UNIT-2

Problem Solving by Searching

- Problem Solving by searching
- Problem-Solving Agents
  - Well defined problem and solutions
  - Formulating problems
- Example Problems
  - Toy problems
- Searching for Solution
- Uninformed Search Strategies
  - Concept of BFS
  - Concept of DFS
  - Depth-limited search
  - Iterative deepening DFS
  - Bidirectional search
- Informed (Heuristic) Search Strategies
  - Concept of Greedy BFS
  - A* search: Minimizing the total estimated solution cost
- Case Study: Applications of AI in transportation.

Unit –2 Problem Solving by Searching

2.1 Problem Solving by Searching:

In Artificial Intelligence searching method is the best technique to solve the problems whether it is defined or not defined. AI used search strategies to solve the specific problem.

To solve the problems in Artificial Intelligence it need following things:

a. Problem Define
b. Problem Analyze
c. Isolate problem and knowledge of task
d. Get techniques
e. Choose techniques to solve problems

To build a system to solve a particular problem first we need to specify the initial system (I) what exactly the problem is and the acceptable final solutions to the problem. Isolate the problem and read about the problems then get multiple possible techniques that are necessary to solve the problem, after that we can get the best technique and apply it (them) to the particular problem.

Define the problem as a **state space search.** All the possible state for solving any particular problem is called as **state space**. For searching the particular state from the state space we use state space search. The structure of state space is of two types:

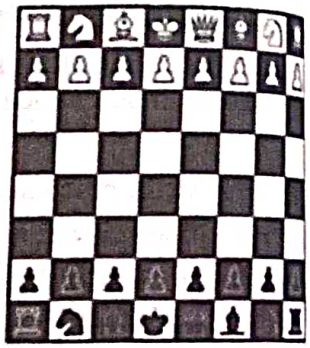
- In some problems set of states and rules are already defined and well structured organize. We have to convert some given condition into the required condition using a set of operations which are already decided.
- In some problems these kinds of representation occurs naturally and are not well structured. In this we have to make the desired structure with the help of some techniques to find the path from the initial state to goal state. Search is an important procedure for the problems to find the, to and Forth solution no direct techniques are used in it.

2.2 Problem solving Agents:

Problem solving agents are the goal based agents which are also called as **problem solving agents**. Simple Problem solving agents have limited strategies to solve the problems. They solve the problems on current perceptions they don't know what they want and what are the goals. Problem solving agents are such that they create the sequential environment to maximize the performance of searching method to solve any problem. Let illustrate the problems **play chess.game** and **milk conman problem.**

❖ Play chess Problem:

We have to build a program which can play chess. Here the required conditions a already defined which can be the position of the players, moves, operations, initial sta and goal state, legal game to win and lost rules.



In the **play chess game** the rules are already provided and the starting position is defines as an 8 by 8 position where each box contains its value and has some opening rules with the legal move. There are many ways to apply this rules because of that it is too difficult and lengthy one could not get the goal without any mistakes. To solve this problem if possible one should look for a way to write the rules with legal moves.

❖ Milk Conman Problem:

We are given two conman (utensil to pour liquid), a 6 liter and 4 liter. We don't hav measuring cup or any mark. We are given a tap to fill the Conman. How can we get exac 2liter of milk into the 6 liter of Conman?

One solution of milk Conman problem	
Liters in the 6 liter of Conman	Liters in the 4 liter of Conman
0	0
0	4
4	0
4	4
6	2
0	2
2	0

The above two problems can be solve with the strategies we discussed already. Both the problems are different and can be solved by using some rules, strategies, approximation, some combinations, and movements until the goal state is found. Play chess game can be solved by using single state search while for milk conman problem different combinations of searching is being used.

Problem
input:
static:
state: ←
if
g -
probl em
s -
action -
s ←
return

❖ Formulatin

Formulating pro problem. We ca

a. Single stat
b. Multiple st
c. Contingen
d. Exploratio

Let us learn the first, when the i net action, sequ **problem**. If all of the action th can be known solving the pro solve single pr depends on the real time probl state can be kn

Problem Solving Agents	
input:	p, Assumption
static:	s, action sequence, empty state, description of present state,
	g, goal, null
	problem, problem changed
state: ←	UPDATE-STATE (state,p)
if	s is empty then
g -	GOAL, (state) CHANGED
probl em	<- PROBLEM CHANGED (state, g)
s -	SEARCH (problem)
action -	recommendation (s, state)
s ←	REMINDER (s, state)
return	action

❖ Formulating problems:

Formulating problem is depending on the knowledge, environment and perception of the problem. We can divide the problem as-

a. Single state problem
b. Multiple state problem
c. Contingency problem
d. Exploration problem

Let us learn the problem with assumption before that we should understand the problem first, when the initial or present state is known with that one should get to know about the net action, sequence of action or sometimes we get goal state which is called **single state problem**. If all the action are known but because of some rules, we get the limited access of the action then the possibility of happening the particular state is greater than 1, thus it can be known as **multiple state problem.** When there is multiple state problem then to solving the problem the process can also be more than one. Probability can be used to solve single problem thus this type of problem requires the sequence of action taken. It depends on the possible contingency which can also call as **contingency problem**. The real time problems are not exact predicted but some additional information like problem state can be known.

2.2.1 Well defined problems and solutions:

Problem is a special part to solve the problem. It is the collection of information that one can decide what exactly we want and what to do next. There are elements of problem: STATE, ACTION, PATH and GOAL.

State can be defined as initial state or end state both. In initial state the problem could be known to be itself. When the problem is already defines b some rules then one could get end state. When the problem is already defined by some rules then one could get end state. After the initial state the set of possible actions available for the problem can be determined. In action state space is used and the path of state space is any sequence of actions which can apply on one state to another. The path is the sequence of any action from the initial state to reach the goal or End state.

Input
datatype: PROBLEM
elements: INITIAL STATE, ACTION, PATH, GOAL, ENDSTATE

2.3 Example problems:

❖ Toy problems:

- 8- puzzle problem-

Start State

1	2	3
4	8	-
7	6	5

Goal State

1	2	3
4	5	6
7	8	-

Down
Up
Left
Right

The 8– Puzzle problem is 3×3 board with 8 tiles and a blank space in which tiles slides from one place to another only where the blank space is available. In the puzzle we have some rules and conditions of moving the tiles in which every tiles should be move only Up, Down, Left, Right. Tiles cannot move diagonally. We can lead following formulation:

- States: A state describes the location of each eight tiles and also of the blank tiles.
- Operators: Up, Down, Left, Right (blank tiles only).
- Start: Describes the initial state.
- Goal: Describes the goal state.

- Path cost: Number of steps taken to reach the goal.

Following are the conditions and states to solve the 8-puzzle problem.

Down-

{(1,2,3), (4,8,0), (7,6,5)}

{(1,2,3), (4,8,5), (7,6,0)}

Left-

{(1,2,3), (4,8,5), (7,0,6)}

Up-

{(1,2,3), (4,0,5), (7,8,6)}

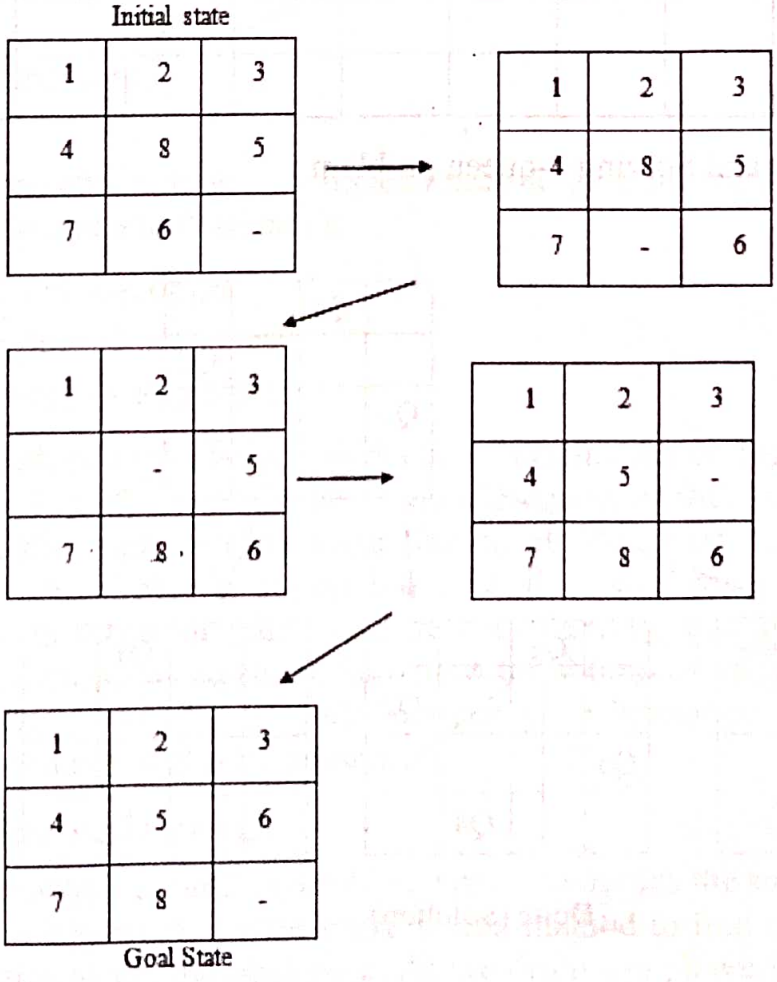
Right-

{(1,2,3), (4,5,0), (7,8,6)}

Down-

{(1,2,3), (4,5,6), (7,8,0)}

Path cost= ("5") i.e. no. of states taken to solve the problem.



