INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

NPTEL

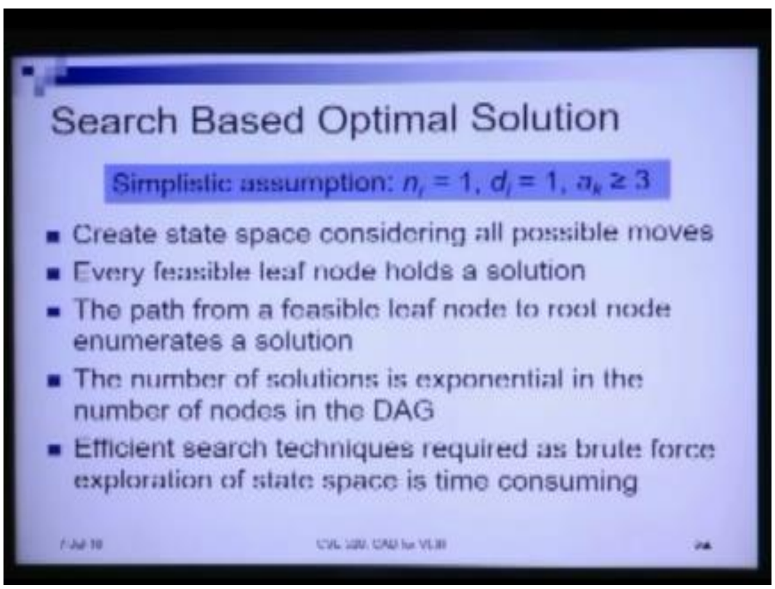
NPTEL ONLINE CERTIFICATION COURSE
An Initiative of MHRD

VLSI Design, Verification & Test

Dr. Arnab Sarkar
Department of CSE
IIT Guwahati

So we start module 1 of lecture 4 in this lecture we will look at a few more resource constraint scheduling strategies the first among which is a first search based optimal scheduling strategy based on the branch and bound technique here we will take the simplistic assumption.

(Refer Slide Time: 00:51)



That number of resource types is one each operation has a unit delay and we have just three operators or resources that are available so as we discussed in the last lecture this takes it to the NP-complete domain so on two resources I can do it in polynomial time but not on three resources so how does this solution approach proceed first we create a state space considering all possible moves right.

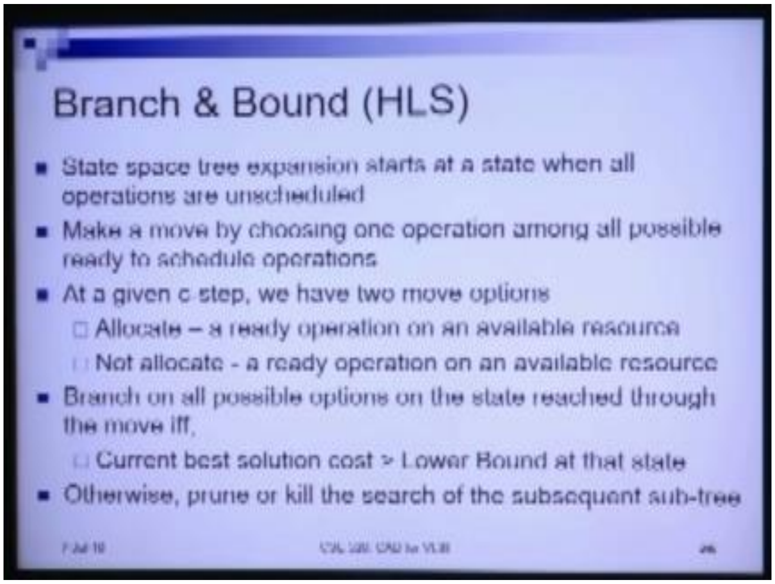
And what do we mean by a state space here as we said the state space gives me the different ways of allocating operations to resources at different time steps so all the different possible ways the exhaustive set of possibilities of allocating operations to resources at different possible time steps is the total state space now we do not explicitly create a state space and then search it here the state space gets implicitly created a state space tree gets implicitly created as we proceed through the search.

