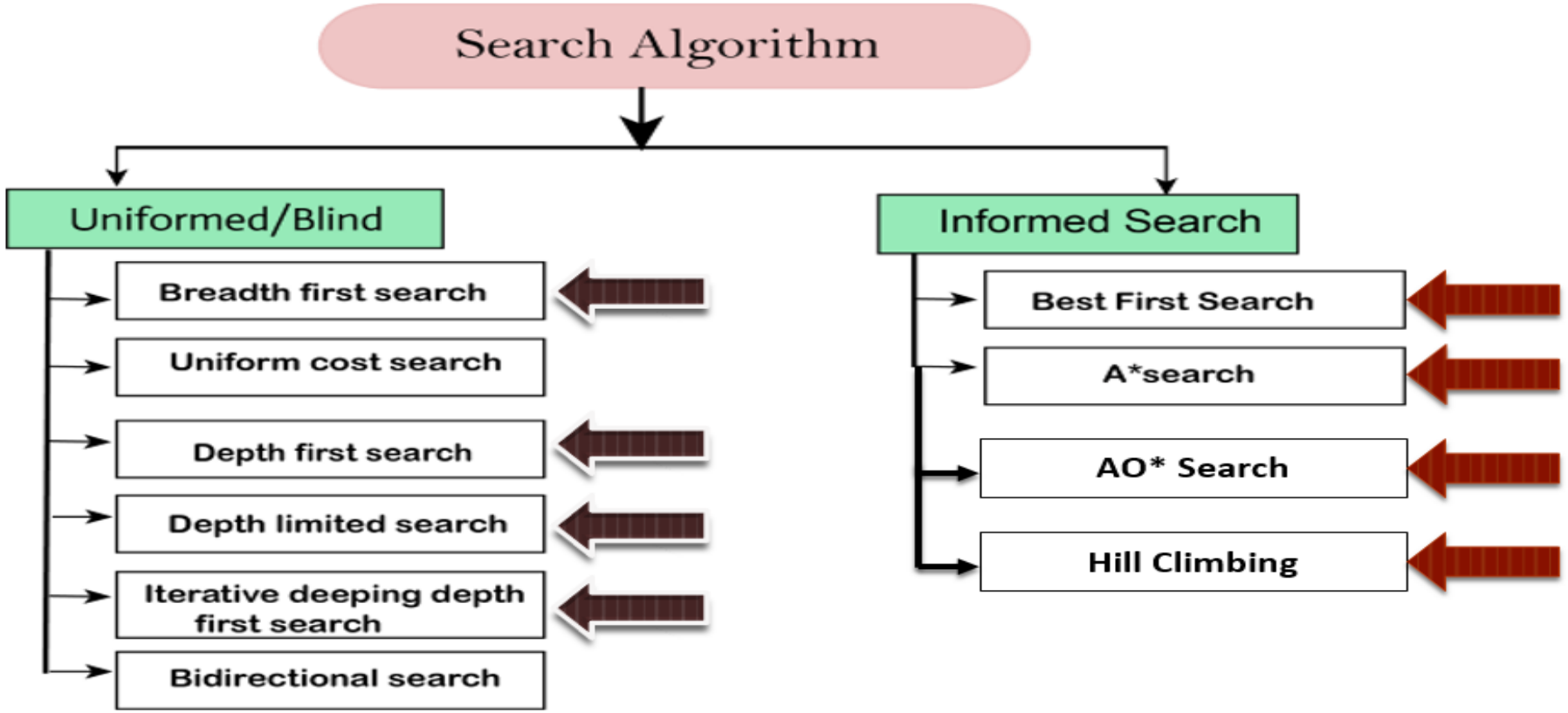
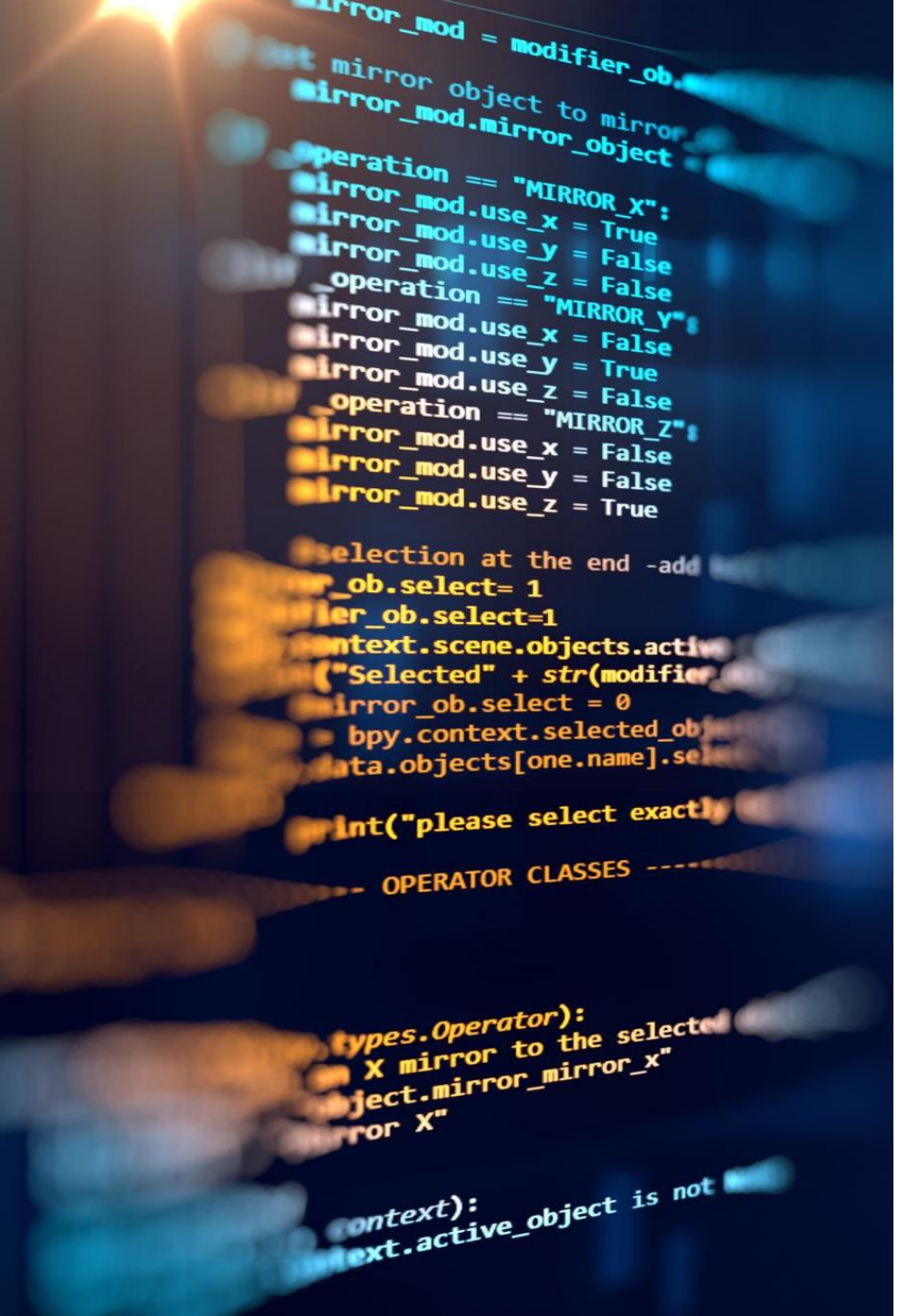
Graphic Era Deemed to be University DEHRADUN

