Problem Solving As Search

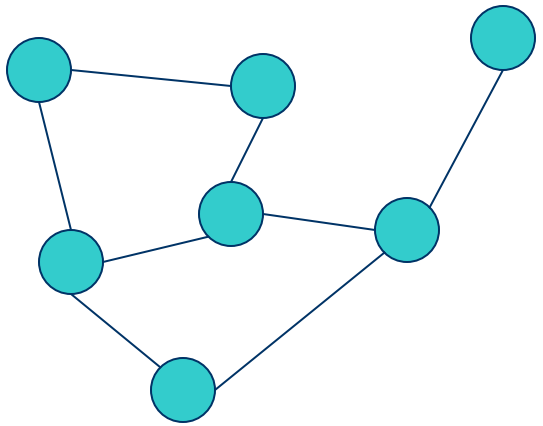
Scott Wallace
CS 440
WSU Vancouver

Outline

- Review of Search
- Problem Solving Agents
- Formulating Problems for Search

Search – in the Classic CS Sense

- (Graph) Search is a set of systematic methods for visiting nodes in a graph



Problem Solving

- Many (most?) problems can be formulated as follows:
  - We have the world as it is now (initial state)
  - We have a vision of the world as we want it (desired state, or goal state)
  - We want to change the world to meet our desires (goals) in the most efficient way possible

Problem Formulation

- For an agent, the goals and the performance measure are closely linked.
- Assuming goals are given, to solve the problem we need to:
  - Determine what types of actions are available
  - Determine how potential actions would change the world
  - Identify a sequence of actions that perform the desired changes

Why is Search Exciting to AI?

- Many problems can be formulated as:
  - Initial state & goal state
  - Potential actions & how they change the world
- Now, we have a set of states (nodes) linked together by actions (edges) that the agent may take in a particular state.
- Solving a problem is thus reduced to graph search.

Search as a Tool

- Search is a general method for solving problems
  - **Upside:** can be used to solve just about any problem.
  - **Downside:** approach is very general and thus not tailored to particularities of the task.
- General methods (like search) are often called **weak methods** in AI

The Problem-Solving Agent

- One type of **goal based agent** can be formulated as follows:
  - Determine goal from current state
  - Search for action sequence to achieve goal
  - Execute action sequence in world
- Last two phases are prominent in AI

Simple Problem Solving Agent

```
function SIMPLE-PROBLEM-SOLVING-AGENT(percept) returns an action
  static: seq, an action sequence, initially empty
          state, some description of the current world state
          goal, a goal, initially null
          problem, a problem formulation

  state ← UPDATE-STATE(state, percept)
  if seq is empty then do
      goal ← FORMULATE-GOAL(state)
      problem ← FORMULATE-PROBLEM(state, goal)
      seq ← SEARCH(problem)
  action ← FIRST(seq)
  seq ← REST(seq)
  return action
```

Environments

- Our agent makes implicit assumptions about environment:
  - Static (doesn’t change during search)
  - Observable (initial state is known)
  - Discrete (can enumerate choices)
  - Deterministic (one action sequence returned)
- Agent acts much like a dung beetle