Now however as we said we will look as to how we obtain this tree but given that we obtain a tree each leaf of this tree so the state's pastry each leaf of the state space tree holds a feasible solution how does this hold a feasible solution the path from this leaf node to the root enumerates a distinct solution because a distinct path from the root to leaf is a distinct set of allocations of operations to resources at particular time steps if you have a different path that different paths will give you a different allocation of operations on resources at time steps hence if a leaf node is reached then we have a feasible solution.

That means we only say that we have not pruned and reached a leaf of the state space search tree of the state's pastry only if let us say my latency bound meets if the if my latency bound does not mean given the resource constraint then I have an infeasible solution that incredible solution will not be contained in the state's pastry the leaf node will not hold a valid solution then so the path from a feasible leaf node to root node enumerates.

The solution the number of solutions is exponential in the number of nodes in the DAG as we can very intuitively understand that this will be true and because we can have possibly a huge search space of the state's pastry efficient search techniques are required as a brute force method can be very time consuming and hence we arrive at the branch and bound solutions technique solution technique.

(Refer Slide Time: 04:25)



The state's pastry expansion starts at a state when all operations are unstable so all operations in the operation constraints graph are firstly unscheduled we have not scheduled any operation not ascribed any time step to any operation now we take a set of possible moves and we can take a set of possible moves depending on the possible choices.

That we have at a given time make a move by choosing one operation among all possible ready to schedule operations with respect to my with respect to my operation constraints graph here what do we have we choose one of the operations among all the possible operations to schedule so let us say if we have three resources and let us say seven operations ready we can at any given time step we can take any three of these seven ready operations and this will give me a distinct allocation.

So at any given C step or time step we have two possible moves what do we do we have a set of possible operations we have a constraint on resources we take an operation and we can either allocate the operation at this time step if a resource is available and even if it resources available we may choose not to allocate an allocated different operation so for each operation we have two possibilities at a given time step either you allocate a ready operation on an available resource or you do not allocate a radio operation on an available resource.

Now whatever decision you take you suppose I have four operations that are available and I choose to choose not to allocate operation number two I have four operations 1, 2, 3, 4 and we choose not to allocate operation number two but allocate operation number three now on my decision not to choose operation number two I need to branch on all possible options that I now have now what do we have I have three more operations and whatever in which way in which other ways I can branch in all possible ways I will branch from that state what is that state my move that I will not allocate operation number two.

That brings me to a distinct state and at that distinct state I have a set of possibilities and I will branch on all those set of possibilities now however this branching will actually be carried out if the current best solution as we said the solution is of is obtained at certain leaf nodes maybe that an initial solution we either we can take the initial solution to be infinite or we can take the initial solution as the output of a heuristic method and get an initial bound on the solution and that solution will be what for example if we take the same objective as the ILP that we defined in the last lecture is to minimize the start time of the in it operation that means is to minimize the length of the entire schedule.

If that is the objective then the current best solution will have a certain time value on what is the schedule length of that solution now the current based solution in terms of each a to length is lower than is greater than the lower bound at that state that means that at a given state I have a I have a set of unscheduled operations now I can obtain a lower bound on the minimum number of time steps that will be required for me to share do the remaining operations starting from that time step why is that so firstly because on the remaining number of nodes I have dependency constraints and also on the remaining number of loads I also have resource constraints.

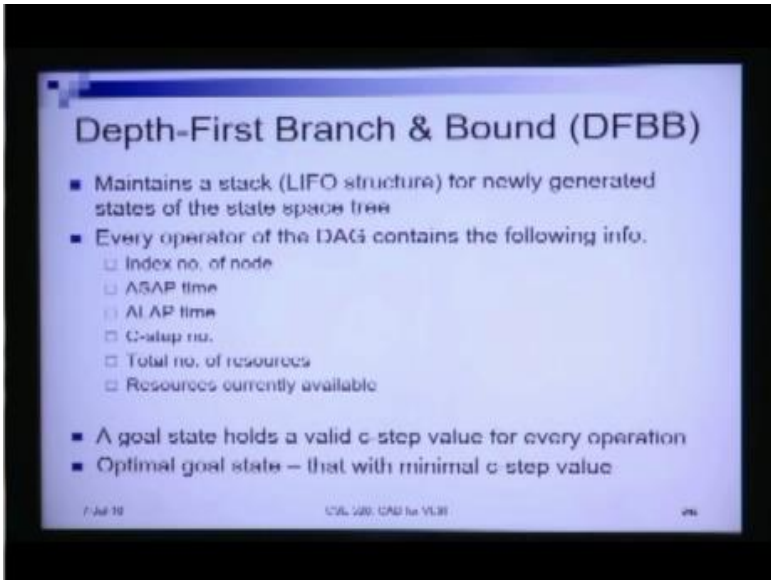
So I will have I can obtain I can obtain a quick lower bound on the number on the number of time steps that will be required now I already know how much time I have taken come from the root node to the current load that means operations that have already allocated I know what is the time that has been taken now the total estimated time for the solution will therefore be the actual time that have been that I have taken to come from the root node to the current state and the plus the estimated time the lower bound on the estimated time.