❖ N- Queen Problem:

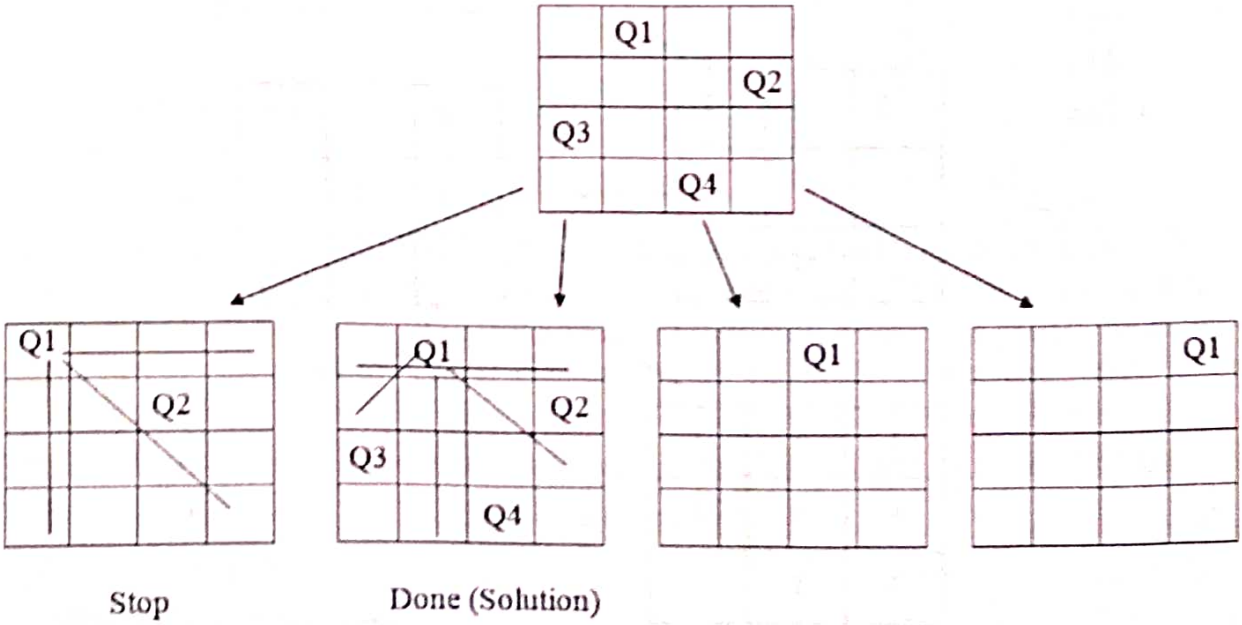
This is the type of constraint satisfactory problem in AI. It has N×N square grid board in which N number of queens is placed. Some constraints/rules are-

1. No row should contain more than one queen.
2. No column should contain more than one queen.
3. No diagonal should contain more than one queen.
4. There should be no row or column without any queen.

Here is the goal state for 8-Queen problem –

	Q1						
			Q2				
						Q3	
				Q4			
		Q5					
Q6							
					Q7		
							Q8

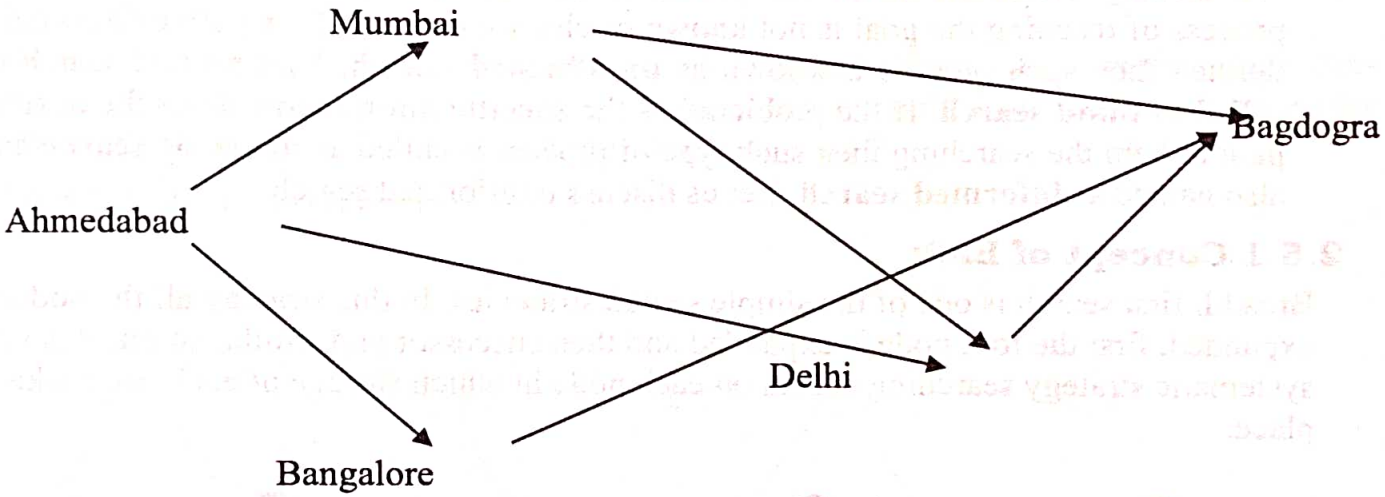
Let we discuss and Solving 4-queen problem



- State: initial position
- Goal: no queen attack on each other
- Operations: place the queen in an square
- Path cost: zero

❖ **Real world Problems:**

Route finding is defined for specified location, source, destination, path and the links between the states.



Let illustrate from the above diagram a traveler need to go from Ahmedabad to Bagdogra. He has three paths to reach to destination.

Ahmedabad →Delhi → Bagdogra

Ahmedabad →Banglore → Bagdogra

Ahmedabad →Mumbai → Bagdogra

It depends on the path and the distance to reach the destination on less time and low cost. If we talk about road transport then it takes more distance, traffic, low cost of petrol and driving time. We use route finding route finding algorithm on both the road and air transportation. On roads it may be simple but for airline travel route finding algorithm is somehow complex in terms of path, cost, seat availability, quality, time, type of air transport, frequency, mileage and so on. Somehow the actions of this problem are also get known outcomes, flight may be delayed, emergency, maintenance, late or overbooked, fog delayed, connections can also be missed.

2.4 Searching for solution:

We have seen the problem definition, problem types, recognize the solution to finding the solution we used state space and state space search method to find the solution with the path and the sequence to get the goal state. As we discussed above in 4-queens problem

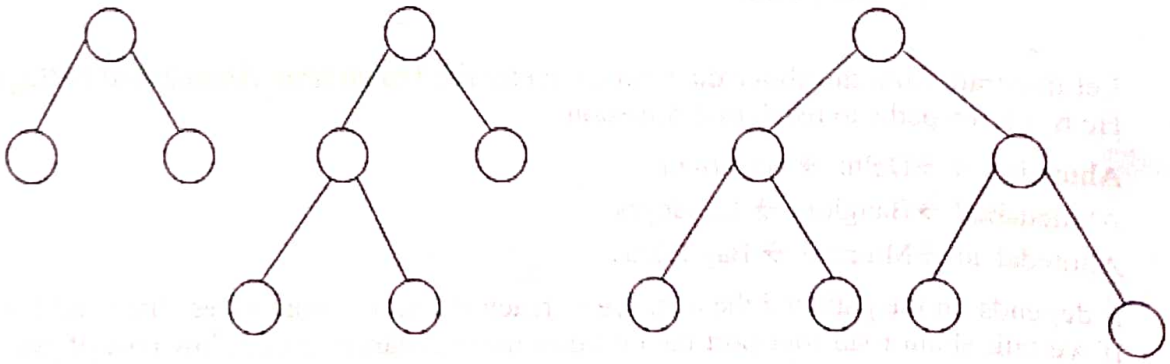
the first method didn't get us the solution so we stopped and try another to get the solution. If the first choice does not lead to the solution we can put that option for later and choose another option to get the solution. We can continue choosing and exploring until the solution is found or the next state is found this process is also known as search strategy. Many search trees are used over state space to find the route and solution and that we will discuss later.

