Search Problems

Russell and Norvig:
Chap. 3, Sect. 3.1 – 3.2

Slides by Jean-Claude Latombe, from an introductory AI course given at Stanford University (used with permission).
Goal-Based Agent

Goal: One way towards maximizing performance measure (to be rational)
Goal-Based Agent

**Goal:** One way towards maximizing performance measure (to be rational)

Can it find a **sequence of actions** achieving its goals, when no single action will do?
Problem-Solving Agent

- Agent
- Environment
- Sensors
- Actuators

- Actions
- Initial state
- Goal test

It can do this through graph searching!
Problem as a Search Problem

- Actions
- Initial state
- Goal test

state space
transition model
Initial State

- Actions
- Initial state
- Goal test

state space

transition model
Goal Test

- Actions
- Initial state
- Goal test

state space

transition model
Example: 8-puzzle

State: Any arrangement of 8 numbered tiles and an empty tile on a 3x3 board
Example: 8-puzzle

<table>
<thead>
<tr>
<th></th>
<th>8</th>
<th>2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>4</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>6</td>
<td></td>
</tr>
</tbody>
</table>

Action set and transition model is knowledge about the 8-puzzle game, but it does not tell us which transition to take.

Search is about the exploration of alternatives to find a goal state.
Example: 8-puzzle

Size of the state space = 9! = 362,880 (but half reachable)

15-puzzle  \( \sim 2 \times 10^{13} \)

24-puzzle  \( \sim 1 \times 10^{25} \)

10 millions states/sec

0.36 sec

24 days

40 billion years
Search of State Space

state graph
Search of State Space

state graph
Search of State Space

state graph
Search of State Space

state graph
Search of State Space

state graph
Search of State Space

search tree

state graph
Search of State Space

solution

state graph
Search Problem

- Initial state
- Actions
- Transition model
- State space
- Goal test
- Path cost
Search Problem

- Initial state:
  - usually the current state
  - sometimes one or several hypothetical states ("what if ...")

- Actions
- Transition model
- State space
- Goal test
- Path cost
Search Problem

- Initial state
- Actions:
  - possible actions available to agent
- Transition model
- State space
- Goal test
- Path cost
Search Problem

- Initial state
- Actions
- Transition model:
  - Result of doing action in state
  - Successor: reachable state by single action from current state
- State space
- Goal test
- Path cost
**Search Problem**

- **Initial state**
- **Actions**
- **Transition model**
- **State space** (implicitly defined by the above):
  - each state is an abstract representation of the environment
  - the state space is discrete
- **Goal test**
- **Path cost**
Search Problem

- Initial state
- Actions
- Transition model
- State space
- Goal test:
  - sometimes the description of a state
  - usually a condition
- Path cost
Search Problem

- **Initial state**
- **Actions**
- **Transition model**
- **State space**
- **Goal test**

**Path cost:**
- [path $\rightarrow$ positive number]
- usually, path cost = sum of step costs
- e.g., number of moves of the empty tile
Simple Agent Algorithm

Problem-Solving-Agent

1. **formulate**: (abstraction!)
   1. initial-state \(\triangleleft\) sense/read state
   2. goal \(\triangleleft\) select/read goal
   3. actions \(\triangleleft\) select/read action models
   4. transition model \(\triangleleft\) select/read model
   5. problem \(\triangleleft\) (initial-state, goal, actions, transition model)

2. **solution** \(\triangleleft\) search(problem)

3. **perform**(solution)
Example: 8-queens

Place 8 queens in a chessboard so that no two queens are in the same row, column, or diagonal.

