Advanced Business Decision Support Systems

Professor Deepu Philip

Department of Industrial Engineering and Management Engineering

Indian Institute of Technology, Kanpur

Professor Amandeep Singh

Imagineering Laboratory

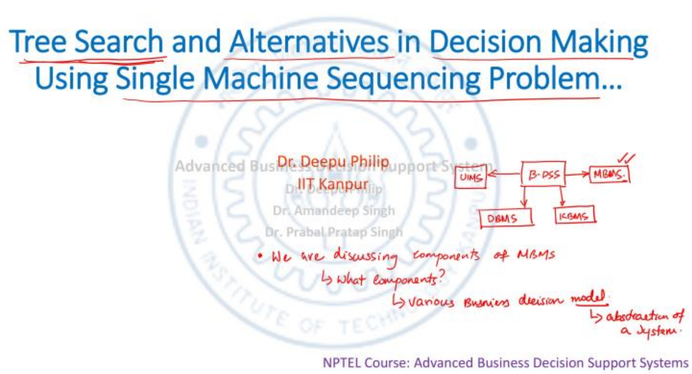
Dr. Prabal Pratap Singh

Indian Institute of Technology, Kanpur

Lecture 12

Tree Search and Alternatives in Decision Making Using Single Machine Sequencing Problem (Part 4 of 4)

Good afternoon everyone, welcome to yet another lecture of the Advanced Business Decision Support System course under the NPTEL MOOC's program of IIT Kanpur. I am Dr. Deepu Philip from IIT Kanpur and along with me Dr. Amandeep Singh Oberoi and Dr. Prabal Pratap Singh is also co-teaching this course and we are now in the week 3 and we are going through certain decision models or business decision models. So, without any further delay, let us look into the slides.



So, we have been looking into the tree search problem and also we are explaining the Tree Search problem using the Single Machine Sequencing problem. You have already seen that how do we make the decision? What is the decision making process in a Single Machine Sequencing problem using Tree Search and the second part we are trying to do is what are the alternatives associated with this. So, remember, I told you earlier in the course that this is the business DSS , it has a UIMS (User Interface Management System) you have DBMS (Database Management System) you have a KBMS (Knowledge Base Management System) and then the last one is we discussed was MBMS (Model Based Management System). So, what we are looking into is the components of the Model Based Management System.

So, we are discussing components of MBMS and what are the components? What components answer to that is various Business Decision Models or we already seen what is a Model? Model is an abstraction of a system. So, we use Models to make business decisions and we are looking into various type of Business Decision Models and one of them we have discussing today is or we have discussed in the previous class using the Tree Search to find an optimal solution. So, continuing further we have already seen what is total enumeration, we have worked on that problem of 4 jobs and the each job has a processing time, each job has a due date and it also has its own penalty lateness, we have seen that. So, the question today is, we enumerated before we discuss what we enumerated, let us take a quick brief. Look on that problem, I will just do this right here.

So, that we can remember, there were 4 jobs. Each job has a processing time Pi and the processing times were 37 your time units, 27 time units, 1 time unit and 28 time unit respectively for job 1, 2, 3 and 4 and the each job has a due date Di the first job due date was 49, second job due date was 36, third job due date was 1 and the fourth job due date was 37 and there was a penalty. If the job is late, lateness penalty and for job if it is late, the penalty is 1, second job late penalty is 5, third job late penalty is 1 and the fourth one, the penalty was 5. So this was the problem setup. So, if you look into how the system actually works there is a single machine and there are 4 sequence positions on this and you decide which job will go into the which sequence position.

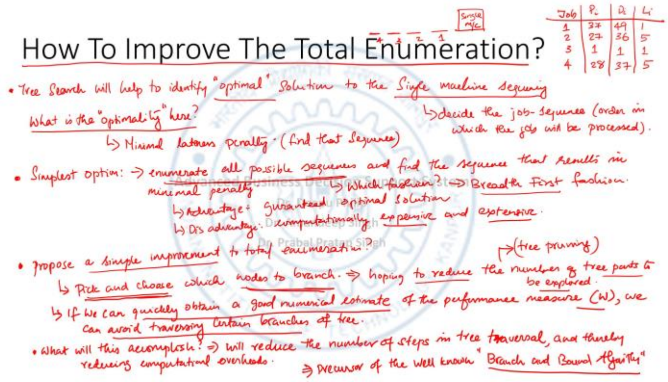
And, we said that this kind of a problem setting, where you have to decide which job will go where is the easiest way for us to do, not easiest actually, one of the most Tree Search will help to identify optimal solution to the Single Machine Sequencing Problem. So, you have to decide on what is the production sequence? The Single Machine Sequencing Problem is decide the job sequence. Job sequence means order in which the job will be processed. So, you have to decide this. You have to find out the Single Machine Sequence as part of it and the Tree Search will help us to identify the optimal solution.

