Artificial Intelligence

Specifying the task environment

- In designing an agent, the first step must always be to specify the task environment as fully as possible
- We call this the PEAS (Performance, Environment, Actuators, Sensors) description

PEAS description of the task environment for an automated taxi

Agent Type	Performance Measure	Environment	Actuators	Sensors
Taxi driver	Safe, fast, legal, comfortable trip, maximize profits	Roads, other traffic, pedestrians, customers	Steering, accelerator, brake, signal, horn, display	Cameras, sonar, speedometer, GPS, odometer, accelerometer, engine sensors, keyboard

Properties of task environments

- **Fully observable vs. partially observable**: If an agent's sensors give it access to the complete state of the environment at each point in time, then we say that the task environment is fully observable
- An environment might be partially observable because of noisy and inaccurate sensors or because parts of the state are simply missing from the sensor data
- For example, a vacuum agent with only a local dirt sensor cannot tell whether there is dirt in other squares, and an automated taxi cannot see what other drivers are thinking

- If the agent has no sensors at all then the environment is unobservable. One might think that in such cases the agent's plight is hopeless, but the agent's goals may still be achievable, sometimes with certainty
- **Single agent vs. multiagent**: For example, an agent solving a crossword puzzle by itself is clearly in a single-agent environment, whereas an agent playing chess is in a two-agent environment
- **Deterministic vs. Stochastic:** If the next state of the environment is completely determined by the current state and the action executed by the agent, then we say the environment is deterministic; otherwise, it is stochastic

Solving problems by searching

- Reflex agents base their actions on a direct mapping from states to actions
- Such agents cannot operate well in environments for which this mapping would be too large to store and would take too long to learn
- Goal-based agents, on the other hand, consider future actions and the desirability of their outcomes
- One kind of goal-based agent called a problem-solving agent

- **Uninformed search algorithms**—algorithms that are given no information about the problem other than its definition
- Although some of these algorithms can solve any solvable problem, none of them can do so efficiently
- **Informed search algorithms**, on the other hand, can do quite well given some guidance on where to look for solutions
- Intelligent agents are supposed to maximize their performance measure

Problem solving

- **Goal formulation**, based on the current situation and the agent's performance measure, is the first step in problem solving
- **Problem formulation** is the process of deciding what actions and states to consider, given a goal

Well-defined problems and solutions

A problem can be defined formally by five components:

