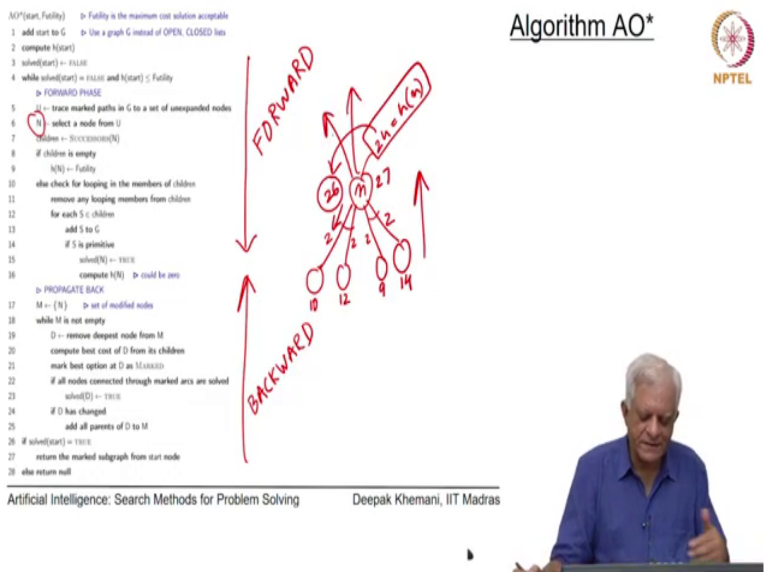
Artificial Intelligence: Search Methods for Problem Solving

Prof. Deepak Khemani
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Chapter – 06
A First Course in Artificial Intelligence
Lecture – 71
AO*: An Illustration

(Refer Slide Time: 00:14)



Algorithm AO*

```
AO*(start, Futility)        ▷ Futility is the maximum cost solution acceptable
add start to G              ▷ Use a graph G instead of OPEN, CLOSED lists
compute h(start)
solved(start) ← FALSE
while solved(start) = FALSE and h(start) ≤ Futility
    ▷ FORWARD PHASE
    U ← trace marked paths in G to a set of unexpanded nodes
    N ← select a node from U
    children ← SUCCESSORS(N)
    if children is empty
        h(N) ← Futility
    else check for looping in the members of children
        remove any looping members from children
        for each S ∈ children
            add S to G
            if S is primitive
                solved(N) ← TRUE
                compute h(N)    ▷ could be zero
    ▷ PROPAGATE BACK
    M ← {N}        ▷ set of modified nodes
    while M is not empty
        D ← remove deepest node from M
        compute best cost of D from its children
        mark best option at D as MARKED
        if all nodes connected through marked arcs are solved
            solved(D) ← TRUE
        if D has changed
            add all parents of D to M
if solved(start) = TRUE
    return the marked subgraph from start node
else return null
```

Artificial Intelligence: Search Methods for Problem Solving   Deepak Khemani, IIT Madras

Welcome back, we have been looking at this algorithm AO star which is used to solve an and or graph and as we discussed in the last session. The algorithm maintains a graph G and it works in two phases in the forward phase it traverses the graph and it traverses at using markers that it is maintained saying that this is the best choice at any given point of time.

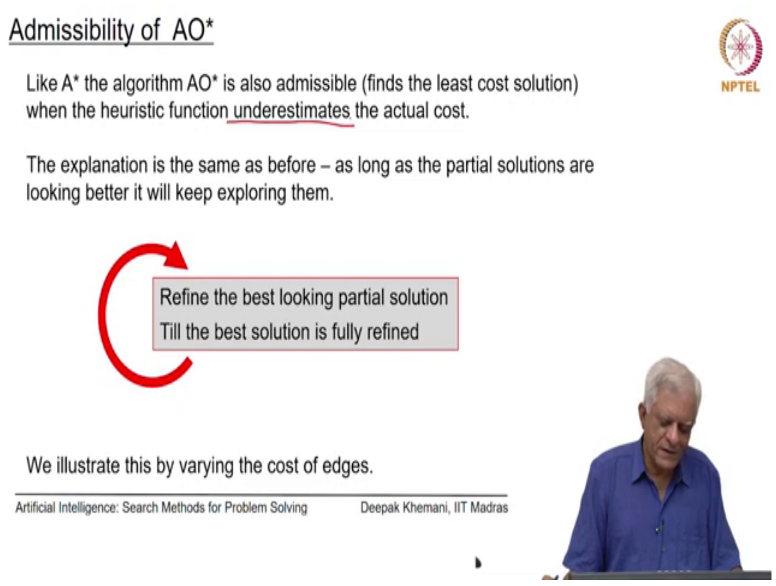
It takes it goes to a set of unsolved nodes U that we said and it picks one node N from U and refines that node. So, it is pick N and it has refined it and evaluated there heuristic values as the case may be so this is a forward phase. In the backward phase which is here I am just drawing an arrow upwards to reflect the fact that it is backward.

