**Chapter 3**

**Problem Solving**

**The chapter consist of Short type Questions & Answers , Descriptive Question & Answer and MCQs & answers.**

Contents

[Short type Questions & Answers 1](#_Toc14943609)

[Q What is Problem Solving in AI ? 1](#_Toc14943610)

[Q Define Problem-Solving Agent 2](#_Toc14943611)

[Q Write in short about State–space search Method 2](#_Toc14943612)

[Descriptive Question & Answer 2](#_Toc14943613)

[Q Explain problem characteristics in AI 2](#_Toc14943614)

[Q Explain in detail Problem-Solving Agent 3](#_Toc14943615)

[Q Solve Water jug problem using state space search Method for the given problem 4](#_Toc14943616)

[Q Solve eight tile puzzle using state space search problem 8](#_Toc14943617)

[Q Solve Vacuum cleaner problem using state space search method 9](#_Toc14943618)

[Vacuum cleaner 9](#_Toc14943619)

[Q Explain Problem Solving agent with types of single-state problem multiple-state problem contingency problem and. exploration problem with example. 11](#_Toc14943620)

[Problem Solving Agent 11](#_Toc14943621)

[Q Explain Tic Tac Toe problem until Win /Loose/ Draw condition is not achieved. 14](#_Toc14943622)

[MCQs & Answers 18](#_Toc14943623)

# Short type Questions & Answers

## Q What is Problem Solving in AI ?

In Artificial Intelligence (AI), the term *problem solving* is given to the analysis of how computers can be made to find solutions in well-circumscribed domains. Since puzzles and games have explicit rules, these are often used to explore ideas in AI (whether the insights thus gained will transfer to less formal domains is another issue).

*Problem solving* is defined as the way in which an agent finds a sequence of actions that achieves its goals, when no single action will do. That means an agent has to execute not a single action but set of actions in sequence so as to achieve the goal. Problem formulation usually requires abstracting away real-world details to define a state space that can feasibly be explored

## Q Define Problem-Solving Agent

Intelligent agents are supposed to act in such a way that the environment goes through a sequence of

states that maximises the performance measures (Please frame: Performance measure is discussed later in the

text for now it is a term that defines how successful the agent is. Hence, we need to maximise the performance

measure.) In its full generality, this specification is difficult to translate into successful agent design.

Problem-solving agent is the form of goal-based agent

## Q Write in short about State–space search Method

*State–space search* is a process used in the field of computer science, including AI, in which successive configurations or states of an instance are considered, with the goal of finding a *goal state* with a desired property.

To deal with AI problems such as water–jug problem, eight tile puzzle, wumpus world problem, etc., can

be solved using following steps:

**1.** Describe the states

**2.** Identify the initial state

**3.** Identify the goal state

**4.** Identify set of rules (all possible actions)

**5.** Find the solution path in the state space

When these steps are followed in the problem formulation then a problem is said to be formulated by an approach called as *State–Space Approach*.

# Descriptive Question & Answer

## Q Explain problem characteristics in AI

A problem may have different aspects of representation and explanation. In order to choose the most appropriate method for a particular problem, it is necessary to analyze the problem along several key dimensions. Some of the main key features of a problem are given below.

1. Is the problem decomposable into set of sub problems? Can the solution step be ignored or undone?
2. Is the problem universally predictable?
3. Is a good solution to the problem obvious without comparison to all the possible solutions? Is the desire solution a state of world or a path to a state?
4. Is a large amount of knowledge absolutely required to solve the problem?
5. Will the solution of the problem required interaction between the computer and the person?

The above characteristics of a problem are called as 7-problem characteristics under which the solution must take place.

## Q Explain in detail Problem-Solving Agent

Intelligent agents are supposed to act in such a way that the environment goes through a sequence of

states that maximises the performance measures (Please frame: Performance measure is discussed later in the

text for now it is a term that defines how successful the agent is. Hence, we need to maximise the performance

measure.) In its full generality, this specification is difficult to translate into successful agent design.

Problem-solving agent is the form of goal-based agent.

*Structure of Problem-Solving Agents*

Consider a goal to be a set of world states in which a goal is satisfied.