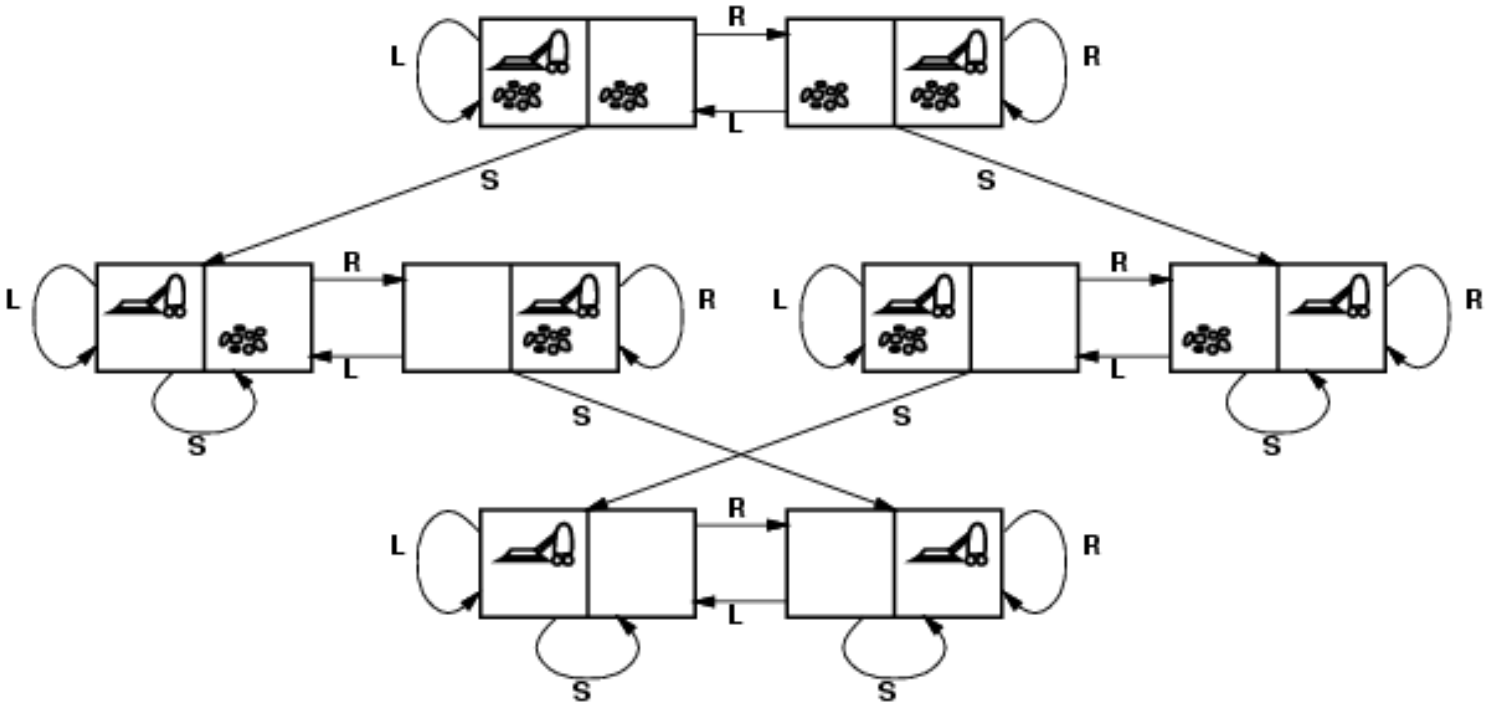
Problem Formulation II

- Initial State
- Available Actions & Their Effects
  - Usually defined as a **successor** function that can be applied to a state and will return a list of **<action, resulting state>** pairs
- Goal Test
  - Usually a function that can be applied to the current state to see if it meets the goal condition
- Path Cost
  - A function assigning cost to a path through the state space. Usually the sum of individual (non-negative) step costs.

Result of Searching

- A path (or paths) through state space
  - Describes the action sequence required to achieve the goal
- The cost of each path (solution)
  - Based on the cost function for the problem
  - Usually we want a minimum cost solution
- Some agents may only care about whether the goal is satisfied, not the process required to satisfy it…

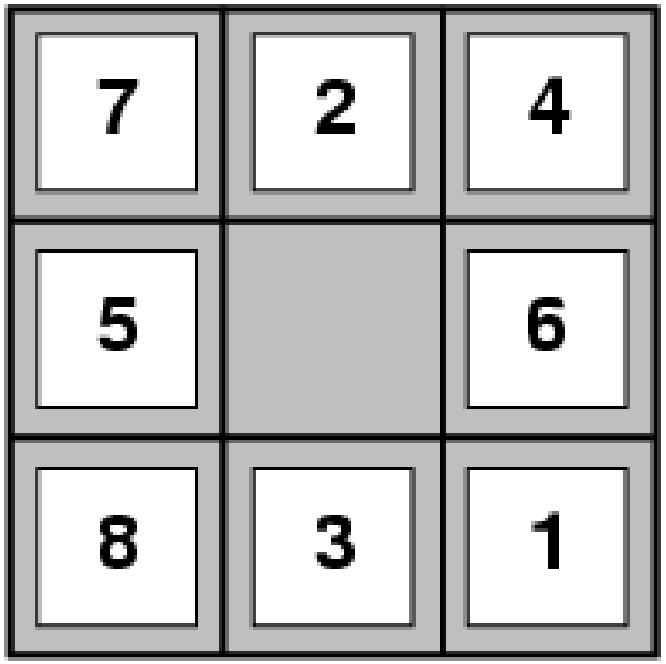
Vacuum World



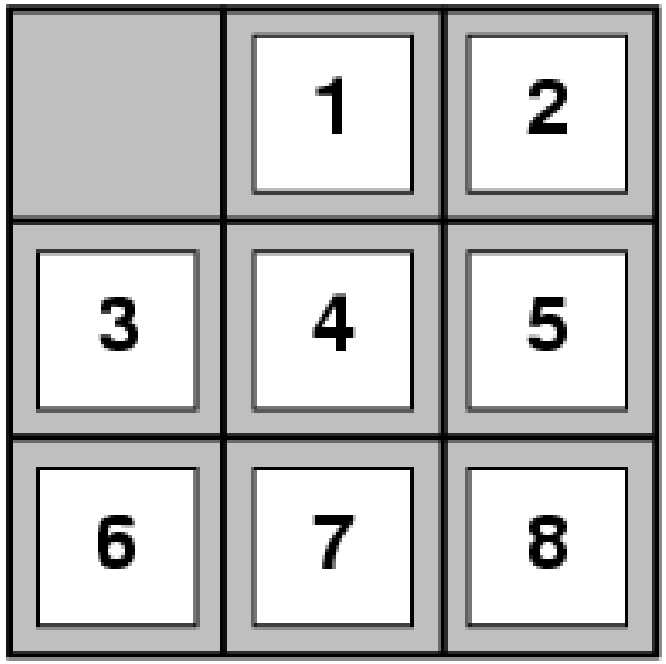
Vacuum World

- States
  - Pair: Robot location and Dirt Location list
  - Representation often worth thinking carefully about
- Actions
  - Right, Left, Suck
- Goal Test
  - All rooms are clean
- Path Cost
  - Positive constant for each action

