UNIT - IV

Problem Solving

CHAPTER - 2

* **Stochastic Search Methods :-** * Many Games are Unpredictable in Nature, such as those involving dice throw. These Games are called as "Stochastic Games". The outcome of the Game depends on skills as well as luck.

* In the Stochastic Games, the winner of the Game is not only decided by the skill but also by luck.

* Examples are

  → Gambling Game
  → Golf ball Game
  → Backgammon etc

* These methods are designed for Problems with inherit Random Noise or deterministic problems solved by injected Randomness.

* Desired Properties of Search methods are

  → High Probability of finding Near optimal Solutions (Effectiveness)

→ Short Processing time (Efficiency)

* They are usually Conflicting : a Compromise is offered by Stochastic techniques where certain steps are based on Random choice.

* Many Stocastic Search techniques are inspired by Processes found in Nature (Like Temperature).

* Why Stochastic Search?

→ It is the method of choice for solving many hard Combinatorial Problems.

→ Solution of large Propositional Sattisfia bility Problems.

→ Solution of large Travelling salesman Problems Good Results in New Application Areas.

* Ex:- Travelling Salesman Problem.

→ salesman has list of cities, each of which must be visited Exactly once

→ In list, there are direct roads between each pair of cities.

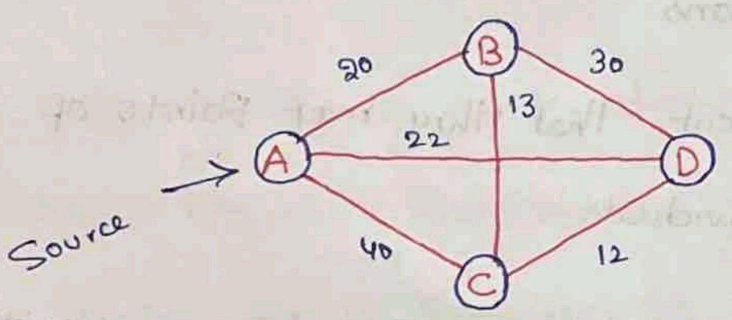
→ To find Route, salesman should follow Shortest possible Round Trip.

→ State is Represented as Pair of any two cities and distance between them.

State Space : Initial State (State A)

↳ (A —20— B), (A —40— C), (A —22— D)

Let us Consider the following Graph



Possible Routes :-

A → B → C → D → A = 67 ✓ optimal

A → B → D → C → A = 102

A → D → B → C → A = 105

A → D → C → B → A = 67 ✓ optimal

A → C → D → B → A = 102

A → C → B → D → A = 105

* Heuristic Search Techniques In AI :-

* Heuristic is a technique that improves the efficiency of a Search Process, possibly by Seacrificing claims of Completeness.

* Using good Heuristic, we can hope to get good solutions to hard problems in less than exponential times.

* There are good to the extent that they point in General interesting directions.

* They are bad to the extent that they may Points of intersts of Particular individuals.

* Some heuristics help to guide a Search Process with out Sacrificing any claims to Completeness.

* Some may occasionally cause an excellent path to be overlooked.

* on an Average, they improve the Quality of the paths that are explored.

* Why Heuristics

1) without it we could become hopelessly ensnared in a Combinatorial explosion.

2) Rarely do we actualy need the optimum solution, a good approximation will usually Serve well.

3) Although the approximations produced by heuristics, may not be very good in the worst case but this case rarely arise the Real world.

4) Trying to understand why a heuristic works or why it doesn't work, often leads to a deeper understanding of a Problem.

* **Heuristic Function :-** * It is a function that maps from problem state descriptions to measures of desirability Usually Represented as numbers.

* Value of a heuristic function at a given node in the Search Process depends on

1) Aspects of the Problem State Considered.

2) Evaluation of those Aspects

3) Weights given to the individual Aspects

* The purpose of a heuristic function is to guide the Search Process in the most profitable direction by Suggesting which path to follow first when more than one is available.

* The more accurate function estimates the true merits of each node in the Search Tree or Graph, the more direct the Solution Process.

* Heuristic Search is also called as Informed Search Strategies. It can be catogorized into following types.

