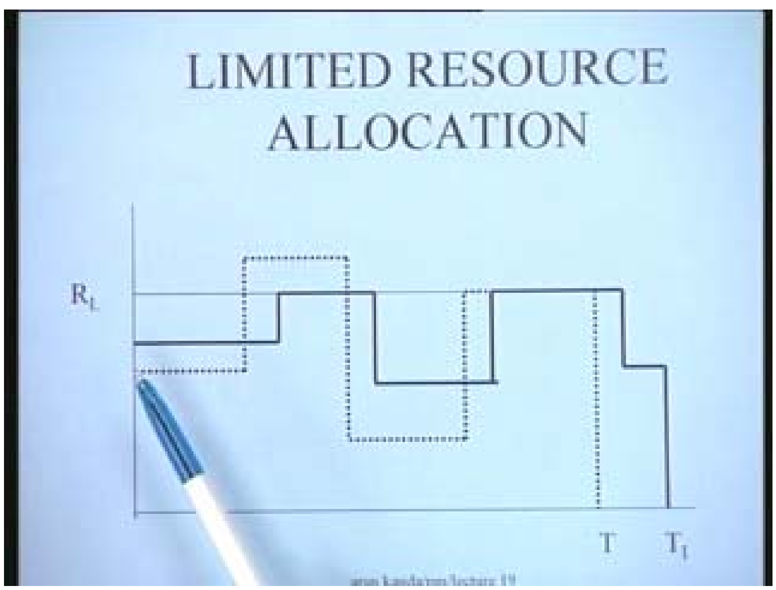
Project and Production Management

Prof. Arun Kanda
Department of Mechanical Engineering
Indian Institute of Technology, Delhi

Lecture - 15
Limited Resource Allocation

Today we are going to be talking about limited resource allocation. In our last lecture you would recall we had talked about the problems of resource aggregation and the problems of resource leveling. The objective in resource leveling was to be able to smoothen the resource usage profile so that it was as well leveled as possible. Today we are looking at the limited resource allocation problem and if you look at the essential problem and how it differs from resource leveling is, in resource leveling we are trying to obtain favorable resource usage profile without worsening the project duration t but what may happen is that even the best leveled resource profile might exceed the limited resource availability. If this happens this schedule would not be easy to implement because during this particular time period you would be exceeding the resource availabilities that you have. You would have to make special provisions either to hire additional resources or to make some other arrangements of sub contracting and so on. If this was not possible we would have to look for means by means of which we could accomplish the project within the limited resource availability.

(Refer Slide Time: 2:40)



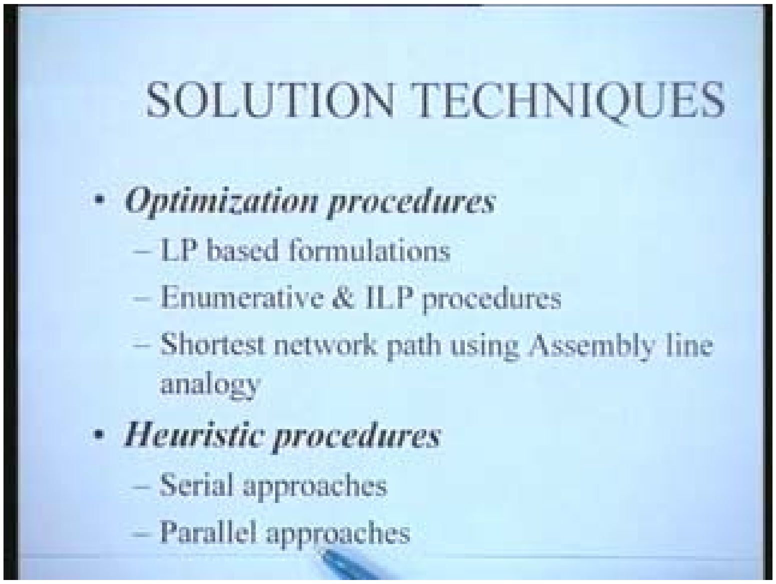
The essential problem in limited resource allocation is to be able to find out which are the activities which are going on during this particular period and may be to defer them beyond this peak so that the new resource usage profile actually conforms to the resource restrictions that we have and in this process inevitably the project duration increases from the original value T to this particular new T_I, the increased value of the project duration.

The major problem in resource allocation is to be able to determine this resource usage profile in such a manner that this increase in duration is the minimum. This is an optimization problem in which we are interested in finding out the minimum project duration. Subject to the availability of resources there could be not only one resource but there could in fact be multiple resources and we are trying to find out a particular value of the duration such that all the resource availabilities are satisfied. Mind you resource leveling is often referred to as unlimited resource leveling because there we are not specifically concerned with the availabilities of the resources.

In the limited resource allocation problem, there are two ways of looking at the problem as far as the solution techniques are concerned. They are number one the optimization procedures which guarantee an optimal solution and some typical examples of these are they are generally some LP based formulations in which constraints are imposed saying that the precedence restrictions are met, that the resource availabilities are not violated in each period. Constraints are written for these individual constraints and then the formulation is solved. The other approach to optimization procedures is enumerative and integer linear programming approaches. The problem of determining a project schedule is a combinatorial problem. Why is it a combinatorial problem; simply because there are many possible options for the starting date of a particular activity. Whenever you look at all the possible combinations of starting dates of various activities you get a variety of schedules and therefore it's essentially an enumerative technique and in fact lot of refinements in enumerative algorithms have been proposed especially by using some branch and bound techniques and searching over a tree and there are other procedures which are based on analogies.

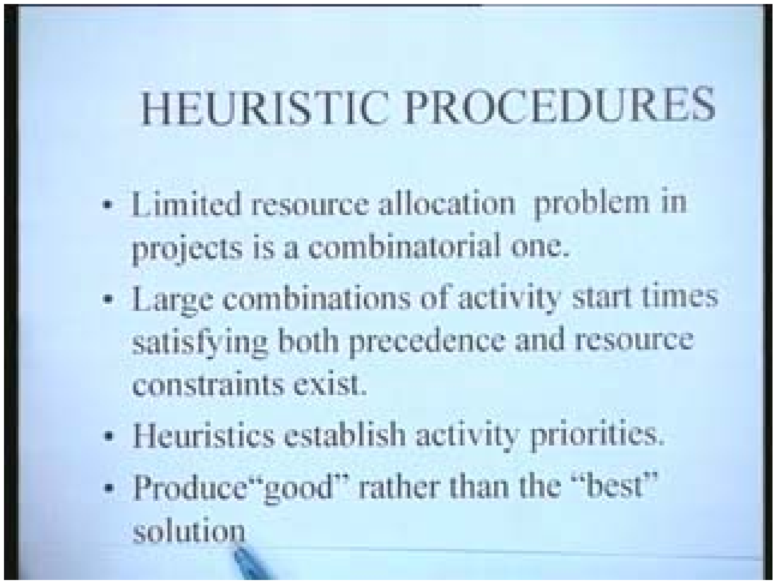
It's a very interesting network path. That means developing a network and trying to find out the shortest path in the network by drawing an analogy between the assembly line balancing problem and the limited resource allocation problem. But the difficulty with all these approaches is that they are not good for real life, real sized problems. They are demonstrated to be good for problems which may involve 10 or utmost 20 activities because the size of the LP formulation or the size of the enumerative number of solutions becomes very, very large and it's not possible to use these procedures for real life situations. Although these particular procedures have been useful in research and a lot of research has been done on these procedures they haven't found their way into practical algorithms in commercial software which is generally used for these kinds of purposes. The commonly used procedures are essentially heuristic procedures and one way to classify these procedures is serial approaches and parallel approaches.

(Refer Slide Time: 7:04)



These differ in terms of how the heuristic is used to basically select out of the many possible combinations the one particular combination that is to be scheduled. Let us now examine some of the problems where heuristic procedures and the kinds of approaches which are available. The limited resource allocation problem in projects is a combinatorial one. That I have already mentioned. By this what we mean is there are a large number of combinations of activity start times satisfying both precedence and resource constraints. These combinations exist and because of these large combinations you have a large number of heuristic procedures possible. What do these heuristics do? The word heuristic simply means a rule of thumb. A rule of thumb is actually used to establish activity priorities in these schedules and these produce good rather than the best solution as we have seen.

