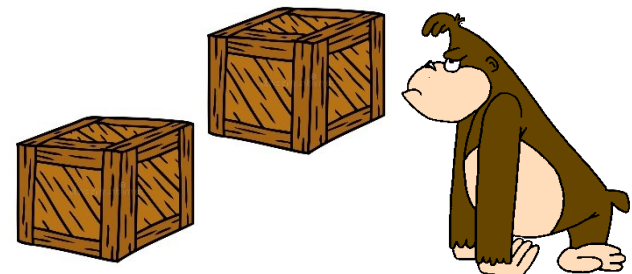
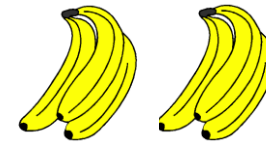
Forneça o Estado Inicial, o Estado Objetivo, a Função Sucessor (ações possíveis) e o Custo de Caminho para cada um dos itens a seguir:

- a. Você tem de colorir um mapa plano usando apenas quatro cores, de tal modo que não haja duas regiões adjacentes com a mesma cor.



Dan Klein and Pieter Abbeel  
ai.berkeley.edu

- b. Um macaco com um metro de altura está em uma sala em que algumas bananas estão presas no teto, a 2,5 metros de altura. Ele gostaria de alcançar as bananas. A sala contém dois engradados empilháveis, móveis e escaláveis, com um metro de altura cada.



# Aplicações em Problemas do Mundo Real

- **Cálculo de Rotas:**

- Encontrar a melhor rota de um ponto a outro
  - Planejamento de rotas de aviões;
  - Sistemas de planejamento de viagens;
  - Caixeiro viajante;
    - visitar cada cidade exatamente uma vez
    - encontrar o caminho mais curto
  - Rotas em redes de computadores;
  - Jogos de computadores (rotas dos personagens).

# Aplicações em Problemas do Mundo Real

- **Circuitos Eletrônicos:**

- Posicionamento de componentes;
- Rotas de circuitos.

- **Robótica:**

- Navegação e busca de rotas em ambientes reais;
- Montagem de objetos por robôs.

# Aplicações em Problemas do Mundo Real

**Considere os problemas de viagens aéreas que devem ser resolvidos através de um site da Web de planejamento de viagem:**

**Espaço de Estados:** Cada estado, inclui uma posição (por exemplo, um aeroporto) e o tempo presente.

Além disso, como o custo de uma ação (um segmento de voo) pode depender de segmentos anteriores, das suas bases de tarifa e da sua condição de ser nacional ou internacional, o estado deverá ter registro de informações adicionais sobre esses aspectos “históricos”.



# Aplicações em Problemas do Mundo Real

**Considere os problemas de viagens aéreas que devem ser resolvidos através de um site da Web de planejamento de viagem:**



- **Estado inicial:** Especificado pela pergunta do usuário.
- **Ações Possíveis:** Pegar qualquer voo a partir da posição atual, em qualquer classe de assento, partindo após o instante atual, deixando tempo suficiente para traslado no aeroporto, se necessário.
- **Estado objetivo (Estado Final):** Estar no destino final especificado pelo usuário
- **Custo do caminho:** Depende do valor da moeda, do tempo de espera, horário do voo, procedimentos de imigração e de atendimento ao cliente, qualidade do assento, horário do dia, tipo de avião, milhagem acumulada etc.

Ok, formulamos alguns problemas, mas  
como resolvê-los?