That I have the for scheduling the operations from the current step we shall do the remaining unscheduled nods now that gives me an estimated solution now we say that if the current best solution is better than the lower bound on the estimated solution it is never possible that I will explore the rest of my search space the sub tree that I have and get a better solution in such cases we can prove the rest of the search tree from this state and look at other places and backtrack to possibly search to search in other areas of the state space.

So this is the pruning criteria in the branch and bound method so this is what we said that if the current best solution is greater than is greater than the lower bound then what is what do we do then only we search the state space but if the current best solution is lower here lowered means what here lower means that the schedule length is lower so the solution is better if the current based solution if the cost of the current based solution is greater that means the schedule length is higher than I then I will schedule the rest of my search space starting from this state because there is a possibility that I will get a solution that has a that gives me a schedule length that is lower than the current based solution.

However if the current based solutions value that means the schedule length is lower than the lower bound that I have at the current state I will not expand from the current state anymore rather die will backtrack and search a different part of the state space this gives me the pruning criteria otherwise if this is so if the we prove or kill the search of the subsequent search tree and this saves me time with respect to exhausted brute force searching.

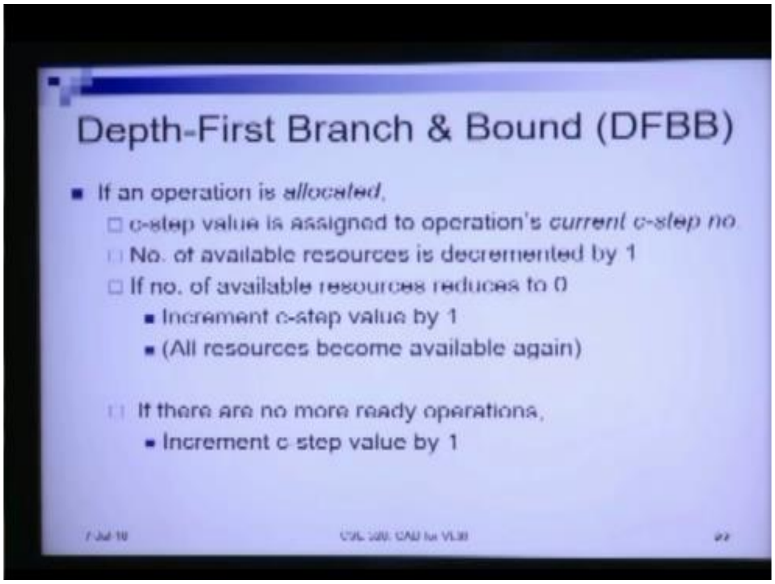
(Refer Slide Time: 11:37)



In the depth first branch and bound what do we maintain we maintain a stack of newly generated states in the state's pastry right every operation of the DAG it is not operated but operation every operation of the DAG contains the following information the index number of the node that means what is the number of the node of the operation the ASAP time the a lap time the current time step number the total number of resources and the current number of resources available at this time step which means that I have a total number of certain number of resources but I have already allocated a few operations to a subset of the resources say then we I only have a subset of the resources available this information is also contained in a state.

A goal state holds a valid C - step value for each operation that is what we already said the leaf node holds a valid cease their value for all operations now what is the optimal goal state that goal state that leaf node that gives me the minimal C - step value that means that will give me the least schedule length.

