UNIVERSITY OF WATERLOO

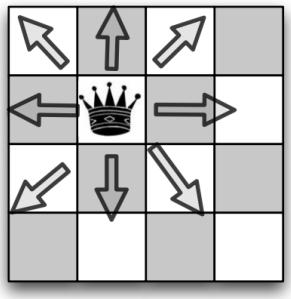
Department of Electrical and Computer Engineering

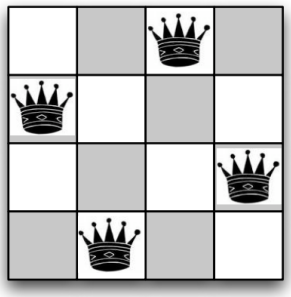
ECE 457A Adaptive and Cooperative Algorithms

Instructor: Dr. Otman A. Basir

Problem Formulation

Example 1. The n-queens problem is a popular problem originally proposed by a chess player in 1848. It requires putting n queens on an n x n chess board such that none of the queens is attacking any of the other queens. In case you're unfamiliar with chess, a queen can move in any direction for any number of squares:

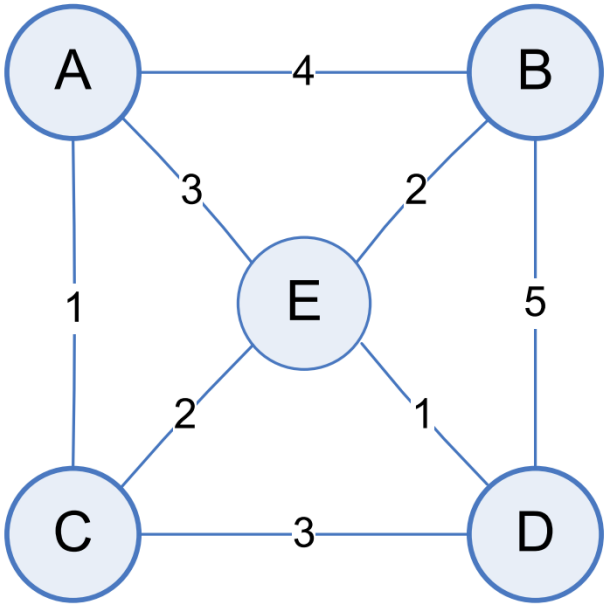




The board on the right, above, shows a solution to the 4-queens problem. Finding a solution to the n-queens problem can be thought of as a search problem.

- Formulate n-queens as a search problem. This means you should specify: a. Start state: empty board
- Successor function: add a queen to an empty square if doing so results in n or fewer queens
- Search space (what does a valid state in the search tree look like): chess board with n or fewer queens, each on a different square
- Goal test: a state is the goal if it has exactly n queens and no queen is attacking any other queen (obviously, this can be stated more formally)
- Like many search problems, the n-queens problem can be parameterized in a number of different ways to create a valid search problem. Come up with a different way of formulating it as a search problem, specifying the same four items as in (1).

Example 2. Traveling Salesman Problem:
The travelling salesman problem (also called the traveling salesperson problem[1] or TSP) asks the following question: "Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?" It is an NP-hard problem in combinatorial optimization, important in theoretical computer science and operations research.

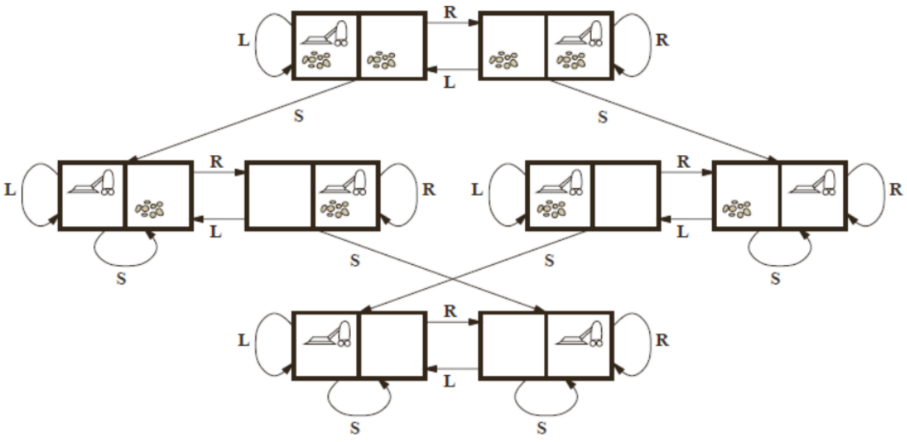


Formulation:
– States: cities
– Initial state: A
– Successor function: Travel from one city to another connected by a road
– Goal test: the trip visits each city only once that starts and ends at A.
– Path cost: traveling time