**1. Goal state:** A state that describes the objective that the agent is trying to achieve is called as the **goal**

**state** of the agent.

**2. Action:** When there is a transition between the world states an **action** is said to be performed.

So, the agent has to find out which action will take it to the goal state, but before it can do this, agent

has to decide what sorts of actions and states to consider.

*Steps in Problem Solving Performed by Problem-Solving Agent*

**1. Goal formulation:** On the basis of the current situation and agent’s performance measure, it is the first

step in problem solving. We could consider goal to be a single world state or set of world states, exactly

those states in which goal is satisfied. Agent’s task is to find out sequence of actions that will get it to

the goal state. Goals help organise behaviour by limiting the objectives the agent is trying to achieve.

**2. Problem formulation:** It is a process of deciding what actions and states to consider given a goal. This

step is discussed in detail in the next section.

**3. Choosing the best sequence:** An agent with several immediate options of unknown value can decide

what to do by first examining different possible sequences of actions that lead to the states of known

value and then choosing the best sequence. This means that whether an agent is in a particular state,

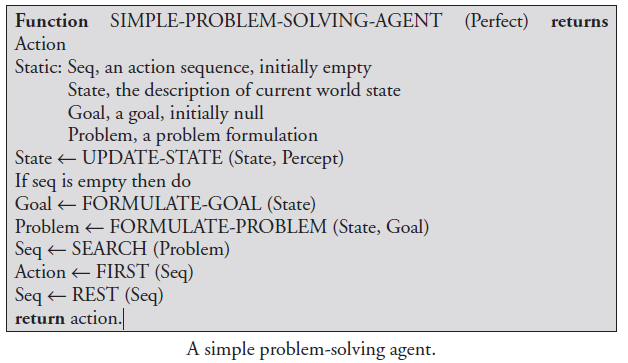
the agent can take many possible actions to move to the next state, but it has to take an action that

will make the agent closest to the goal. Next action is called a *best action*. And sequence of such a best

actions is called as a *best sequence*.

The process of looking for such a **best sequence** is called **search**. A search algorithm taken a problem

as input and returns **solution** in the form of an **action sequence**.



## Q Solve Water jug problem using state space search Method for the given problem

**Problem:**You are given two jugs, a 4-gallon one and a 3-gallon one. Neither has any measuring mark on it.There is a pump that can be used to fill the jugs with water.How can you get exactly 2 gallons of water into the 4-gallon jug.

**Solution:**

The state space for this problem can be described as the set of ordered pairs of integers **(x,y)**

Where,

X represents the quantity of  water in the 4-gallon jug  **X= 0,1,2,3,4**

Y represents the quantity of water in 3-gallon jug **Y=0,1,2,3**

**Start State: (0,0)**

**Goal State: (2,0)**

Generate production rules for the water jug problem

**Production Rules:**

|  |  |  |
| --- | --- | --- |
| **Rule** | **State** | **Process** |
| 1 | (X,Y | X<4) | (4,Y)  {Fill 4-gallon jug} |
| 2 | (X,Y |Y<3) | (X,3)  {Fill 3-gallon jug} |
| 3 | (X,Y |X>0) | (0,Y)  {Empty 4-gallon jug} |
| 4 | (X,Y | Y>0) | (X,0)  {Empty 3-gallon jug} |
| 5 | (X,Y | X+Y>=4 ^ Y>0) | (4,Y-(4-X))  {Pour water from 3-gallon jug into 4-gallon jug until 4-gallon jug is full} |
| 6 | (X,Y | X+Y>=3 ^X>0) | (X-(3-Y),3)  {Pour water from 4-gallon jug into 3-gallon jug until 3-gallon jug is full} |
| 7 | (X,Y | X+Y<=4 ^Y>0) | (X+Y,0)  {Pour all water from 3-gallon jug into 4-gallon jug} |
| 8 | (X,Y | X+Y <=3^ X>0) | (0,X+Y)  {Pour all water from 4-gallon jug into 3-gallon jug} |
| 9 | (0,2) | (2,0)  {Pour 2 gallon water from 3 gallon jug into 4 gallon jug} |

**Initialization:**

Start State: (0,0)

