Module: 3.1

Problem Solving and Search

Motivation:

This module 3.1: will brief about problem formulation approach in AI.

Syllabus:

Lecture no	Content	Duration (Hr)	Self-Study (Hrs)
1	Problem solving agent	1	1
2	Formulating Problems, Example Problems	1	1

Learning Objective:

- Learner should know about various real time problems solved with the help of problem solving agent.

- **Theoretical Background:**

Searching is the universal technique of problem solving in AI. There are some single-player games such as tile games, Sudoku, crossword, etc. The search algorithms help you to search for a particular position in such games.

Key Definitions:

- **Problem Space** − It is the environment in which the search takes place. (A set of states and set of operators to change those states)
- **Problem Instance** − It is Initial state + Goal state.
- **Problem Space Graph** − It represents problem state. States are shown by nodes and operators are shown by edges.
- **Depth of a problem** − Length of a shortest path or shortest sequence of operators from Initial State to goal state.
- **Space Complexity** − The maximum number of nodes that are stored in memory.
- **Time Complexity** − The maximum number of nodes that are created.
- **Admissibility** − A property of an algorithm to always find an optimal solution.
- **Branching Factor** − The average number of child nodes in the problem space graph.
- **Depth** − Length of the shortest path from initial state to goal state.

Course Content:

Lecture : 1

Problem solving agent

In Artificial Intelligence, Search techniques are universal problem-solving methods. **Rational agents** or **Problem-solving agents** in AI mostly used these search strategies or algorithms to solve a specific problem and provide the best result. Problem-solving agents are the goal-based agents and use atomic representation. In this topic, we will learn various problem-solving search algorithms.

Brute-Force Search Strategies

They are most simple, as they do not need any domain-specific knowledge. They work fine with small number of possible states.

Requirements −

- State description
- A set of valid operators
- Initial state
- Goal state description

Let's check the take away from this lecture

Exercise

Q. 1 Problem-solving agents are the…………………

(a) Simple reflex agents

(b) Model based agents

(c) goal-based agents

(d) Learning agents

Learning from this lecture: Learners will be able to understand about problem solving agents

Lecture : 2

Formulating Problems, Example Problems

Formulating problems : A problem is defined by four items:

1. initial state e.g., "at Arad"
2. actions or successor function or operators

   S(x) = set of action–state pairs

   – e.g., *S(Arad) = {Arad Zerind, Zerind , ... }*

3. goal test, can be
   - explicit, e.g., x = "at Bucharest"
   - implicit, e.g., *Checkmate(x)*
4. path cost function (additive)
   - e.g., sum of distances, number of actions executed, etc.
   - $c(x,a,y)$ is the step cost, assumed to be ≥ 0

A solution is a sequence of actions leading from the initial state to a goal state.

A search strategy is defined by picking the order of node expansion. Strategies are evaluated along the following dimensions:

- completeness: does it always find a solution if one exists?
- time complexity: time taken to find solution
- space complexity: maximum number of nodes in memory
- optimality: does it always find a least-cost solution?

Time and space complexity are measured in terms of

- b: maximum branching factor (state expanded to yield new states) of the search tree
- d: depth of the least-cost solution
- m: maximum depth of the state space (may be ∞)

Example Problem

The 8-puzzIe: The **8-puzzle** consists of a 3x3 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 the configuration shown on the right of the figure. One important trick is to notice that rather than use operators such as "move the 3 tile into the blank space," it is more sensible to have operators such as "the blank space changes places with the tile to its left." This is because there are fewer of the latter kind of operator. This leads us to the following formulation:

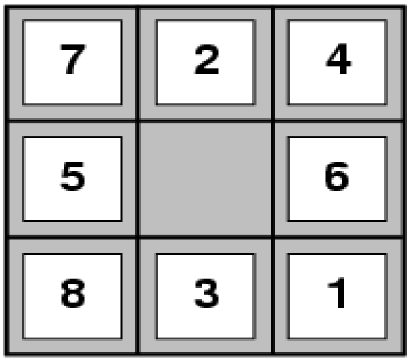
States: a state description specifies the location of each of the eight tiles in one of the nine squares. For efficiency, it is useful to include the location of the blank.

Operators: blank moves left, right, up, or down.

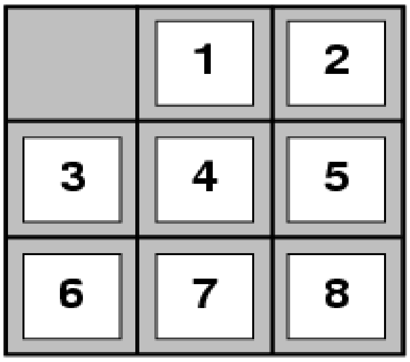
Goal test: state matches the goal configuration shown in Figure below

Path cost: each step costs 1, so the path cost is just the length of the path.