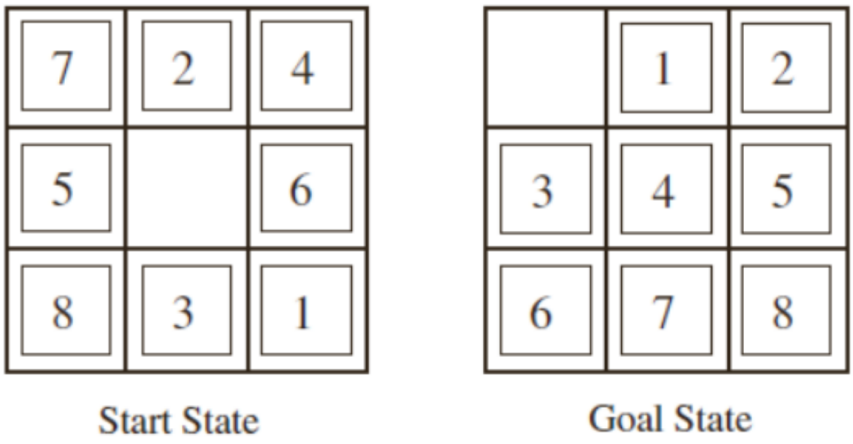
Example 3. Vacuum world
States:
• The state is determined by both the agent location and the dirt locations.
• The agent is in one of two locations, each of which might or might not contain dirt.
• Thus, there are 2×2
$2 = 8$ possible world states.
Initial state:
Our vacuum can be in any state of the 8 states shown in the picture.
State description:
Successor function generates legal states resulting from applying the three actions Left, Right, and Suck.
The states space is shown in the picture, there are 8 world states.
Goal test:
Checks whether all squares are clean.
Path cost:
Each step costs 1, so the path cost is the sum of steps in the path.

vacuum world: state space for the vacuum world

- Links denote actions: L = Left, R =Right, S = Suck.



Example 4. The 8-puzzle consists of a 3×3 board with eight numbered tiles and a blank space.
- A tile adjacent to the blank space can slide into the space.
- The object is to reach a specified goal state.

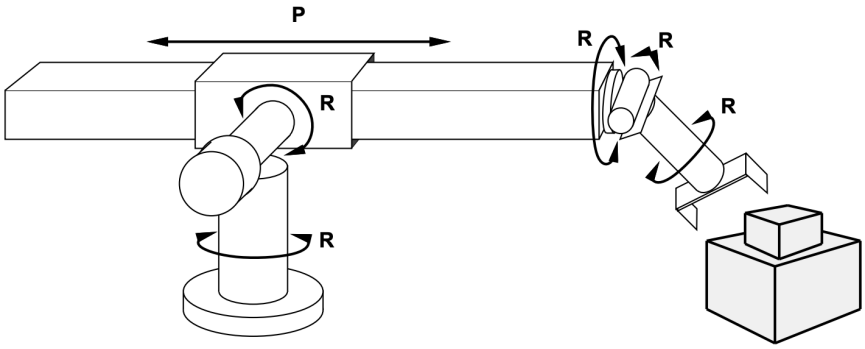


States: A state specifies the location of each of the eight tiles and the blank in one of the nine squares.
Initial state: Any state can be designated as the initial state.
- Note that goal can be reached from exactly half of the possible initial states.

Actions: Movements of the blank space Left, Right, Up, or Down.
- Different subsets of these are possible depending on where the blank is.

Transition model: Given a state and action, this returns the resulting state;
Goal test: This checks whether the state matches the goal configuration
Path Cost: Each step costs 1, so the path cost is the number of steps in the path.
- The 8-puzzle belongs to the family of sliding-block puzzles,
- This family is known to be NP-complete.
- Optimal solution of n-Puzzle family is NP-hard. ie NO polynomial solution for the problem.
- The 8-puzzle has 9!/2=181, 440 reachable states.
- The 15-puzzle (on a 4×4 board) has around 1.3 trillion states,
- The 24-puzzle (on a 5 × 5 board) has around 1025 states

Example 5. Robot Arm Path Planning



States real-valued coordinates of robot joint angles and parts of the object to be assembled

Actions continuous motions of robot joints

Goal test assembly complete?

Path cost time to execute