Course: Introduction to AI

Prof. Ahmed Guessoum
The National Higher School of AI

Chapter 3

Search

Outline

- Problem-Solving Agents
  - Well-defined problems and solutions
  - Formulating problems
- Example Problems
- Searching for Solutions
- Infrastructure for search algorithms
  - Measuring problem-solving performance
- Uninformed Search Strategies
  - Breadth-first search
  - Uniform-cost search
  - Depth-first search
  - Depth-limited search
  - Iterative deepening depth-first search
  - Bidirectional search

Outline (cont.)

- Informed (Heuristic) Search Strategies
  - Greedy best-first search
  - A* search
  - Memory-bounded heuristic search
  - Learning to search better
- Heuristic Functions
  - The effect of heuristic accuracy on performance
  - Generating admissible heuristics from relaxed problems
  - Learning heuristics from experience

Problem-Solving Agents

- Goals help organize behaviour by limiting the objectives that the agent is trying to achieve and hence the actions it needs to consider.
- **Goal formulation**: first step in problem solving. It is based on the current situation and the agent’s performance measure.
- **Goal**: *the* state of world in which the goal is satisfied.
- **Agent’s task**: find out how to act, now and in the future, so that it reaches a goal state ➔ what sorts of actions and states it should consider.
- **Problem formulation:** process of deciding what actions and states to consider, given a goal.

Case: Travelling in Romania

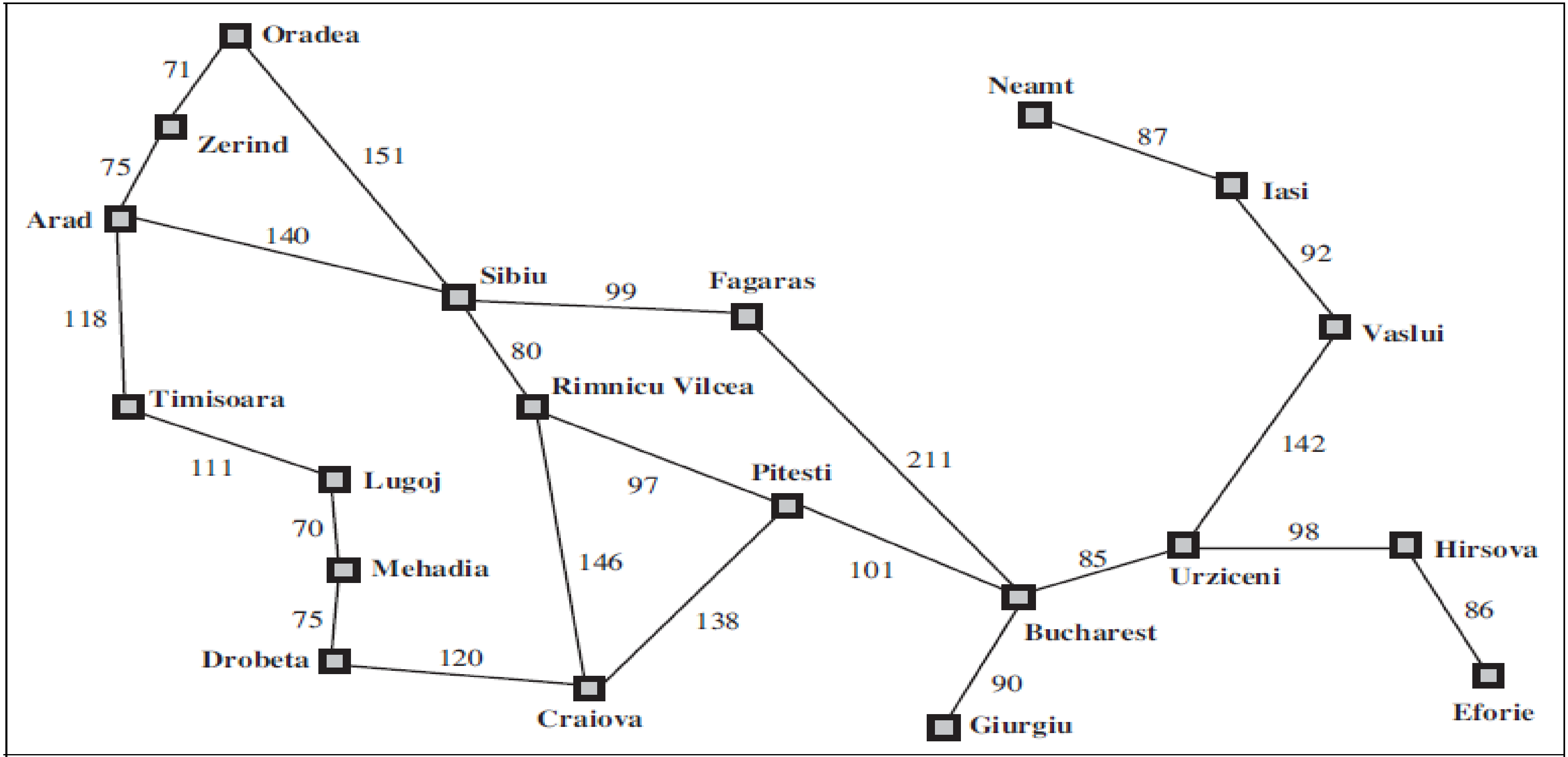


Figure 3.2 A simplified road map of part of Romania.