The 8-puzzle belongs to the family **of sliding-block puzzles.** This general class is known to be NP-complete, so one does not expect to find methods significantly better than the search algorithms described in this module and the next. The 8-puzzle and its larger cousin, the 15-puzzle, are the standard test problems for new search algorithms in Al.



Start State



Goal State

- states? locations of tiles
- actions? move blank left, right, up, down
- goal test? = goal state (given)
- path cost? 1 per move

The 8-queens problem: The goal of the 8-queens problem is to place eight queens on a chessboard such that no queen attacks any other. (A queen attacks any piece in the same row, column or diagonal.) Figure shows an attempted solution that fails: the queen in the rightmost column is attacked by the queen at top left. Although efficient special-purpose algorithms exist for this problem and the whole *n*queens family, it remains an interesting test problem for search algorithms. There are two main kinds of formulation:

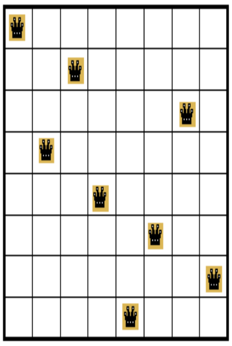
The *incremental* formulation involves placing queens one by one, whereas

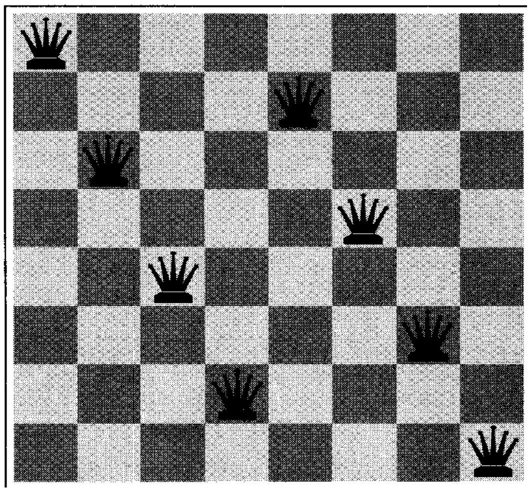
The *complete-state* formulation starts with all 8 queens on the board and moves them around. In either case, the path cost is of no interest because only the final state counts; algorithms are thus compared only on search cost. Thus, we have the following goal test and path cost:

Goal test: 8 queens on board, none attacked.

Path cost: zero.

- The eight queens problem is to place eight queens on the empty chessboard in such a way that no queen attacks any other queens.





There are also different possible states and operators. Consider the following simple-minded formulation:

States: any arrangement of 0 to 8 queens on board.

Operators: add a queen to any square.

In this formulation, we have 64^8 possible sequences to investigate. A more sensible choice would use the fact that placing a queen where it is already attacked cannot work, because subsequent placings of other queens will not undo the attack. So we might try the following:

States: arrangements of 0 to 8 queens with none attacked.

Operators: place a queen in the left-most empty column such that it is not attacked by any other queen.

It is easy to see that the actions given can generate only states with no attacks; but sometimes no actions will be possible. For example, after making the first seven choices (left-to-right) in Figure, there is no action available in this formulation. The search process must try another choice. A quick calculation shows that there are only 2057 possible sequences to investigate. The right formulation makes a big difference to the size of the search space. Similar considerations apply for a complete-state formulation. For example, we could set the problem up as follows:

States: arrangements of 8 queens, one in each column.

Operators: move any attacked queen to another square in the same column. This formulation would allow the algorithm to find a solution eventually, but it would be better to move to an unattacked square if possible.

Let’s check the take away from this lecture

Exercise

Q.1 Following is not a state of problem solving……………

(a) State description

(b)A set of valid operators

(c) Initial state

(d) Idle

Q.2 Goal State can be …………..

(a) Implicit

(b) Explicit

(c) None of above

(d) both of above

Learning from this lecture: Learners will be able to understand how to Formulate Problems with Examples

Conclusion

This chapter was introduction to intelligent systems which covered Definition, History, Categorization, Components, Foundation, , Applications and recent trends in IS (i.e AI).

Short Answer Questions:

Q. 1 Define problem solving agent with example.

Q.2 Illustrate problems can be solved using problem formulation approach.

Long Answer Questions:

Q.2 Explain in detail problem formulation with any three examples.

References:

Books:

	Title	Authors	Publisher	Edition	Year	Chapter No
1	Artificial Intelligence a Modern Approach	Stuart J. Russell and Peter Norvig	McGraw Hill	3rd Edition	2009	
2	A First Course in Artificial Intelligence	Deepak Khemani	McGraw Hill Education (India)	1st Edition	2013	

Online Resources:

- https://onlinecourses.nptel.ac.in/noc20_cs81/preview
- https://nptel.ac.in/courses/106/102/106102220/
- https://www.coursera.org/learn/introduction-to-ai/
- https://www.coursera.org/learn/mind-machine-problem-solving-methods
- https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligent_systems.htm
- https://www.javatpoint.com/history-of-artificial-intelligence
- https://people.eecs.berkeley.edu/~russell/
- https://en.wikipedia.org/