CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

UNIVERSITY INSTITUTE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGG.

Bachelor of Engineering (Computer Science & Engineering)

Principles of Artificial Intelligence (20CST-258)

AO* Algorithm

DISCOVER . LEARN . EMPOWER

Outline

- Informed Search
- Heuristics function
- AO* Algorithm

The Informed Search

- Informed search methods use knowledge about the problem domain and choose promising operators first.
- These heuristic search methods use heuristic functions to evaluate the next state towards the goal state.

  For finding a solution, by using the heuristic technique, one should carry out the following steps:-

  1. Add domain: specific information to select what is the best path to continue searching along
  2. Define a heuristic function h(n): that estimates the ‗goodness' of a node n. Specifically, h(n) = estimated cost(or distance) of minimal cost path from n to a goal state.
  3. The term, heuristic means "serving to aid discovery" and is an estimate, based on domain specific information that is computable from the current state description of how close we are to a goal.

Heuristics function

- Heuristic is a function which is used in Informed Search, and it finds the most promising path.
- It takes the current state of the agent as its input and produces the estimation of how close agent is from the goal.
- Heuristic function estimates how close a state is to the goal.
- It is represented by h(n), and it calculates the cost of an optimal path between the pair of states.
- Admissibility of the heuristic function is given as:

$$h(n) <= h^*(n)$$

- Here h(n) is heuristic cost, and h*(n) is the estimated cost. Hence heuristic cost should be less than or equal to the estimated cost.

Characteristics of heuristic search

- Heuristics are knowledge about domain, which help search and reasoning in its domain.
- Heuristic search incorporates domain knowledge to improve efficiency over blind search.
- Heuristic is a function that, when applied to a state, returns value as estimated merit of state, with respect to goal.
- Heuristic evaluation function estimates likelihood of given state leading to goal state.
- Heuristic search function estimates cost from current state to goal, presuming function is efficiency

Pure Heuristic Search

- Pure heuristic search is the simplest form of heuristic search algorithms.
- It expands nodes based on their heuristic value h(n). It maintains two lists, OPEN and CLOSED list.
- In the CLOSED list, it places those nodes which have already expanded and in the OPEN list, it places nodes which have yet not been expanded.
- On each iteration, each node n with the lowest heuristic value is expanded and generates all its successors and n is placed to the closed list. The algorithm continues unit a goal state is found.

Heuristic Search

- **Pure Heuristic Search**
  - In the informed search, there are two main algorithms:
    - Best First Search Algorithm(Greedy search)
    - A* Search Algorithm
- **Hill Climbing**
    - Simple hill Climbing
    - Steepest-Ascent hill-climbing
    - Stochastic hill Climbing
  - AO* Algorithm

AO* (AND-OR) Algorithm

- The Depth first search and Breadth first search given earlier for OR trees or graphs can be easily adopted by AND-OR graph.
- The main difference lies in the way termination conditions are determined, since all goals following an AND nodes must be realized; where as a single goal node following an OR node will do. So for this purpose we are using AO* algorithm.
- Like A* algorithm here, AO*uses two arrays and one heuristic function.
- **OPEN:** It contains the nodes that has been traversed but yet not been marked solvable or unsolvable.
- **CLOSE**: It contains the nodes that have already been processed.

AO* (AND-OR) Algorithm

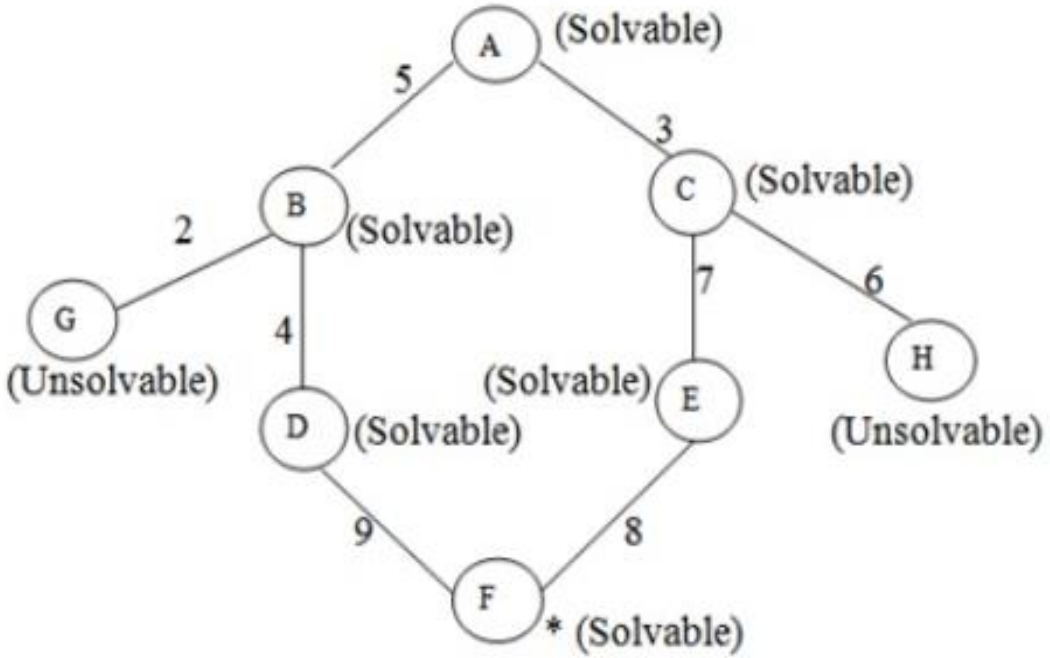
- In an AND-OR graph AO* algorithm is an efficient method to explore a solution path.
- AO* algorithm works mainly based on two phases. First phase will find a heuristic value for nodes and arcs in a particular level. The changes in the values of nodes will be propagated back in the next phase.
- In order to find solution in an AND-OR graph AO* algorithm works well similar to best first search with an ability to handle the AND arc appropriately.
- The algorithm finds an optimal path from initial node by propagating the results like solution and change in heuristic value to the ancestors as in 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.