TCS 421

INTRODUCTION TO COURSE – FUNDAMENTAL OF STATISTICS AND AI

Searching


Search Algorithm
Uniformed/Blind
Breadth first search
Uniform cost search
Depth first search
Depth limited search
Iterative deeping depth first search
Bidirectional search
Informed Search
Best First Search
A*search
AO* Search
Hill Climbing



Topics Covered

- AO* Search algorithm
- Hill Climbing algorithm

AO* Search algorithm

AO* search Algorithm is based on problem decomposition (Breakdown problem into small pieces) When a problem can be divided or decomposed into a set of sub problems, where each sub problem can be solved separately and for each sub problem, sub solution is evaluated and a combination of these sub solutions will be a whole solution, AND OR graphs or AND OR trees are used for representing this solution.

AO* Search algorithm

AO* is informed search algorithm, work based on heuristic. We already know about the divide and conquer strategy, a solution to a problem can be obtained by decomposing it into smaller sub-problems.

Each of this sub-problem can then be solved to get its sub solution. These sub solutions can then recombined to get a solution as a whole. That is called is Problem Reduction. AND-OR graphs or AND – OR trees are used for representing the solution.

AO* Search algorithm

Algorithm:

Step 1: Place the starting node into OPEN.

Step 2: Compute the most promising solution tree say T0.

Step 3: Select a node n that is both on OPEN and a member of T0. Remove it from OPEN and place it in CLOSE.

Step 4: If n is the terminal goal node then leveled n as solved and leveled all the ancestors of n as solved. If the starting node is marked as solved then success and exit.

Step 5: If n is not a solvable node, then mark n as unsolvable. If starting node is marked as unsolvable, then return failure and exit.

Step 6: Expand n. Find all its successors and find their h (n) value, push them into OPEN

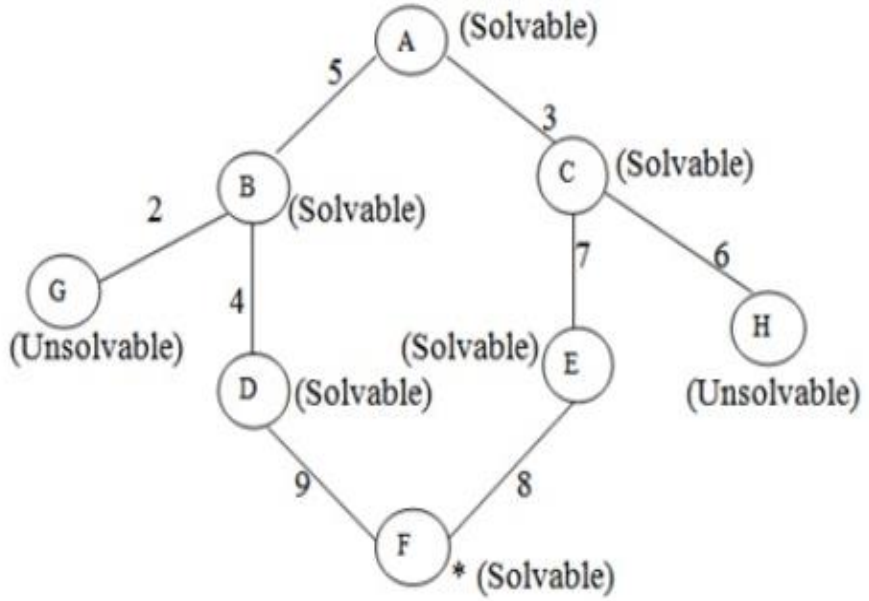
Step 7: Return to Step 2.

Step 8: Exit.

AO* Search algorithm

Example:

Let us take the following example to implement the AO* algorithm.



AO* Search algorithm

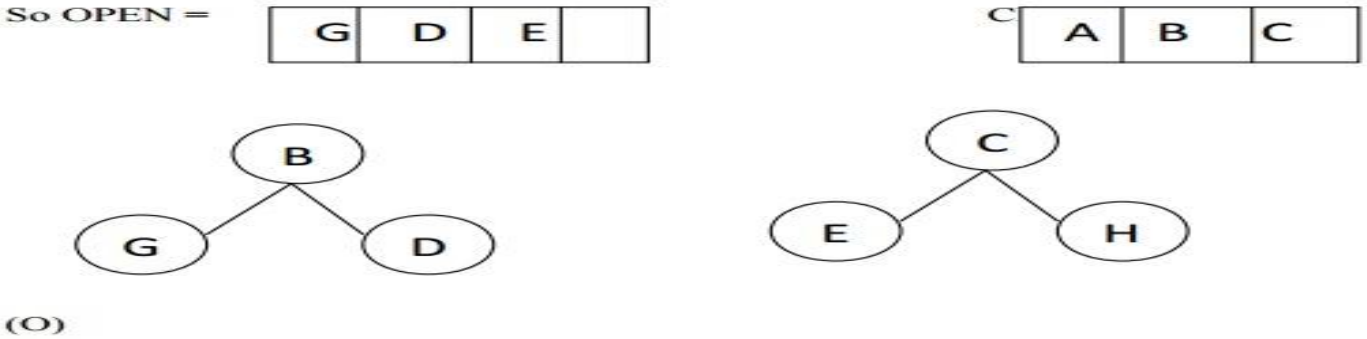
Step 1: In the above graph, the solvable nodes are A, B, C, D, E, F and the unsolvable nodes are G, H. Take A as the starting node. So place A into OPEN.