(Refer Slide Time: 12:52)

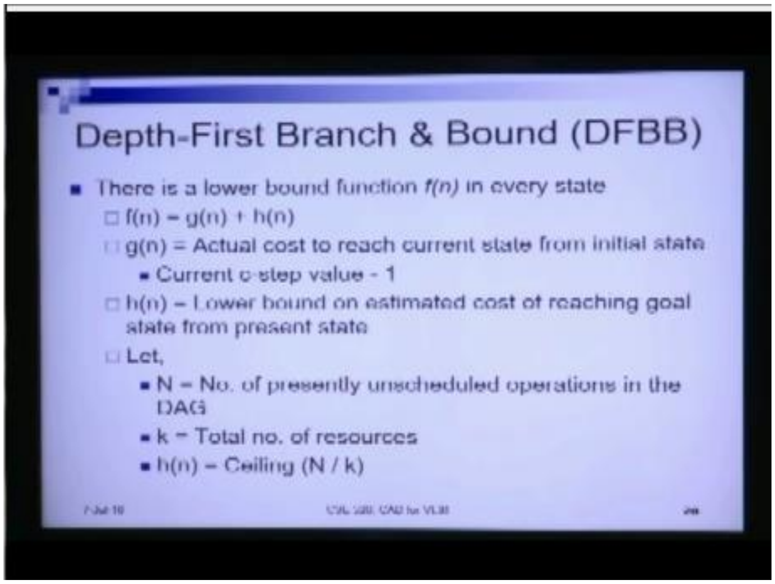


So firstly I have no operations allocated I have a set of nodes that are ready independent at the first step then what do what do I do I choose to allocate one of them now what happens then if we choose to allocate the steed step value is assigned to the operations current C – step number so that operation is now allocated that system because I have chosen to allocate it at that C – step the number of available resources is decremented by one.

The number if the number of allocated resources reduces to zero we increment time step by one and then all resources become available again why because we have assumed here that all my operations take unit delay have unit delay so therefore every time step every operation takes only one times the time step to execute hence if I have all resources exhausted at a time step and then I increment time then all my resources become available again if there are no more radio operations then increment C - step by one.

Now what is the other way around is this because this is the DAG the maximum possible parallelism given the available resources it may not be possible to attain at every C – step because I do not have that many ready to schedule operations because of my dependency constraints in the graph and hence if that happens I have to increment time in order to complete scheduling or complete the execution of the currently allocated operations so that a few more operations become independent and ready to be scheduled.

(Refer Slide Time: 14:46)



So as we have been talking about the function in mathematical terms how do we obtain the cost the estimated cost at any given state the it is obtained as follows there is a lower bound function f(n) in every state and f(n) is defined as a summation over g(n) + h(n) what is g(n) is the actual cost to reach the current state from the initial state that means currents c is the value – 1 if we have because we have not already allocated at the current step we have only actually allocated till the last step.

So the schedule length till now of the operations of operations that have already been scheduled that schedule length is current c-step - 1 one now what is h(n) is a lower bound or estimated cost of reaching the goal state from the parent state for example how can such an estimate be obtained the estimate let us say n be the number of presently unscheduled operations in the DAG so I have already allocated and chilled a set of operations but there are a set of operations that are that are left to be scheduled that I have to schedule and that is denoted by n let k denote the total number of resources that are available.

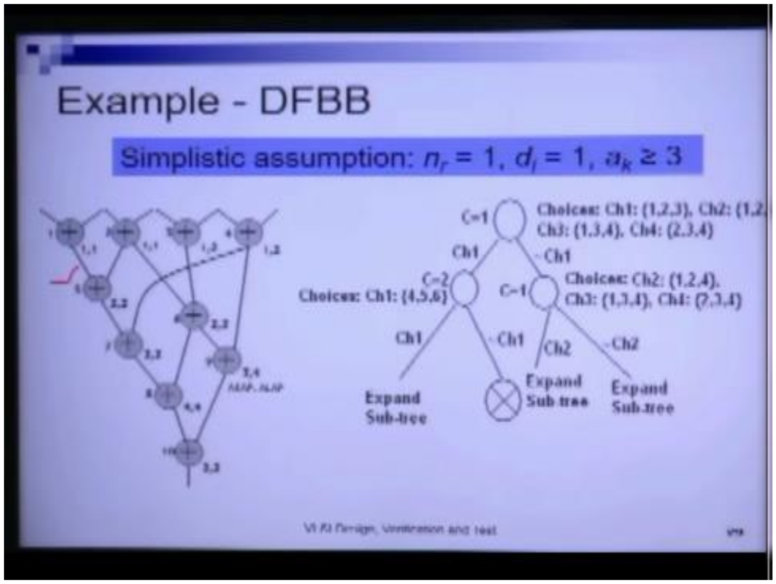
So then because every operation takes only one time step to execute so what is a lower bound on the number of time steps that will be required to show you these remaining n nodes it is given by the ceiling of n / K assuming that there will be no constraints due to the dependency and we will always be able to achieve the maximum parallelism that is available then this is true this ceiling n/ K now obviously we see that this can be this h(n) the value of this h(n) can be obtained in constant time.