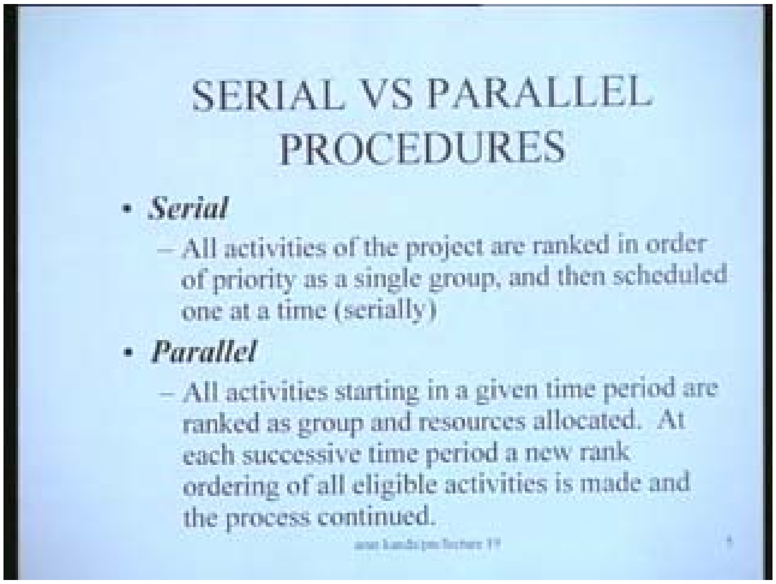
(Refer Slide Time: 8:13)



Talking about the two broad categories of heuristic procedures namely serial and parallel procedures, serial procedures are those procedures in which all activities of the project are ranked in order of priority as a single group and then scheduled one at a time. That is serial. What happens in serial procedures is that you establish the priority of the job only once and you do not change it during the course of the application of the algorithm. It is something like you say activities are born with. It's something in their genes. The moment you pick up a project you can identify the ranks of the activities and these ranks stay with them as long as the activities are being scheduled during the particular process. This is what is being done.

As opposed to this we have parallel procedures. In parallel procedures what happens is that all activities starting in a given time period are ranked as a group and the resources are allocated and at each successive time period a new rank ordering of all the eligible activities is made and the process continued. The key difference between serial and parallel procedures is that in the parallel procedures in each period you set up a competitive examination as it were and you allow the activities to perform on that and it's on the basis of their karma that the individual activities perform and you make a selection based on that. In each activity the activity can in fact improve its ranking or may be worsen its ranking depending upon the situation.

(Refer Slide Time: 10:09)



Out of these two kinds of procedures parallel procedures have proved to be more efficient generally than serial procedures and we will take a look at one such parallel procedure during the course of this particular ……. There are a number of commonly used heuristics but here is a partial list of these heuristics. Minimum activity slack is perhaps one of the most commonly used heuristics.

That means if an activity is critical it should be given greatest priority. If it's less critical it should be given lesser priority and so on and this is determined by the total float of the activity which is I think intuitively quite obvious. The minimum late finish time of an activity is another criterion which is quite commonly used and it is shown that both these criteria are actually one and the same in terms of their application. A resource scheduling method has been proposed in which a priority is given to a particular activity based on its early start and the late finish time of the activity. So you can attach some weights and you can develop a priority sequence for the various activities. The greatest resource demand means that suppose there are three activities competing to be scheduled on a particular day. The activity which has the greatest resource demand which requires the maximum crew for instance would be the one which would be picked up by this particular heuristic.

Then we have the greatest resource utilization. Greatest resource utilization means that whenever there is a set of activities which has to be scheduled you try to pick up a subset of these activities in such a manner that the maximum amount of resource is scheduled. So it's lie trying to pick up. It's like a typical classical knapsack problem in linear programming where you are trying to pack a bag with the maximum value assuming that each item which you have to pick up has the maximum value subject to the resource availability that is the capacity of the bag. The asterisk here shows this. What can be done is a small LP can be solved to determine which particular combination of activities will in fact give you the greatest resource utilization.

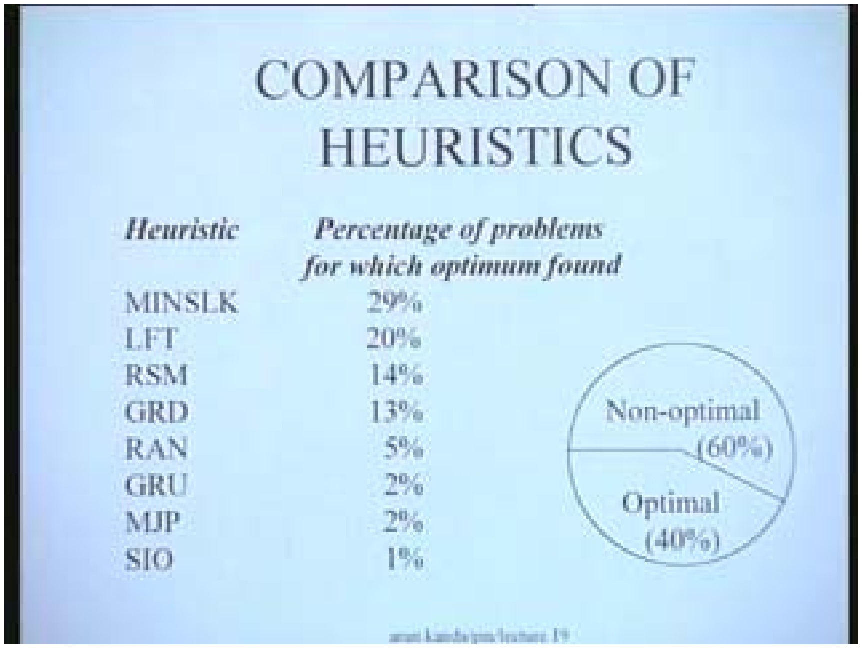
The shortest imminent operation out of the various activities the one which is likely to take the shortest time is given a priority here. This is a rule which has performed very well in job shops especially because it leads to low inventory levels. But as we will see this does not perform so well in project scheduling. Another heuristic which is quite often used in scheduling of projects is most jobs possible.

(Refer Slide Time: 13:27)

Most jobs possible means that if there is a list of 5 jobs and all the 5 cannot be scheduled because of resource availabilities you try to pick up a schedule in which the maximum number of jobs is scheduled; may be 3 or 4 can picked up. That can be done. This again can be set up as a linear programming problem and you can identify this set of jobs which should be picked up and another very commonly used heuristic is random activity selection where out of the various jobs one can be selected at random and you can do this.

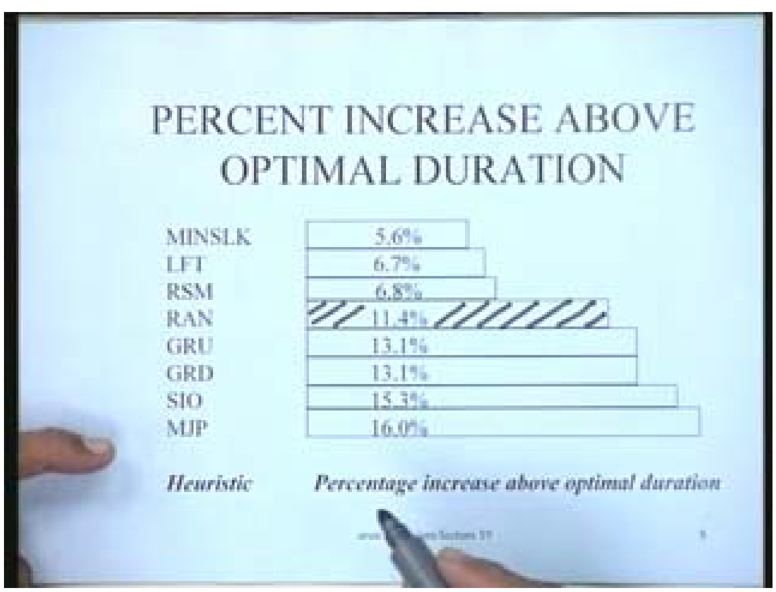
In one particular study these 8 heuristics were compared and it's interesting to see the results that were obtained. If we look at these 8 heuristics we find that this is a ordering of the heuristics in terms of the percentage of the problems for which the optimum was found. It was found that the min slack heuristic for that set of problems led to an optimum in 29% of the cases, LFT in 20% cases, RSM in 14% cases, GRD greatest resource demand in 13%, random in only 5%, greatest resource utilization in only 2%, most jobs possible in 2% and shortest imminent operation was in fact only 1% which is the worst performance as far as this is concerned. Since these rules did not give the optimum for the same set of problems it showed actually for the entire problem set optimal solutions were obtained in 40% of the cases by the application of one or more rules and for 60% of the cases the non-optimal solutions were found out by the heuristics.