The algorithm of course, goes from top to down, in the backward phase it backs up the values from the children of N. So, first it will decide what are the cost associated with the children.

So, let us say if this cost is 10 and this cost is 12 and this cost is 9 and this cost is 14 and if the edge cost is let us say for sake of argument is all 2. So, then this is 10 plus 12 22 and 4 26 and this is 9 plus 14 23 plus 4 is 27. So, it says that this is the best choice essentially it starts by doing that.

So, let us say value of n was 24 h of n as given to us by some heuristic function. So, this 24 has now gone to 26 and therefore, it needs to propagate this up to all the ancestors and that is what the backward phase does essentially.

(Refer Slide Time: 02:47)



Admissibility of AO*

Like A* the algorithm AO* is also admissible (finds the least cost solution) when the heuristic function underestimates the actual cost.

The explanation is the same as before – as long as the partial solutions are looking better it will keep exploring them.

Refine the best looking partial solution
Till the best solution is fully refined

We illustrate this by varying the cost of edges.

Artificial Intelligence: Search Methods for Problem Solving   Deepak Khemani, IIT Madras

Now, let us talk about admissibility of this algorithm of the algorithm AO star, like the algorithm a star the algorithm AO star is also admissible that is it finds the least cost solution when heuristic function underestimates the actual cost.

Now if you think about this algorithm you will see that it is following the broad schema that we have been talking about all along which is to refine the best looking partial solution till the best solution is fully refined and the best solution is refined will be reflected by the fact that the root node would be labeled as solved essentially.

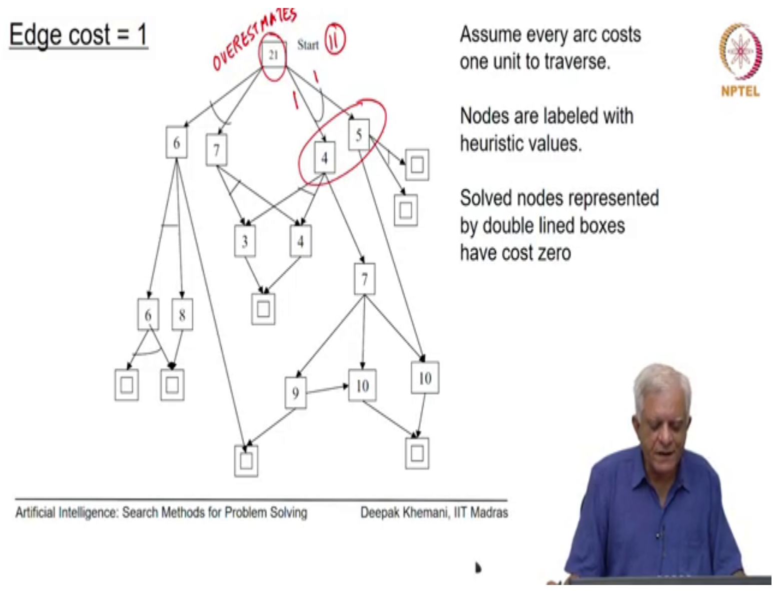
But now what we are saying is that this algorithm would be admissible when the heuristic function underestimates the actual cost essentially and the argument for that which we will not go into detail is of the explanation is same as before that as long as the partial solutions which are yet to be explored are looking better it will keep exploring them better than the fully refined solution even if you have found one essentially.

So, it will not terminate with the first solution it finds because some other solution may looks like it might be cheaper. So, it could explore that before actually terminating, we saw that in the case of branch and bound and also in the case of A star. We will do this by taking a problem another small example and modifying the problem in two ways by changing the edge cost associated with transformation.

In one case we will keep the edge cost very low which would mean that it is the heuristic function is actually overestimating the effort that is required. In another case we will keep edge cost as high which would mean that the heuristic function would be under estimating the actual effort required and we will see how the algorithm performs differently in both the cases, like we did in the case of for example, A star when we looked at W A star and so on.

We saw that the more emphasis to give to the heuristic function in the faster it converges, but it may not converge with an actual the optimal solution and that is the point you want to illustrate with AO star as well essentially.