Optimal solution means, in this case, what is the optimality here? What are the optimality? We are talking about here. The optimality here is minimal lateness penalty. So, find the sequence that minimizes the lateness penalty. Find that sequence. That is what our job was.

So, what did we do to get this done? So, the one option is that the simplest option is enumerate all possible sequences and find the sequence that results in minimal penalty.

So, this is what we enumerate all possible sequences. So, the advantage is, that the solution will be advantage guaranteed optimal solution. Disadvantage computationally expensive and extensive. So, you have lot of computational expense and it will take a lot of time to actually solve this. So, this is the one that we have done.

So, now the question is how to improve and what we did is, we enumerate all possible



sequences in which fashion? We did Breadth First Fashion. We elaborated Breadth First Fashion is elaborate all nodes in a tree in a particular depth before moving to the next depth. So, we iterate through, we visited all nodes once for a particular depth before moving to the next depth. So, the question is how can we improve this option? Suppose a simple improvement to total enumeration.

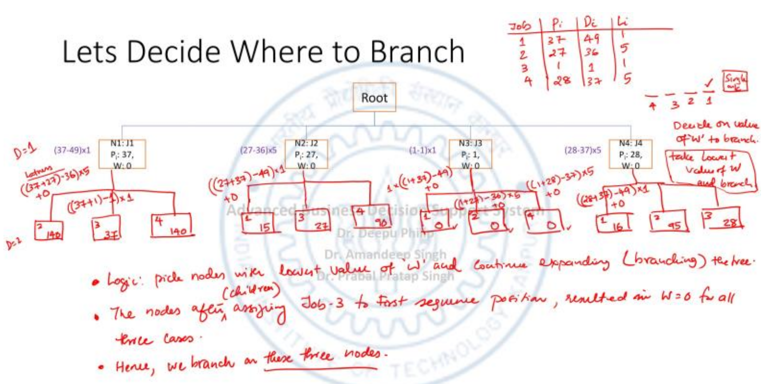
How do you propose a simple improvement? I am not asking about very sophisticated or something a simple improvement, what is one option to do this? And, the one way to do this is, pick and choose instead of going through every one pick and choose which nodes to branch We pick and choose where we decide, we choose instead of going through every nodes.

The intention of this is hoping to reduce the number of tree parts to be explored or hope is that by in doing this we may be able to get to the solution that we want much faster. So, the idea is, that if we can quickly obtain a good numerical estimate of the performance measure. In this case, the performance measure was W, the lateness penalty. We can avoid traversing certain branches of tree.

So, the idea of this picking and choose or why we pick and choose instead of going through every branches would be to quickly obtain a good numerical estimate. Good numerical estimate of W, so that if we know that okay, fine here is a good numerical estimate and at some any point of time, you find that branch of a tree or a particular depth is giving a solution worse than this, then there is no point in continuing that part. So, the idea here is, you can avoid traversing certain branches, certain specific portions of the tree, we can avoid this and what will this do? What will this accomplish? By avoiding certain traversing the branches of tree what are we going to accomplish. It will reduce the number of steps in Tree Traversal. It will reduce the number of steps in Tree Traversal and thereby reducing use of computational overheads.

So, you know this is like we can say this is the precursor of the well known branch and bound approach branch and bound algorithm. So, there is a very popular tree source algorithm. It is called branch and bound. We are trying to get to branch and bound, but I am more interested in explaining to you the details or the specificalities of branch and bound how branch and bound would actually result in or how actually branch and bound algorithm evolved or how the decision making process actually evolved. So, without much further delay, our main aim in this regard is to reduce or avoid the enumerating all possible sequences.