(Refer Slide Time: 15:18)



That gives you an idea about the performance of these heuristics. Another study was done on how bad or good these heuristics were in terms of their capability to give an optimal solution. This is the increasing order of the percentage increase above the optimal duration. You can see from this graph is that the random rule gave a percentage increase above the optimal duration of about 11.4%. This was the kind of performance of the random rule and all these rules had been arranged in this fashion.

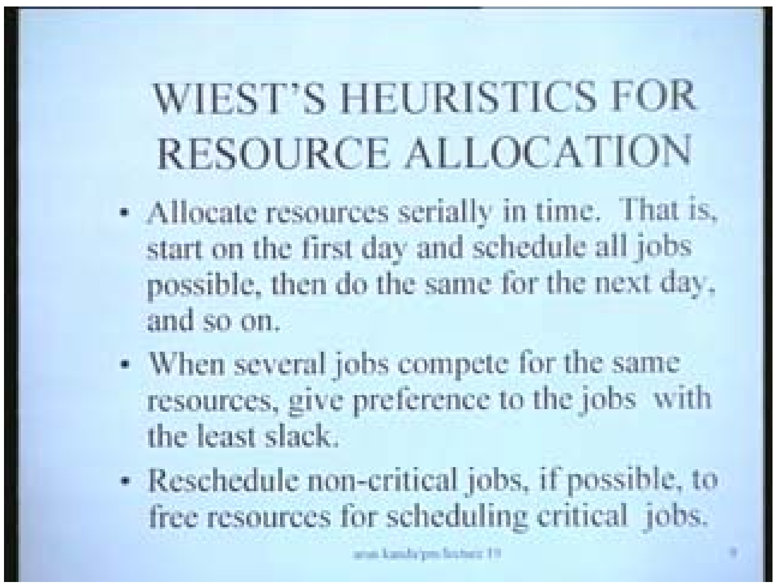
(Refer Slide Time: 16:09)



As a general indicator you can see that these are the 3 rules which are better than random and these are the 4 rules which are worse than random. The min slack rule gave an increase in optimum duration of only 5.6% above the optimum, LFT 6.8%, RSM 6.8%. They were fairly close and pretty good and better than random and if random is taken as a boundary these were all much higher; 13.1%, 13.1% and going up to 16%. The conclusion here is that rules based on min slack or latest finish times are generally performing the best as far as the heuristics are concerned.

We will now take up heuristic procedure suggested by Wiest for resource allocation. This particular procedure of Wiest is based on very simple 3 rules of thumb and these 3 rules are the following.

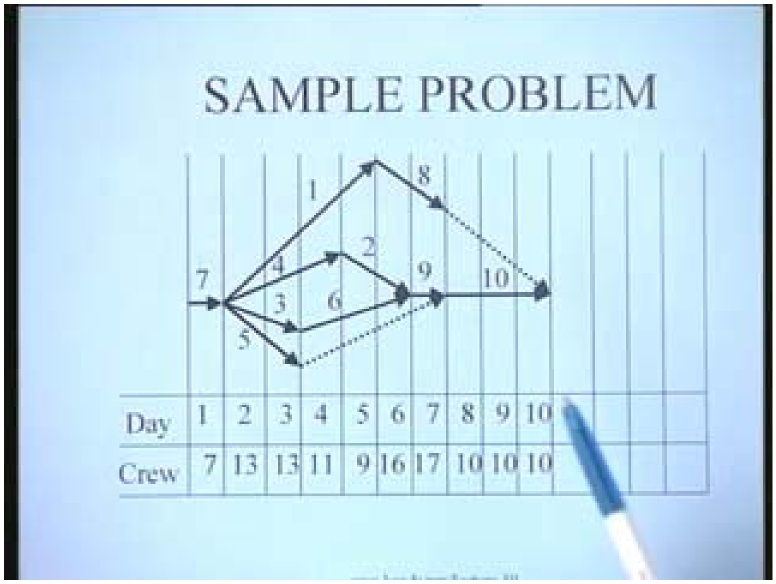
(Refer Slide Time: 17:25)



The first rule simply says allocate the sources serially in time. That is, go from 1 day to the next day. That is, start on the first day and schedule all jobs possible. Then do the same for the next day and so on. Basically what we are doing is we are moving from the first day to the second day to the third day and so on, very simple. That's the first heuristic. The second heuristic says that when several jobs compete for the same resources give preference to the jobs with the least slack or the least total float. What we are trying to do here is that we are using the heuristic of least slack to pick up the jobs and third one is a slightly interesting rule. It says try to reschedule non-critical jobs if possible to free resources for scheduling the critical jobs which means that at any particular stage if a situation occurs when a non-critical job is already scheduled but there is a critical job which is demanding some resources. If it's possible to sacrifice the non-critical job then try to do it in favor of the critical job if possible; otherwise not. Simply we call it rule 1, rule 2 and rule 3. These are the 3 heuristics on which the wiest procedure is based for resource allocation.

We will look at a small problem and try to apply these three rules to that problem systematically and try to see how we can obtain a solution to the limited resource allocation problem. First let us state the problem. Let us take a network which has 10 jobs.

(Refer Slide Time: 19:38)

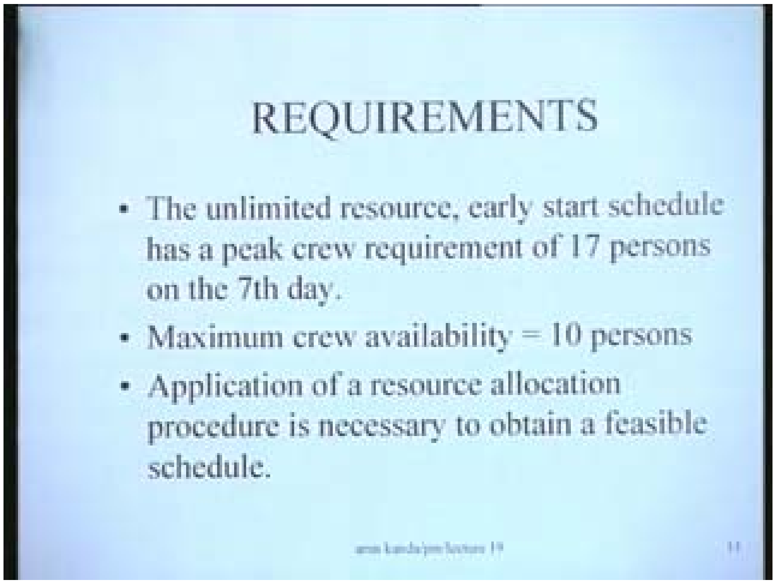


You can see that jobs are numbered 1, 2, 3, 4, 5, 6, 7, 8, 9 and 10. There are 10 jobs in this network and the number of the job also indicates the crew requirement of the job. For instance job number 7 requires 7 men. This job number 1 requires 1 man. This job 4 requires 4 men and so on. The durations of each of these activities are actually indicated on this time scaled network. For instance 7 takes only 1 day. 1 takes 4 days. 4 takes 3 days, 3 takes 2 days, 5 takes 2 days, 6 takes 2 days, 2 takes 2 days, 9 takes only one day and 10 takes 3 days. We have developed in this particular case the early start schedule for this particular project. All activities are scheduled at their earliest.

These dotted lines show you actually the amount of total float which is available to the various activities. For instance activities 1 and 8 have a total float of 3. Activity 5 has a total float of 1, 2, 3, 4 days and in fact in this particular case there are 2 critical paths, 7, 4, 2, 9, 10; this is one and 7, 3, 6, 9 and 10. These are the two paths which are critical and the project duration without any resource restrictions is now 10 days as you see here and you can do a resource aggregation exercise and see that the total resource requirement is 7 on the first day, 13 on the second day, 13 on the third day, 11 on the fourth day, 9 on the fifth day, 16 on the seventh day and 17 on the seventh day and subsequently 10 on days 8, 9 and 10. You find that the peak resource requirement is 17 which occurs on the 7th day and this is in fact the problem data as it is. The unlimited resource allocation problem will give us this.

For this particular problem what are our requirements? The requirements are stated as follows. The unlimited resource early start schedule has a peak crew requirement of 17 persons on the 7th day as we have seen from this example. The maximum crew availability is only 10.

