Optimality of A* search –

A* has following properties →

(1) → The tree search version of A* is optimal if h(n) is admissible.

(2) → The graph search version of A* is optimal if h(n) is consistent.

→ lets try to prove the 2nd claim :-

If h(n) is consistent, then the values of f(n) along any path are non-decreasing.

* for any node n, and for any successor n' of n the following inequality should hold

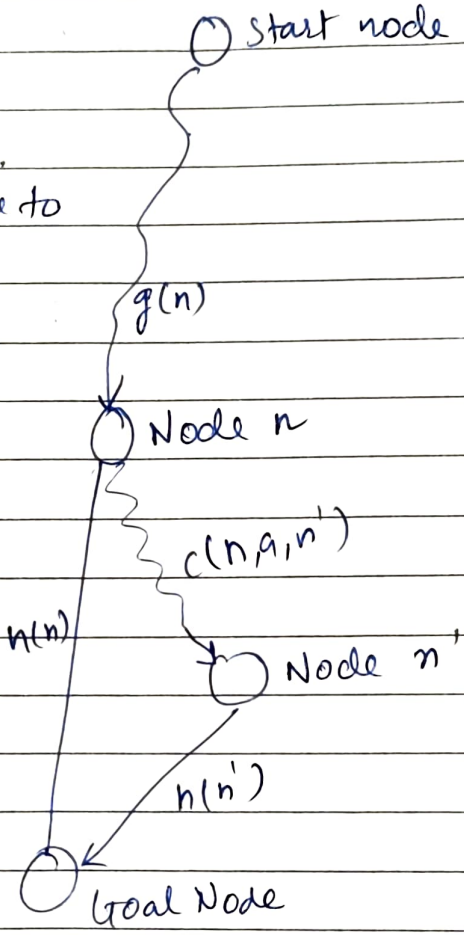
$$h(n) \leq c(n,a,n') + h(n') \quad \text{—— (A)}$$

where c is the cost to go from n to n' by performing action a.

Lets say during our search process, we have reached node n from start node, and the cost for start node to Node n is g(n).

Now consider a node n' which is a successor of node n. Let c(n,a,n') be the actual cost to go from n to n'.

we also have a goal node & let h(n) be the estimate of cost from node n to goal node & let h(n') be the estimate of the cost from node n' to goal node.



Now, in case of A* search we know, that the evaluation function is

$$f(n) = g(n) + h(n) \quad \text{---} \quad (3)$$

Similarly

$$f(n') = g(n') + h(n') \quad \text{---} \quad (1)$$

from the above dig → $g(n') = g(n) + c(n,a,n')$ — (2)

in eqn (1) replace value of $g(n')$ by eqn (2)

$$f(n') = g(n) + c(n,a,n') + h(n') \quad \text{--- (3)}$$

In the above eqn $c(n,a,n') + h(n')$ is a part of inequality of eqn (A).

In eqn (3), so, now if we replace $c(n,a,n') + h(n')$ with $h(n)$, it means we are replacing with a smaller value since eqn (A)

$$h(n) \leq c(n,a,n') + h(n')$$

so in eqn (3)
RHS will become less than $f(n')$, mean

$$f(n') \geq g(n) + h(n)$$

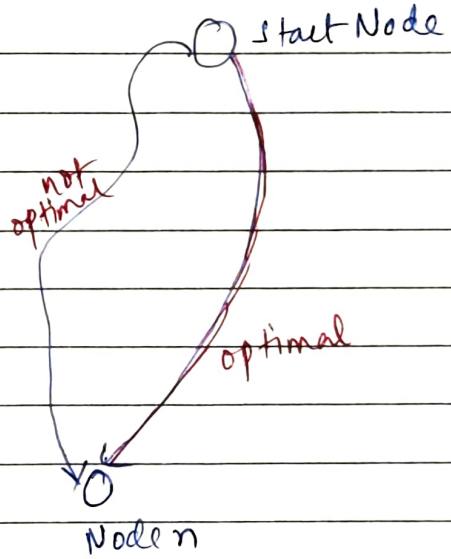
↳ replacing this with eqn (B)

$f(n') \geq f(n)$ → so along any path if we have a node n of we have a successor n' of n the $f(n') \geq f(n)$, so we see the f value only increase or remain the same means it is non-decreasing function.

Now we make another claim —

** Whenever A* selects a node for expansion, the optimal path to ~~the~~ that node has been found.

Proof →

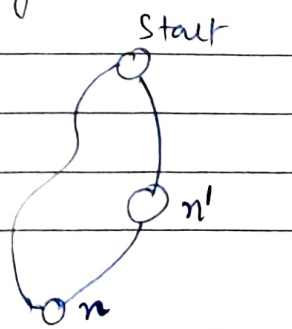


lets say we have 2 different paths from start node to Node n.

Now out of these two paths lets consider one to be a optimal path & other to be a sub-optimal path

optimal means → g-value along the path will be less than the g-value along the sub-optimal path

Now lets say we have selected node n through the non-optimal path.
BUT, this cannot happen. why?



That's because there must be a node n' in the optimal path which is in the frontier list.
lets calculate f-values.

$f1$ = f-value of node n along optimal path
$f2$ = f-value of node n along sub-optimal path.
$f3$ = f-value of node n'

$f3 <= f1$ { we have seen, Along any path f-value is non-decreasing }

We will see that
$f_1 < f_2$

$f(n) = g(n) + h(n)$
h(n) value doesn't depend on which path we followed to reach upto node n. It is the estimate to the goal node.

since h-value is same, which f-value (through optimal path or through non-optimal path) is smaller, will depend on g-value.

Since g value of optimal path is less than the g value of non-optimal path, the f-value calculated for node n along this optimal path $(f1)$ is less than the f-value calculated through sub-optimal path $(f2)$

So, $f_1 < f_2$

Now Based on, $f_3 <= f_1$ and $f_1 < f_2$
we can say -

$f_3 < f_2$

which means, the f-value of node n' is less than the f-value of node n calculated along the sub-optimal path.

Remember, Node n' is in the frontier list, & A* algo always selects node to expand from frontier (open) list, it will select the node which has the lowest f-value.

so as long as we have node n' in the frontier list, whose f-value is less than the f-value of node-n through suboptimal path → then definetly node n' will be selected instead of node n for the expansion

So, it can not happen that we have selected the node n through the sub-optimal path when we actually have a node n' in the frontier list.

So, we have proved -

i) If h(n) is consistent, then the values of f(n) along any path are non-decreasing.
ii) whenever A* selects a node n for expansion, the optimal path to that node has been found.

Based on above two claims, we can say

* The sequence of nodes expanded by A* using GRAPH-SEARCH is in non-decreasing order of f(n).

Hence, the first goal node selected for expansion must be the optimal solution because f is the true cost for goal nodes (which have h=0) and all later goal nodes will be at least as expensive.

If C^* is the cost of the optimal solution path, then we can say the following:

- A* expands all the nodes with $f(n) < C^*$.
- A* might then expand some of the nodes with $f(n) = C^*$ before selecting a goal node.

* A* is also complete as long as there are only finite number of nodes with cost less than or equal to C^* (considering all step costs exceed some finite ϵ and b is finite)

b → branching factor

* A* is optimally efficient as no other optimal algorithm is guaranteed to expand fewer nodes than A*. A* expands only those nodes with $f(n) <= C^*$

Any algorithm that does not expand all nodes with $f(n) < C^*$ runs the risk of missing the optimal solution.