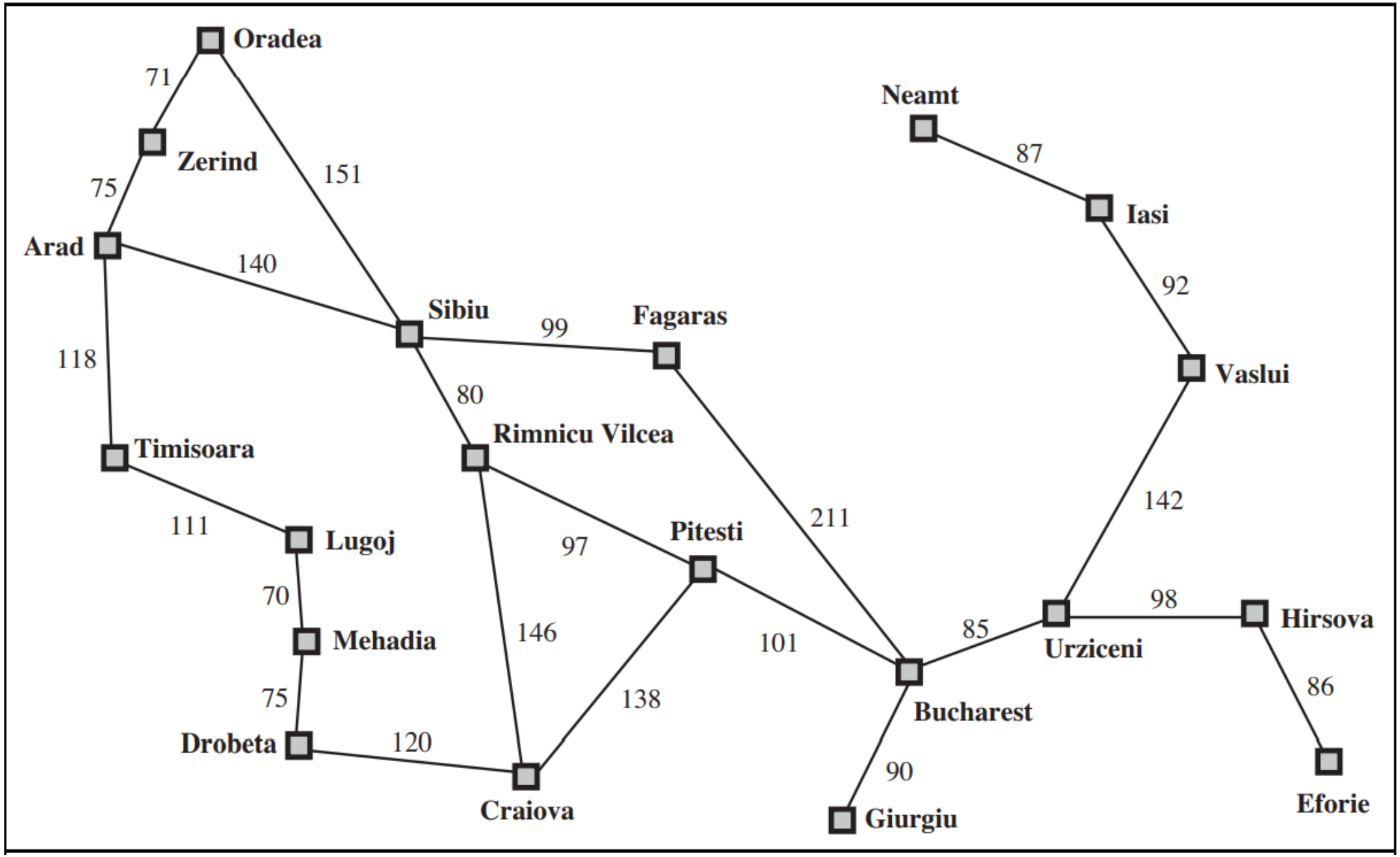
- The initial state that the agent starts in. For example, the initial state for our agent in Romania might be described as In(Arad)
- A description of the possible actions available to the agent. Given a particular states, ACTIONS(s) returns the set of actions that can be executed in s
- We say that each of these actions is applicable in s. For example, from the state In(Arad), the applicable actions are {Go(Sibiu), Go(Timisoara), Go(Zerind)}

- A description of what each action does; the formal name for this is the transition model, specified by a function RESULT(s, a) that returns the state that results from doing action a in state s

  RESULT(In(Arad),Go(Zerind)) = In(Zerind)

- 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 state space forms a directed network or graph in which the nodes are states and the links between nodes are actions

- A path in the state space is a sequence of states connected by a sequence of actions
- The goal test, which determines whether a given state is a goal state. Sometimes there is an explicit set of possible goal states, and the test simply checks whether the given state is one of them. The agent’s goal in Romania is the singleton set {In(Bucharest)}


Oradea
71
Neamt
Zerind
87
151
75
Iasi
Arad
140
92
Sibiu
99
Fagaras
118
Vaslui
80
Rimnicu Vilcea
Timisoara
142
111
211
Pitesti
Lugoj
97
98
70
Hirsova
Mehadia
146
85
Urziceni
101
75
86
Bucharest
138
Drobeta
120
90
Craiova
Giurgiu
Eforie

- A path cost function that assigns a numeric cost to each path
- The problem-solving agent chooses a cost function that reflects its own performance measure
- For the agent trying to get to Bucharest, time is of the essence, so the cost of a path might be its length in kilometers
- A solution to a problem is an action sequence that leads from the initial state to a goal state
- Solution quality is measured by the path cost function, and an optimal solution has the lowest path cost among all solutions

Formulating problems

- State of the world includes so many things: the traveling companions, the current radio program, the scenery out of the window, the proximity of law enforcement officers, the distance to the next rest stop, the condition of the road, the weather, and so on
- All these considerations are left out of our state descriptions because they are irrelevant to the problem of finding a route to Bucharest
- The process of removing detail from a representation is called abstraction
- The actions are being abstracted. Besides changing the location of the vehicle and its occupants, it takes up time, consumes fuel, generates pollution, and so on

- Our formulation takes into account only the change in location
- Also, there are many actions that we omit altogether: turning on the radio, looking out of the window, slowing down for law enforcement officers, and so on
- And of course, we don’t specify actions at the level of “turn steering wheel to the left by one degree"