(Refer Slide Time: 05:12)


Edge cost = 1
OVERESTIMATES
Start
11
21
6
7
4
5
3
4
7
6
8
9
10
10
1
1
Artificial Intelligence: Search Methods for Problem Solving
Deepak Khemani, IIT Madras
NPTEL

Assume every arc costs one unit to traverse.

Nodes are labeled with heuristic values.

Solved nodes represented by double lined boxes have cost zero

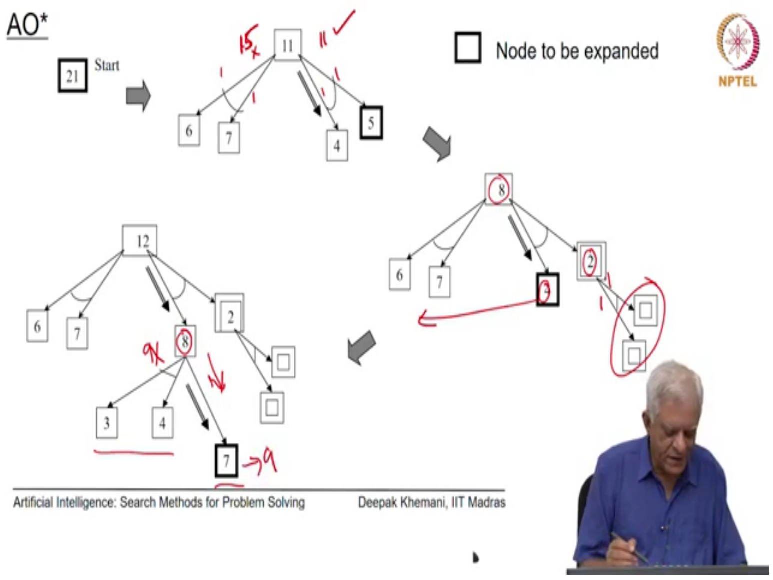
So, here is the problem that we are trying to solve, the values in the nodes are the heuristic values as given by some heuristic function and we are not mention the edge cost here because we have going to look at two edge cost, but in the first case we will assume the edge cost to be 1. So, let us look at its sample here. So, if you are looking at this branch where it is the heuristic values are 4 and 5 if you add 1 and 1 here then it becomes 4 plus 1 5 plus 5 plus 1 6 and so this would become 11.

You can see that the actual value of 21 is over estimates, but you can also realize that because of the fact that it is over estimating the cost when it looks at a more refined version and the more refined version appears to be cheaper it would kind of be committed towards that and it would rapidly go towards whichever solution it finds first in some sense and that is exactly like what best first search algorithm would have done without actually taking into account the actual cost (Refer Time: 06:41).

So, here in this example we will assume that the every arc cost is 1 nodes all the nodes internal nodes are labeled with heuristic values and solved nodes are represented by double line squares and we will assume for the sake of argument that the cost of solving the primitive node is 0.

So, the only cost is associated with the edge cost which is how much transformation you do before you reach a set of solved node essentially. So, let us work first with edge cost equal to 1 and remember that we are starting with this root node which is a value of 21.

(Refer Slide Time: 07:23)



AO*
Start
Node to be expanded
NPTEL
Artificial Intelligence: Search Methods for Problem Solving
Deepak Khemani, IIT Madras

And this is how the algorithm begins, we will show the node that is to be expanded next this is node n which is selected from the set of unexpended nodes u in bold square. And proceed to expand that in the first case of course, there is only the start node. So, you expand the start node you generate the 4 children as we saw in the previous example. The better option is on the right hand side and it is it results in a cost of 11 which is 4 plus 5 plus 1 plus 1 remember.

The other side would have been 7 plus 6 13 plus 2 15 the 15 is of course, not selected, 11 is selected and that is how what is reflected in the in the partial solution or the graph that we have expanded so far. So, we also have the marker which says that this is the better of the two edges. So, it algorithm will in the next cycle go down this marker and it will have to choose between one of the two nodes 4 and 5 let us say for the.

Let us say it chooses node 5 and if it does that then it would have expanded the node that follows. Now you can see that node 5 what was originally the value 5 is now been reduced 2 and that is because it has got 2 children and they are both solved and we said that solved nodes have cost 0. So, the only cost 2 that we have is a cost of these two edges 1 and 1.