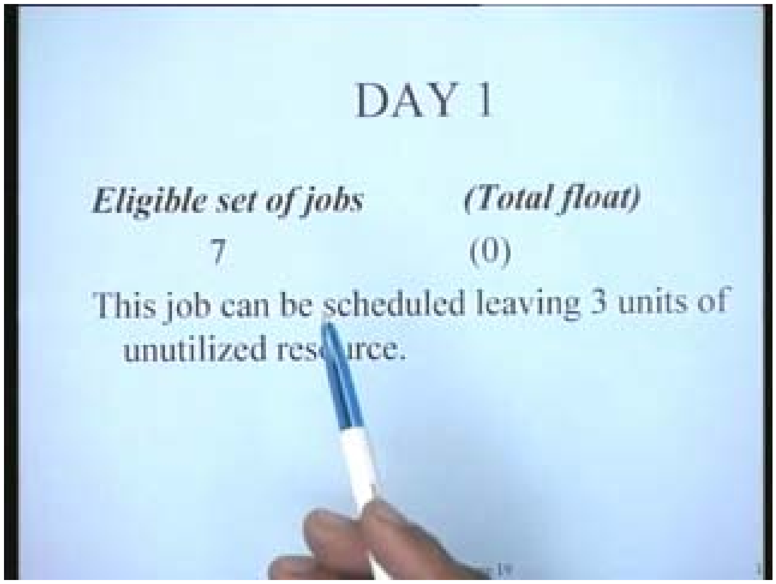
(Refer Slide Time: 22:28)



That's what we are seeing and the application of a resource allocation procedure is therefore necessary to obtain a feasible schedule within the limit of 10 men. For this example it's easy to see that you cannot - ? find a schedule with less than 10 men because there is one particular job that is activity 10 which requires 10. The maximum resource requirements of any activity in a particular project will in fact be the lower bound on the resources needed for that particular job. There would be other factors as well but in this case we are looking for a schedule with only 10 men. What we do is we keep this original problem in mind which defines our structure, which defines our precedence relationships and what do we see in this problem. We find that there is only one activity which has no predecessors. That is activity number 7. This particular activity is the only candidate on day one for scheduling.

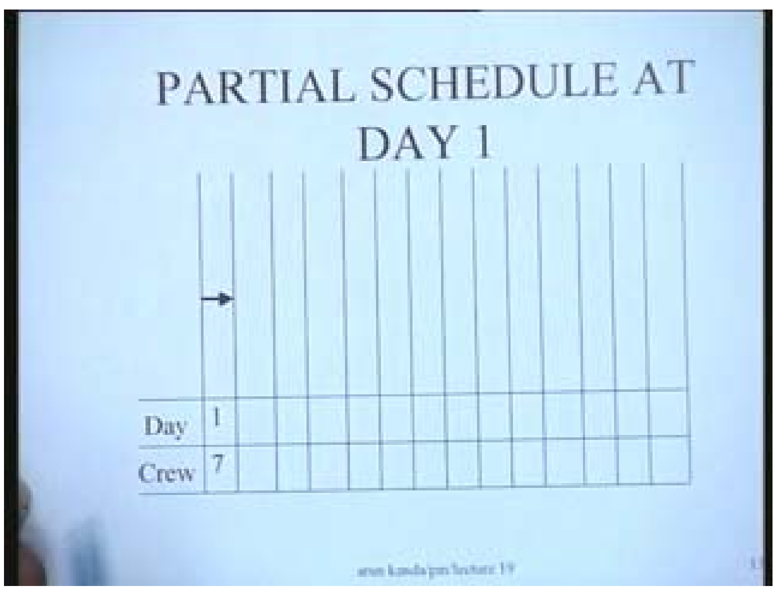
We proceed systematically as per the first heuristic day by day. On day one what happens? On each day we construct a list of eligible set of jobs and their total floats. We find that job 7 is the only eligible set of jobs because it has no predecessors and it has a total float of zero because it's a critical job and this job can be scheduled leaving 3 units of unutilized resource because there is nothing else, no other job. So this job can be scheduled.

(Refer Slide Time: 24:20)



The decision on the first day is this. On the first day we will now build up the schedule day by day. On the first day we take a decision that we schedule this particular job 7 and once we schedule this job on this particular day, on the first day the total crew that we have used up is 7 and after this we move to day 2.

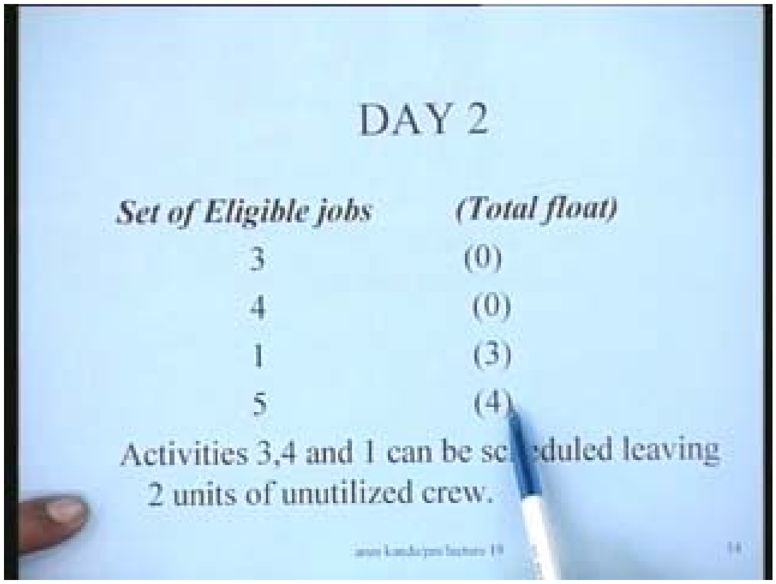
(Refer Slide Time: 24:48)



As you move to day 2, job number 7 has duration of only 1 day. So job number 7 would be over. When job number 7 is over all the successors all the immediate successors of 7 become candidates for placement on the eligibility list. Let’s find out the eligibility list on day 2. Mind you that we are doing a parallel heuristic.

So on day 2, we have these 4 jobs, 3, 4, 1 and 5 which have now become eligible for placement and we order them first according to their total float because this is going to be our tie breaking heuristic.

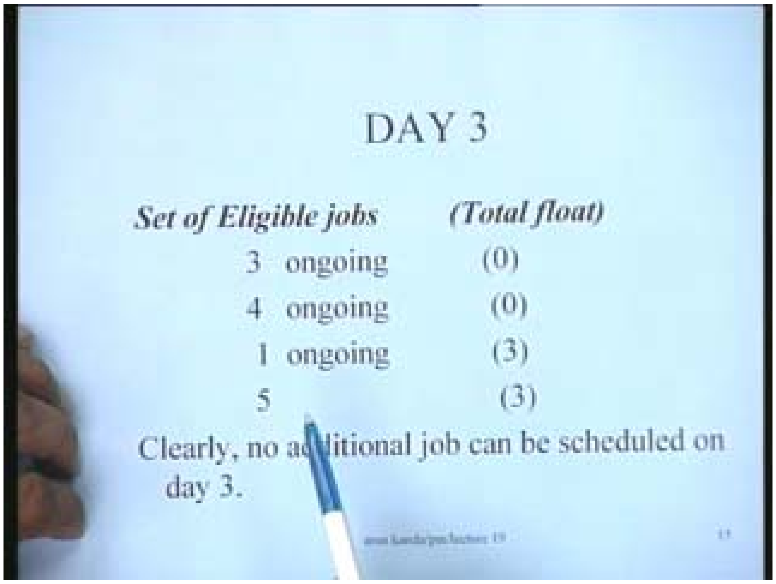
(Refer Slide Time: 25:37)



So 3 and 4 are both critical jobs. They have a total float of zero. Job number 1 has a float of 3. You can see for instance job number 1 has this float of 1 2 3. Both 1 and 8 can be slided by 3 days and job number 5 has a float of 4 days. This is the order of the jobs. Now we have to allocate a total of 10 and we will pick up in this order. 3+4 is 7; 7+1 is 8 and this cannot be scheduled. Activities 3, 4 and 1 can be scheduled leaving 2 units of unutilized resource. That means 8; 4+3=7; 7+1=8. 8 units of resource would be consumed. That is the crew consumed would be 8 in number and 2 units of unutilized crew could be left over. Out of these if you see job number 3 has duration of 2 days. Since job number 4 has duration of 3 days, job number 3 has duration of 2 days and job number 1 has duration of 4 days. The first job to be completed out of this will be job number 3 which takes only 2 days. Since this is started in day 2 it will be over in day 3. That's what it means.