→ Greedy Best First Search

→ A* Search

→ AO* Search

→ Problem Reduction

→ Hill Climbing

* Informed Search Uses domain knowledge i.e The problem information is available which can guide the Search.

* Informed Search Strategies find the solution in terms of more efficient than Uninformed.

* A Heuristic is a way which might not always gurantee for the best solution but guranteed to find a good solution in a Reasonable time.

* It can solve much more complex problems which could not solved in another way.

* AO* Search :- * AO* Algorithm is a heuristic Search Algorithm in AI.

* AO* Algorithm uses the concept of AND – OR Graphs to decompose any complex problem given into smaller set of Problems which are further Solved.

* The AO* Algorithm works on the following formula

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

where

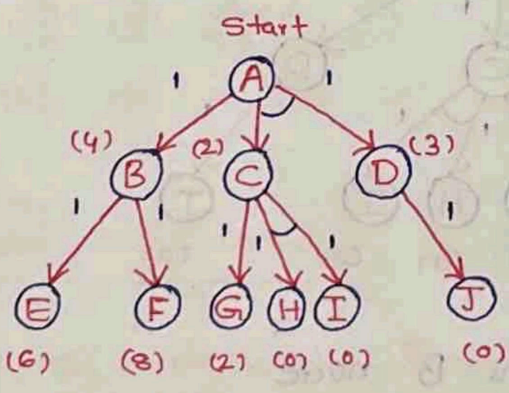
g(n) : The actual cost of traversal from initial State to the current State.

h(n) : The estimated cost of traversal from the current State to the Goal State.

f(n) : The actual cost of traversal from the initial State to the Goal State.

* Ex:- In the below Example all numbers in brackets are heuristic values i.e h(n)

* Each Edge is considered to have a value of 1 by default.



* Step - 1 : Starting from Node A, we first calculate the best Path.

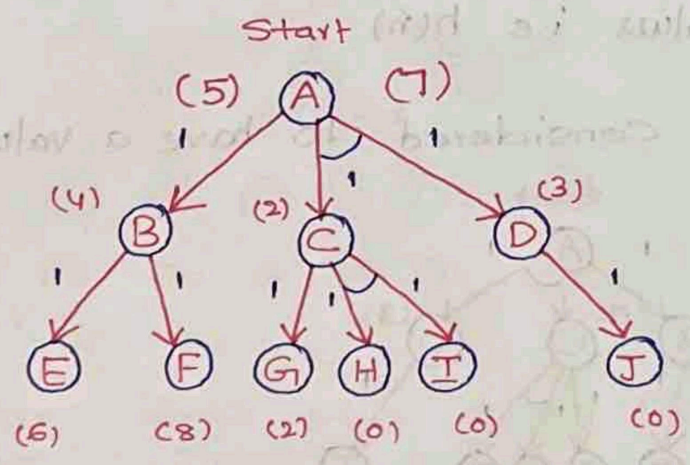
$$f(A-B) = g(B) + h(B)$$
$$= 1 + 4 = 5$$

where 1 is the default cost value of travelling from A to B and 4 is the estimated cost from B to Goal State.

$$f(A-C-D) = g(C) + h(C) + g(D) + h(D)$$
$$= 1 + 2 + 1 + 3 = 7$$

Here we are calculating the path cost as C and D because they have the AND - Arc.

The default cost value of travelling from A-C is 1 and from A-D is 1, but the heuristic value given for C and D are 2 and 3 Respectively. Hence making the cost as 7.



* Step - 2 : from the B node

$$f(B-E) = g(E) + h(E)$$
$$= 1 + 6 = 7$$

$$f(B-F) = g(F) + h(F)$$

$$= 1 + 8 = 9$$

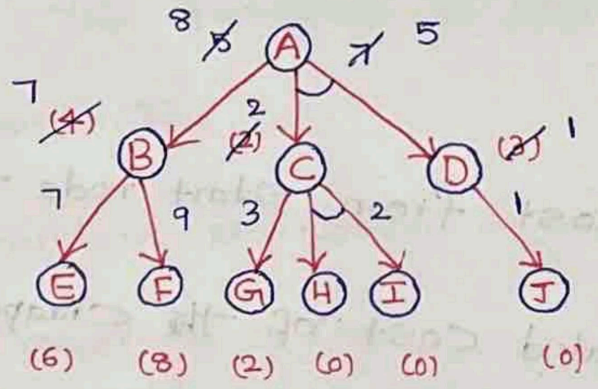
* Hence, the B-E path has lesser cost. Now the heuristic have to be updated since there is a difference between actual and heuristic value of B.