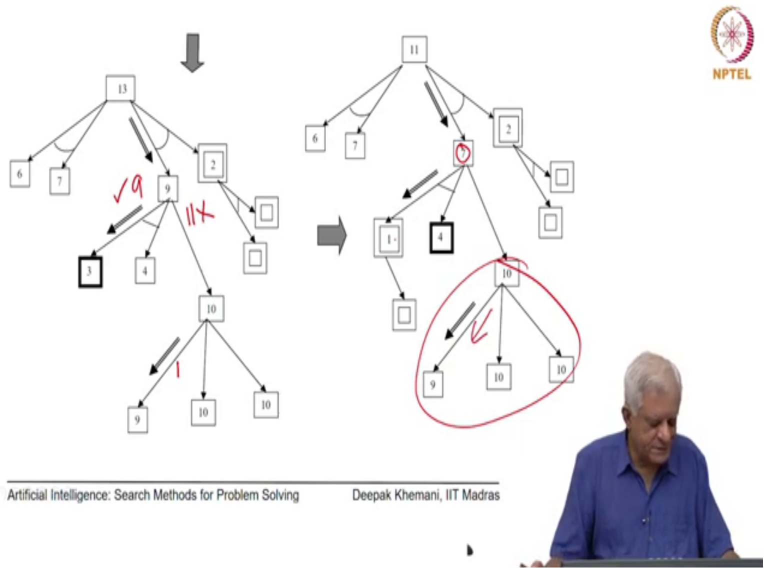
And the value has gone down from 5 to 2 and corresponding the value of root has gone down from 11 to 8 essentially it was 11 in the previous case and it is 8 now. So, you can see that the effect of overestimating is to as you refine one particular solution more and more it becomes cheaper and cheaper and you get more and more committed towards that essentially.

So, you can see that the one can think about why it does not find the optimal solution. I will leave this as a small exercise for you to find out what is the optimal cost for this problem that we have given optimal solution and does the algorithm find it.

So, here of course, there is only one more node left which is the node which has value 4 we refine that and in this particular example this value 4 has now gone to 8, it has actually gone up and that is because in this particular node the heuristic function is not under estimating is not over estimating, it is it actually underestimated the cost and therefore, we will see that it will end up doing a bit more work essentially.

So, this node can be solved in two ways one is with 2 children 3 and 4 the other is with one child 7. So, 7 plus 1 is 8 and that is what is reflected here. So, this is a best choice as marked here this would have been 3 plus 4 7 plus 2 9, but 9 would not be selected. So, 7 is a best choice to be marked here.

(Refer Slide Time: 10:35)



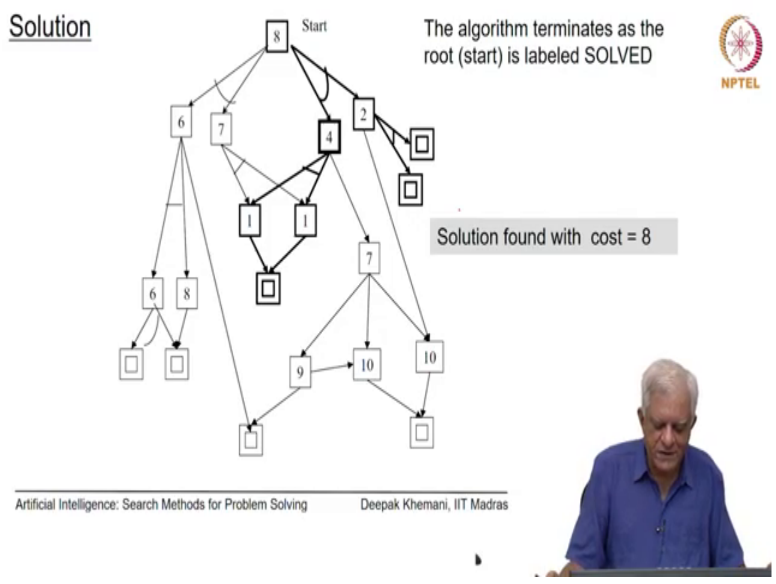
We had said that this node value 4 was underestimating we find that the revise value of 8 also was an underestimate it refined this node which had a value of 7 and that is gone to 9. So, I should draw an arrow like this.

So, this as you will see in the next slide it has gone to 9. And therefore, its attention would shift to the other side essentially or rather it is gone to 10 that is because its style was 9. So, 9 plus 1 10 and 10 plus 1 this side is 11 and that is side was 3 plus 4 7 plus 2 9. So, 11 has been deducted and 9 has been selected and attention shifts to the other side essentially.

