Solving problems by Searching

Problem Solving Agents

Agents

- Problem-solving agents: Atomic representation
- Planning agents: Factored or Structured representation

Problem-solving agents

- Goal formulation: Based on current situation and agent's performance measure.
- Problem formulation: Process of deciding what actions and states to consider, given a goal
- Search
- Execution

- Open loop system
- Closed loop system

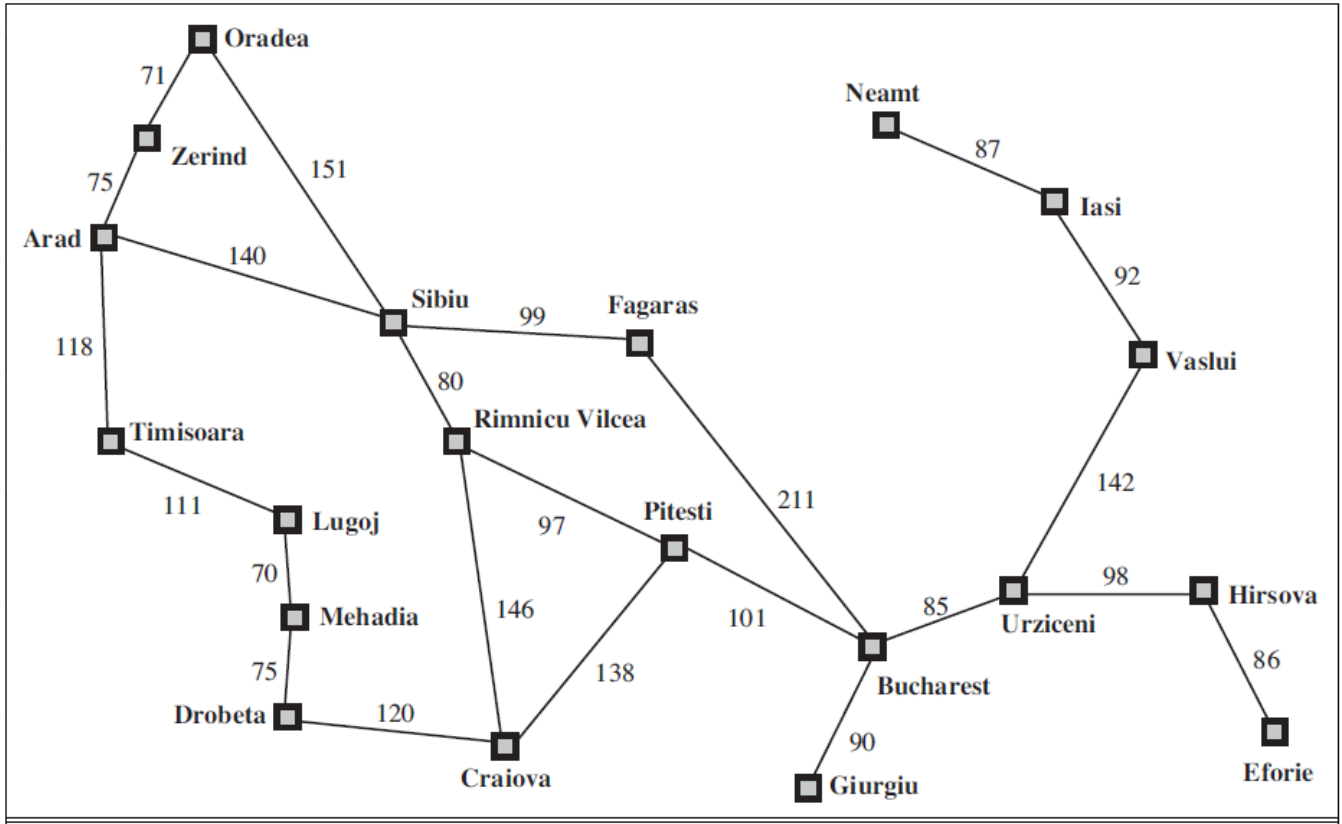

Figure 3.2 A simplified road map of part of Romania.

Formulate, Search, Execute design

```
function SIMPLE-PROBLEM-SOLVING-AGENT(percept) returns an action
  persistent: 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
      goal ← FORMULATE-GOAL(state)
      problem ← FORMULATE-PROBLEM(state, goal)
      seq ← SEARCH(problem)
      if seq = failure then return a null action
  action ← FIRST(seq)
  seq ← REST(seq)
  return action
```

Figure 3.1 A simple problem-solving agent. It first formulates a goal and a problem, searches for a sequence of actions that would solve the problem, and then executes the actions one at a time. When this is complete, it formulates another goal and starts over.