2.5 Uniformed search strategies:

We already discussed about the problems and searching. When the actions and the process of reaching the goal is not known or else we can say if the path of the goal is not defined then such process is known as **uninformed search**. Uninformed search is also called as **blind search**. If the problem has the specific information about the next step or path to help the searching then such type of process is called as **heuristic search** and it is also named as **informed search**. Let us discuss uninformed search.

2.5.1 Concept of BFS:

Breadth first search is one of the simple search strategies. In this strategy all the nodes are expanded, first the root node is expanded and then successor and similar so on. It is very systematic strategy searching occurs on each node in which storage of each node takes place.



Circle represents nodes (0, 1, 2, 3, 4, 5, 6, 7, 8.... so on)

- **Levels of BFS:**

Function BFS (problem)

If Goal

Then Return, Failure

Or else return search (problem, Goal)

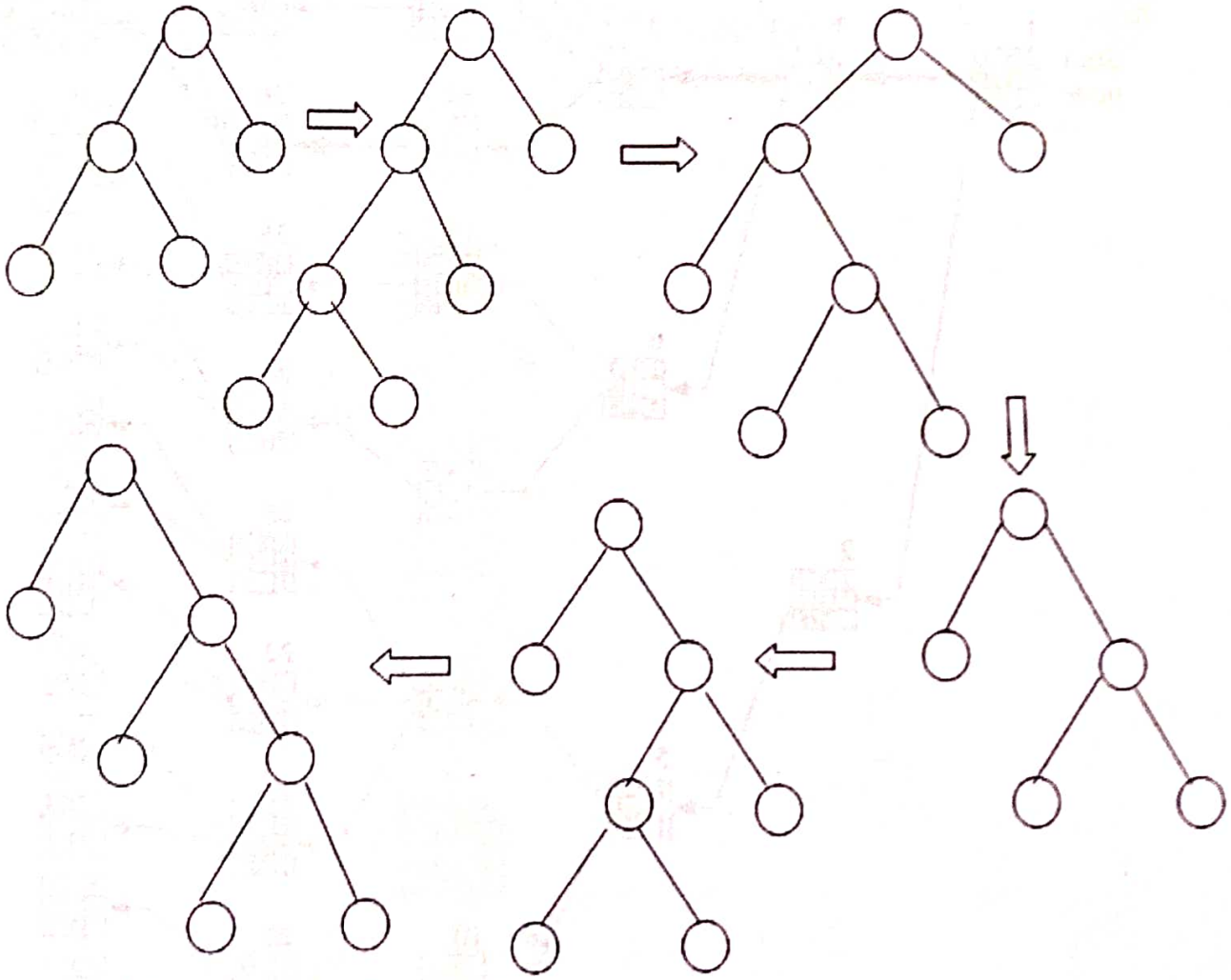
Return

If there is a solution for a problem then BFS always find not so deep goal state first, then the second one and so on. But it takes so much of memory and time sometime BFS is not been chosen strategy among all. Because of the searching of every single node, BFS takes time and also memory. Assume 100 nodes can be checked in a second and one node requires 10 bytes of storage thus the more memory and time required which is problem for BFS. To solve any random problem using BFS one could wait for less than 10 minutes and because of the more memory the cost also increase to solve the problem and to reach out to the Goal state.

2.5.2 Concept of DFS:

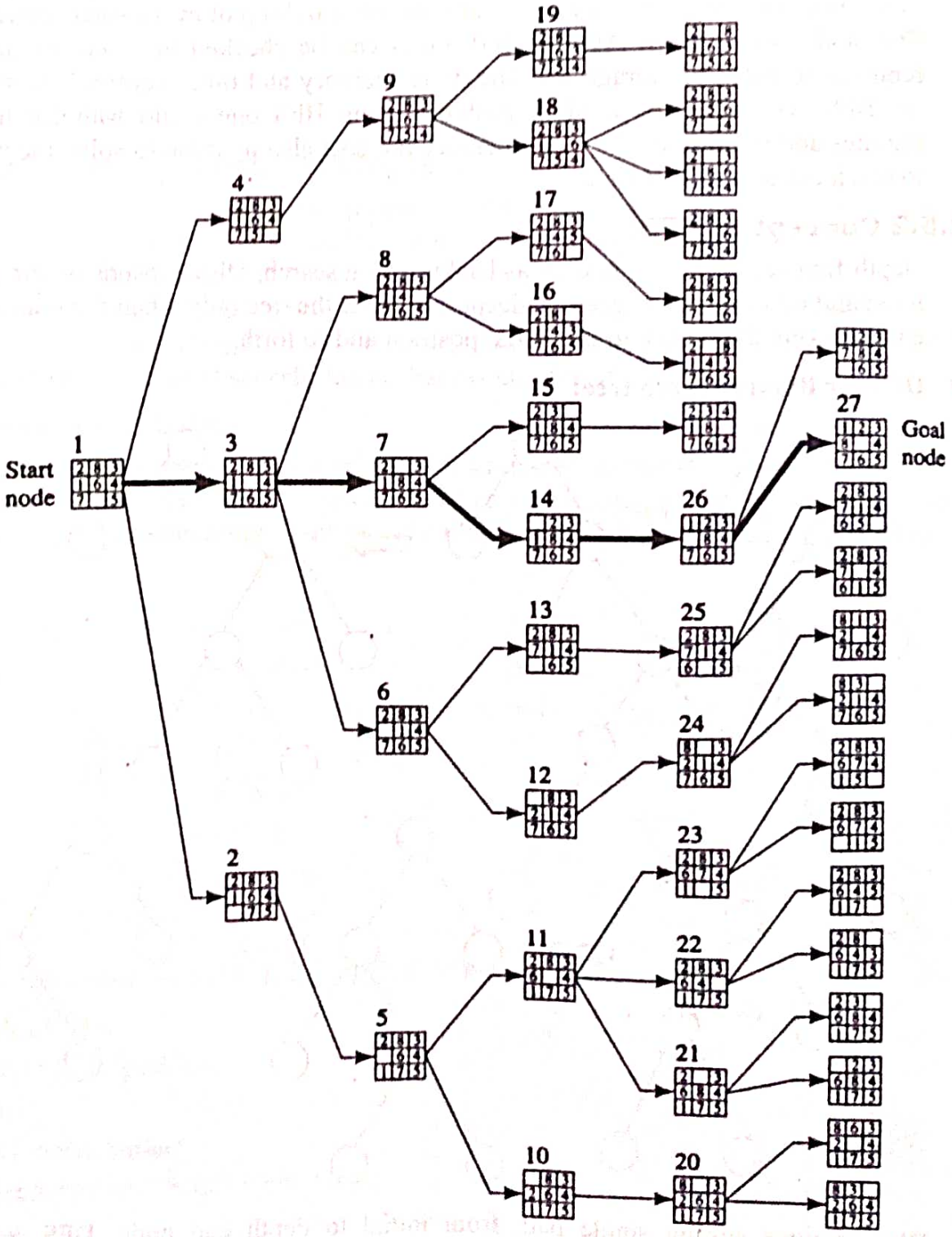
Depth first search is also known as backtracking search. DFS expands on the nodes, ones it expand on one node it goes on deepest level of the tree only when the search get on the endpoint then it go back to the initial position and so forth.