Let us say we select this node which is the value 3 to be this thing and we find that it leads to a solved node in one step its values has gone down from 3 to 1 and therefore, this its parent has gone down from 9 to 7 and its own parent has gone down from 13 to 11 essentially.

And so the mark path is shown here observe that the markers are maintained in other parts of the graph as well, but we will never reach this part of the graph unless this marker work to somehow shift this thing we will see that can happen if we underestimate the cost at the moment we are over estimating the cost so this phenomena is not likely to be seen essentially. So, once it has gone down this path it will stick to this path it say it refines this node 4 next.

(Refer Slide Time: 12:21)



Solution

The algorithm terminates as the root (start) is labeled SOLVED

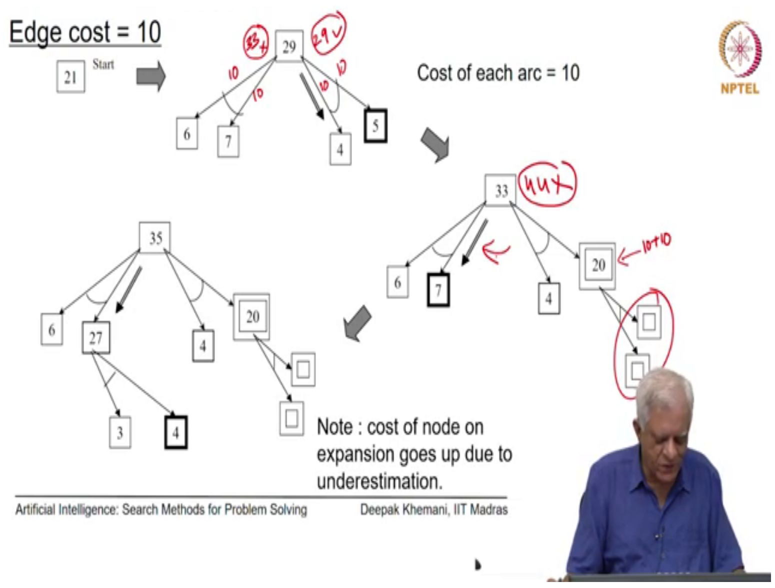
Solution found with cost = 8

Artificial Intelligence: Search Methods for Problem Solving Deepak Khemani, IIT Madras

And we find the solution now it has also been reduced to a cost of 1. So, you can see that the total cost is now become 8 and this 8 has come because there are 8 edges in the solution 1 2 3 4 5 6 7 8, the only cost is cost of transforming and this is a solution it is found.

So, you can see that except for the fact that is 4 that we see here was actually an underestimate it had kind of committed to whatever it is exploring first and dominated rather fast essentially. Let us now take the same problem and assume that the edge cost is 10 which means that the original values may are likely to be underestimates essentially.

(Refer Slide Time: 13:20)



Edge cost = 10

Cost of each arc = 10

Note : cost of node on expansion goes up due to underestimation.

Artificial Intelligence: Search Methods for Problem Solving Deepak Khemani, IIT Madras

So, we start as before with the root node which has the estimated costs of 21, but now we assume that the edge costs are 10. So, of course, the first nodes will be refined in the same way the same sub graph would be generated, but now the edge costs are 10. So, we have 10 here, 10 here, 10 here, 10 here. So, this is 4 plus 5 9 and 20 which is 29 this side. And this is 7 and 6 13 and 20 which is 33 this side. So, 33 is; obviously, rejected and 39 is selected.

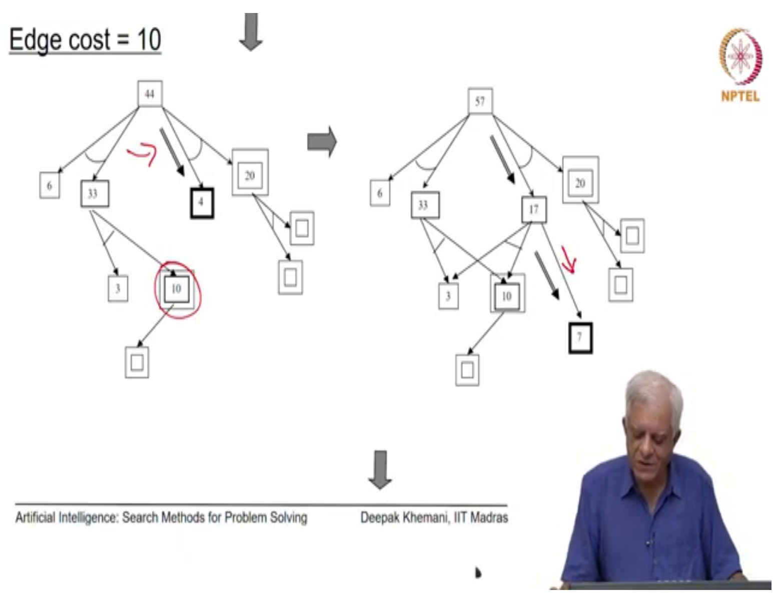
So, like before in this case because it is a symmetric situation each hyper arc had two edges. So, each of them contributed to 20 20 each to the transformation problem. And 4 plus 5 is cheaper than 6 plus 7. So, it also marks the same this thing, but notice that from 21 the root has gone up to 29 and that is the property that we saw in a star also is that as you refine the solution further or as you move closer to the goal the estimates which is the f values in that case monotonically increase.