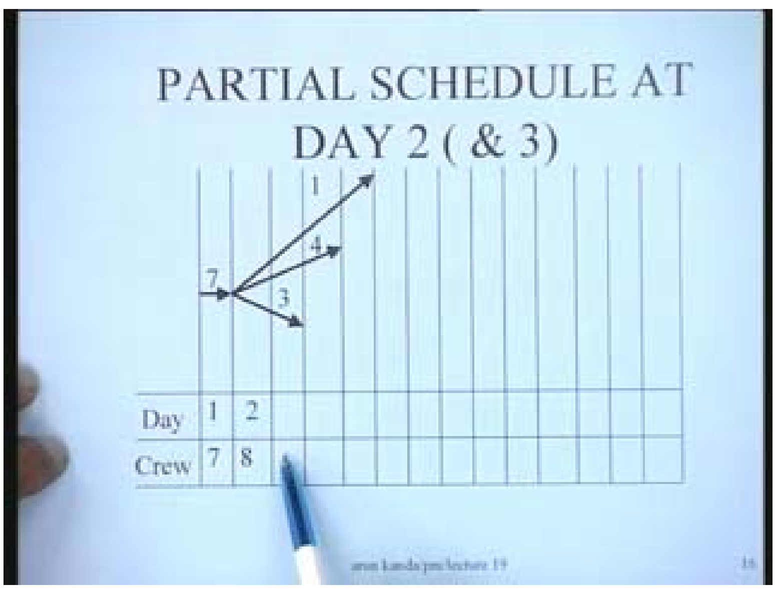
Let's look up the situation as to what would change on day 3. On day 3 when you look at the whole thing 3 is on going job because we had scheduled it in the previous period. 4 is on going and 1 is on going because these are the 3 jobs that were scheduled earlier on day 2. This was a critical job. This was a critical job and this particular job had a float of 3 days and 5 now has a float which is 1 day less than what it had previously. It previously had a float of 4 days but now it has a float of only 3 days because we have already postponed the start of 5 by 1 day. This is what has happened. On this particular day clearly no additional job can be scheduled because this cannot be accommodated with these on going jobs.

(Refer Slide Time: 28:05)



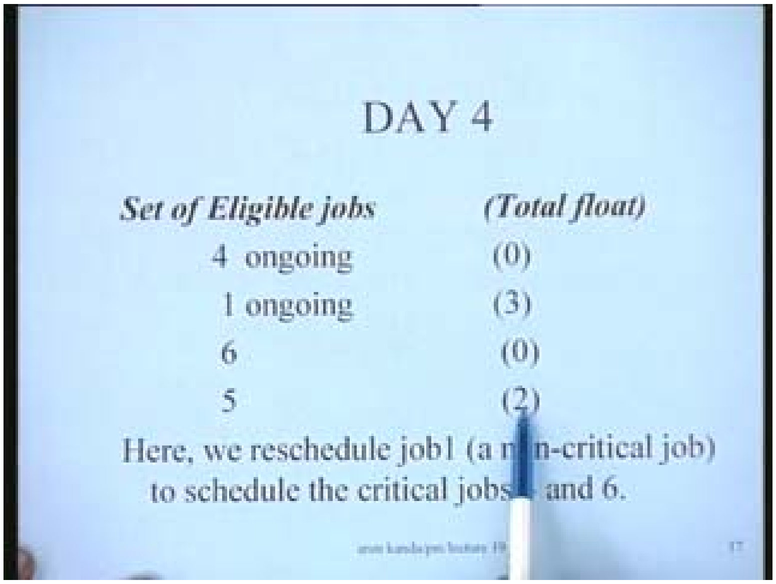
As far as day 2 and 3 are concerned, on 2 we took a decision. On day 2 we took a decision to schedule 3, 4 and 1 and the implications of that decisions would be felt in the future depending upon the durations of these activities because we have now committed resources to these activities by virtue of the decision that we have taken on day 2 and on day 3 these are continuing as things are and therefore the partial schedule for day 2 and 3 would be something like this and even on day 3 we are actually consuming 8 units of resource.

(Refer Slide Time: 28:51)



Let us see what would happen now when we come to day 4. Look at day 4. On day 4, what has really happened is that job number 3 is over but job number 4 and 1 are on going. So 4 is on going and 1 is on going by virtue of the previous decisions taken on day 2 and this is a zero and job number 1 has a float of 3 days and what has happened now is job number 6 becomes a candidate for placement. 6 is the immediate successor of job number 3 and it's a critical job and job number 5 has now a float of only 2 days. It is reduced by 1 more day.

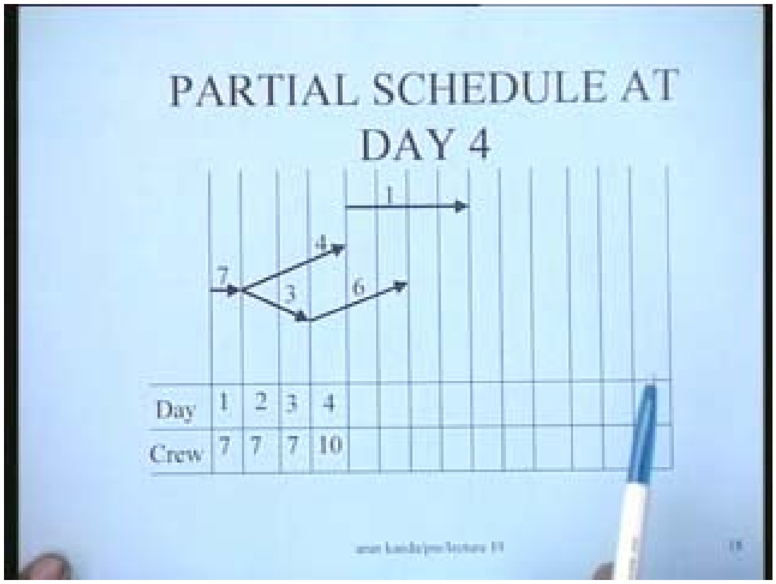
(Refer Slide Time: 29:48)



This is the ordered list that we have. These are two ongoing jobs and here is a critical job. The critical job is clamouring for resources and at this stage we can invoke heuristic 3, rule number 3 of Wiest procedure. Because what we notice is that here we have an instance of 1 job which is actually a non-critical job which is consuming 1 unit of resource only but here is a critical job which is asking for resource and cannot be accommodated along with this. If we give up 1 then 6 and 4 can be accommodated within the resource availability of 10 days. That's what the third rule says that here if this is possible let's try to do that. So here we reschedule job 1. That means job 1 which was being done we sort of forget about it, take it off and we try to reschedule the critical jobs 4 and 6 on site like this. If that happens on day 4, what are the decisions? Let us try to depict on day 4 the decisions that we have taken.

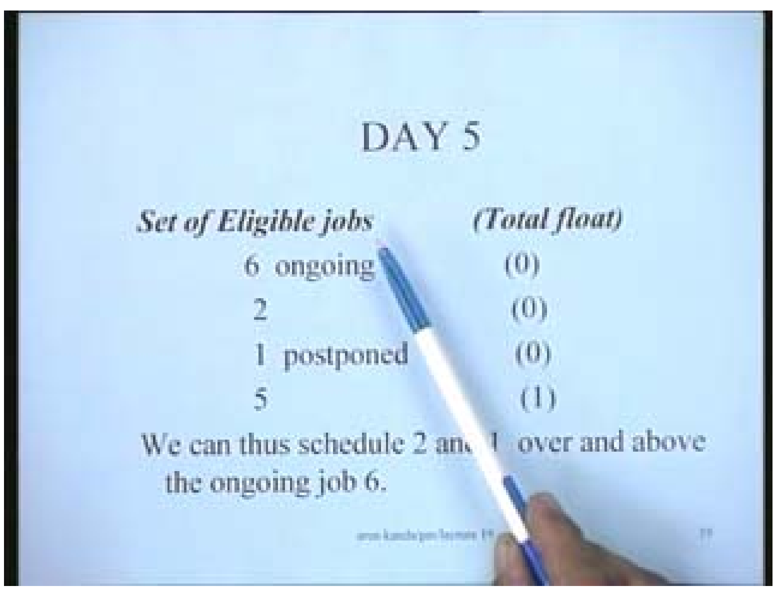
The situation on day 4 is job 1 which was done earlier, we have forgotten about it. We have taken it off and we say it can be done some other time. Earliest it can be done is in day 5. But on day 4 we have rescheduled; we have scheduled job number 6 and job number 4 is already going on. The total resource requirements are 10 and we have updated the crew requirements here by virtue of the fact that 1 has been rescheduled to 7, 7 and 7.