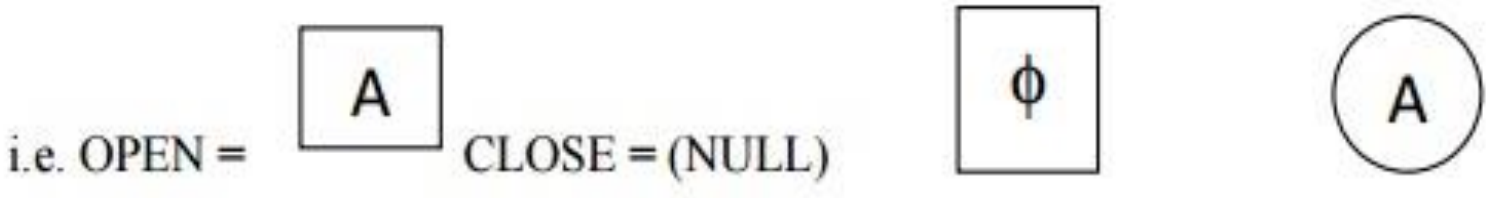
Example

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



Example

- **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.



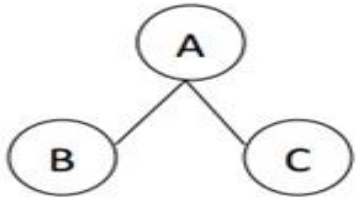
Step 2:

Example

Step 2:

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

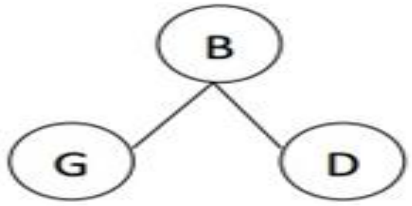
i.e. OPEN = | B | C | CLOSE = | A |

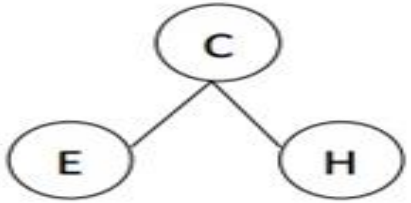


Step 3:

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.

So OPEN = | G | D | E | | C | A | B | C |





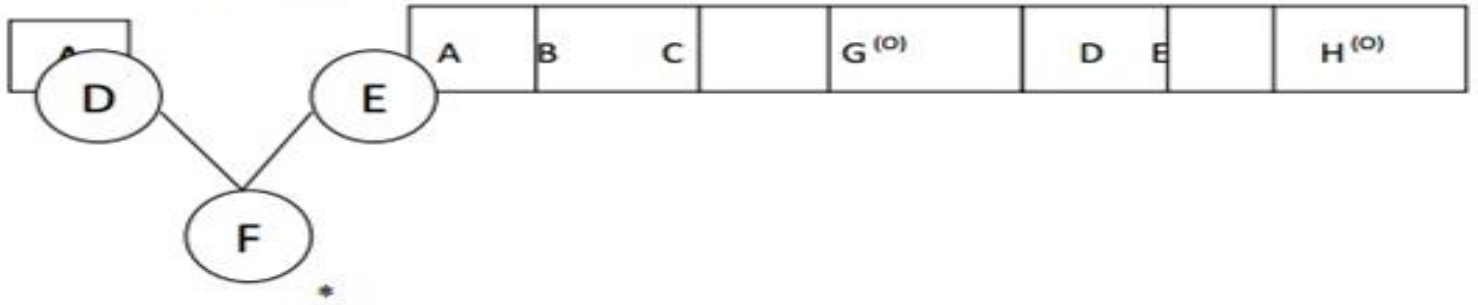
'O' indicated that the nodes G and H are unsolvable.

Example

Step 4:

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

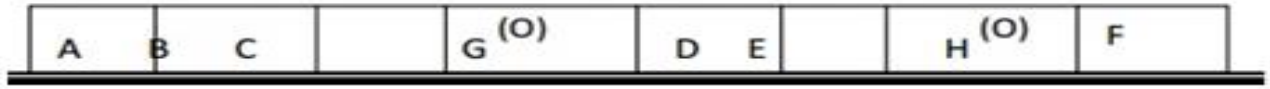
i.e. OPEN = CLOSE =



Step 5:

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

i.e. CLOSE =

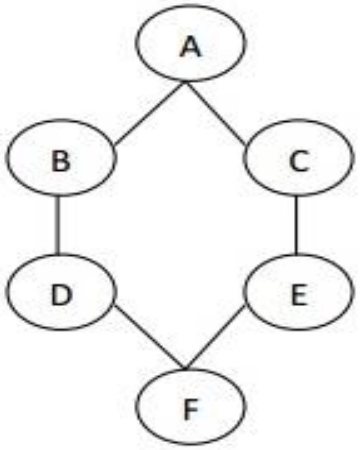


Step 6:

Success and Exit

AO* Graph

AO* Graph:



- **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

Time to Think..

- Analyze the difference between A* and AO* Search Algorithms.

THANK YOU