Step 2:

The children of A are B and C which are solvable. So place them into OPEN and place A into the CLOSE.

Step 3:

Now process the nodes B and C. The children of B and C are to be placed into OPEN. Also remove B and C from OPEN and place them into CLOSE.



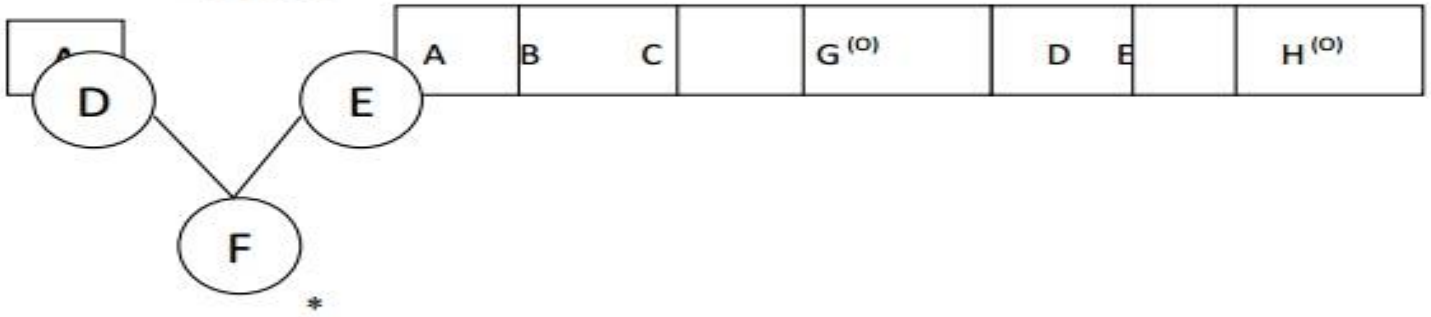
AO* Search algorithm

'O' indicated that the nodes G and H are unsolvable.

Step 4:

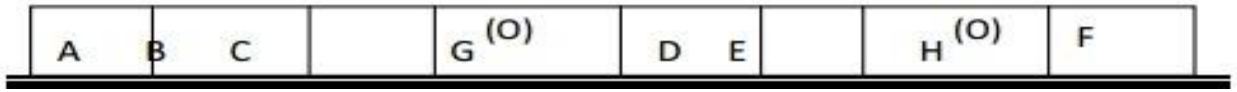
As the nodes G and H are unsolvable, so place them into CLOSE directly and process the nodes D and E.

i.e. OPEN = CLOSE =



Step 5:

Now we have been reached at our goal state. So place F into CLOSE.



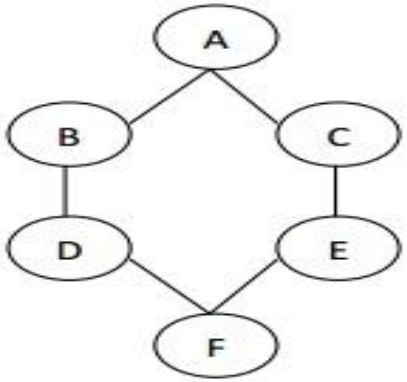
AO* Search algorithm

i.e. CLOSE =

Step 6:

Success and Exit

AO* Graph:



Figure

AO* Search algorithm

Advantages:

It is an optimal algorithm.

If traverse according to the ordering of nodes. It can be used for both OR and AND graph.

Disadvantages:

Sometimes for unsolvable nodes, it can’t find the optimal path. Its complexity is than other algorithms.

Hill Climbing

- 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.

Hill Climbing

- 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.