- Suppose Agent needs to go from Arad to Bucharest.
- Three possible alternative first steps.

Problem-Solving Agents

- If the agent has no additional information, i.e. the environment is **unknown** (Chap. 2), then only choice: try one of the actions at random.
- Suppose the agent has a map of Romania, i.e. the agent has information about the states it might get itself into and the actions it can take, it can find a sequence of states to reach the goal.
  - Once it has found a path on the map from Arad to Bucharest, it can achieve its goal by carrying out the driving actions.

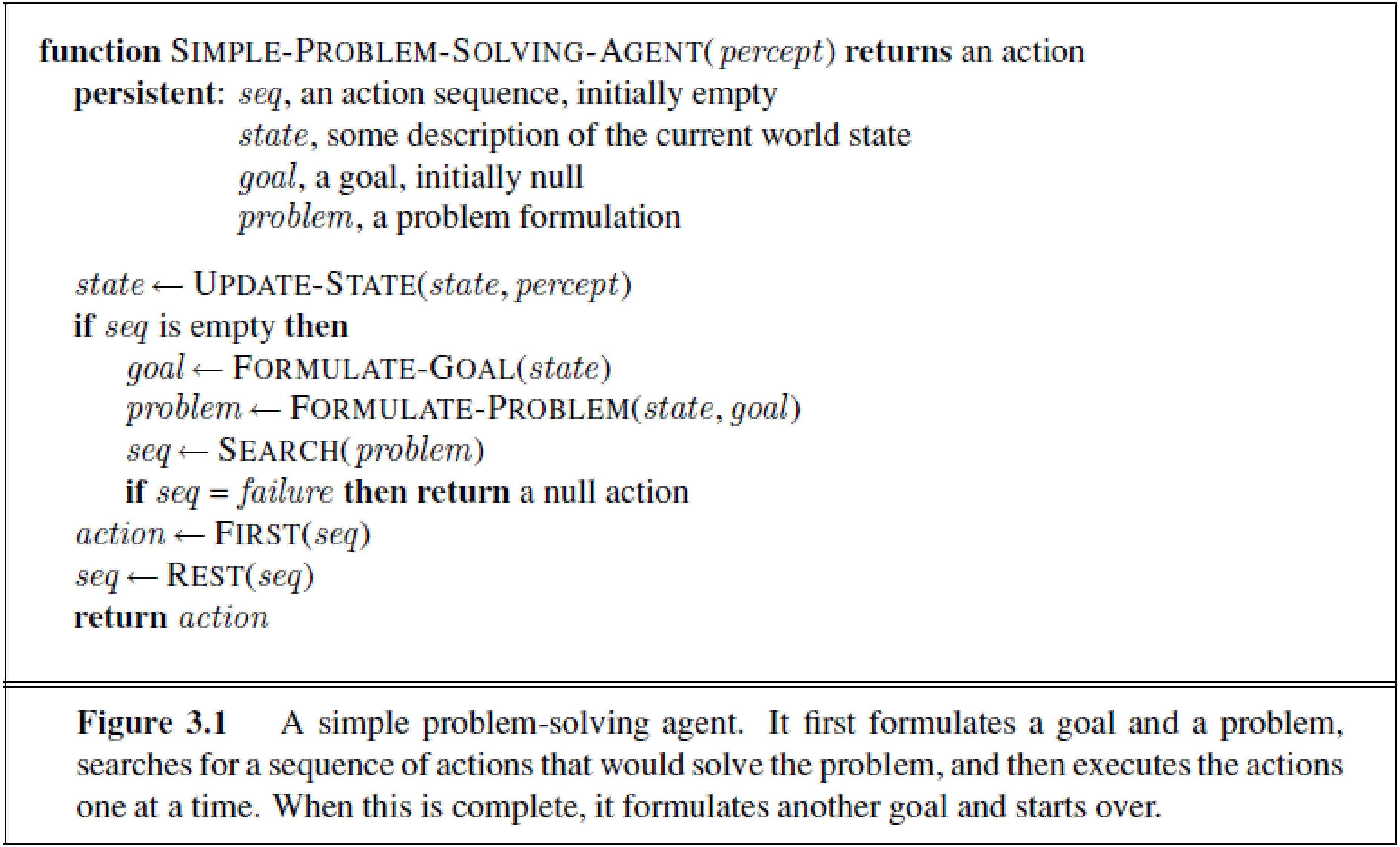
Problem Assumptions

We will assume the environment is :