- **DFS for Binary search tree:**

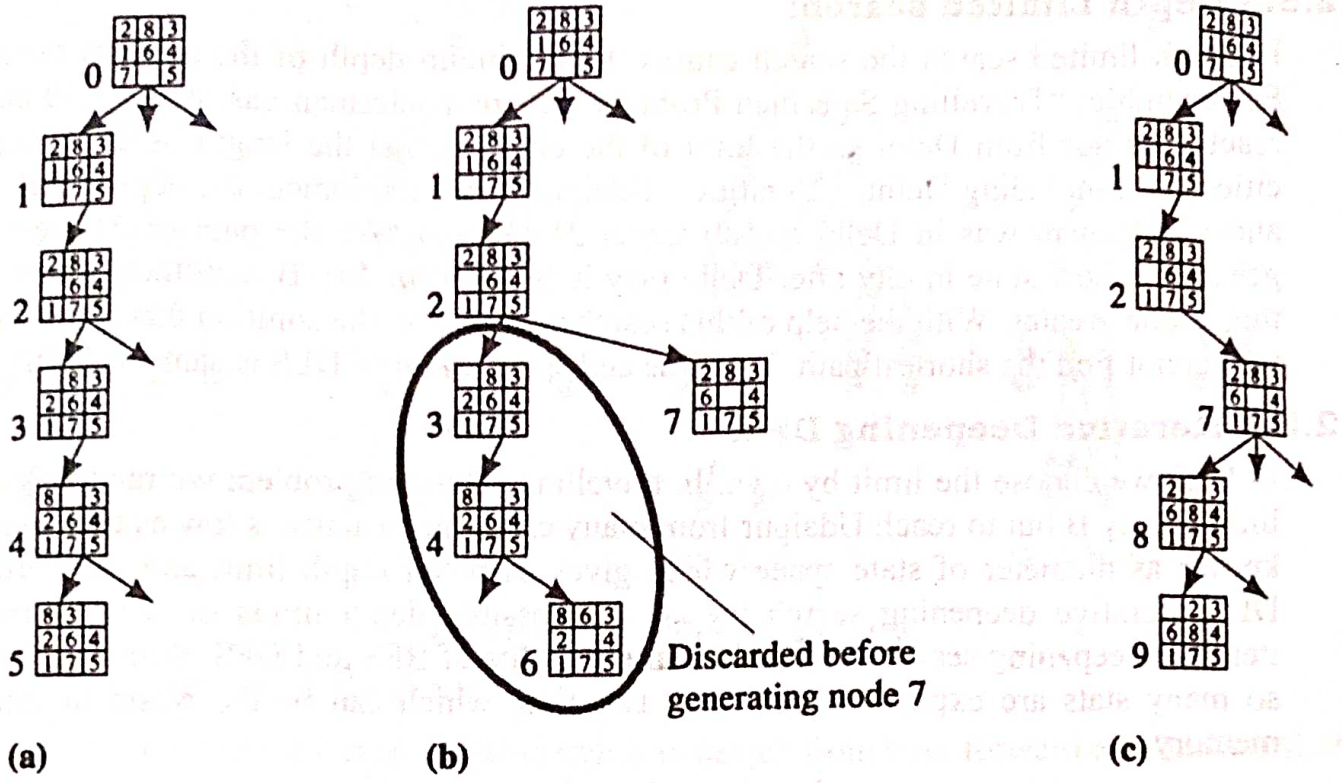


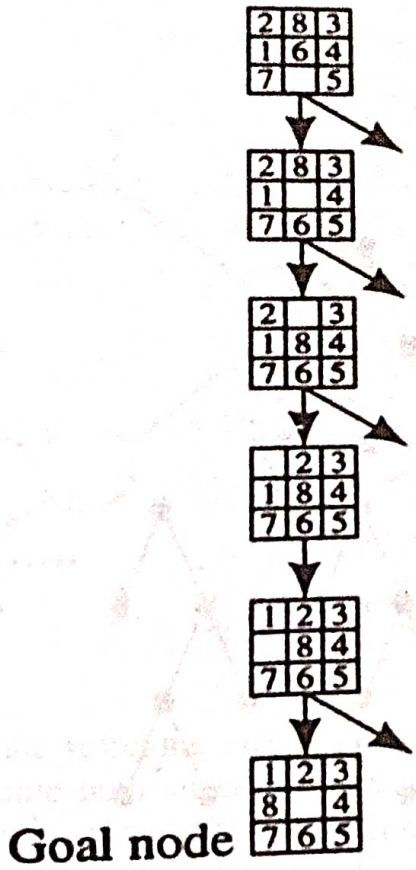
DFS requires storing single path from initial to depth end node. DFS requires less memory to store the data. DFS may find the solution without examine the search space of

all. DFS is getting stop when one is formed. The only demerit of DFS is it may stick into the depth of the nodes and it's difficult to come back on the position (initial node).