Apply Rule 2:

|  |  |
| --- | --- |
| (X,Y | Y<3)    -> | (X,3)  {Fill 3-gallon jug} |

Now the state is **(X,3)**

**Iteration 1:**

Current State: **(X,3)**

Apply Rule 7:

|  |  |
| --- | --- |
| (X,Y | X+Y<=4 ^Y>0) | (X+Y,0)  {Pour all water from 3-gallon jug into 4-gallon jug} |

Now the state is **(3,0)**

**Iteration 2:**

Current State : **(3,0)**

Apply Rule 2:

|  |  |
| --- | --- |
| (X,Y | Y<3)    -> | (3,3)  {Fill 3-gallon jug} |

Now the state is **(3,3)**

**Iteration 3:**

Current State:**(3,3)**

Apply Rule 5:

|  |  |
| --- | --- |
| (X,Y | X+Y>=4 ^ Y>0) | (4,Y-(4-X))  {Pour water from 3-gallon jug into 4-gallon jug until 4-gallon jug is full} |

Now the state is **(4,2)**

**Iteration 4:**

Current State : (4,2)

Apply Rule 3:

|  |  |
| --- | --- |
| (X,Y | X>0) | (0,Y)  {Empty 4-gallon jug} |

Now state is (0,2)

**Iteration 5:**

Current State : (0,2)

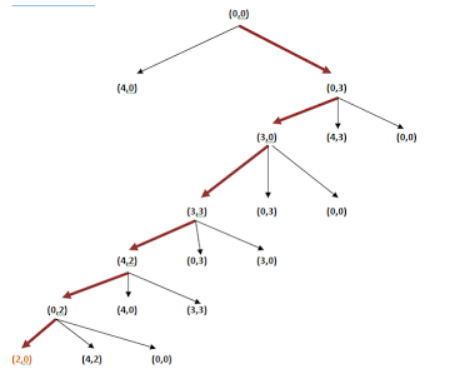
Apply Rule 9:

|  |  |
| --- | --- |
| (0,2) | (2,0)  {Pour 2 gallon water from 3 gallon jug into 4 gallon jug} |

Now the state is **(2,0)**

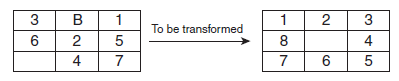
**Goal Achieved.**

**State Space Tree:**



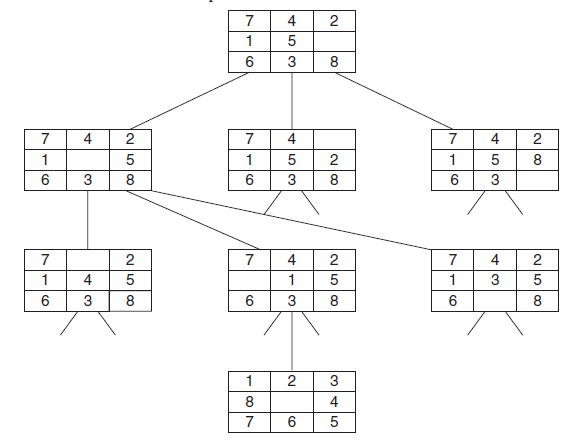
## Q Solve eight tile puzzle using state space search problem

The eight tile puzzle consist of a 3 by 3 (3 × 3) square frame board which holds eight movable tiles numbered 1 to 8. One square is empty, allowing the adjacent tiles to be shifted. The objective of the puzzle is to find a sequence of tile movements that leads from a starting configuration to a goal configuration.



The states of eight-tile puzzle are the different permutations of the tiles within frame.

Let’s do a standard formulation of this problem now.

