Problem-Solving Agents

(PSAs)

What is a PSA?

- Problem-Solving Agents are goal-based agents!
- They find a sequence of actions to maxize their performance measurment to achieve their goal.
- **Idea**: initial state → goal state
- **Goal** = set of states that maxize their performance measurment.

More about PSAs

- **Problem formulation** = actions and states to be considered, given a goal.
- **Solution**: Sequence of actions.
- **Solution Space** = space of states and actions.
- **Solution method** = search in the space state of actions and states.
- **Search process** = process that analyze all the posible sequences of actions that can end in the better sequence (goal and solution).

General algorithm of a PSA

```
function PSA-SIMPLE(p) returns action
    input: p (perception)
    static: sec (sequence of actions), e (state), m (goal),
      pr (problem)

    e ← UPDATE-STATE(e,p)
    if sec = ∅ then do
        m ← FORMULATE-GOAL(e)
        pr ← FORMULATE-PROBLEM(e, m)
        sec ← SEARCH(pr)
    action ← FIRST(sec)
    sec ← REST(sec)
    return action
```

Environment

- Static
- Fully observable
- Discrete
- Deterministic

Problem formulation

- **Initial State** = State where the agent starts to solve the problem.
- **Set of possible actions:**
  - Succesor Function = given a state it returns a set of all possible and valid succesor states
  - State Space = set of all the possible states reachable from the initial state by any sequence of actions.
  - Representation of the Space of States = directed network or graph.
  - Path = a sequence of stares connected by a sequence of actions.
- **Goal test** = which determines whether a given state is a goal state.
- **Step cost and path cost** = function that assigns a numeric cost to each path.

Types of problems

- Toy problems
  - They are intended to illustrate or exercise various problem-solving methods.
- Real-world problem
  - It is one whose solutions people actually care about.

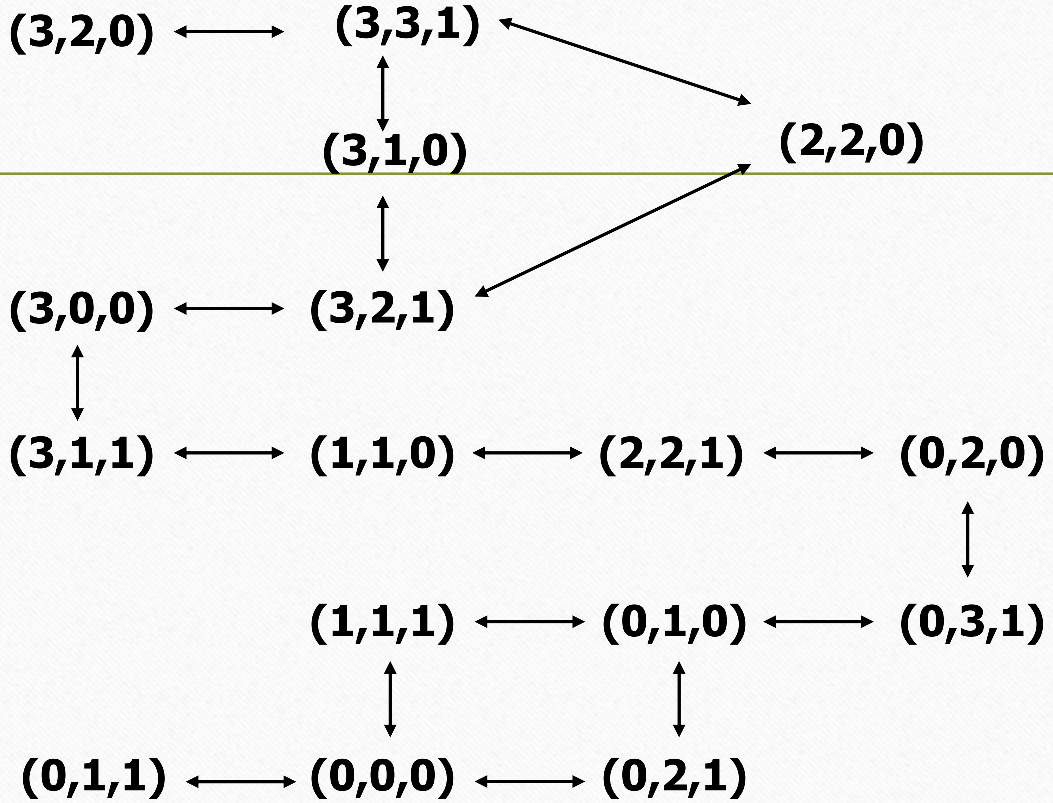
Missionaries and Cannibals

- Three missionaries and three hungry cannibals are on one side of a river.
- They have a boat to go to the other side of the river, but only two persons can travel on this boat.
- Missionaries have to find the way they can cross the river all of them.
- But in any moment should be more cannibals than missionaries in the same side.

