ARTIFICIAL INTELLIGENCE ASSIGNMENT- 4

Ques1. Explain hill climbing algorithm in detail?

Ans. Hill climbing algorithm is a local search algorithm which continuously moves in the direction of increasing elevation/value to find the peak of the mountain or best solution to the problem. It terminates when it reaches a peak value where no neighbor has a higher value.

Hill climbing algorithm is a technique which is used for optimizing the mathematical problems. One of the widely discussed examples of Hill climbing algorithm is Traveling-salesman Problem in which we need to minimize the distance traveled by the salesman.

It is also called greedy local search as it only looks to its good immediate neighbor state and not beyond that.

A node of hill climbing algorithm has two components which are state and value.

Hill Climbing is mostly used when a good heuristic is available.

In this algorithm, we don't need to maintain and handle the search tree or graph as it only keeps a single current state.

Ques2. Describe A* search technique. Prove that A* is complete and optimal?

Ans. A* Search algorithm is one of the best and popular technique used in path-finding and graph traversals.

Informally speaking, A* Search algorithms, unlike other traversal techniques, it has “brains”. What it means is that it is really a smart algorithm which separates it from the other conventional algorithms. This fact is cleared in detail in below sections.

And it is also worth mentioning that many games and web-based maps use this algorithm to find the shortest path very efficiently (approximation).

Prove that A* is complete and optimal:

- A* is complete (finds a solution, if one exists) and also optimal (finds the optimal path to a goal) if:

• the branching factor is finite

• arc costs are > 0

• h(n) is admissible which means an underestimate of the length of the shortest path from n to a goal node.

This property is known as the admissibility of A*

Ques3. Write a note on AO* search technique?

Ans. AO* is a part of informed search technique and use heuristic values to solve the problem.

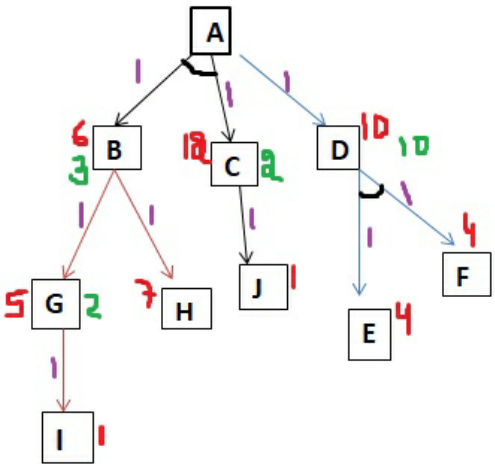
The solution is guaranteed in both algorithm.

AO* always gives an optimal solution (shortest path with low cost) But It is not guaranteed to that AO* always provide an optimal solutions.

Reason: Because AO* does not explore all the solution path once it got solution.

How AO* works

Let's try to understand it with the following diagram



The algorithm always moves towards a lower cost value.

Basically, We will calculate the cost function here (F(n)= G (n) + H (n))

H: heuristic/ estimated value of the nodes. and G: actual cost or edge value (here unit value).

Here we have taken the edges value 1 , meaning we have to focus solely on the heuristic value.

Ques4. "Game playing is a search technique with a difference". explain the statement?

Ans. Game Playing is an important domain of artificial intelligence. Games don't require much knowledge; the only knowledge we need to provide is the rules, legal moves and the conditions of winning or losing the game.

Both players try to win the game. So, both of them try to make the best move possible at each turn. Searching techniques like BFS(Breadth First Search) are not accurate for this as the

branching factor is very high, so searching will take a lot of time. So, we need another search procedures that improve –

Generate procedure so that only good moves are generated.

Test procedure so that the best move can be explored first.

Ques5. Discuss Alpha-Beta pruning with example?

Ans. Alpha-beta pruning is a modified version of the minimax algorithm. It is an optimization technique for the minimax algorithm.

As we have seen in the minimax search algorithm that the number of game states it has to examine are exponential in depth of the tree. Since we cannot eliminate the exponent, but we can cut it to half. Hence there is a technique by which without checking each node of the game tree we can compute the correct minimax decision, and this technique is called pruning. This involves two threshold parameter Alpha and beta for future expansion, so it is called alpha-beta pruning. It is also called as Alpha-Beta Algorithm.

Alpha-beta pruning can be applied at any depth of a tree, and sometimes it not only prune the tree leaves but also entire sub-tree.

The two-parameter can be defined as:

Alpha: The best (highest-value) choice we have found so far at any point along the path of Maximizer. The initial value of alpha is -∞.

Beta: The best (lowest-value) choice we have found so far at any point along the path of Minimizer. The initial value of beta is +∞.