 Tree diagram showing the search space

**States:** It specifies the location of each of the eight tiles and the blank in one of the nice squares.

**Initial state:** Any state can be designated as the initial state.

**Goal:** Many goal configurations are possible in one such is shown in Figure 3.1.

**Legal moves (or state):** They generate legal states that result from trying the four actions:

**1.** Blank moves left

**2.** Blank moves right

**3.** Blank moves up

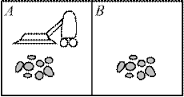
**4.** Blank moves down

**Path cost:** Each step costs 1, so the path cost is the number of steps in the path.

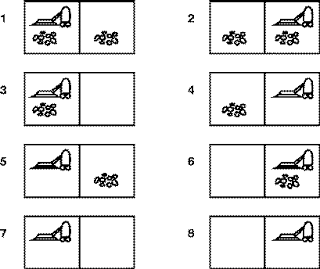
## Q Solve Vacuum cleaner problem using state space search method

### Vacuum cleaner

Consider a Vacuum cleaner world

[](https://1.bp.blogspot.com/-eKv3yq93ZyE/VsVcMn-QrGI/AAAAAAAAAFs/6nJZ0lCEcg4/s1600/va1.png)

Imagine that our intelligent agent is a robot vacuum cleaner.   
Let's suppose that the world has just two rooms. The robot can be in either room and there can be dirt in zero, one, or two rooms.

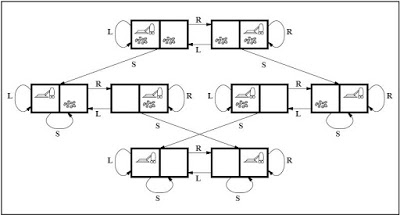
[](https://3.bp.blogspot.com/-awZN_dv2vLI/VsVcMux__HI/AAAAAAAAAF0/-1ZRtzWXu9A/s1600/fig03.02.gif)

Goal formulation: intuitively, we want all the dirt cleaned up. Formally, the goal is

 { state 7, state 8 }.

Problem formulation(Actions):Left,Right,Suck,NoOp

 State Space Graph:

[](https://4.bp.blogspot.com/-ts1PGzUzl6E/VsVcMgfAEGI/AAAAAAAAAFw/iMvPRc9ANsg/s1600/VacuumWorld.JPG)

Measuring performance

With any intelligent agent, we want it to find a (good) solution and not spend forever doing it.

The interesting quantities are, therefore,

* the search cost--how long the agent takes to come up with the solution to the problem, and
* the path cost--how expensive the actions in the solution are.

The total cost of the solution is the sum of the above two quantities.

## Q Explain Problem Solving agent with types of single-state problem multiple-state problem contingency problem and. exploration problem with example.

### Problem Solving Agent

1. Goal Formulation-Set of one or more (desirable) world states.(eg.Checkmate in Chess)
2. Problem Formulation-What actions and states to consider given a goal and an initial state
3. Search for solution-Given the problem, search for a solution--a sequence of actions to achieve the goal starting from initial state
4. Execution of the solution

Goal Formulation

Specify the objectives to be achieved  
goal - a set of desirable world states in which the objectives have been achieved  
current / initial situation - starting point for the goal formulation  
actions - cause transitions between world states

Problem Formulation

Actions and states to consider  
states - possible world states  
accessibility -  the agent can determine via its sensors in which state it is  
consequences of actions - the agent knows the results of its actions  
levels -  problems and actions can be specified at various levels  
constraints -  conditions that influence the problem-solving process  
performance  -  measures to be applied  
costs - utilization of resources

Problem Types

Not all problems are created equal

single-state problem  
multiple-state problem  
contingency problem  
exploration problem

Single-state problem

exact prediction is possible  
state - is known exactly after any sequence of actions  
accessibility of the world all essential information can be obtained through sensors  
consequences of actions are known to the agent  
goal - for each known initial state, there is a unique  
goal state that is guaranteed to be reachable via  
an action sequence  
simplest case, but severely restricted

Example:

Vacuum world,

 Limitations:

·         Can’t deal with incomplete accessibility

·         incomplete knowledge about consequences changes in the world

·         indeterminism in the world, in action

Multiple-state problem

* semi-exact prediction is possible  
  state is *not*known exactly, but limited to a set of possible states after each action  
  accessibility of the world *not*all essential information can be obtained through sensors  
  reasoning can be used to determine the set of possible states  
  consequences of actions are not always or completely known to the agent; actions or the environment might exhibit randomness  
  goal due to ignorance, there may be no fixed action sequence that leads to the goal  
  less restricted, but more complex
* Example:
* Vacuum world, but the agent has *no*sensors  
  The action sequence right, suck, left, suck is guaranteed to reach the goal state from any initial state  
  Limitations:
* Can’t deal with changes in the world during execution (“contingencies”)