Search problems and solutions

- State space
- Initial state
- Goal state
- Actions ACTIONS (Arad) = {Go(Sibiu); Go(Timisoara); Go(Zerind)}
- Transition model RESULT(In(Arad), Go(Zerind)) = In(Zerind)
- Action cost function (s, a, s')
- Path
- Solution (Optimal)

Graph (Map of Romania)

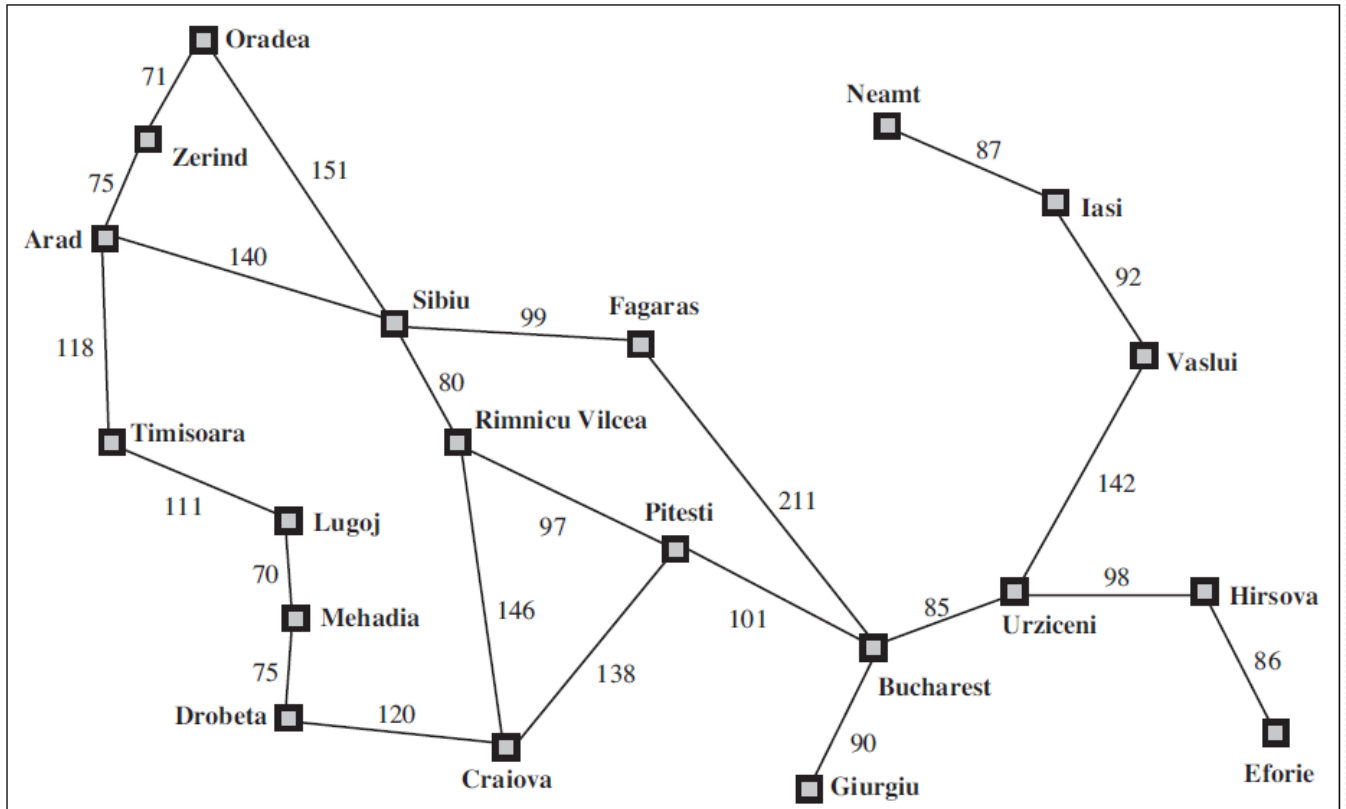

Figure 3.2 A simplified road map of part of Romania.

Formulating problems

- Model
- Abstraction
- Level of Abstraction