However this estimate is a bit crude we can still obtain better estimates but obviously at the cost of cost of higher time complexity because why do we talk of time complexity at this point because at each step to calculate each of this heuristic functions at each step we need to calculate this heuristic function and the calculation of this heuristic function will therefore consume time at each step so these heuristic function the calculation of this heuristic function cannot take significant amount of time however if the estimate is better than we have higher chances of pruning.

Therefore the heuristic function that we will choose is a trade-off between how good the estimate is because although this is a lower bound the closer this lower bound is to the actual solution that I will get by searching the actually searching the state space below and the closer I will get higher is my chance that I will be able to prune apart of the state space because if the lower bound is low then that then I have a higher chance that the current based solution will tell me that you have a possibility of searching the remaining state space is from this state and get a better solution than the current based solution however a better estimates increases the probability of higher pruning.

However again as I said there is a cost of generating this solution this estimate this heuristic estimate and hence finding this estimate how good it will be is a trade-off so one good lower bound estimate has been provided in the book.

(Refer Slide Time: 19:21)

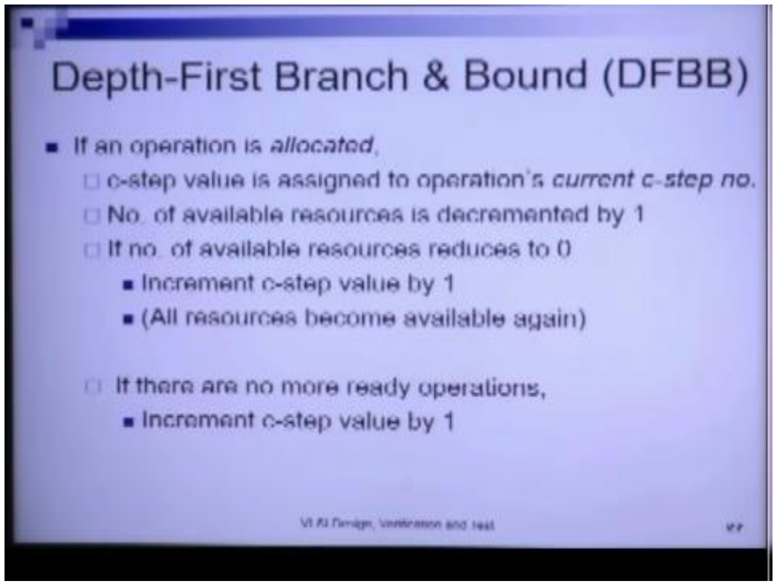


Now with this we will take a small example we have this operation constraints graph where there are ten operations 1, 2 , 3, 4, 5, 6, 7, 8, 9, 10 and we have the ASAP and a lap times that are given in this comma separated these comma-separated numbers between beside each node tells me the Slap and lap times now we will take an example to show how a solution strategy that uses branch and bound progresses from the beginning it is a small partial example a full example will take lot of space and time.

The DAG that we have on the left hand side here we will use this all operations take unit time to execute and we have three resources we said that for a general DAG this is this is where is the boundary where the solution goes to the np-complete space for two resources of one type we can schedule it in polynomial time we are using this simplistic assumption that we have only one type of resource but we are now working with a the NP complete solution and hence we are trying to search a large state space systematically using depth first branch and bound.