(Refer Slide Time: 31:40)



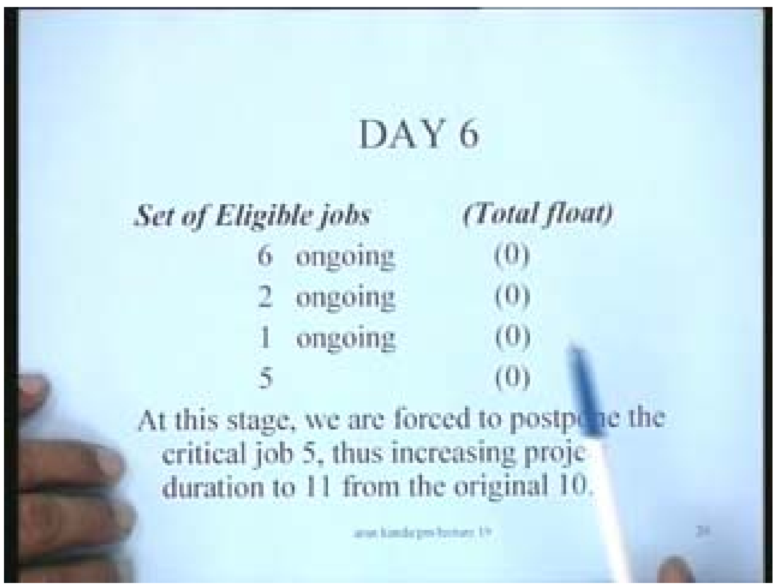
This is the change that is made on day 4. Up to day 4, the partial schedule shows that activity 7, 3, 4 and then 6 have been scheduled and on day 5 we will see what can be done about the various jobs which are candidates for placement. On each day you have to maintain a list. Let us now look at day 5. What happens on day 5? On day 5, you can see from this partial schedule that job number 4 is going to be over. The moment job number 4 is going to be over, job number 2 which is the immediate successor of job number 4 would become a candidate for placement on the list and it being a critical job, it would in fact ask for a greater place. On day 5 our situation is something like this.

(Refer Slide Time: 32:43)



6 is currently ongoing; it's a critical job. 2 is a critical job. So we place it here and 1 is a postponed job. Earlier it had a float of 3 but now if 1 is kept the way it is, then it is postponed to day 5. The earlier postponed job 1, will become a critical job at this stage and 5 has a float of only 1 at this stage. 5 had a total float of 4. Gradually we have been postponing it on every day and therefore its slack has become lesser and lesser. So a stage would come when it would become critical and would become top priority. At this particular stage 6 is ongoing and 2 can be accommodated. This is also a critical job. So 6+2 is 8; 8+1 is 9. We will try to schedule all these 3 jobs. Thus we can schedule 2 and 1 over and above the ongoing job 6 on day 5. That's what we will try to do and on day 6 what's going to happen?

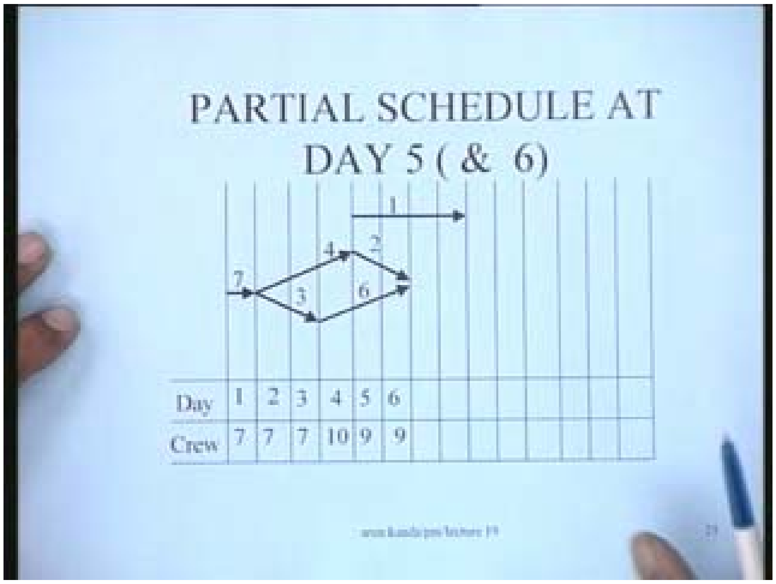
(Refer Slide Time: 34:10)



On day 6 again nothing very significant is going to happen. Because 6, 2 and 1 are the jobs which were scheduled earlier they would now be ongoing jobs on day 6. They would be critical jobs and at this stage job 5 has also become critical. Job 5 has become critical and these are 3 critical jobs going on. But we cannot accommodate this critical job. At this stage on day 6 we are forced to postpone the critical job 5 thus increasing the project duration to 11 from the original 10. This is what happens. Whenever we take a decision to postpone a critical job the project duration increases by 1 day. That's what happens here on day 6 because we had no other option. Because we are now postponing job number 5 which is in fact a critical job, the project duration increases to 11 from the original value of 10.

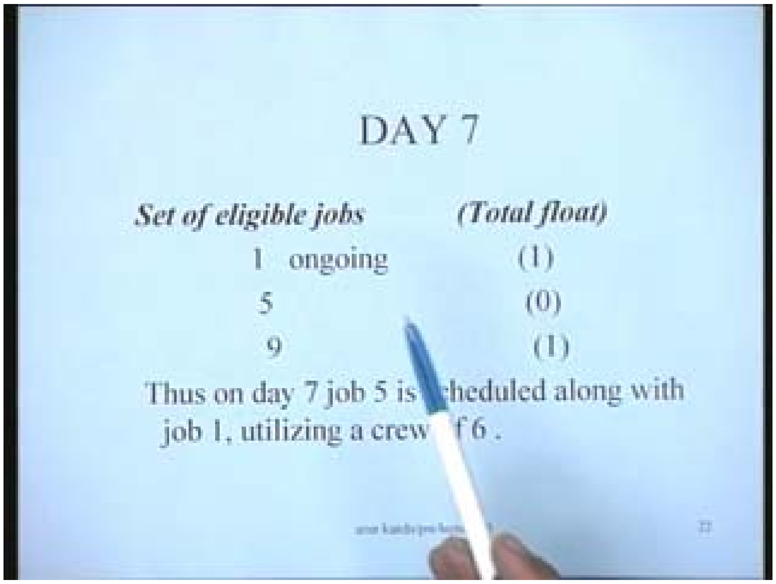
At this stage based on the decisions in day 5 and 6 let's see what our partial schedule looks like?

(Refer Slide Time: 35:24)



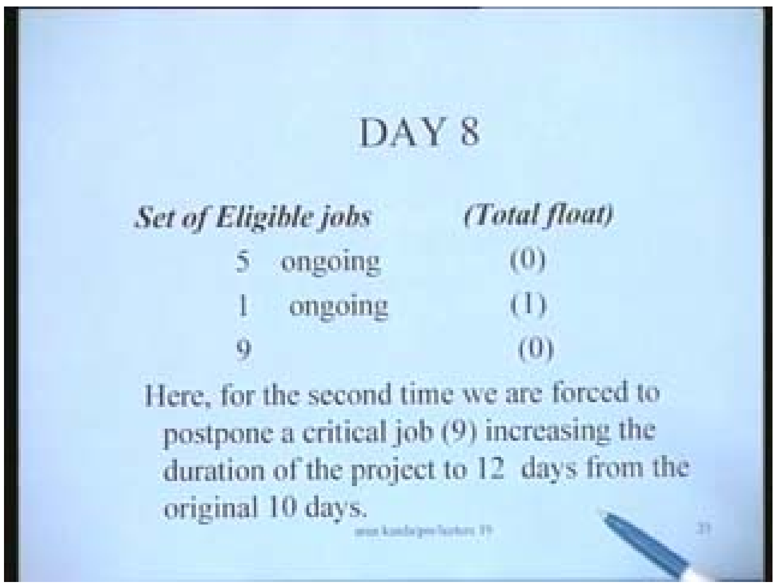
You find that on days 5 and 6 the decisions were 6 and 2 are ongoing and we had scheduled 1 also in parallel and 1 will have duration of 4 days like this. During days 5 and 6, the total resource requirement will be 9 and 9 respectively and this is what is going to continue at the end of the 5th and 6th day. When 2 and 6 are completed On the 7th day we would find that being successors to job 2 and 6, job 9 would also become a candidate for placement. Let us see how that happens? Let us update the list on day 7. Look at day 7. 1 is ongoing. It's good to write down ongoing here because that tells you that these resources are already committed and you have only 9 more resources to accommodate the other jobs.