BFS of Eight -Puzzle





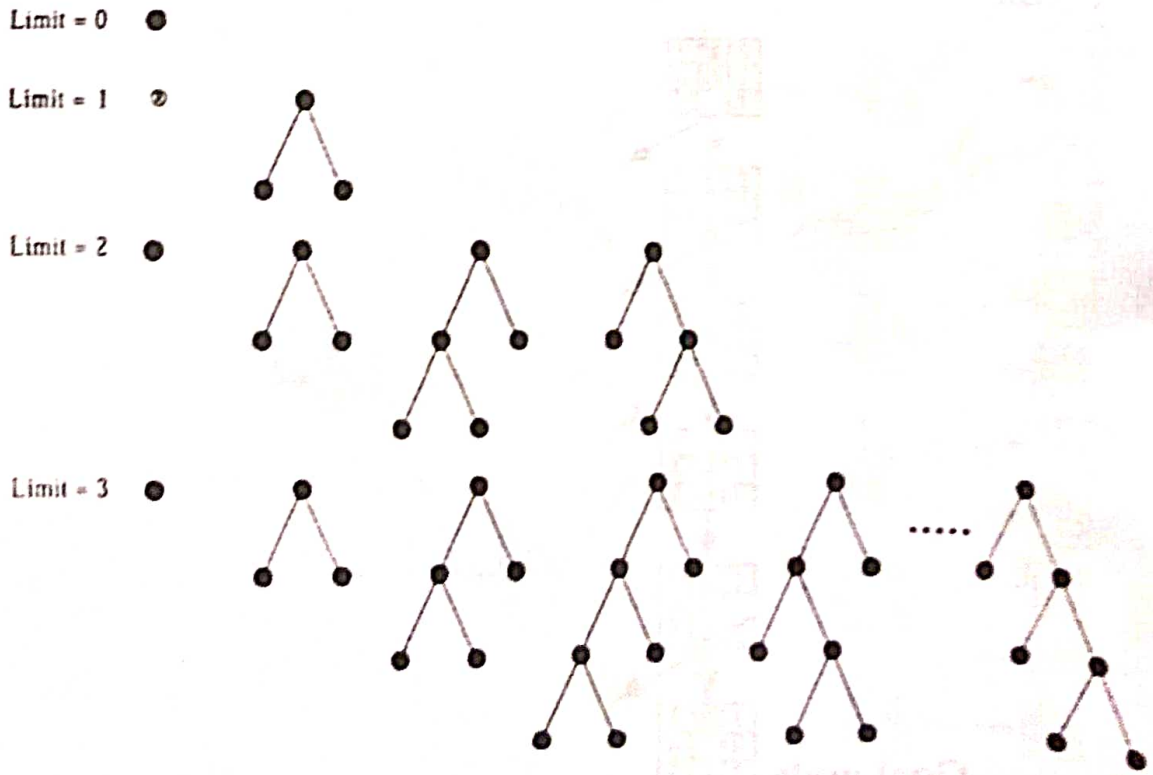
DFS of Eight Puzzles

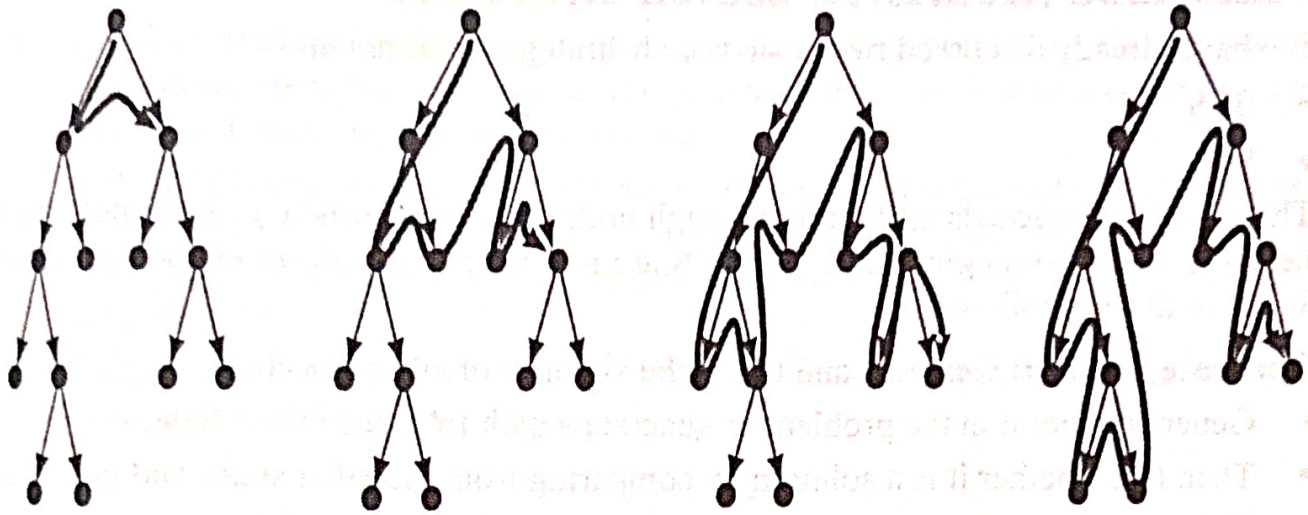
2.5.3 Depth Limited search:

In depth limited search the search causes the minimum depth of the path on the node. For example: "Travelling Salesman Problem" where a salesman has to cross 20 cities to reach Udaipur from Delhi so the limit of the cities we get the length of 21 as the no. of cities is 22 including Delhi – 20 cities – Udaipur. If we implement the depth limit search and if salesman was in Delhi and to travel 21 cities it take the path of 21 steps, the generate a new state in city after Delhi (say it as Agra) or city B with the path of length that is one greater. With the help of this search we can find the solution that exists but still we cannot find the shortest path. The time and space taken of DLS is same as DFS.

2.5.4 Iterative Deepening DFS:

In DLS we choose the limit by own. In travelling salesman problem we random take the limit of city B but to reach Udaipur from many cities the distance is low as this number is known as diameter of state space which gives us better depth limit and more efficient DLS. Iterative deepening search try all the possible depth limits or we can say the iterative deepening search is a combination benefits of BFS and DFS. Sometimes in IDS so many stats are expanded more than two time which can be the waste of time and memory.

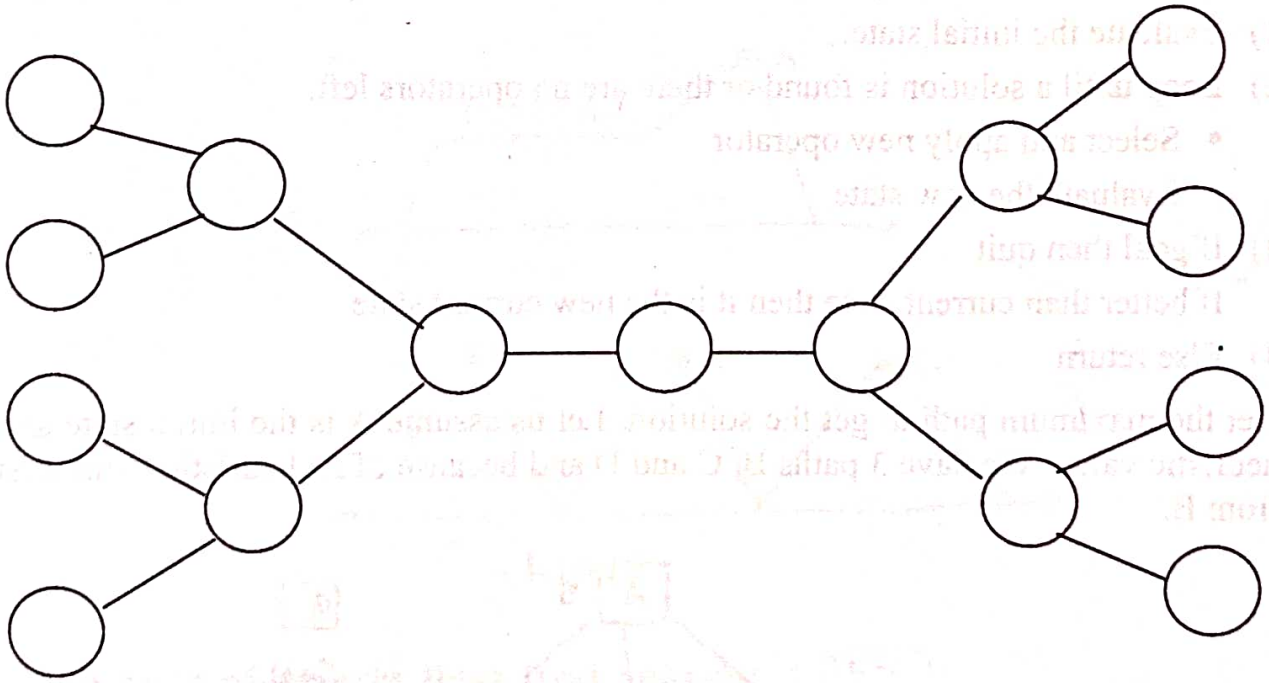




Stages in Iterative –Deepening Search

2.5.5 Bidirectional Search:

The idea behind the bidirectional search is to search from both forward and backward, it can be search from initial state and also from end goal state.



When both the searching met in the middle of the problem it makes a big difference. It has to generate both successor and predecessor nodes to search from both the sides, theoretically the address of bidirectional search can be succeeded. Both the predecessor and successor are identical. Bidirectional search can reduce time complexity.

2.6 Informed (Heuristic) search strategies:

We have already discussed two basic search strategies before i.e.

- BFS
- DFS

The searching proceeds preferably through nodes that are heuristic or informed are called heuristic search strategies. BFS and DFS are non heuristic (uniformed) search strategies. We here discuss following:

Generate and test: Generate and test is the simplest of all approaches.

- Generate a point in the problem or generate a path from the initial state.
- Then test whether it is a solution by comparing from the other states and goal state.
- If the solution is found then go to end point or else return to initial state.