For doing that, what we do is, we will pick and choose which nodes to branch in the tree. We will decide which nodes to branch and from there, by doing that, we are hoping that we will reduce the number of tree parts to be explored. And, this approach, reducing the number of tree parts to be exposed the classical term for this is Tree Pruning. Pruning is like cutting certain branches out. So, that is the idea here, we are pruning certain branches of the tree.



So, without further delay, let us look into the problem statement. The earlier problem that we did and I have written expanded the first 4 nodes just for your easy reference. But looking into the problem, there are 4 jobs, job number 1, 2, 3, 4. Each one of them has its own processing times and that is 37, 27, 1 and 28. Each one of them have a due date this is the processing the due date my Pi and Di looks exactly the same 49, 36, 1 and 37 and there is the lateness penalty Li 1, 5, 1 and 5.

So, we were saying that we were enumerating. So, the root node is the first node and from there, the problem statement is, as I said earlier, Single Machine you have 4 points

1, 2, 3, 4, the 4 sequence position. So, now we are talking about the first sequence position. That is your depth equal to 1. So, we expand the first depth and we assign job 1, the first sequence position job 2, job 3, job 4 and in this case, all of the value of W is equal to 0.

So, we decide on value of W to branch. So, the idea is to find take lowest value of W and branch. So in this case, all the values of W are 0, 0, 0. So, we have no other option but to branch on all of them. So, without further delay, we will start branching on them.

So, the first one, now we will branch to the next level, which is D equal to 2, the second depth. So, we know that each one of them, there will be 3 branches below because once you fix here the sequence position, then the depth 2 is the second sequence position where you only have 3 options. So, if you fix job 1 here, then the next option in front of us is job 2, job 3, job 4. That is the option. The first 3, this depth 2 the next sequence position.

Bunch below this one you will have this will be job 1 because 2 is already assigned, so you cannot assign 2 anymore job 3 or job 4. These are the 3 options in front of us then, if you take the third option where job 3 is assigned to the first sequence position. Then we have job 1, job 2, job 4. 3 is already assigned and the last one is job 4 is assigned to the first sequence position job 1, job 2, job 3. So, now for each one of them, we will now calculate what is the lateness W value? We already done in the previous class but just to remind you, I will explain this to you how we calculate.

So, let us look at this one so job 1 processing time is 37, job 2 processing time is 27. So, the total time will the job 2 will completed will be 37 plus 27. That will be the time and from there, the due date of that will get completed by the time period job 2 in the second sequence position. The due date of job 2 is 36. So, we subtract 36 out of this and that will be the total lateness of this. This is the estimate of lateness.

Then, what we do is we multiply that lateness with the penalty for since, it is job 2, the penalty is 5. So, that is 5 multiplied by 5. So, that is the lateness penalty associated by assigning job 2 to the second sequence position. Now, with that, the previous penalty is 0.

So, we add the previous penalty plus 0. All of that put together, we get a value of 140 and same way, for job 3, if you look into this, 37 will be the processing time of the first job which is in sequence position 1. Then, to this, we add the job 3, which is the processing time is 1, sequence position minus the due date of job 3 is 1, so that is minus 1 multiplied by the penalty. Penalty for this is also 1, so that will be this 37 plus the previous one is 0.

So, you get a penalty of 37. Using the same approach, we get the job 4 penalty. If you calculate, it as we calculated as 140 yesterday. Similarly, if you look at the next sequence position, job 2 is assigned to the first sequence position which will take 27 time units to complete. If you take this, the first sequence position will be 27 plus the time required to process job 1 is 37. 27 plus 37 then minus, that will be the time when job 1 will get completed the due date of job 1 is 49.

So, it is 49, that will be the lateness you multiply that with the Li value which is 1, that will be the lateness plus the previous delay which is 0. If you add all of that, you get a value of 15 here, that will be the value of lateness 15. Same way, use the same approach job 3, we would actually get the value of 27 and the 4, we will get a value of 90. I have explained this entire calculation in the previous stuff, so if you have confusion please refer that to see how we got these numbers.