Problem Formulation

- **States**: sequence of three numbers representing the number of missionaries, cannibals and boat in the initial side.
- **Initial state**: (3, 3, 1)
- **Succesor function**: To cross the river with a missionary, with a cannibal, with two missionaries, with two cannibals, with one missionary and one cannibal.
- **Goal test**: current state = (0,0,0)
- **Step cost**: each step costs 1, so the **path cost** is the number of steps in the path.

Space of states



(3,2,0) (3,3,1)

(3,1,0) (2,2,0)

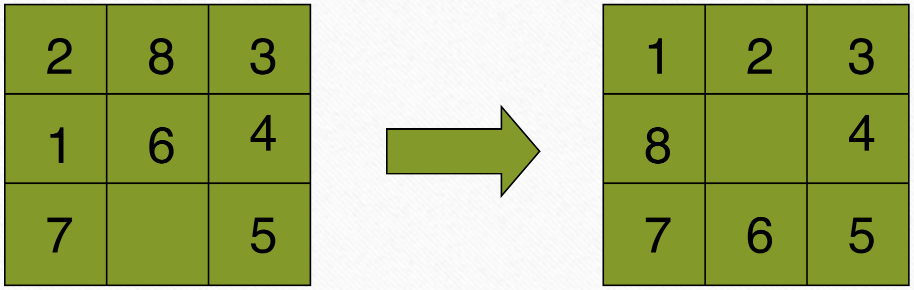
(3,0,0) (3,2,1)

(3,1,1) (1,1,0) (2,2,1) (0,2,0)

(1,1,1) (0,1,0) (0,3,1)

(0,1,1) (0,0,0) (0,2,1)

8-PUZZLE (another exmple)