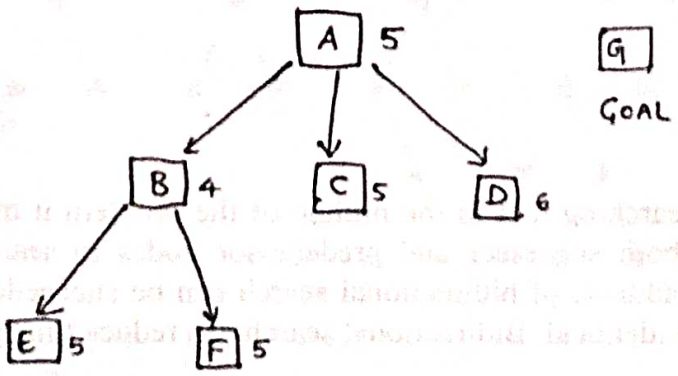
So if the possibility of solution is done systematically then this strategy can find many solutions. The best ways to apply generate and test strategy as a DFS with the backtracking method.

➢ **Hill Climbing Algorithm:**

This is the local search algorithm which has no backtracking and has a greedy approach.

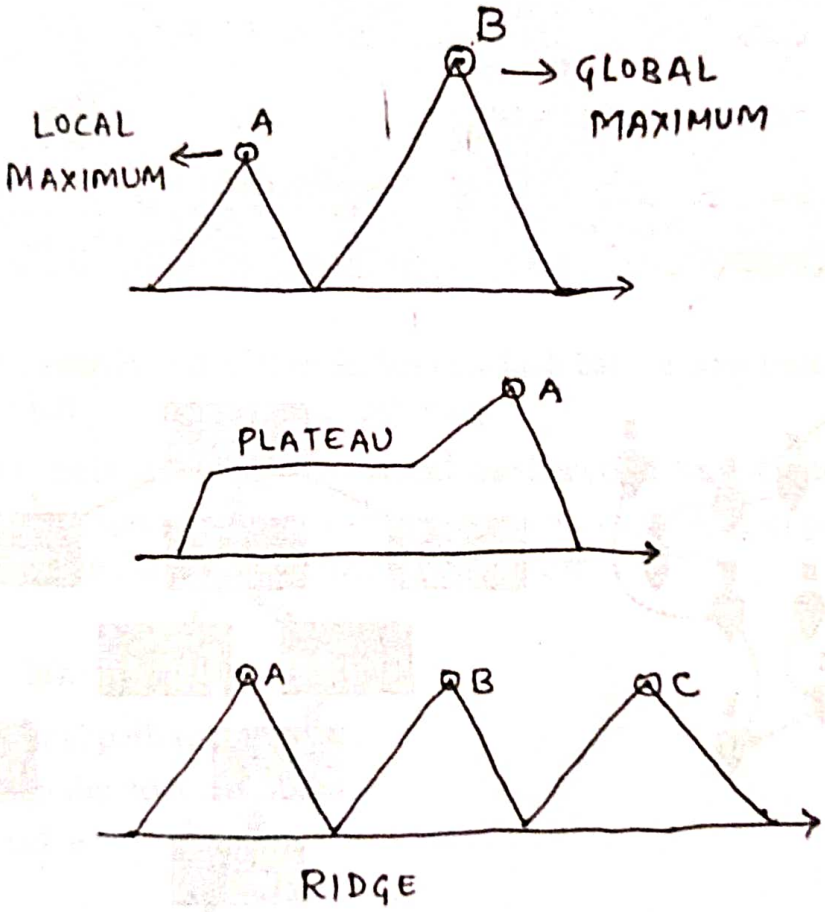
1) Evaluate the initial state.
2) Loop until a solution is found or there are no operators left.
   - Select and apply new operator
   - Evaluate the new state
3) If goal then quit
   If better than current state then it is the new current state
4) Else return

Get the maximum path to get the solution. Let us assume A is the initial state and 5 is its heuristic value. We have 3 paths B, C and D and because of its heuristic value best path is from B.



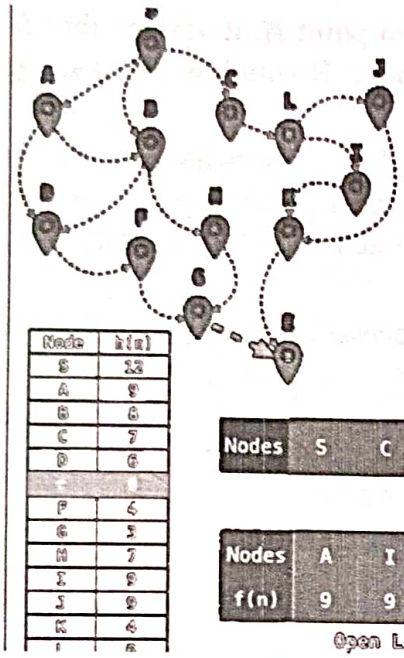
➢ **Limitations:**

1. Local maximum- Let us suppose the algorithm get on point A, it assume that A is the destination point but when it goes down then it get point B which is the exact goal we have to achieve and it is the best solution.
2. Plateau- Plateau means a point where the values of all successors have same value but after sometime it become higher but for particular time algorithm get same value and it stopped and react that this is the last value which achieved but actually this is not the goal state.
3. Ridge- Ridge means at all the point where all the successors have same values and the first point searching gets stopped.



2.6.1 Concept of greedy Best first search: CBFS

In BFS the best part is selected with the best evaluation. In the greedy search according to the smallest heuristic value the number of steps taken to reach the goal node is heuristic value and the path which has small heuristic value picks the best node out of them as the next node to expand according some rules. We can say that h(n) is the cost value of heuristic.



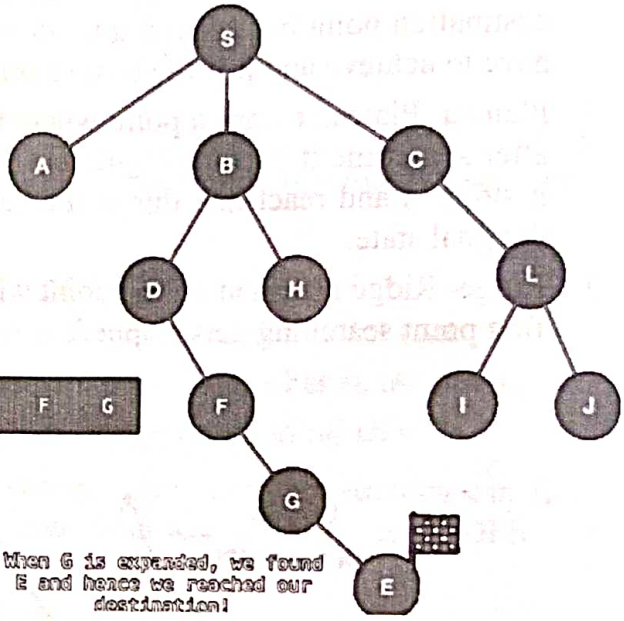
Node	h(n)
S	12
A	9
B	8
C	7
D	6
E	0
F	4
G	3
H	7
I	9
J	9
K	4
L	6

Nodes	S	C	L	B	D	F	G

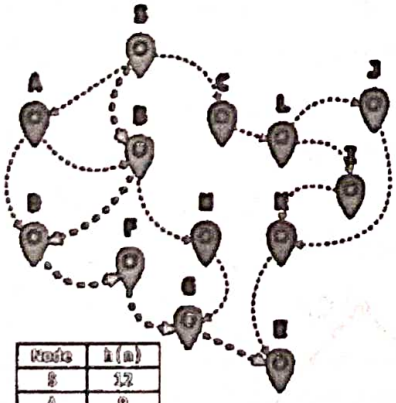
Closed List

Nodes	A	I	J	H
f(n)	9	9	9	7

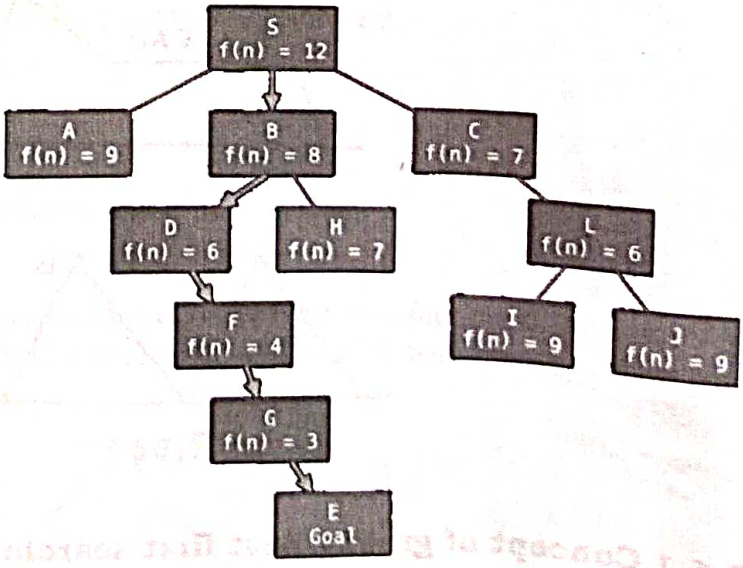
Open List



When G is expanded, we found E and hence we reached our destination!