I will only do one node of each one of these in the interest of time and if we look at job 3 where you have already assigned job 3 to first sequence position, then the time with which the job 1 will complete will be 1 plus the processing time of it will be 37. 1 plus 37 minus the due date is 49 and multiply that by 1 will be the lateness for that particular thing and plus at the previous lateness 0. So, 1 plus 37 is 38 minus 49 is negative, so that means, the job is not late. So, this whole place the lateness will be 0. Same way, the job 2, if you look into this, 2 is calculated as job 1 is already assigned 1 plus 27, that will be the processing time. When the job 2 is over minus the due date is 36, it is negative, the weightage is 5 plus 0.

That value you will get is 0 and the same way, if you look at the job 4, it is 1 plus the processing time of job 4 is 28 minus 37 multiplied by 5. That whole lateness value is 0. Now, it is negative actually that means the job is not late and you are the previous 0. So, all these 3 nodes will have the W value as 0 and the last one I will just do the first of the last one I just showed this purely because the third branch, the assigning job 3 to first sequence position the second one also is 0. So, there is no lateness so it gives much more promising option in front of us.

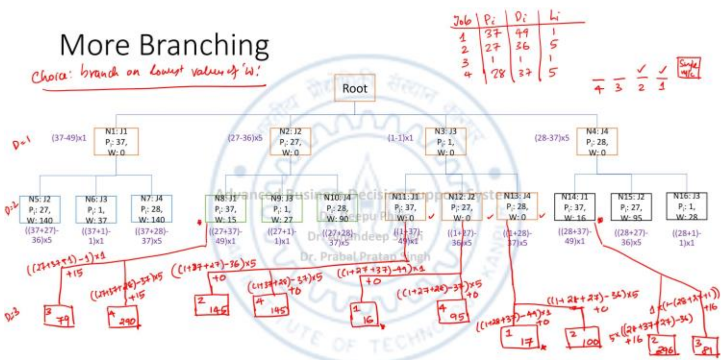
The processing time of job 4 is 28. So, They assign 1 to this one. That will be 28 plus 37 minus 49 that is the delay multiplied by 1 and plus 0. So, if you do all that math, you will actually get the value to be as 16 and same way for job 2 you get 95 and then you get this job is 28. Now, you have different values of W. All of these are W values 140, 37, 140, 15, 27, 90.

Now, our job is, our decision earlier case was to take the lowest value of W and branch on this, so we have 3 lowest values of W is 0 in 3 cases. So, what we have to do is, we have to branch on all these 3 cases and what we do is, we will elaborate on this further.

So, I will move to the next slide. So, that it makes easy for you. Let us for that, we can branch further down below this.

So, just to remember again logic, pick nodes with lowest value of W and continue expanding that is called branching. You continue to expand the tree or continue branching the tree. So, you expand on those ones. So, the nodes after assigning job 3 to first sequence position resulted in the nodes after you can called as child or children resulted in W equal to 0 for all 3 cases. Hence, we branch on these 3 nodes. So, we branch on these 3 nodes at this point to make our life easy.

I have elaborated all these step whatever we have done, so that we can branch nicer below this just for our quick reference, I am going to summarize the problem right here again once more. So, the values are available right here jobs Pi, Di and Li 1, 2, 3, 4 is the job processing times are 37, 27, 1 and 28, the due dates are 49, 36, 1 and 37, the lateness is 15, the lateness penalty or weight penalty is 1, 5, 1, 5 and now we branch below from here and the current option in front of us is, you have a Single Machine and you have



sequence position 1, 2, 3 and 4. So, this current branch shows us 1 is assigned and 2 is assigned.So, the depth equal to 1 is completed here, depth equal to 2 is completed here and our option, the decision the choice was branch on lowest values of W.

This was our choice. So, the lowest values of W were these 3, as we explained earlier. So, now the option in front of us is to branch on them. So, we branch to what we call as depth equal to 3, the next depth, we branch instead of elaborating all nodes like the Breadth First, we branch on the promising nodes. So, first we will branch on this.

So, I am just going to do this as best as I can . I hope I can do a better job, it is not very easy, it looks easy but it is not easy So, we have 2 options here from this branch. So, this branch says, job 3 is assigned to first sequence position, job 1 is assigned to second sequence position, so then the option in front of us is job 2 or job 4 is what we can assign at this point. Then, what we have next is the second one, where we can branch 2 more. So, I am going to do fashion just to save space, that is all the main name of this is saving space. So, you have already assigned to job 3 to first sequence position, job 2 is second sequence position, so the option in front of you now to assign is 1 or 4.