A solution

Not a solution
**Example: 8-queens**

Formulation #1:
- **States:** any arrangement of 0 to 8 queens on the board
- **Initial state:** 0 queens on the board
- **Actions:** add a queen in any empty square
- **Transition model:** board contains queen
- **Goal test:** 8 queens on the board, none attacked

- $64^8$ states with 8 queens
Formulation #2:
- **States:** any arrangement of $k = 0$ to 8 queens in the $k$ leftmost columns with none attacked
- **Initial state:** 0 queens on the board
- **Actions:** add a queen to any square in the leftmost empty column such that it is not attacked by any other queen
- **Transition model:** board contains queen
- **Goal test:** 8 queens on the board

> 2,057 states
Example: Robot navigation

What is the state space?
Example: Robot navigation #1

Cost of one horizontal/vertical step = 1
Cost of one diagonal step = \( \sqrt{2} \)
Example: Robot navigation #1

This path is the shortest in the discretized state space, but not in the original continuous space.
Example: Robot navigation #2
Example: Robot navigation #2
Example: Robot navigation #2

states
Example: Robot navigation #2

transition model
A path-smoothing post-processing step is usually needed to shorten the path further.
Example: Robot navigation #3

Cost of one step: length of segment
Example: Robot navigation #3

Visibility graph

Cost of one step: length of segment
Example: Robot navigation #3

The shortest path in this state space is also the shortest in the original continuous space.
Example: Assembly Planning
Example: Assembly Planning
Example: Assembly Planning

- **State:** Collection of sub-assemblies
- **Initial state:** All sub-assemblies are individual parts

- **Goal state:** Complete assembly
- **Actions:** Merge two subassemblies (check for collision)
- **Transition model:** Merged assembly
- **Cost function:** Longest sequence of assembly operation
Example: Assembly Planning
Assumptions in Basic Search

- The environment is **static**
- The environment is **discretizable**
- The environment is **observable**
- The actions are **deterministic**
Search Problem Formulation

- Real-world environment -> Abstraction
Search Problem Formulation

- Real-world environment -> Abstraction
  - Validity:
    - Can the solution be executed?
Search Problem Formulation

Real-world environment -> Abstraction

- Validity:
  - Can the solution be executed?
  - Does the state space contain the solution?
Search Problem Formulation

Real-world environment -> Abstraction

- Validity:
  - Can the solution be executed?
  - Does the state space contain the solution?

- Usefulness
  - Is the abstract problem easier than the real-world problem?
Search Problem Formulation

- Real-world environment ✖ Abstraction
  - Validity:
    - Can the solution be executed?
    - Does the state space contain the solution?
  - Usefulness
    - Is the abstract problem easier than the real-world problem?

Without abstraction an agent would be swamped by the complexity of the real world
Search Problem Variants

- One or several initial states
- One or several goal states
- The solution is the path or a goal node
  - In the 8-puzzle problem, it is the path to a goal node
  - In the 8-queen problem, it is a goal node
Search Problem Variants

- One or several initial states
- One or several goal states
- The solution is the path or a goal node
- Any, or the best, or all solutions
Important Parameters

- Number of states in state space

<table>
<thead>
<tr>
<th>Problem</th>
<th>States</th>
</tr>
</thead>
<tbody>
<tr>
<td>8-puzzle</td>
<td>362,880</td>
</tr>
<tr>
<td>15-puzzle</td>
<td>$2 \times 10^{13}$</td>
</tr>
<tr>
<td>24-puzzle</td>
<td>$1 \times 10^{25}$</td>
</tr>
<tr>
<td>8-queens</td>
<td>2,057</td>
</tr>
<tr>
<td>100-queens</td>
<td>$10^{52}$</td>
</tr>
</tbody>
</table>

There exist techniques to solve N-queens problems efficiently!

Stating a problem as a search problem is not always a good idea!
Important Parameters

- Number of states in state space
- Distribution of goal states
- Size of memory needed to store a state
Important Parameters

- Number of states in state space
- Distribution of goal states
- Size of memory needed to store a state
- Running time of the successor function
Applications

- Route finding: airline travel, networks
- Pipe routing, VLSI routing
- Pharmaceutical drug design
- Robot motion planning
- Video games
Summary

- Problem-solving agent
- State space, actions, transition model
- Search!
- Examples:
  - 8-puzzle, 8-queens, route finding, robot navigation, assembly planning
- Assumptions of basic search
- Important parameters