So, here also if the heuristic function is under estimating you will find that the estimated costs will increase essentially and that is a property of an under estimating function. So, as before let us say it selects this node which had a value of 5 and again it found two nodes which were the solved nodes, but now in earlier this instead of this 20 it was 2, but now it is 20 because it is 10 plus 10 for the two edges. So, its cost has gone up and suddenly we find let this side has 20 plus 4 24 plus 20 is 44 and that is rejected and the marker has shifted to this side.

So, this is the phenomena that we will see more often when the heuristic function is under estimating because as you refine a solution it becomes more expensive or it start looking more expensive and some alternate solutions may start looking better.

And by doing that the algorithm is going to explore more of the search space and not leave out the optimal solution at any stage and which is what would end up in it finding the optimal solution because it does more exploration then when you are overestimating. So, you refine this node with a value 7 and you find that even that has gone up, but in this example it has only one choice. So, there are two nodes left 3 and 4.

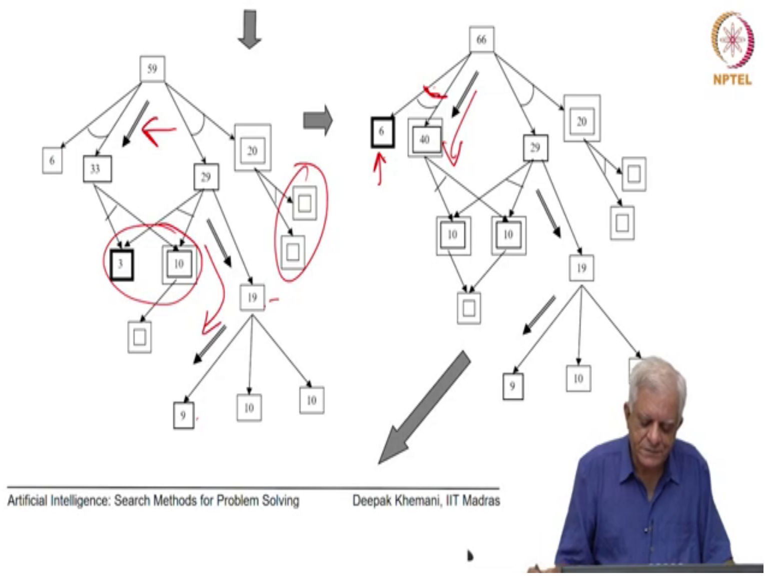
(Refer Slide Time: 16:27)


Edge cost = 10
44
6
33
4
20
3
10
57
6
33
17
20
3
10
7
NPTEL
Artificial Intelligence: Search Methods for Problem Solving
Deepak Khemani, IIT Madras

So, let us say it picks the node 4 and refines that. So, that was a node that it refined last, but because its value has gone up and from 4 to 10 the overall solution from that time has gone up and the pendulum is again shifted towards this side essentially.

So, it refines this thing which is also was labeled as 4, but that has now two choices now when is when we looked at this solution last time the left choice was better, but now the right choice seems better because this one because it has only one edge and remember the edge cost are the ones which are contributing more.