(Refer Slide Time: 36:29)



At this particular case job 1 will have a total float of 1. Why does it have a float of 1? Earlier it had a float of zero because now you have postponed a critical job. Your project duration has become 11 which mean the float of all the jobs has increased by 1 day. That's what has happened except the critical job which is zero. 5 is now a critical job. So it has a float of zero and job number 9 which was earlier a critical job now has a float of 1 by virtue of the increase in duration of the jobs. Our list is like this. Thus on day 7, job 5 because this is a critical job it has to be scheduled and once it's put along with this, this job cannot be scheduled. Job 5 is scheduled along with job 1, utilizing a crew of only 6. That is the decision that we take on day 7. Is that clear? We have now decided to schedule 1 and 5 together and 9 will have to wait. Let us see what happens on day 8? You see it's a dynamic list.

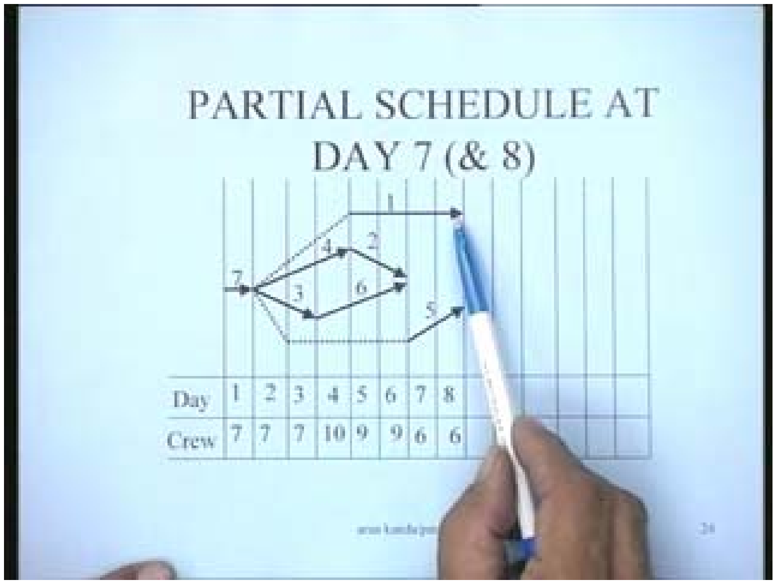
(Refer Slide Time: 37:51)



Each day the priorities keep changing. If you have not looked after a particular job which has been crying in the past it becomes more and more critical and at a certain point of time it asks for attention. That's the kind of thing. On day 8, job number 5 is ongoing, job number 1 is ongoing. This has a float of zero and this has a float of 1 and 9 because we have not scheduled previously, it has now become a critical job. Here for the second time what happens is on day 8 neither can we accommodate job 9 along with these two jobs and nor can we reschedule 1 because it will not release enough resources to accommodate job 9. Even rule 3 will not help us. Here for the second time we are forced to postpone a critical job 9 increasing the duration of the project to 12 days from the original 10 days. This is what has happened.

Whenever you take a decision to postpone a critical job, like we are doing this for the second time in this procedure, the project duration has now increased from the value 11 that we had had to 12. It has increased by 2 days from the original project duration. Based on these decisions that we have taken on day 7 and 8 let us see what the partial schedule is like? You can say that on day 7 and 8 we have taken these decisions basically to schedule job 5 along with job 1.

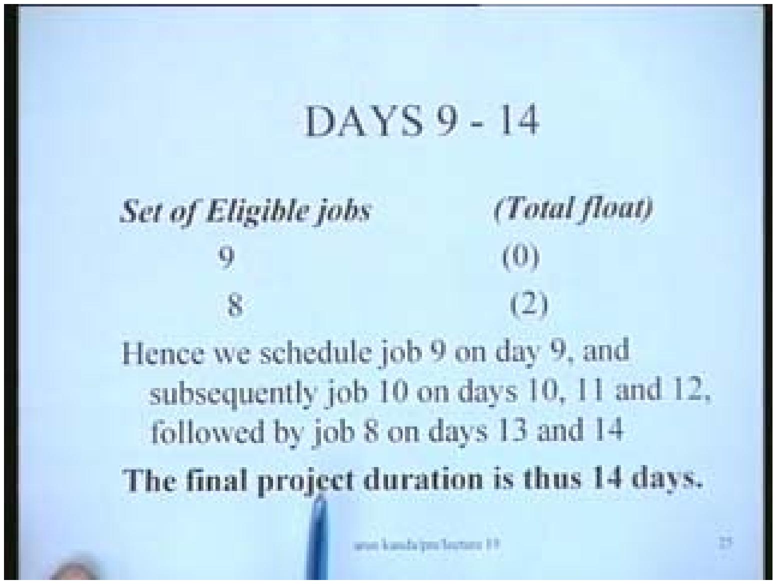
(Refer Slide Time: 39:40)



The crew requirement is 6 and 6 on both the days and we have been able to build up the schedule up to this particular point in time where we know that the total crew availability touched a figure of 10 on the 4th day, 9, 9 on day number 5 and 6 and thereafter it is being controlled within this particular figure as shown here. On day 7 and 8 we see that job number 1 and job number 5 both of them are going to be completed at the end of day 8. When you go to day 9 what happens? Let us see what happens at the end of day 9.

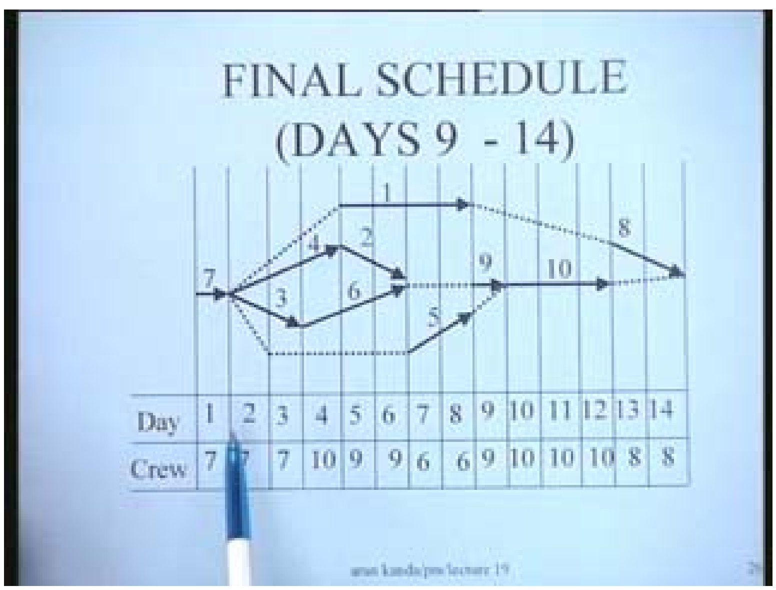
On day 9, the two jobs which are eligible are number 9 which is now a critical job with project duration of 12 and job number 8 which has a total float at this time of 2. If you recall job number 8 was a successor of job number 1. But at this stage it has a total float of 2 with this. Hence we schedule job number 9 on day 9 and 9 only takes 1 day. 9 is a job which has only a duration of 1 day and subsequently what's going to happen is that job number 10 on days 10, 11 and 12, which it takes 3 days, followed by job 8 on days 13 and 14because all these jobs 8, 9 and 10 their resource requirements are such that only 1 job can be scheduled on a particular day. You cannot have combinations of these jobs being scheduled by virtue of the limited resource availability that you have for these particular jobs. These are the decisions and in fact as a consequence what you find is that we schedule 9 on day 9 and 10 on day 10, 11 and 12. Subsequently job 8 has to be delayed. It becomes a critical job subsequently and therefore it will further delay the project by 2 more days.

(Refer Slide Time: 42:01)



The project will now end after the job 8 is scheduled on days 13 and 14 and the final project duration is thus 14 days. The final schedule for this particular case would look something like this. This is the final schedule.