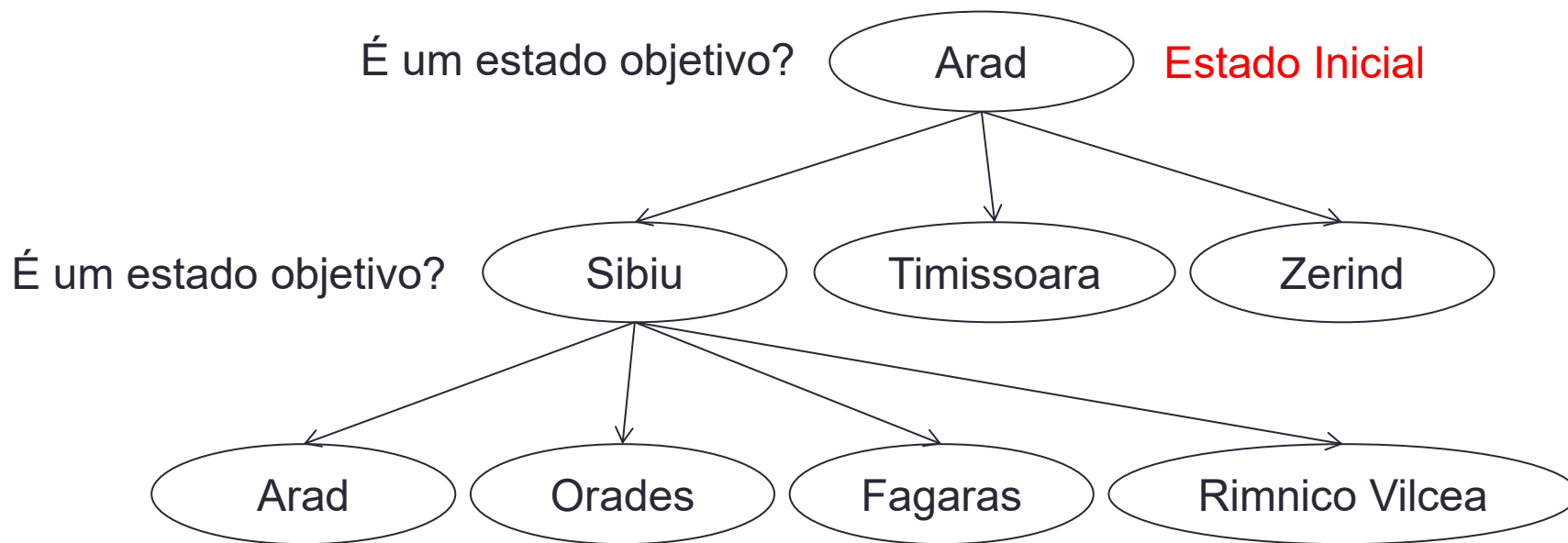
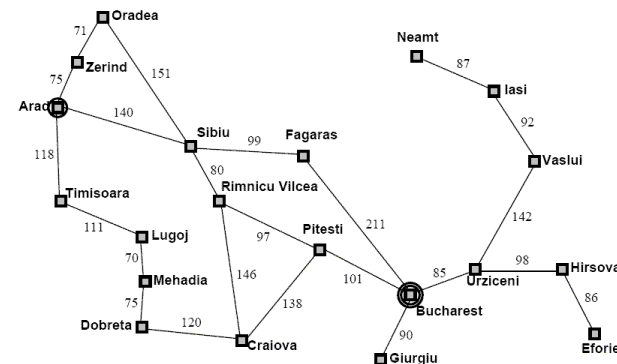


# Como Resolver?

- Quando temos o problema bem formulado, o estado final (objetivo) deve ser “**buscado**” no espaço de estados.
- **Busca** é representada em uma **árvore de busca**, onde:
  - **Raiz**: corresponde ao estado inicial;
  - Expande-se o estado atual, gerando um novo conjunto de estados;
  - Escolhe-se o próximo estado a expandir seguindo uma **estratégia de busca**;
  - Prossegue-se até chegar ao estado final (solução)
    - Se falhar na busca pela solução, volta e escolhe outras opções, então, testa e expande até ser encontrada uma solução ou até não existirem estados a serem expandidos.

