

Assignment - 2

Title : Searching algorithm (Informed search)

Aim : Implement A star algorithm for any game search problem.

Theory : A\* search is the most commonly known form of best first search. It uses heuristic function  $h(n)$ , cost to reach the node  $n$  from the start  $g(n)$ . It has combined features of UCS and greedy best first search, by which it solve the problem efficiently. A\* search algorithm finds the shortest path through the search space using the heuristic function. This search algorithm expands less search tree and provides optimal result faster. A\* algorithm is similar to UCS except that it uses  $g(n) + h(n)$  instead of  $g(n)$ .

In A\* search algorithm, we use search heuristic as well as the cost to reach the node. Hence we can combine both costs as following, and this sum is called as a fitness number.

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

Estimated cost of the cheapest Solution	cost to reach node $n$ from start state	Cost to reach from node $n$ to goal node
---	---	--

## Algorithm of A\* search :

- Step 1 : Place the starting node in the OPEN list.
- Step 2 : Check if the OPEN list is empty or not, if the list is empty then return failure and stop.
- Step 3 : Select the node from the OPEN list which has the smallest value of evaluation function ( $g+h$ ), if the node  $n$  is goal node then return success and stop, otherwise.
- Step 4 : Expand node  $n$  and generate all of its successors, and put in into the closed list. For each successor  $n'$ , check whether  $n'$  is already in the OPEN or CLOSED list, if not then compute evaluation function for  $n'$  and place into open list.
- Step 5 : Else if node  $n'$  is already in OPEN and CLOSED, then it should be attached to the back pointer which reflects the lowest  $g(n')$  value.
- Step 6 : Return to step 2.

## Advantages :

1. A\* search algorithm is the best algorithm than other search algorithms.
2. A\* search algorithm is optimal and complete.
3. This algorithm can solve very complex problems.

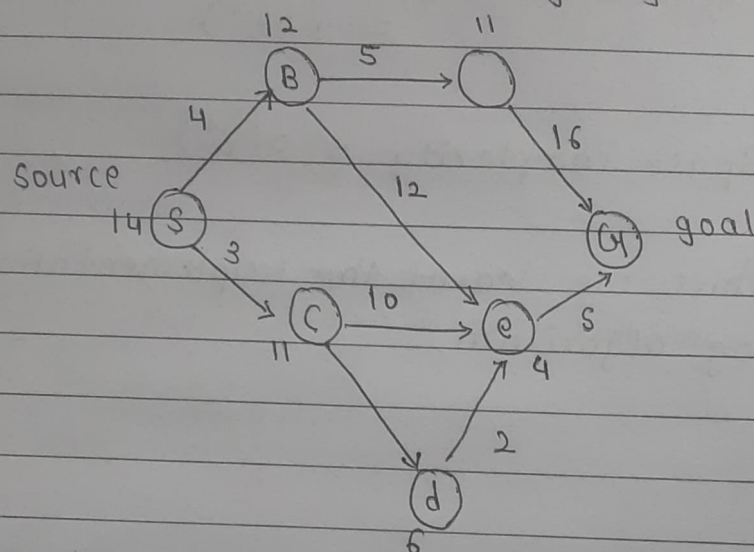
## Disadvantages :

1. It does not always produce the shortest path as it



- mostly based on heuristics and approximation
2. A\* search algorithm has some complexity issues.
  3. The main drawback of A\* is memory requirement as it keeps all generated nodes in the memory, so it is not practical for various large scale problems.

Example of A\* searching algorithm:



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

Actual cost from  
start node to n

Estimation cost from  
n to goal node

$$f(s) = 0 + 14$$

$$= 14$$

$$s \rightarrow B$$

$$= 4 + 12 = 16$$



$$sB \rightarrow F$$

$$= 4 + 5 + 11$$

$$= 20$$

$$sB \rightarrow E$$

$$4 + 12 + 4$$

$$= 20$$

$$s \rightarrow C$$

$$= 3 + 11 = 14 \checkmark$$



$$sC \rightarrow E$$

$$= 3 + 10 + 4 = 17$$

$$sC \rightarrow D$$

$$= 3 + 7 + 6 = 16 \checkmark$$

$$\begin{aligned} \text{scd} &\rightarrow e \\ &= 3+7+2+4 = 16 \end{aligned}$$

$$\begin{aligned} \text{scde} &\rightarrow G \\ &= 3+7+2+5 = 17 \end{aligned}$$

$$\begin{aligned} \text{Time complexity} &= O(4+E) \\ &= O(b^d) \end{aligned}$$

$$\text{Space complexity} = O(b^d)$$

Conclusion: Thus we learnt the implementation of A slot se  
ing algorithm.