In branch and bound the basic approach is to enumerate all possible choices at any given state for each choice we will branch on all possible options by considering that when we take that choice what are all the possible ways in which to get the solution and if you do not take that particular choice what are the possible ways in which the solution can be obtained so both the things will be will be considered and solution will proceed with both the considerations.

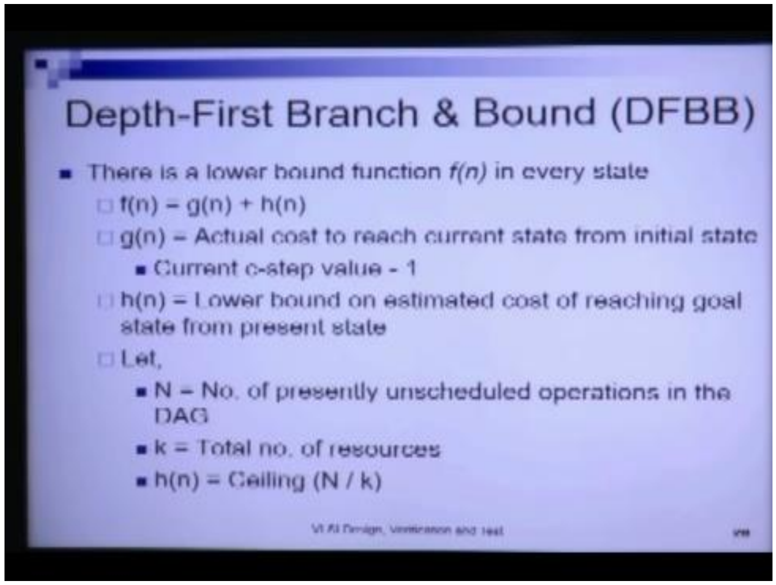
(Refer Slide Time: 21:10)



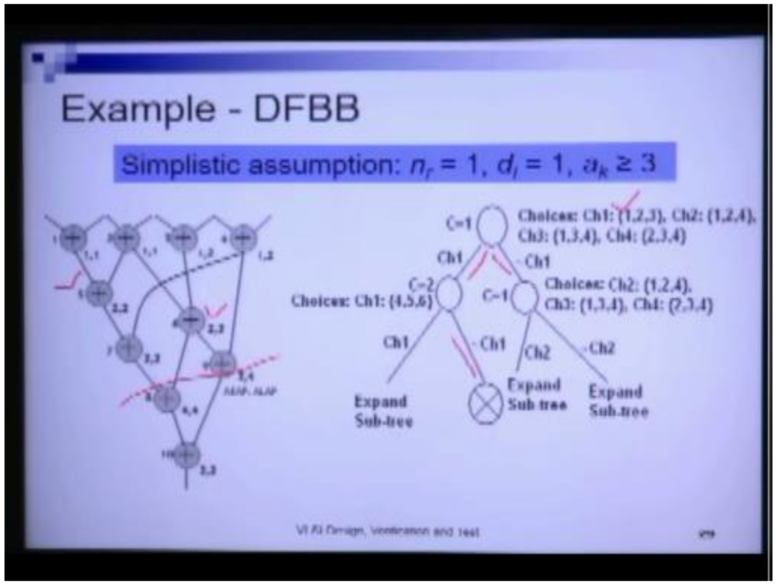
In the previous approach we said that when we actually allocate a choice a choice is what for us here a choice is the allocation of an operation to a particular resource now when we take a particular choice that is we actually allocate a certain operation on a resource we assign a c step value with the operation the number of available resources at that time step it decreases by 1 and if the number of resources decreases to zero that means at this time step I have exhausted all my resources then I must increase time to get back all my resources and if there are no more ready operations also I have resources but I do not have any more radio operations.

Then also I have to increment time to get back my resources because after execution of the current operations that are allocated to the resources the resources will again become free and currently ready of unscheduled operations can be shared on those resources.

(Refer Slide Time: 22:15)



(Refer Slide Time: 22:18)



In this example we can also expand the state space you know you know in a slightly different way this is a resource constraint problem so at any point in time I have to exhaust all my resources so instead of progressing a teach time step operation by operation we can progress time step by time step in this case because all operations take unit time then for example at the first time step at C = 1 which de notes the first time step and nothing has been scheduled then our choices are as follows I can use three resources and I can allocate three operations to this resources.