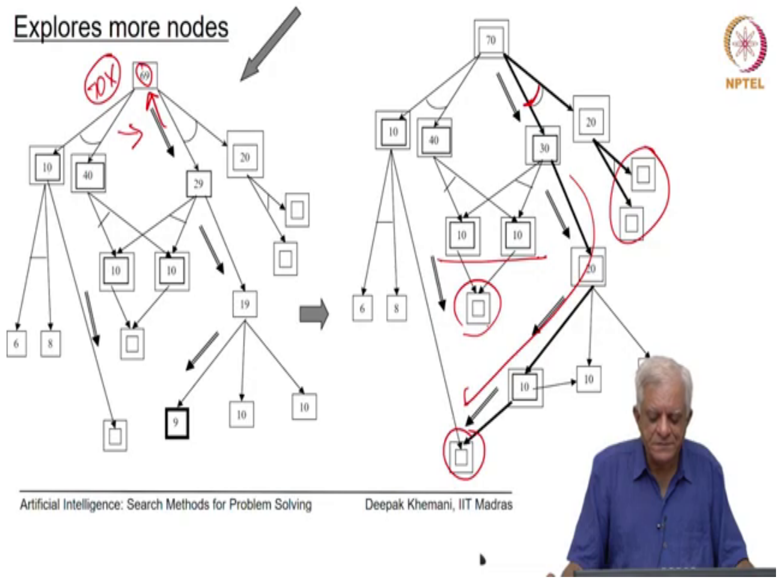
(Refer Slide Time: 17:10)


59
6
33
29
20
3
10
19
9
10
10
66
6
40
29
20
10
10
19
9
10
NPTEL
Artificial Intelligence: Search Methods for Problem Solving
Deepak Khemani, IIT Madras

So, it goes ahead and refines that and this remains the preferred choice as opposed to this when we solved the problem last time the solution from this side was composed from here. And the other solution came from here and that was the solution, but now the preferred path is along this single edge and you can see that because this is 9 this is 19 and this is 29 which is cheaper than going on the other side essentially.

So, and so on it explores more and more it has made 29 but now the pendulum has shifted as you can see here. So, it is now going in this direction and it is refined this it has solved one side of this and edge here and now its shifts its attention to the other side which has the heuristic estimate of 6 here.

(Refer Slide Time: 18:13)


Explores more nodes

Artificial Intelligence: Search Methods for Problem Solving

Deepak Khemani, IIT Madras

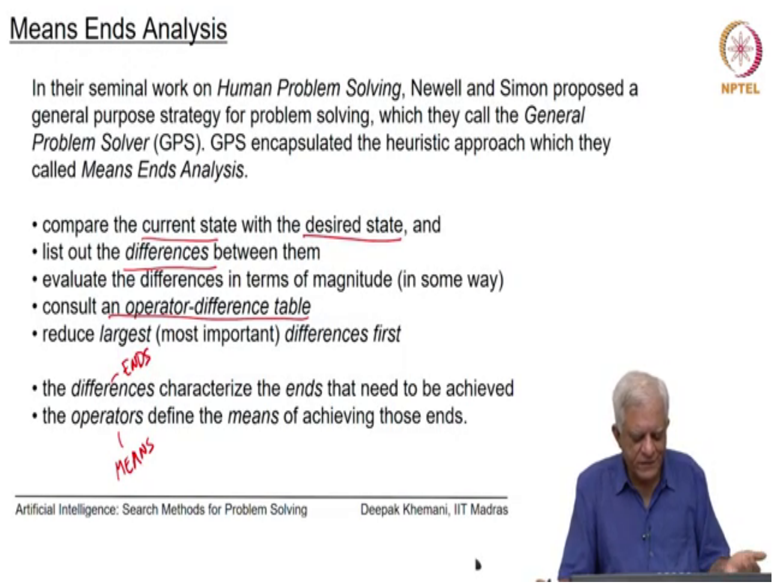
So, when it refines that 6 it turns out that that 6 has gone up to 10, but still that is and again the pendulum is shifted to this side here essentially. So, you must keep track I am not worked out the exact values that are changing, but you must do this on pen and paper yourself and see that indeed the pendulum is shifted, this value 69 that we see is actually coming from here and we are not written the other side what the value was.

We can imagine that this is 0 plus 10 so, this is 10 so and this is 40. So, that is 50 plus 2 would be 70 this side and that would be rejected. So, 69 is coming from this side. So, you go and refine this node 9 and we will see that this node 9 now has gone up to 10 and it is gone to value 70 and at this point the root gets labeled as 70 and it is found the solution which is a different set of nodes. So, to solve the root node here we have this and arc these two forms still part of the solution, but the other side is formed by one single node here.

So, there are three nodes in the final solution and which is different from the solution which was found earlier, when it was over estimating these two nodes were part of the solution or rather this node was part of the solution, but it is a different solution.

What is important here is to realize that the number of expansions that (Refer Time: 19:54) are did when it was under estimating was more. So, in some sense it is made sure that it is not missing out on the optimal solution before finally, terminating with a solved root node. So, this gives as a general picture of the algorithm AO star it is quite a neat and simple algorithm.