- **Observable**: the agent always knows the current state.
- **Discrete:** at any given state there are only finitely many actions to choose from.
- **Known**: the agent knows which states are reached by each action.
- **Deterministic**: each action has exactly one outcome.
- N.B.: The above are ideal conditions.
- The process of looking for a sequence of actions that reaches the goal is called ***Search***.
- A search algorithm takes a problem as input and returns a **solution** in the form of an action sequence to execute.

Formulate Goal → Formulate problem → Search → Execute

Simple Problem-Solving Agent



```
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.

Well-defined problems & solutions

Formal definition of a problem:

- The **initial state:** where the agent starts
- A description of all the possible **actions** available to the agent in a given state.
- The **transition model:** description of what each action does.

Together, the initial state, actions, and transition model implicitly define the **state space** of the problem—*the set of all states reachable from the initial state by any sequence of actions*

- The **goal test**: determines whether a given state is a goal state.
- A **path cost** function: assigns a numeric cost to each path. It reflects the agent’s performance measure.

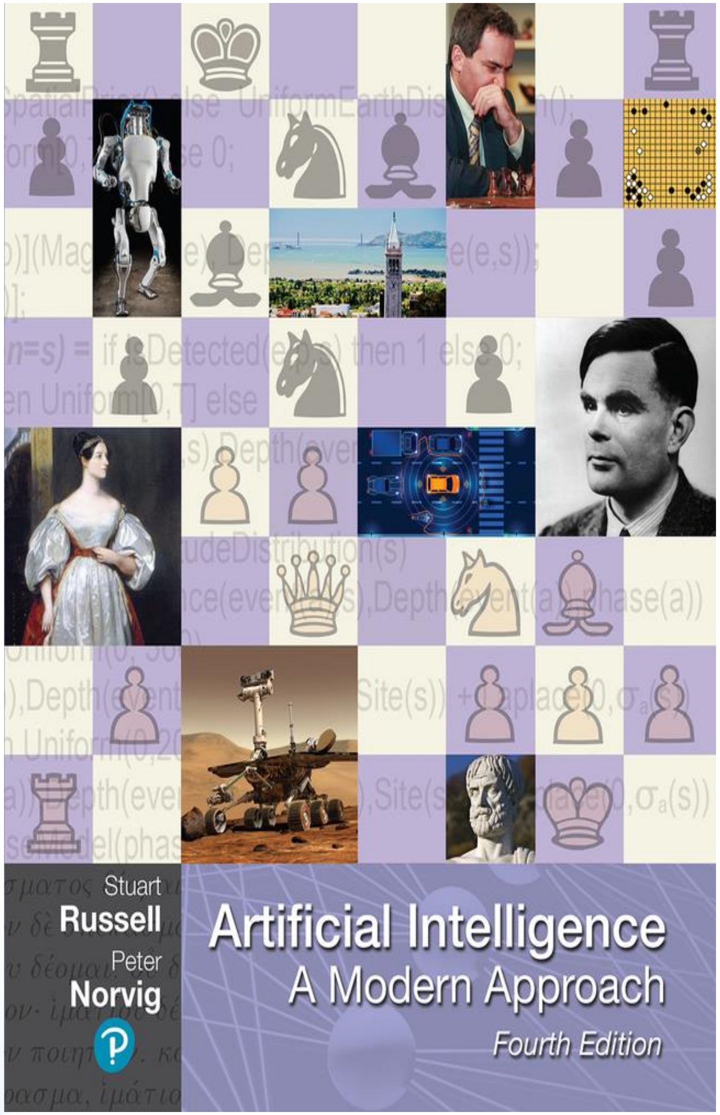
Case of travelling in Romania

- **Initial state:** Arad
- Possible **actions** from *Arad*: {*Go(Sibiu), Go(Timisoara), Go(Zerind)*}
- The **transition model** contains all descriptions like RESULT(In(Arad),Go(Zerind)) = In(Zerind)
- **Goal test**: singleton set {In(Bucharest)}
- **Path cost:** might be length in kilometres.

(In this chapter) We will assume that the cost of a path can be described as the *sum* of the costs of the individual actions along the path.

- The **step cost** of taking action a in state s to reach state s' is denoted by c(s, a, s').

Slides based on the textbook



- Russel, S. and Norvig, P. (2020) Artificial Intelligence,

A Modern Approach (4th Edition), Pearson Education Limited.