# Busca de Soluções

- Exemplo: Ir de **Arad** para **Bucharest**

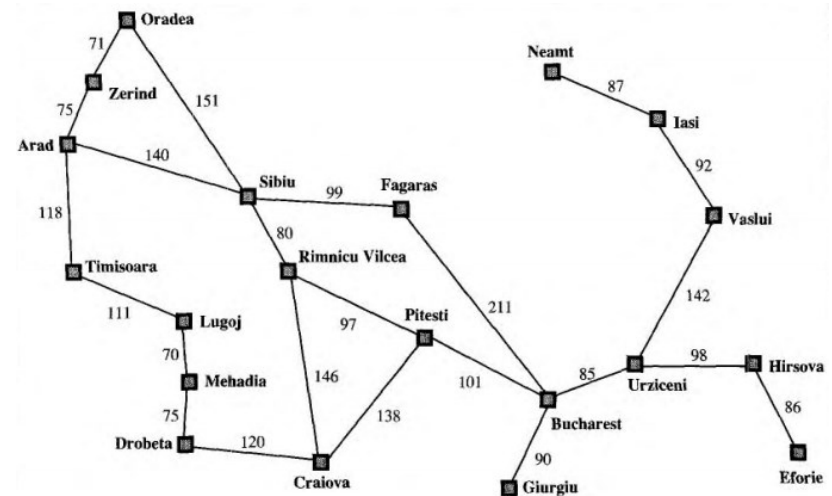


A escolha de qual estado expandir é determinada pela estratégia de busca.



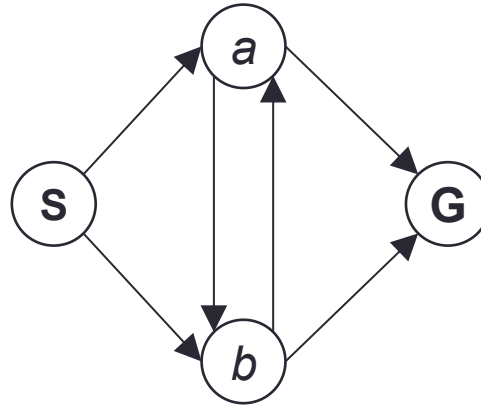
# Busca de Soluções

- A árvore de buscas é **diferente** do espaço de estados.
- Exemplo:
  - Há **20 estados no espaço** de estados no mapa da Romênia.
    - Um para cada cidade.
  - Porém há um número infinito de caminhos.
  - Portanto, a **Árvore de Busca tem infinitos nós**;
  - Caminho infinito:
    - Arad-Sibiu; Arad-Sibiu-Arad-...



## Exercício: [Dan Klein and Pieter Abbeel; ai.berkeley.edu]

- Considere o seguinte espaço de estados:



- Qual é o número de nós para a árvore de busca acima, considerando como o estado inicial S?
  - a) 0
  - b) 4
  - c) 7
  - d) 16
  - e) 258
  - f)  $\infty$  (infinito)

# Medida de Desempenho do Algoritmo de Busca

- Uma estratégia de busca é definida pela escolha da **ordem da expansão de nós**
- Estratégias são avaliadas de acordo com os seguintes critérios:
  - **Completeza:** o algoritmo sempre encontra a solução se ela existe?
  - **Otimização (Custo de Caminho):** a estratégia encontra a solução ótima? - Qualidade da solução
  - **Complexidade de Tempo (Custo de Busca):** quanto tempo ele leva para encontrar a solução? - Número de nós gerados
  - **Complexidade de Espaço (Custo de Busca):** quanta memória é necessária para executar a busca? - Número máximo de nós na memória.

## Custo Total

Custo do Caminho + Custo de Busca.

# Métodos de Busca

- **Busca Cega ou Exaustiva:**

- Não tem nenhuma informação adicional sobre os estados, isto é, não sabe qual o melhor nó da fronteira a ser expandido. Apenas distingue o estado objetivo dos não objetivos.

- **Busca Heurística:**

- Ou busca com informação, estima qual o melhor nó da fronteira a ser expandido baseado em funções heurísticas.

- **Busca Competitiva:**

- Considera que há oponentes hostis e imprevisíveis. Ex: Jogos

- **Busca Local:**

- Operam em um único estado e movem-se para a vizinhança deste estado.
- **Algoritmos Genéticos:**
  - Variante de Busca Local em que é mantida uma grande população de estados. Novos estados são gerados por mutação e por crossover, que combina pares de estados da população.