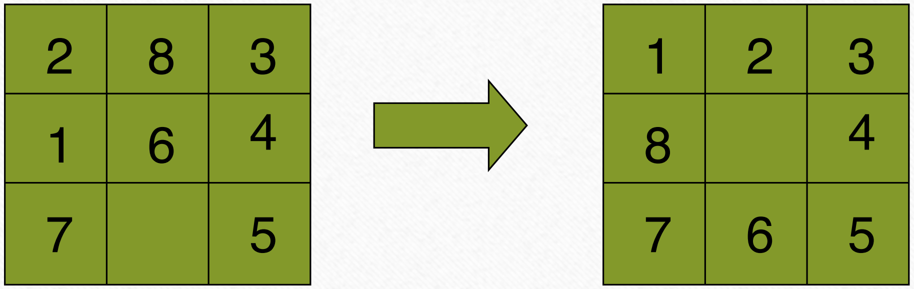


2	8	3
1	6	4
7		5

1	2	3
8		4
7	6	5

Problem Formulation

- Initial state: random configuration
- Goal state:
  curren state = final state

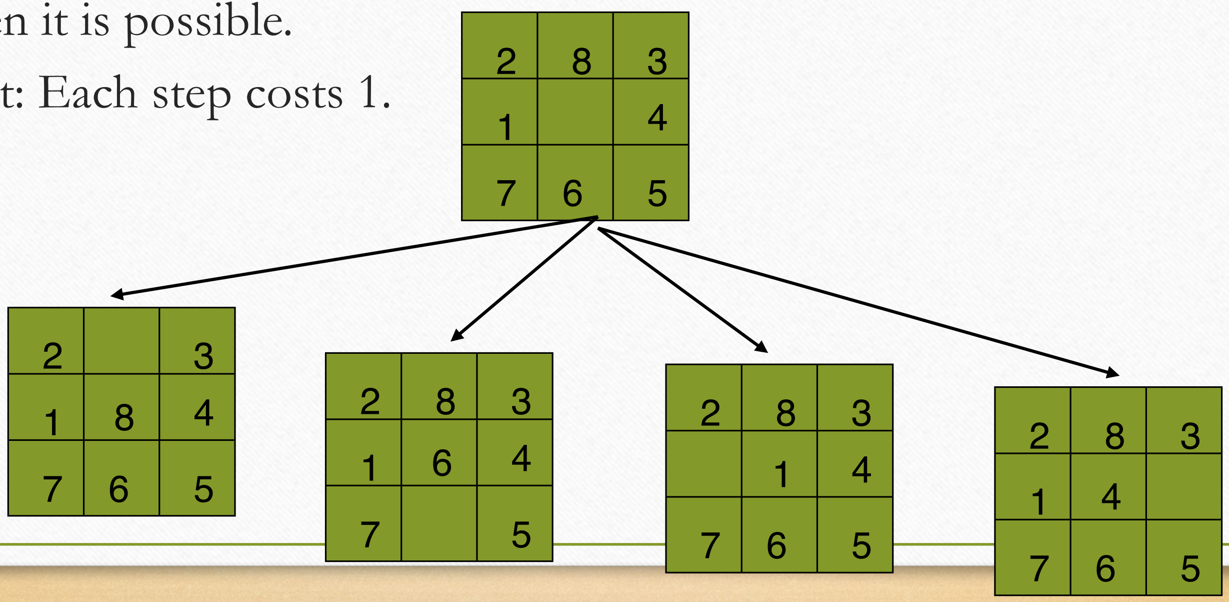


2	8	3
1	6	4
7		5

1	2	3
8		4
7	6	5

Problem Formulation

- Succesor function: moving the empty cell up, down, right, left, when it is possible.
- Step cost: Each step costs 1.