The Eight Puzzle



Start State

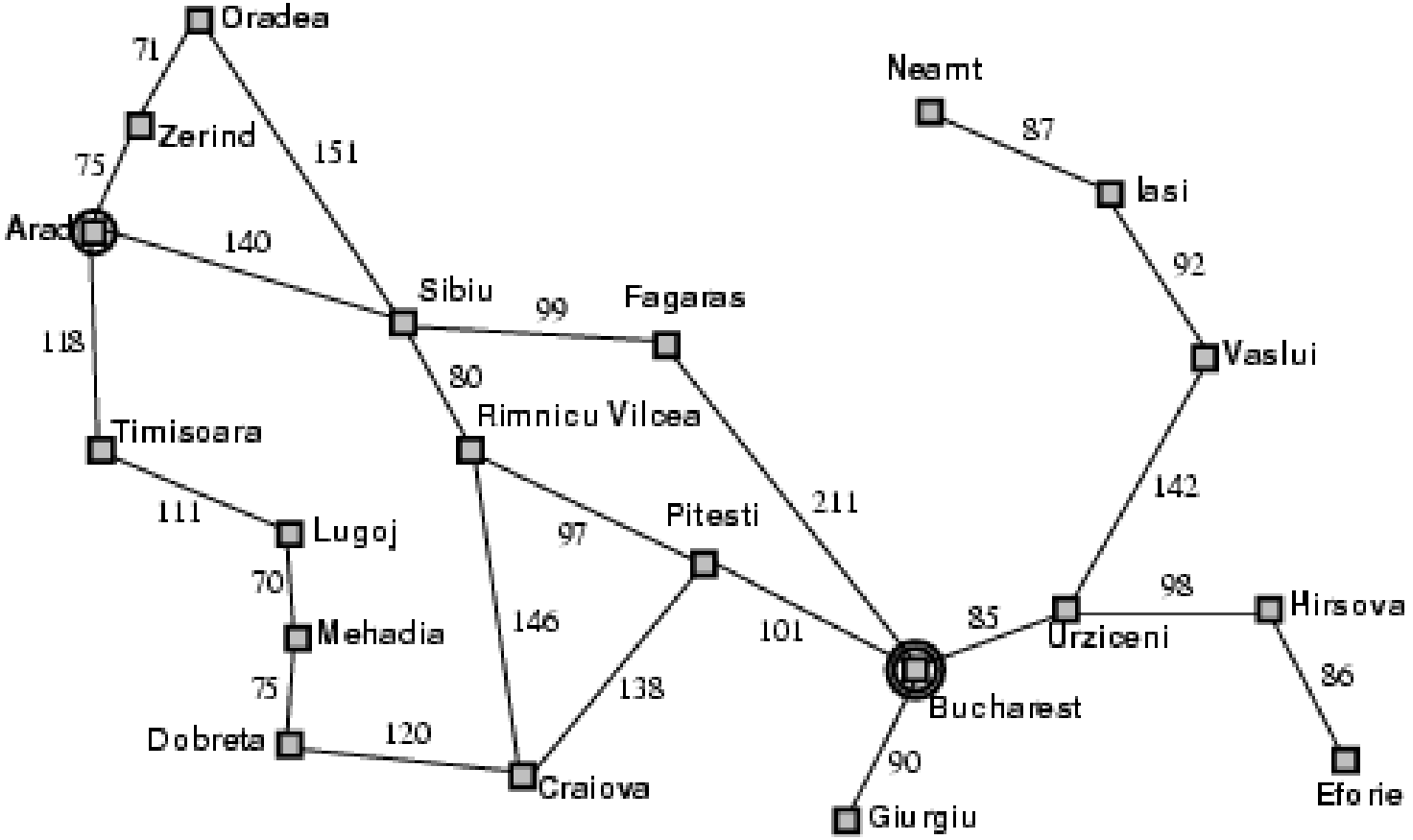


Goal State

The Eight Puzzle

- States
  - Obvious representation of puzzle (9-tuple)
- Actions
  - Move Empty Tile: Up, Down, Left, Right
- Goal Test
  - Are all pieces in appropriate place (state == goal)
- Path Cost
  - Positive constant for each action

Route Finding



Route Finding

- States
  - Traveler's location
- Actions
  - Move to new location (based on current location)
- Goal Test
  - Is traveler at the destination
- Path Cost
  - Distance traveled

Touring Problems

- Visit all cities on a map minimizing some cost
- Visit all cities on a map exactly once minimizing some cost

Recap: Important Concepts

- Many problems can be formulated as search
- Search is a general (weak) method for solving a tricky problem
  - View the problem as a set of states (state space)
  - Actions link the states together
  - Object is to find a low cost path from current state to a goal state