Node	h(n)
S	12
A	9
B	8
C	7
D	6
E	0
F	4
G	3
H	7
I	9
J	9
K	4
L	6



Final tree with Solution Path highlighted in Green

Algorithm

- Step 1: Place the starting node into the OPEN list.
- Step 2: If the OPEN list is empty, Stop and return failure.
- Step 3: Remove the node n from the OPEN list which has the lowest value of h(n), and place it in the CLOSED list.
- Step 4: Expand the node n and generate the successors of node n.
- Step 5: Check each successor of node n and find whether any node is a goal node or not. If any successor node is goal node, then return success and terminate the search, else proceed to Step 6.
- Step 6: For each successor node, algorithm checks for evaluation function f(n), and then check if the node has been in either OPEN or CLOSED list. If the node has not been in both list, then add it to the OPEN list.
- Step 7: Return to Step 2.

Greedy Best first search is useful in Robotics which can be commercial or domestic use. It also useful in Game zones and route planning.

2.6.2 A* search: minimizing the total estimated solution cost:

It is the best known form of Best first search. It avoids expanding of paths that are already expensive and it expandson most promising paths first.

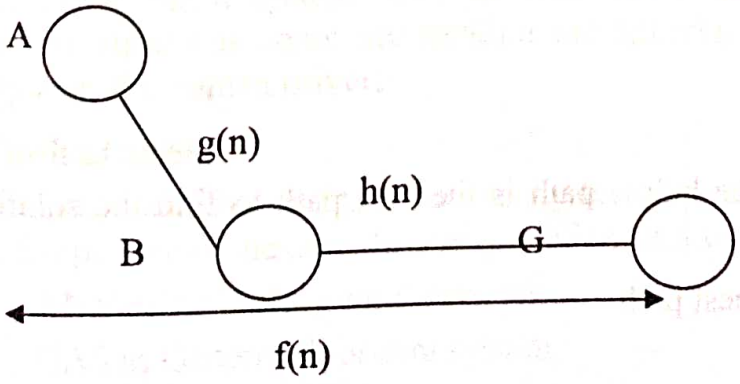
Thus,

$$F(n) = g(n) + h(n)$$

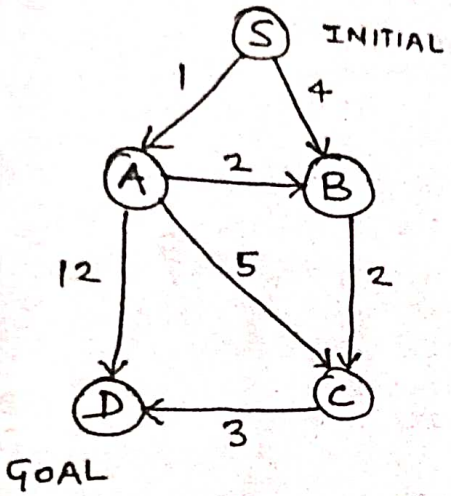
F(n)= estimated total path to reach goal

g(n)= cost to reach the node

h(n)= heuristic value



Let us learn with the following example-



Heuristic Value	
S	7
A	6
B	2
C	1
D	0

S →

f(n) = g(n)+h(n)

= 0+7

=7 & so on..

Similarly we are doing on each state calculate the heuristic value for every path and select the smallest value.

S → A

S → B

S → B → C

S →A → B

S →A → C

S →A → D

S →A → B →C

S →A → B →C →D

S →B → C →D

S →A → C →D

After searching the entire path the below path is the best path to find the solution of a problem.

S →A → B →C →D This is the best path.

2.7 Case Study: Applications of AI in transportation:

❖ Outline:

Artificial Intelligence is changing the transportation field. It helps cars, trucks, trains, ships and airplanes to function automatically to make traffic smooth, safe, clean, smart and more efficient. Due to increase of population, the evolution of transportation increases and it increase the complexity to manage and to analyze the generated data in transportation. To simplify and to maintain the complexity of such requirements AI is being used. AI-based techniques are applied in different form and in different areas whether it is road, aviation, waters or railways. AI helps all transportation safer and smarter. It also reduces human errors which involve traffic accidents and many more. The use of AI in transportation is also known as **intelligent transportation systems (ITS)** which include simplification of huge amount of data generated during transportation whether it is from vehicles as well as drivers.

❖ Method:

Transportation is the basic living standard for the present life. About 305 of the humans spent approx 2 hours of time in a day in travelling. In AI there are many methods to improve the use of transportation system. Particularly the transportation use AI as self control vehicles, predictions of path, and predictions of traffics. It can be divided as-

- Vehicle control system
- Traffic control system
- Traffic prediction system
- Accident prediction system
- Road safety system

The above AI methods decrease the level of accidents and injuries, because of the auto drivers, vehicles controlled so that less accident happens. Some methods are auto driving auto braking system, auto management, and emission control system. Predictions of traffic also reduce the traffic jams and accidents. Road safety systems help to avoid and prevent the possible accidents. Many transports are based on sensors such as GPS, cameras, radar in combination with the devices which transform an input signal into motion. Control systems and software are some of the methods which takeover the drivers functions some are parking the vehicle, Reversing, Driving, Stay, Stop which replaces the human drivers.

❖ Techniques:

AI is the vast area and there are many techniques which are applying at present and also in the past in all the area. In transportation area we grouped some of them are:

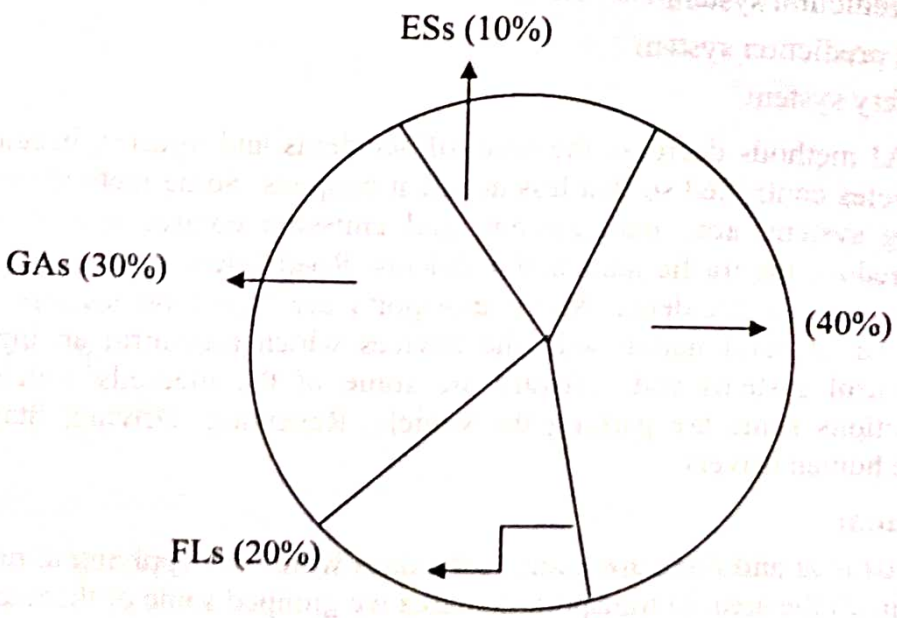
1. ANN as Artificial neural network
2. GAS as Genetic algorithm system

3. FLS as Fuzzy logic system
4. ES as Expert system

Use of AI techniques in different transportation areas

Areas	AI Techniques			
	ANN	GAs	FL	ESs
Vehicle control system	5	8	5	2
Traffic control system	11	4	7	1
Road safety and accident prediction	5	3	6	3