That is what you can assign. Then, the third one is what we are going to do now is somewhere here. Let me draw really low so then, I can draw other things also. Some of I do not want that for. So, you are assigned to job 3 and job 4, the option in front of you is job 1 job 2.

So, we elaborated all the 3 nodes of this. So, now let us calculate what is the W value for each one of them. So, the W value for the hold on, let me do a little bit more beautification, there you go So, for job 2, on the third sequence position, depth equal to 3. You will have 1 plus so the processing time of the job 3 is 1 the other one is 37. So, the job 2 processing time is 27.So, it will be 1 plus 37 plus 27 minus the due date of job 2 is 36, penalty is 5 multiplied by 5 plus the previous lateness of 0.

So, that will give us a value of something close to 145. That is for the second job will be 44. So, it will be 1 plus 37 plus the job 4 is 28 minus the due date is 37, there is a total lateness multiplied by 5 plus the previous lateness of 0. That should also give you a value of 145. So, the 2 jobs will have by the time the third depth, we will have 145 as the value.

Now, the for the second one, it is job 1 is 1, second one is job 2 is 27. So, for job 1, if we assign, so then that will be 1 plus 27 plus job 1 is 37 minus 49, that is the lateness that is multiplied by 1 plus the previous lateness is 0 and you get a value of 16 for that and the job 4, it will be same way, it will be 1 plus 27 plus job 4 is 28, the due date is 37 that is multiplied by the lateness penalty of 5 plus the previous one of 0 and that you get it as 95. Then, the last one is 1 plus 28, that is job 4 is assigned to the second sequence position. So, it will be 1 plus 28 plus job 1 processing time is 37 minus the due date is 49 multiplied by the lateness penalty is 1 plus 0, the previous penalty all of that added to that, you get a value of 17. And then, the last one, same way, what we can do is 1 plus 28 plus job 2 is 27 minus due date is 36 multiplied by the penalty is 5 plus 0 and you get the value there as 100.

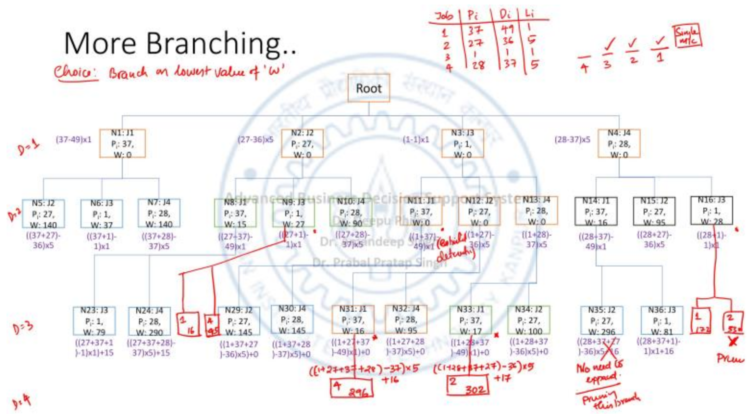
So, these are the values of different lateness. So, now if you look into in this scenario, at this particular point, take the lowest value of W and branch on it. So, now which are the lowest values of W. The next lowest values of W is 15 and then you have two 16s associated with this, one 16 here and one 16 here. So, now we have these 3 as the alternatives. So, now let us just do this is, let us also expand those nodes as well and that is how this process continues because it is a computer program, it blindly follows the algorithm that is being told to it.

So, we do it like this, we expand this and we expand this also all depth equal to 3. So, for this case, you already assigned job 2 and job 1. So, the option in front of us is assigned job 3 or job 4 in this case, you are assigned job 4 and job 1. So, the option in front of us is assigned job 2 and job 3. So, for this case, to calculate then you follow the same procedure, 27 plus 37 that is, 27 plus 37 plus job 3 processing time is 1 minus the due date is 1 multiplied by the penalty is 1 plus the previous one is 15 instead of 0.