So the choice one is I allocate 1, 2 and 3 to the three resources choice two is I allocate 1,2 and 4 to the three resources choice three is allocate 1,3,4 to the three resources and choice four is I allocate 2, 3, 4 the available three resources so the number of choices here will always be number of available radio operations c the number of resources so we here the number of choices are 4c 3 because the number of independent radio operations at the first step is for and the number of resources available is three.

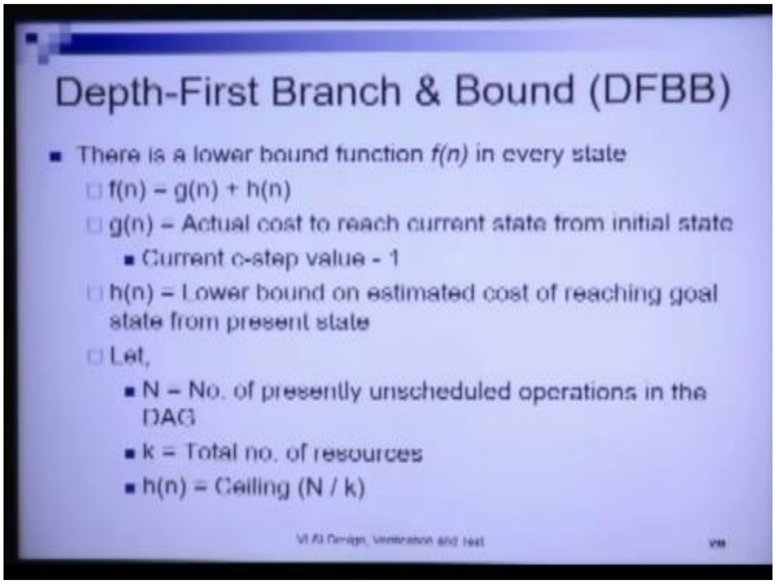
And let us say we first consider choice one and then for choice one we have two options either to take choice one or not take choice one so Ch1 we can take or tilde Ch1 it refers to the case that we do not take choice one so when we take choice one that means that operations 1, 2, 3 have been allocated to all the three resources and hence my time will progress by one here and in the in the second case available choices will be 4,5 and 6 and only one choice will be there why because I have three resources I will always try to maximize my resource use it because this is a resource constraint problem then we want to minimize time.

Because this is our objective we would like to maximize the resource usage at each step and hence my choice my choice will be 4,5,6 and that is only choice now if I do not take choice one what happens I still remain in time step one on the left side because I have actually taken choice one so I progress to time step two on the right side I have not taken choice one so my time is still at one now the available choices are 2,3 and 4 now again I will proceed further I choose to 2 or 3 or 4 any one of them if I if I choose two then whether I select the choice to or do not select wise to I have two options.

Now on the other hand and the left hand side I am at time step two and the only choice is 456 if I take choice one I will again schedule I will make available resource 9 then I will make available resource 7 and 9 will become available at this time steps here at the next time step 3 however if I do not take choice one then I have nothing to do and my solution will just backtrack here right because I there is only one choice at this point so this explains this small partial graph explains how the state space free will be slowly expanded to generate all possible cases and exhaustively search the state space.

Now how do we prove so we said that what is the pruning criteria the pruning criteria as we said was $f(n) = g(n) + h(n)$ where $g(n)$ is the actual amount of time that I have taken to travel from the root node root state to my current state and ancient is a lower bound of the time that I will take to schedule the remaining unscheduled operators from the current state.

(Refer Slide Time: 26:36)



So we said that g(n) is the actual cost to reach the current state from the initial state and h(n)is given by ceiling N/ K what is ceiling N / K N denotes the number of presently unscheduled operators and K is the number of resources so N by K tells me the minimum amount of time that I will require to schedule the rest of the nodes this is how solution will progress.

Centre For Educational Technology
IIT Guwahati
Production

Head CET
Prof. Sunil Khijwania

CET Production Team
Bikask Jyoti Nath
CS Bhaskar Bora
Dibyajyoti Lahkar
Kallal Barua
Kaushik Kr. Sarma
Queen Barman
Rekha Hazarika

CET Administrative Team
Susanta Sarma
Swapan Debnath