* The minimum cost path is choosen and is updated as the heuristic, in our case the value is 7.

* And because of change in heuristic of B there is also change in heuristic of A which is to be calculated again

$$f(A-B) = g(B) + \text{updated } (h(B))$$

$$= 1 + 7 = 8$$



* Evaluation Functions :- * An Evaluation function, also known as "Heuristic Evaluation function" or "static Evaluation Function". It is a function used by game playing Computer Programs to estimate the value or good ness

of a position in a Game Tree.

* In GBFS (Greedy Best First Search), each node is expanded using the following Evaluation function.

$$f(n) = h(n)$$

where

$h(n)$ = Estimated cost of the cheapest path from Node n to a Goal Node.

* we use the "Straight-Line distance" heuristic called h_{SLD} to find a solution for reaching the Goal State

* The Evaluation function of A* Search is

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

where

$g(n)$ = path cost from start node to node 'n'

$h(n)$ = Estimated cost of the cheapest path from node 'n' to the Goal Node.

$f(n)$ = Estimated cost of the cheapest solution to 'n'.

* The Evaluation fuction of AO* Search is

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

where

g(n) : The actual cost of traversal from initial State to the current State.

h(n) : The estimated cost of traversal from the current State.

f(n) : The actual cost of traversal from the initial State to the Goal State.

* Informed Search Techniques Require an Evaluation fuction. This fuction is applied to States (nodes) in the Search Tree.

* This fuction incorporates an estimate of the path cost from the State to a Goal.

* Minmax Algorithm or Alpha Beta Pruning in Game theory each leaf node had a value associated with it. we had stored this value in an Array.

* But in the Real world when we are creating a Program

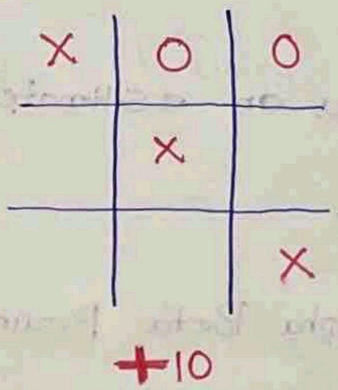
to play Tic - Tac - Toe, Chess, Backgammon etc. we need to implement a function that calculates the value of the board depending on the placement of pieces on the board.

* The evaluation function is Unique for every type of Game.

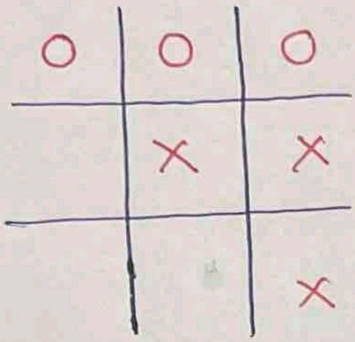
* Let us Consider, evaluation function for Tic - Tac - Toe to give high value for a board if maximizer's Turn or a low value for the board if minimizer's Turn.

* For this Scenario let us Consider X as the maximizer and O as the minimizer.

1) If X wins on the board we give it a positive value of +10

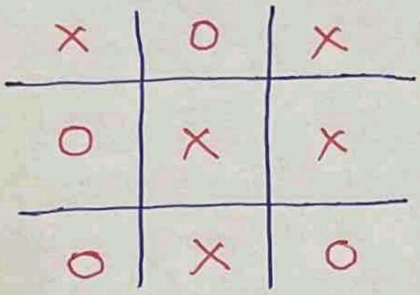


2) If O wins on the board we give it a Negative value of -10



-10

3) If no one has won or the Game Results in a draw then we give a value of $+0$



$+0$

* Adversial Search : Games, optimal decisions in Games, Real Time decisions, Imperfect Decisions

* Alpha - Beta Pruning

— Refer Unit - VI (Chapter - 2)

* Problem solving : Formulating Problems, Problem Types solving Problems by Searching, Heuristic Search Techniques Constraint Satisfaction Problems.

— Refer Unit - III (Chapter - 1 & 2)