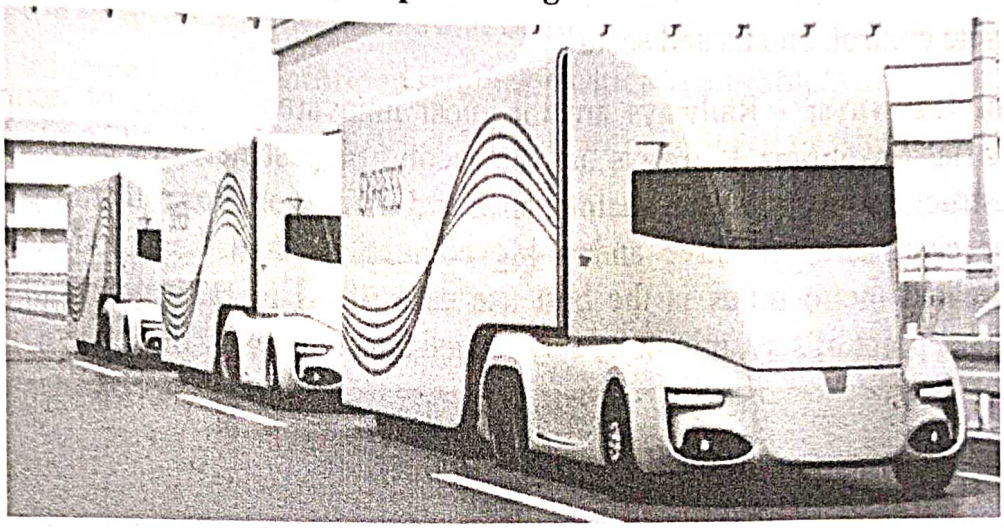
It analyze that artificial neural network technique is the most used techniques among all. If we take an example of roads and traffics ANN and FLs are used to monitoring and to find the stability of vehicles in intelligent system and to avoided traffic jams and predictions of vehicles GAs implemented.



Areas:

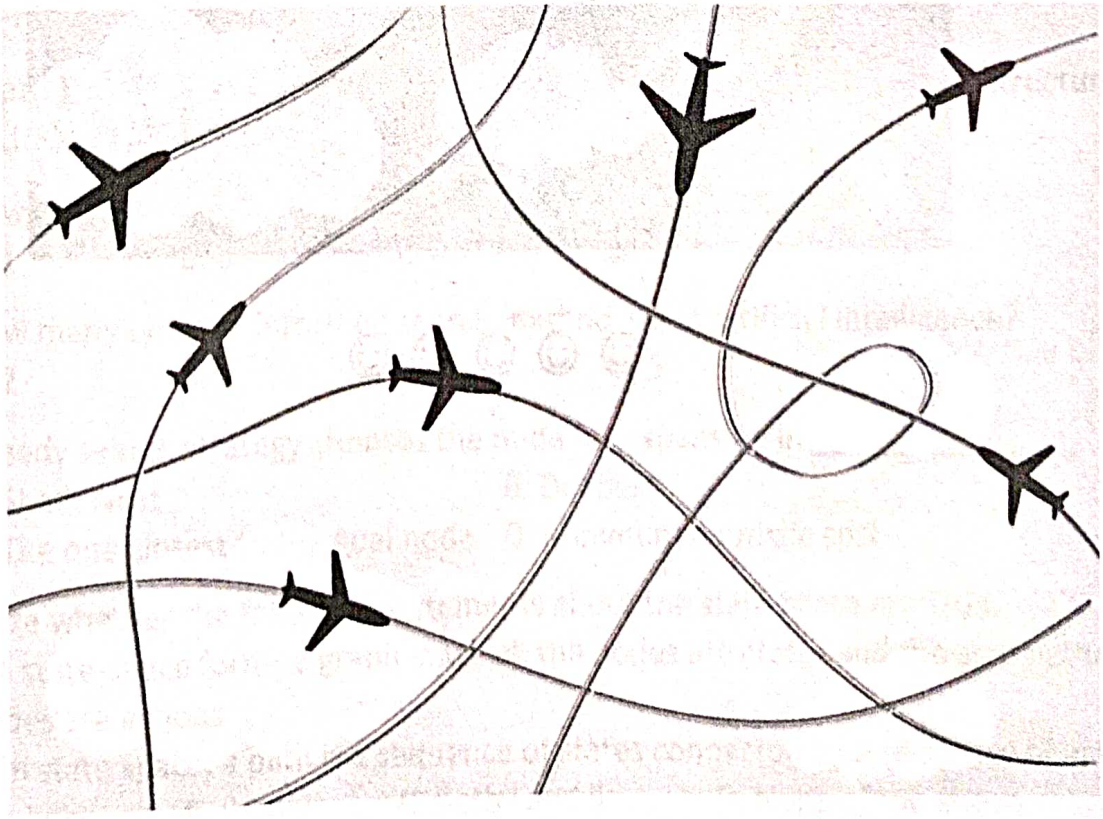
As we already mention with illustrations the areas where AI is used, here we discuss in details.

- **AI in Roads: - Illustration of truck platooning.**



AI makes truck platooning, let 4 vehicles are inline the driver of the first vehicle can control other three trucks or allowing them to let go of the steering wheel and let the truck steer by own or it is also called as automatic driving system. The distance between the vehicles is about to 10-12 meters. This method is very helpful which cause less congestion. Another platform where AI is used in uber drivers were the riders and drivers can recognizes the identity and also the route optimization.

- **AI In Aviation: - Illustration of automated pilots in wars.**



It's about 20 million of flights in 2018. AI is not new in the aviation industry by automating driving, brake, traffic predictions. AI helps aviation systems for safe and secure. AI makes aviation business helpful with machine learning it predict the data what the customer want through google or social media and adjust the content according to customers. Some highly uses cases are fuelling, catering, loading-unloading passengers, baggage control, emails and security.

- **AI in Railways: -** Railways are the most innovative sectors of economy and major industrial revolution. Railways are the second biggest network after web. AI improves infrastructure, manufacture, maintenance and other work for rail transports. It helps with lower cost, safe, effective and a big competition to other modes of transportation. Automated metro trains is the best use case of AI in railways where the source, the distance, point of stay, minute of stay reduce the complexity of this use case.

☺ ☺ ☺ ☺ ☺

Exercises

❖ Answer the following Questions in brief.

1. Explain the difference between BFS and DFS?
2. Explain Informed and Uninformed strategies with a Toy problem
3. What is the concept of Greedy BFS and its algorithm? Explain with real time example?
4. Explain the Applications of Artificial Intelligence in –
   - Gaming
   - Social Media

❖ Multiple Choice Questions:

1. What is the main task of a problem-solving agent?
   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. Both A and B
   D. None of the Above

2. What is Initial state + Goal state in Search Terminology?
   A. Problem Space B. Problem Instance
   C. Problem Space Graph D. Admissibility

3. Depth-First Search is implemented in recursion with _____ data structure.
   A. LIFO B. LILO C. FIFO D. FILO

4. How many types are available in uninformed search method?
   A. 2 B. 3 C. 4 D. 5

5. How many types of informed search method are in artificial intelligence?
   A. 2 B. 3 C. 4 D. 5

6. Greedy search strategy chooses the node for expansion in _____
   A. Shallowest B. Deepest
   C. The one closest to the goal node D. Minimum heuristic cost

7. State whether the following statements about the state space are True.
   i) A state-space forms a graph in which the nodes are states and the arch between nodes are actions.
   ii) In state space, a path is a sequence of states connected by a sequence of actions.
   A. i-only B. ii-only C. Both i and ii D. None of the above

8. The Set of actions for a problem in state space is formulated by which one of the following?

A. Successor function, which takes current action and returns next immediate state

B. Initial state C. Intermediate states D. None of these

9. Which of the following is a touring problem in which each city must be visited exactly once? The purpose is to search for the shortest tour among all the tours.

A. Searching the shortest path between a source and a destination

B. Depth-first search traversal on a given map represented as a graph

C. Map coloring problem D. Travelling Salesman problem

10. Which of the following searching technique takes less memory?

A. Optimal search B. Breadth-First Search

C. Linear Search D. Depth-First Search

Answer:

1. **Ans : C**

   Explanation: The problem-solving agents are one of the goal-based agents.

2. **Ans : B**

   Explanation: Problem Instance : It is Initial state + Goal state.

3. **Ans : A**

   Explanation: Depth-First Search implemented in recursion with LIFO stack data structure.

4. **Ans : D**

   Explanation: The five types of uninformed search method are Breadth-first, Uniform-cost, Depth-first, Depth-limited and Bidirectional search.

5. **Ans : C**

   Explanation: The four types of informed search method are best-first search, Greedy best-first search, A* search and memory bounded heuristic search.

6. **Ans : C**

   Explanation: Sometimes minimum heuristics can be used, sometimes maximum heuristics function can be used. It depends upon the application on which the algorithm is applied.

7. **Ans: C. Both i and ii**

8. **Ans: A** 9. **Ans: D** 10. **Ans: D**