If you add all of that, you get a value of 79. Then, the second one, if you look into it is again, 27 plus 37 plus job 4, the processing time is 28 minus the due date of job 4 is 37 and the penalty is 5 multiplied by 5 plus the previous is 15, all of that are together, you get a value of 290. Now, similarly, this side you can calculate the first one as job 4, job 1 is assigned. So, it is 28 plus 37 plus, now job 2 is being assigned that is 27 minus the due date is 36, the lateness is multiplied by 5 plus the previous penalty of 16 gets added and that you get a value of 296. And, this case, the last case again will be job 3 is assigned. So, that is 28 plus 27 plus 1 minus 1 is the deadline multiplied by the lateness is 1 plus 16 gets added to it, that value gives you something called as 81.

So, now you can see that after expanding these promising one, there is nothing with 0s and there is only one that is left out which is with the value of 16. Everything else is now whatever the promising nodes, we have done so far expanded to the level of depth equal to 3. So, the third depth point we have reached ideally speaking to set up promising nodes. Now, we have one 16 left out and then there is one 17 which looks like another promising one.

So, if you want, we can branch on them at this point. So, what I am going to do is to take it to the next step. So, I will clean this up, I mean, I will use the next slide where these things are cleaned up and provided to you so that you can see the numbers and then proceed further.



So, without further delay, we move to the next branching part and just to remind you again so that because we need these numbers to do the problem. So, it is 1, 2, 3, 4 is the job numbers the processing times Pi's are 37, 27, 1 and 28 and the due dates Di are 49, 36, 1 and 37 and the lateness Li is 1, 5 and 1, 5.

So, we have seen this is the and you have your Single Machine. Your sequence position 1, 2, 3 and 4 and our choice is branch on lowest value of W. That is what our choice is and depth equal to 1 is the first one, we have already done that depth equal to 2 is the second one, we have already elaborated that part and from there, we picked the lowest value of this and we went to depth equal to 3 which is this portion and from there, we have seen that we have 2 lower choices I mean, there are other choices also. But the 2 lowest choices are 16 and then there is a 17 available for us at this point and then there is 28 here and there is some 27 other side but for the time being, these are the 2 lowest values of W for us. So, let us expand to this, if we expand that, it will move to depth equal to 4, the fourth level.

So, we expand this to the next one. So, in this case the only option in front of us is you already assigned job 3, job 2, job 1 the, only option left out of in is job 4. If it is job 4, then what we end up doing is, how it will be calculated? It will be job 3 is 1, so it is 1 plus the next one is 27 then, it is 37, that is job 1 plus the job 4 will be completed and job 4 is 28 that is, 93. All of this put together minus the due date of job 4 is 37 that is the lateness of this multiplied with the penalty of 5 and you add the previous penalty of 16, all of this put together you get what you call as 296. So, at this point, you can see this number, you can easily say that, I do not want to expand this node. So, you can say, no need to expand because it is not going to give you any solution better than this.

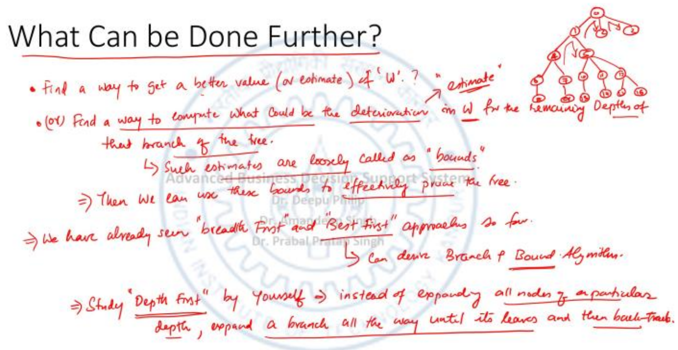
So, going below, there is no need to go below, this so you are pruning this branch, you pruned that particular branch and you do not allow it to proceed any further. So, this 296 is a number that you can use to decide if you see any values greater than or equal to this, you can stop expanding that particular portion of the tree. Now, let us take this particular branch again 17 to see whether we actually get a better number or something like that. So, here you have already assigned job 3, job 4, job 1 and the only option left out for in front of is job 2. So, if you calculate the time for the lateness of that it will be 1 plus 28 plus 37 plus the job 2 will be 27, all of this is equivalent to 93 minus the due date of that it is 36 multiplied by 5 plus the previous lateness is 17, all of this put together which is 302.