2	8	3
1		4
7	6	5

2		3
1	8	4
7	6	5

2	8	3
1	6	4
7		5

2	8	3
	1	4
7	6	5

2	8	3
1	4	
7	6	5

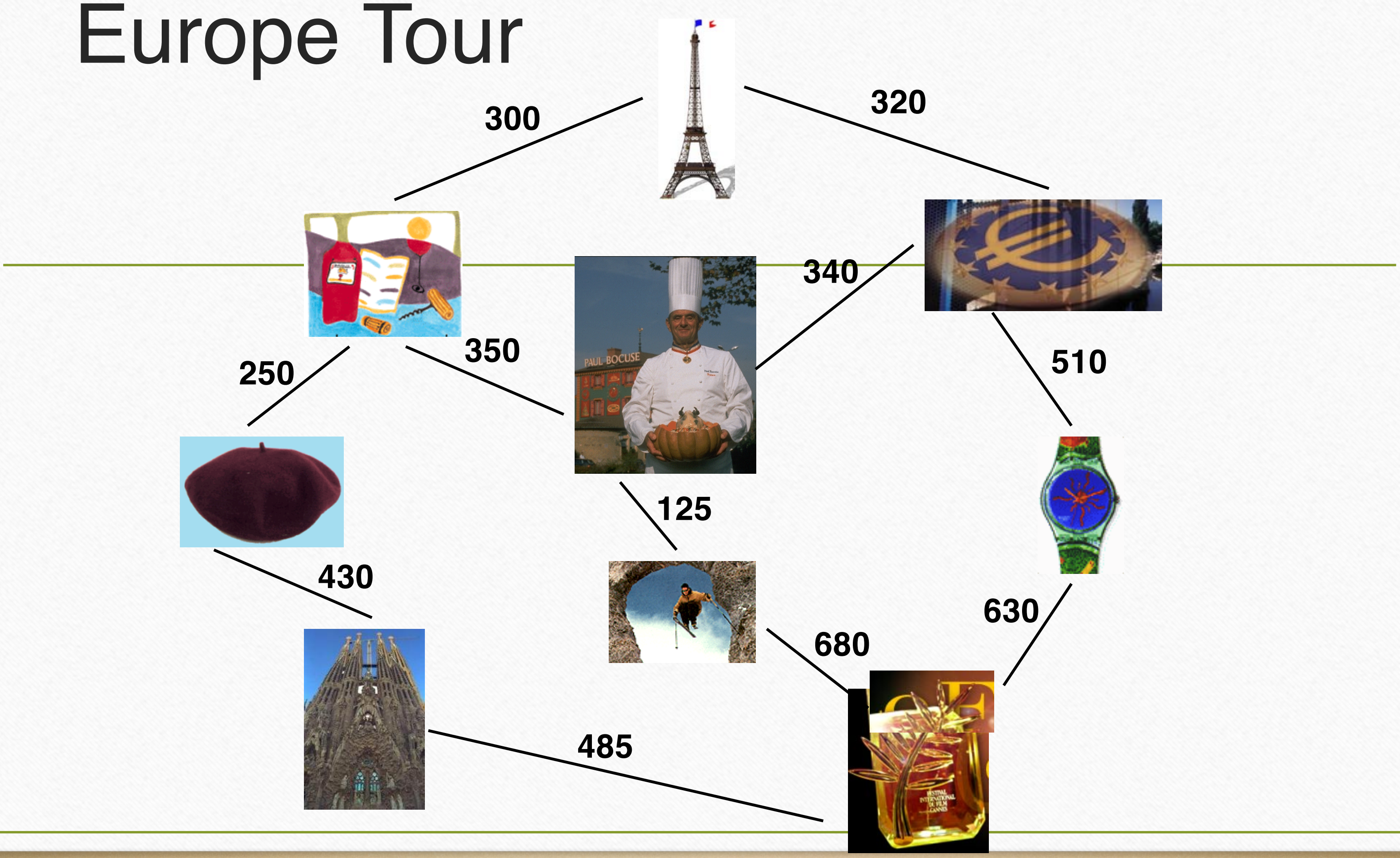
Other toy problems

- Vacuum world from the textbook AIMA (not the Roomba robot that we know).
- 8-queens problem is to place 8 queens on a chessboard such that no queen attacks any other.
- …

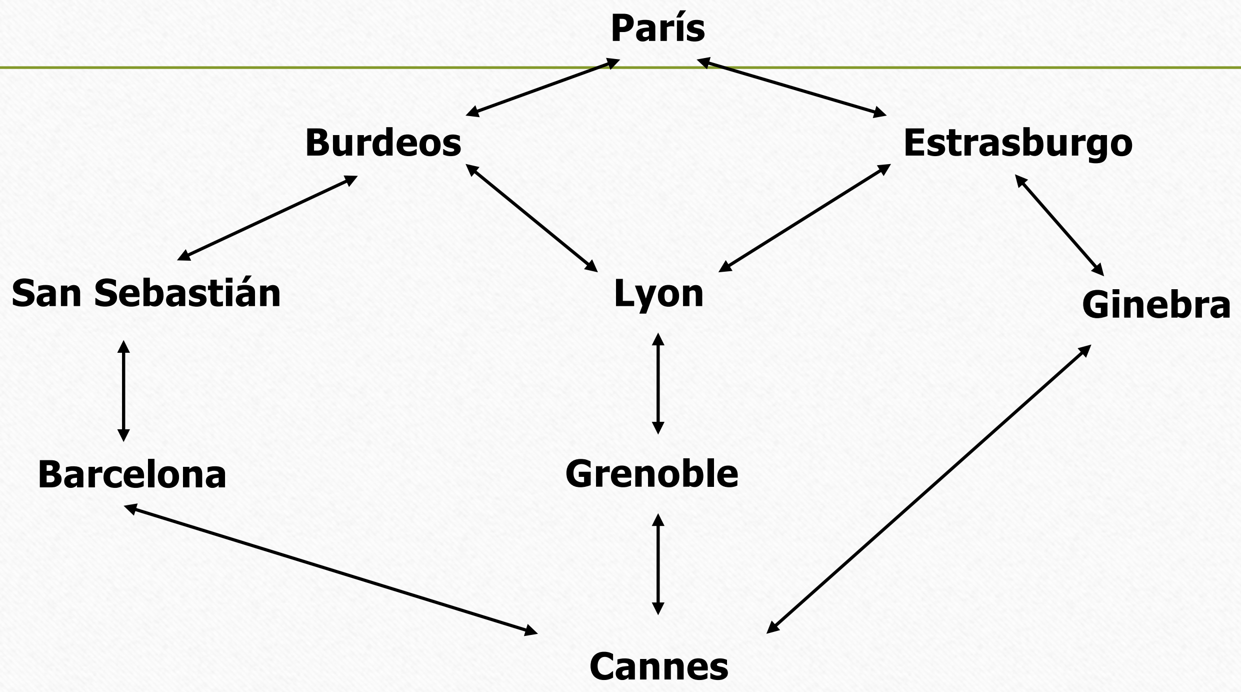
Traveling Salesman Problem (real)

- The agent needs to find a path between two points in a map. For example from Paris to Cannes

Europe Tour



Space of states



Problem Formulation

- **States**: we use a graph where each node represents a city on the map.
- **Initial State**: node of the starting city (Paris).
- **Succesor function**: travel from one node to the next node
- **Goal test**: current state = node of th goal city (Cannes).
- **Step cost**: the distance between any pair of adjacent nodes. So the p**ath cost** is the sum of distances traveled so far from the initial node.

Other real applications

- Travel salesman problem
- VLSI
- Robot navegation
- Internet search
- Logistics
- …