Contingency problem

* exact prediction is impossible  
  state unknown in advance, may depend on the outcome of actions and changes in the  
  environment  
  accessibility of the world some essential information may be obtained through sensors only at execution time  
  consequences of action may not be known at planning time  
  goal instead of single action sequences, there are *trees of actions*  
  contingency branching point in the tree of actions  
  agent design different from the previous two cases:the agent must act on incomplete plans  
  search and execution phases are interleaved
* Example: Vacuum world,  The effect of a suck action is random.  
  There is no action sequence that can be calculated at planning time and is guaranteed to reach the goal state.  
  Limitations: Can’t deal with situations in which the environment or effects of action are unknown

Exploration problem

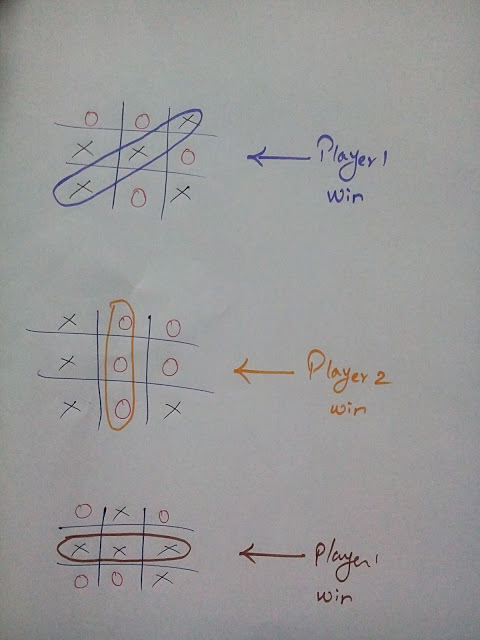
* effects of actions are unknown  
  state the set of possible states may be unknown  
  accessibility of the world some essential information may be obtained through sensors only at execution time  
  consequences of actions may not be known at planning time  
  goal can’t be completely formulated in advance because states and consequences may not be  
  known at planning time  
  discovery what states exist  
  experimentation what are the outcomes of actions  
  learning remember and evaluate experiments
* agent design different from the previous cases: the agent must experiment  
  search requires search in the real world, not in an abstract model  
  realistic problems, very hard

Performance Measuring

* for problem solving  
  success  
  Has a solution been found?  
  quality  
  Is it a good solution?  
  What are the criteria?  
  optimal solution  
  may be difficult to find and not necessary  
  cost  
  sum of  
  *•*search cost (time, resources to find a solution)  
  *•*path cost (as defined above)

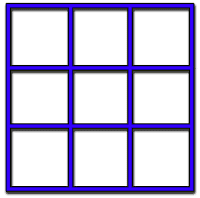
## Q Explain Tic Tac Toe problem until Win /Loose/ Draw condition is not achieved.

The game Tic Tac Toe is also known as Noughts and Crosses or Xs and Os ,the player needs to take turns marking the spaces in a 3x3 grid with their own marks,if 3 consecutive marks (Horizontal, Vertical,Diagonal) are formed then the player who owns these moves get won.  
  
Assume ,  
  
Player 1 - X  
Player 2 - O

[[](https://3.bp.blogspot.com/-gExxjOi6KBs/VriuSlJE3WI/AAAAAAAAAEc/kazMr4z-tBc/s1600/IMG_20160208_203035.jpg)](https://3.bp.blogspot.com/-gExxjOi6KBs/VriuSlJE3WI/AAAAAAAAAEc/kazMr4z-tBc/s1600/IMG_20160208_203035.jpg)

So,a player who gets 3 consecutive marks first,they will win the game .  
  
Let's have a discussion about how a board's data structure looks and how the Tic Tac Toe algorithm works.