So, this value of W is worse than 296. So, there is nothing you can do beyond this. So, you get to a point where you are like okay, fine I have expanded 2 possible options and this is the number I actually got. Now, you have other options in front of you, just to you know now you need to continue. So, then the option in front of you now is, you have another 27 right here and you have a 28 right here. Now, we said, pick the lowest values of W and branch on this is what our argument is.

So, if that is the case, so this is in level D equal to 2, so you need to now branch on this 2, this is you branch it into D equal to 3. So, here you already assigned job 2, job 3. So, the only option in front of you is, you have you can assign 1 or 4 and then, you calculate the W for that 2 particular cases and same way for 28, we take this alright and here you have job 4 and job 3 assigned. So, you have job 1 and job 2 possible, both of them and then using these values same approach, you calculate the W for this and then, you see among all these W's which W will be the lowest. So, sometimes you may get 37 to be the next lowest one, so you branch on that and something like that and at some point of time, you will get a value much better than 296 which prevent you from branching on many other nodes.

So, you will reduce the number of searches in that regard and just for the completeness of this exercise, just to clarify what the values are. Let me just quickly go back to the previous computation of what those values were. So, if we look into this the 27 node, we would actually get this 1 and 4, so you will get values of W equal to 16 and W equal to 95. So, 95 you will get in this case then, same way, if you look at 1 and 2, that 1 will be 172 and 2 will be 330. The minute you get a value 330, you can say, I do not need to continue any further because you get 296, a better one there, so you can stop there that way right.

So, you would prune this branch also and this exercise will continue and you will get into better value of finally, you will not be expanding that much number of leaves of the tree.



So, the question is, before concluding, what are the other options as part of this? So, the simplest way to think about it is or argue on this is find a way to get a better value or estimate of W. Find a way to do a better estimate of W, how can we do this and such kind of things when you are having, or find a way to compute what could be the deterioration in W for the remaining depths of that branch of the tree. Find a best way to compute what could be the deterioration and estimate and estimate that is what we are interested in can you estimate this.

Such type of estimates are loosely called as Bounds. So, if you can find a bound, you can find a estimate of a deterioration of W for the remaining depth of the tree, you say I reached a particular depth. So, if you can find something here okay, I reached this depth, estimated deterioration, what is that? If I can do that then, I say this deterioration estimated is really bad then, there is no point in me going that side. So, then we can use these bounds to effectively search the tree or effectively prune the tree, that is ideal. You can effectively prune if you have a good bound then, you can effectively prune the tree. So, this is one approach, the previous approach, we have already seen Breadth First and "Best First" approaches so far.

So, from Best First, you can derive branch and bound algorithm because same way, as we choose the best value of W to then, instead of that best value of W you find the best bound, which is the deterioration of the one below and then, use that to come up with the approach. But the Best First is something that is used very heavily in lot of the searches and decision making so far. So, it would help you guys to do that and the depth first. So, study depth first by yourself instead of expanding all nodes of a particular depth and a branch all the way until its leaves and then backtrack. So, that is the depth first approach instead of expanding all nodes of a particular depth expand a branch all the way until its leaves and then backtrack.

So, you may be able to get a quick estimate of a bound also, in depth first that is also may be possible. So, these are the possibilities that are in front of us at this point. So, just to elaborate on this, we have already seen the way the Best First search is and now we say that, instead of using W find a way to estimate the deterioration the estimated deterioration of W for the remaining of the branch depth of the tree and that kind such kind of estimates, we loosely call as the bounds and using bounds you can probably effectively prune the tree. So, with this we will come to the conclusion of the lecture and this would help us in understanding majority of the tree searches and how tree search can be used to make decisions etcetera. So, at this point we will stop but,, I would really request you guys to study Depth First algorithm by yourself because as I mentioned, instead of expanding the depth first quickly, just to show you what happens the depth first goes something like this.

It will go 4 and 5 then you have 6, 7, 8, 9 like this and then you are backtracked here, you will have maybe you may have 3 options here 10, 11, 12 then you may have 13,14 you may have 15 and 16 something like this. So, you will go then, you will so you will first come here finish this then, you go back do this then, you go back and do this then, you come back and do this. So, that is the depth first search approach. So, I would request you to do that all by yourself and then try to derive the branch and bound from the best first that I just demonstrated in the class.

Thank you for your patient hearing. I will see you in the next class with probably decision tree approach and other business decision models.

Thank you very much.