(Refer Slide Time: 42:28)



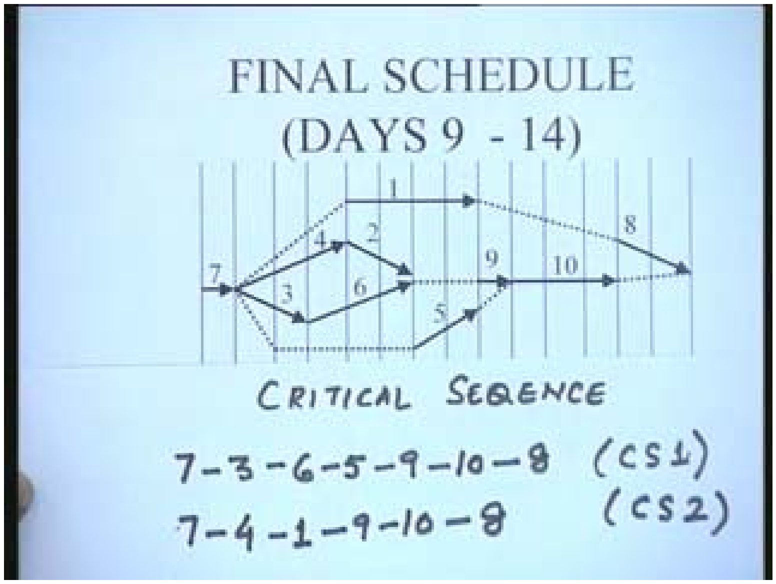
The final schedule shows that we had put job number 9 on the day 9 and after 9 we had put 10 and if we had put 10 we could not have put 8 in parallel with it because it was not possible. Otherwise had it been possible to finish 8 in parallel you could have been able to finish the project by day 12. But now it’s not possible. We have to finish job 8 for job 10 to be finished. Only then job 8 can be done. Job 8 is done in this particular fashion and this is again the slack the signs here and ultimately what you notice is that we have been able to accomplish the project in 14 days and the resource profile or the crew requirements are that on 4th day, the 10^{th}, 11^{th} and 12^{th} day you have a total resource requirement of 10. On other days it's less than 10. This is a feasible schedule for this particular problem.

In fact for this particular problem you work out an optimal schedule. This is in fact the optimal schedule as well. That means there is no schedule with project duration lesser than 14 for a resource availability of 10 for this particular problem. This is a specially constructed problem by Wiest may be to demonstrate his procedure. It is specially designed to generate an optimal solution but the heuristic will not necessarily give you an optimal solution as we indicated just now. If you look at this schedule, this is the final optimal solution. You will notice some very interesting properties of this particular solution. First thing is, if you recall that, the notion of the critical path vanishes. Isn't this something very surprising because earlier a critical path was an unconnected path. It was this one; this, this and this was connected to this, this and that was the critical path. Now there is no path in the problem which doesn't involve some kind of float. There is no path. This one, this one, this one, this one; so is there a critical path in this problem now? What do you think?

This is a very interesting thing which has happened and in fact in all limited resource allocation problems you do not have a critical path in the conventional sense and the notion of the critical path is however replaced by the notion of what is called a critical sequence. What is a critical sequence? Let us try to examine for this particular final schedule. That means can you identify some set of activities whose durations add up to the project duration? You see you can go 7, then you can go 4, then you can go 3, then can you go to anything else? Yes! You can go to 5 and then you can come to 9 and then you can come to 10 and then you can come to 8. Look at this same network and notice the following thing. In this very network let us try to understand the notion of a critical sequence.

Let us see what is a critical sequence? You go to activity 7; then you go to 3, then you go to 6 and from 6 if I add up to 5 and from 5 if I add up to 9 and 9 I add up to 10 and say I add up to 10. If I add up the durations of these activities this is nothing but a critical sequence, critical sequence 1. Isn't it? These are the durations; this, this, this and then you change. It's like saying that there is no direct path. When you come to this station you have to change your train and get into this one and then come here and then change a train again to go to this, this and then change again to this and there could be many critical sequences. For instance this was 3, 6. For instance another way of going is you could have say 7, 4 and then you can go to 1 and from 1 you can go to 9. You can go to 10 and then you can go to 8. This is a critical sequence 2. For the same problem I could have gone like this, this, this and this.

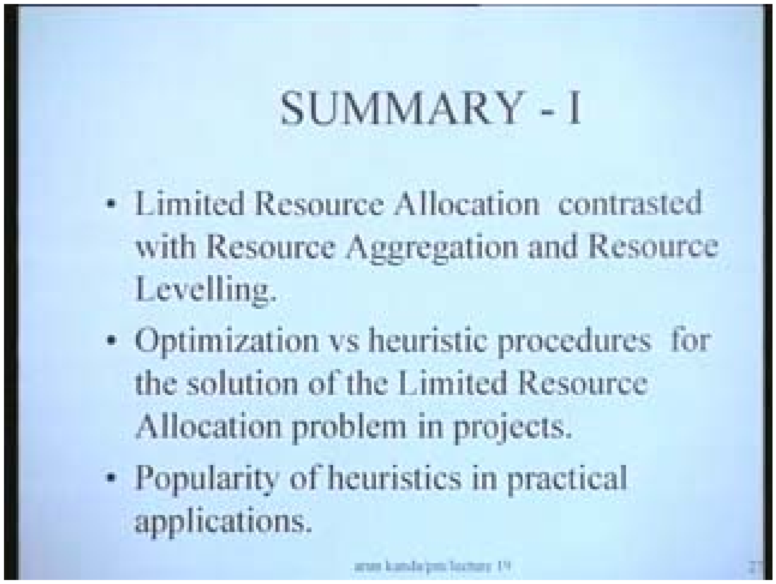
(Refer Slide Time: 47:27)



The sum of the durations of all these activities is 14 but mind you these activities are not strictly related by the precedence that is found in a conventional critical path but the sum of the durations of these activities does lead to the total duration of 14 and it's not the critical path. The term that is used for such things is the critical sequence. There are critical sequences and there could be many critical sequences in a particular problem but you will not have necessarily one connected path which will go like we had in the earlier case. Here there is a vacuum. You go here, then here, then change again here. This is another interesting feature. The notion of the critical sequence is actually used in modeling the problems sometimes. Basically when you are trying to model the problem instead of dealing with the critical path we have to deal with the critical sequences for such problems.

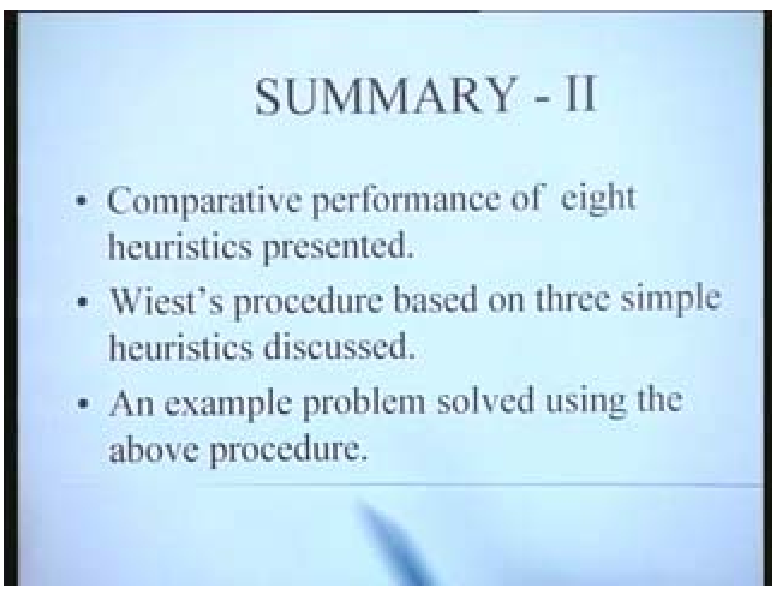
Let's try to summarize our discussions today. In summarizing the discussion today we saw that the limited resource allocation problem was first of all contrasted with the resource aggregation and the resource leveling problems. I think the point that really has to be understood here is that the resource aggregation problem is the simplest problem which does not involve any optimization. Resource leveling requires some optimization but it does not worsen the project duration and this is the most complicated problem namely the limited resource allocation problem where you are trying to conform to limited resource availabilities and the objective is to minimize the project duration. I think that distinction between the three problems must be clear. Then optimization versus the heuristic procedures for the solution of the limited resource allocation problem and the problem in projects were presented. It was seen that the heuristics were typically more popular in practical applications than optimization solutions.

(Refer Slide Time: 49:55)



Then we looked at the comparative performance of 8 heuristics and we found that heuristics based essentially on minimum slack or the latest finish time, perform the best. A procedure due to Wiest based on three very simple heuristics was discussed and an example problem was solved using the above procedure.

(Refer Slide Time: 50:22)



Finally some interesting things that came out from this procedure were that the notion of a critical path is not necessarily valid any longer. This is now replaced by the generalized notion of a critical sequence. I think we will stop here. Thank you!