Hill Climbing

Features of Hill Climbing:

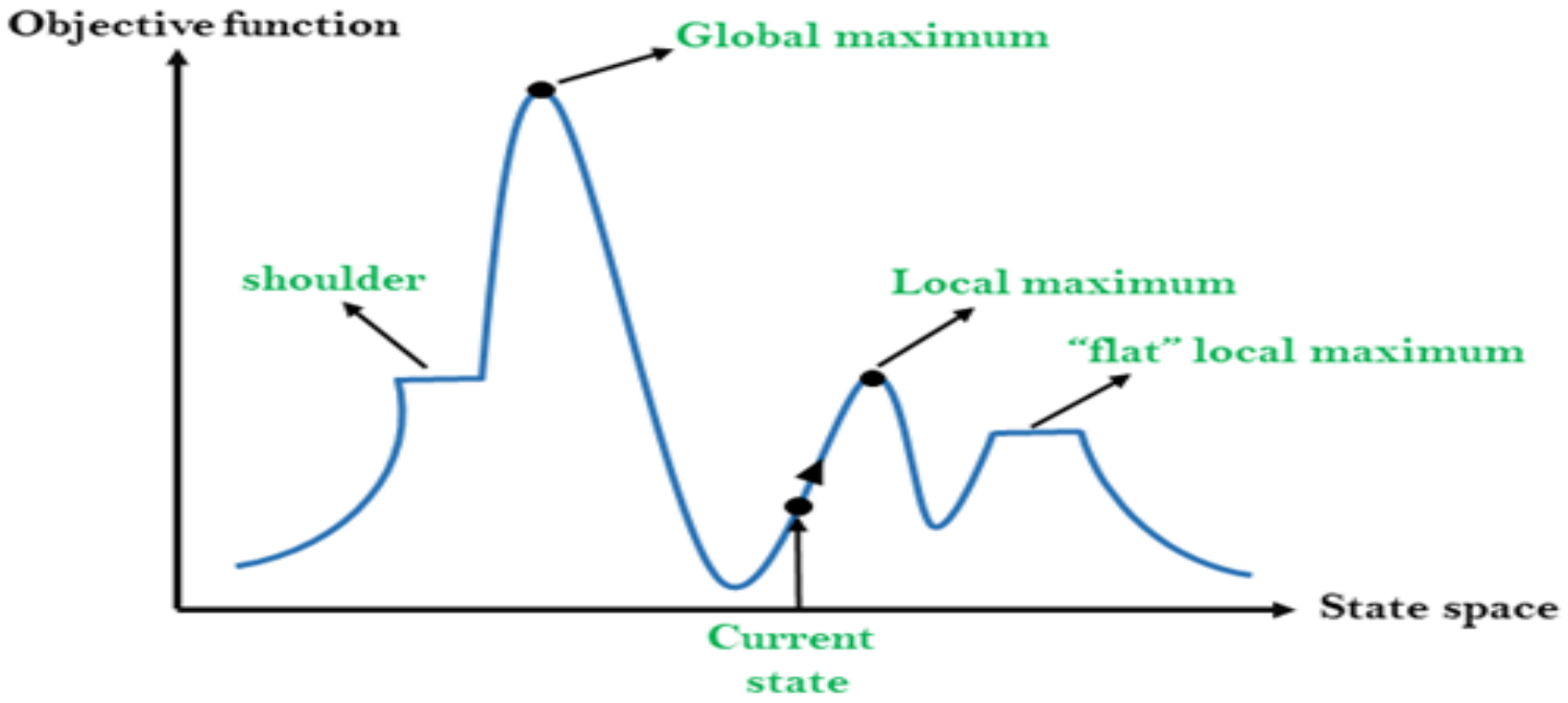
Following are some main features of Hill Climbing Algorithm:

- **Generate and Test variant:** Hill Climbing is the variant of Generate and Test method. The Generate and Test method produce feedback which helps to decide which direction to move in the search space.
- **Greedy approach:** Hill-climbing algorithm search moves in the direction which optimizes the cost.
- **No backtracking:** It does not backtrack the search space, as it does not remember the previous states

State-space Diagram for Hill Climbing:

- The state-space landscape is a graphical representation of the hill-climbing algorithm which is showing a graph between various states of algorithm and Objective function/Cost.
- On Y-axis we have taken the function which can be an objective function or cost function, and state-space on the x-axis.
- If the function on Y-axis is cost then, the goal of search is to find the global minimum and local minimum. If the function of Y-axis is Objective function, then the goal of the search is to find the global maximum and local maximum.

Different regions in the state space landscape:



Types of Hill Climbing Algorithm:

- Simple hill Climbing
- Steepest-Ascent hill-climbing
- Stochastic hill Climbing