(Refer Slide Time: 20:13)



Means Ends Analysis

In their seminal work on *Human Problem Solving*, Newell and Simon proposed a general purpose strategy for problem solving, which they call the *General Problem Solver* (GPS). GPS encapsulated the heuristic approach which they called *Means Ends Analysis*.

- compare the current state with the desired state, and
- list out the *differences* between them
- evaluate the differences in terms of magnitude (in some way)
- consult an *operator-difference table*
- reduce *largest* (most important) *differences first*

- the *differences* characterize the *ends* that need to be achieved
- the *operators* define the *means* of achieving those ends.

ENDS

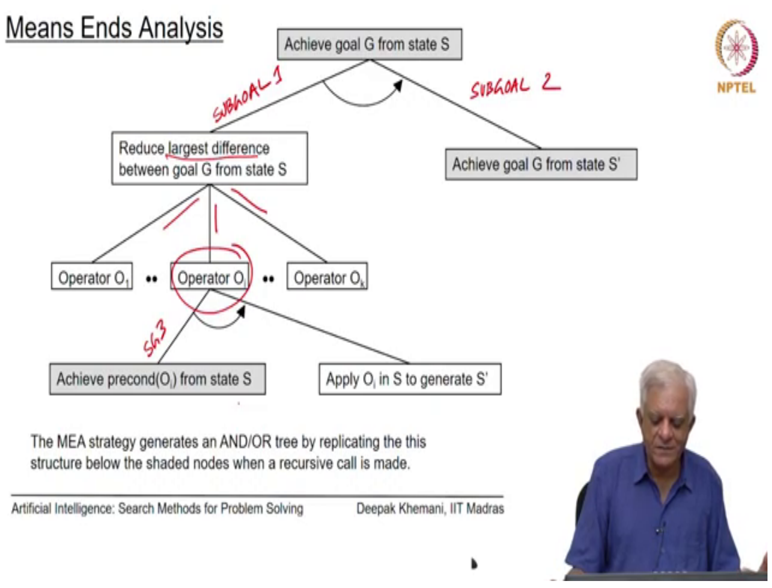
MEANS

Artificial Intelligence: Search Methods for Problem Solving   Deepak Khemani, IIT Madras

But you might recall that we had looked at Means Ends Analysis which was a work of Newell and Simon as part of their general purpose strategy. And you can think of what was happening there as also similar to what is happening in the AO star, in the sense that it is goal based reasoning essentially. So, you are comparing the current state to the desired state and you are looking at the differences between them and then you are trying to reduce the largest difference.

So, you can see it has a backward flavor to this whole thing and we are looking at this operator references tables and to solve it and the means and the operators are the means and the differences are the ends and that is why its strategy was called means ends analysis that you analyze this and form a solution.

(Refer Slide Time: 21:09)



Means Ends Analysis

Achieve goal G from state S
SUBGOAL 1
SUBGOAL 2
Reduce largest difference between goal G from state S
Achieve goal G from state S'
Operator O_1
Operator O_i
Operator O_k
SG3
Achieve precond(O_i) from state S
Apply O_i in S to generate S'
NPTEL

The MEA strategy generates an AND/OR tree by replicating the this structure below the shaded nodes when a recursive call is made.

Artificial Intelligence: Search Methods for Problem Solving — Deepak Khemani, IIT Madras

And if you might recall that we had also looked at this diagram which says that if you want to achieve a goal G from a state S then you solve reduce a largest difference which is the sub goal that you are solving. So, you solve that SUBGOAL 1 and then you move to SUBGOAL 2. So, let us call this sub goal 1 and 2 and it has a same flavor of problem decomposition that you address one part of the problem solve it and then come back to the remaining parts of the problem.

In this case you can see that they were these all choices here and you selected this and this also was broken up into its own sub problems sub goal. So, let us call it sub goal 3 and then move on and so forth. So, the means ends analysis also had a goal directed backward approach to problem solving and that is quite similar to what we saw in the and our star algorithm.

In the next few classes we will look at approaches to problem solving where again you decompose the problem into sub problems and solve them in a piecewise fashion and in doing so we will do two things. We will look at how rule based systems try to solve you know parts of problems eventually working out the whole.

We will also look at theorem proving in logic and reasoning and see that one of the approaches in logic was also these goal directed backward search kind of an algorithm, but we will do that in the next few classes ok. So, we will see you then.