#### Board's Data Structure:

[](https://4.bp.blogspot.com/-5mU7lafsuAA/Vriv9Go2QnI/AAAAAAAAAEk/6ul2vnuXwaE/s1600/tic_tac_toe.gif.png)

The cells could be represent as Center square,Corner,Edge as like below

#### **<https://3.bp.blogspot.com/-NYUZsSXxnGg/VriwshO9QHI/AAAAAAAAAEs/K2qZH-UKcC4/s1600/tic_tac_toe_board.jpg>**

Number each square starts from 1 to 9 like following image

#### Consider a Board having nine elements vector.Each element will contain

* 0 for blank
* 1 indicating X player move
* 2 indicating O player move

Computer may play as X or O player. First player is always X.  
  
Move Table  
It is a vector of 3^9 elements, each element of which is a nine element vector representing board position.  
Total of 3^9(19683) elements in move table  
Move Table  
Index Current Board position New Board position  
0             000000000                 000010000  
1             000000001                 020000001  
2             000000002                 000100002  
3             000000010                 002000010  
.  
.

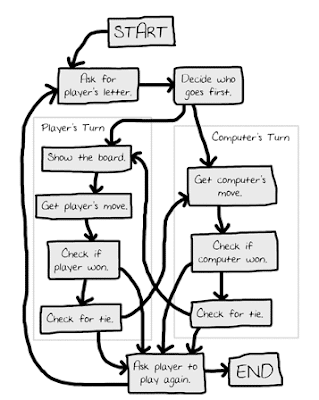
#### Algorithm

To make a move, do the following:

1. View the vector (board) as a ternary number and convert it to its corresponding decimal number.
2. Use the computed number as an index into the move table and access the vector stored there.
3. The vector selected in step 2 represents the way the board will look after the move that should be made. So set board equal to that vector.

Let's start with empty board

 Step 1:Now our board looks like 000 000 000 (tenary number) convert it into decimal no.The decimal no is 0  
 Step 2:Use the computed number ie 0 as an index into the move table and access the vector stored in  
New Board Position.  
The new board position is 000 010 000   
Step 3:The vector selected in step 2(000 010 000 ) represents the way the board will look after the move that should be made. So set board equal to that vector.  
After complete the 3rd step your board looks like\  
   
This process continues until the player get win or tie.     
  
Flowchart:

[](https://4.bp.blogspot.com/-HvnqlNfD-t0/VrjGeuIHLwI/AAAAAAAAAFE/Jna17X6SBcw/s1600/image002.png)

# MCQs & Answers

**1.** The main task of a problem-solving agent is \_\_\_\_\_\_\_\_\_\_\_.

(a) Solve the given problem and reach to goal

(b) To find out which sequence of action will get it to the goal state

(c) All of the above

(d) None of the above

**2.** What is state space?

(a) The whole problem

(b) Your definition to a problem

(c) Problem you design

(d) Representing your problem with variable and parameter

**3.** The problem-solving agent with several immediate options of unknown value can decide what to do by just examining different possible sequences of actions that lead to the states of known value, and then choosing the best sequence. This process of looking for such a sequence is called Search. State True or False.

(a) True

(b) False

**4.** A search algorithm takes \_\_\_\_\_\_\_\_\_ as an input and returns \_\_\_\_\_\_\_\_ as an output.

(a) Input, output

(b) Problem, solution

(c) Solution, problem

(d) Parameters, sequence of actions

**5.** The set of actions for a problem in a state space is formulated by a \_\_\_\_\_\_\_\_\_\_\_.

(a) Intermediate states

(b) Initial state

(c) Successor function, which takes current

action and returns next immediate state

(d) None of the above

**6.** The process of removing detail from a given state representation is called\_\_\_\_\_\_.

(a) Extraction

(b) Abstraction

(c) Information retrieval

(d) Mining of data

**7.** A problem-solving approach works well for\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

(a) 8-puzzle problem

(b) 8-queen problem

(c) Finding an optimal path from a given

source to a destination

(d) Mars Hover (Robot Navigation)

**8.** Web crawler is a/an\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

(a) Intelligent goal-based agent

(b) Problem-solving agent

(c) Simple reflex agent

(d) Model based agent

**Answers**

1. **(c) 2. (d) 3. (a) 4. (b) 5. (c) 6. (b) 7. (d) 8. (a)**