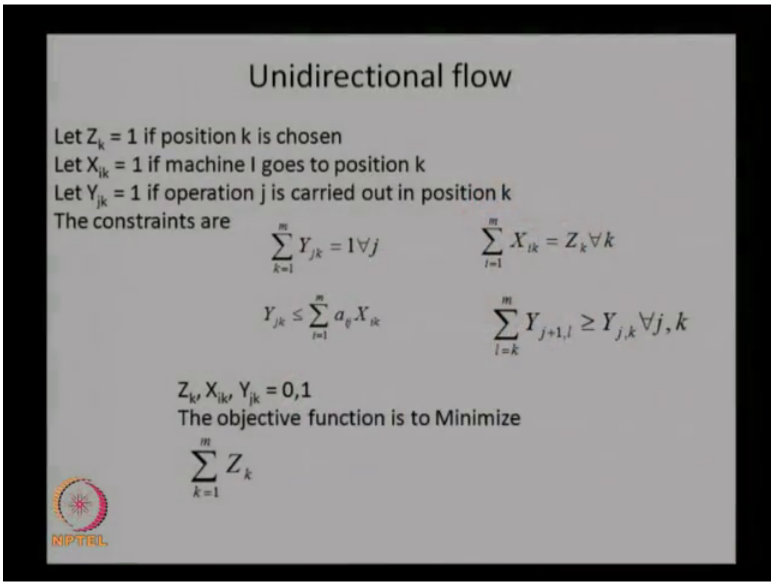
Manufacturing Systems Management
Prof. G. Srinivasan
Department of Management Studies
Indian Institute of Technology, Madras

Lecture - 19
Branching algorithm for product based cells, Operator and task assignment

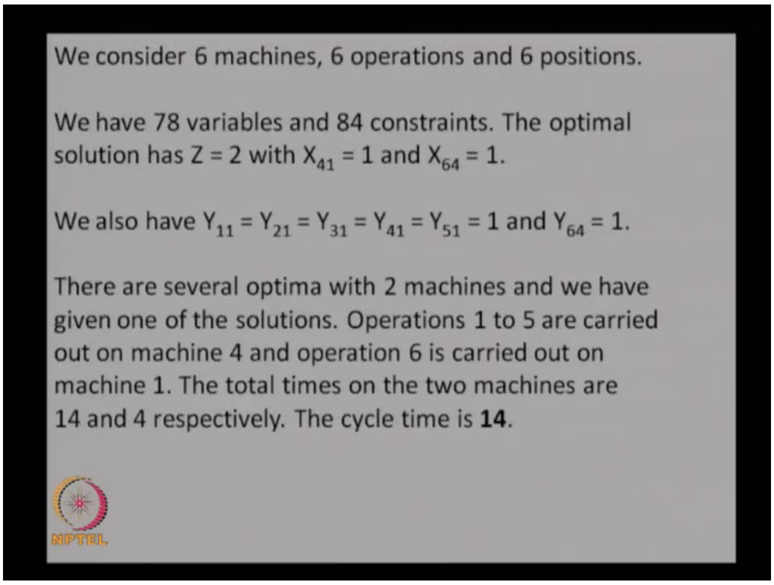
In this lecture we discuss the Branching Algorithm, for cell formation considering cells that are based on products. After discussing the branching algorithm we will discuss the case with multiple products and then we will move on to discuss models that consider operator allocation and assignment.

(Refer Slide Time: 00:40)



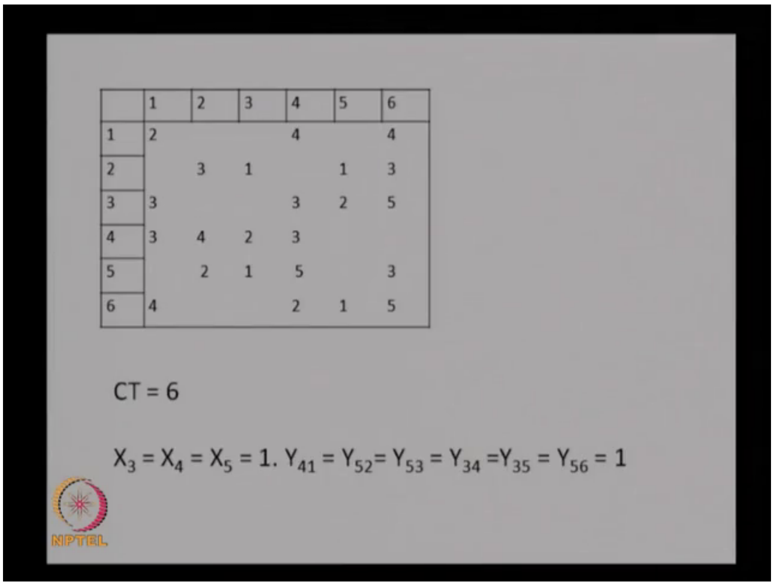
Now in the earlier lecture we saw a model for unidirectional flow.

(Refer Slide Time: 00:43)



We saw the integer programming model which had 78 variables and 84 constraints. We also mentioned that this problem can become very large if the numbers of operations are large. Now instead of solving it by integer programming, now can we look at a branching algorithm which can give us an optimum solution to this problem; perhaps considering or taking less time than the integer programming formulation? We will see one such branching algorithm.

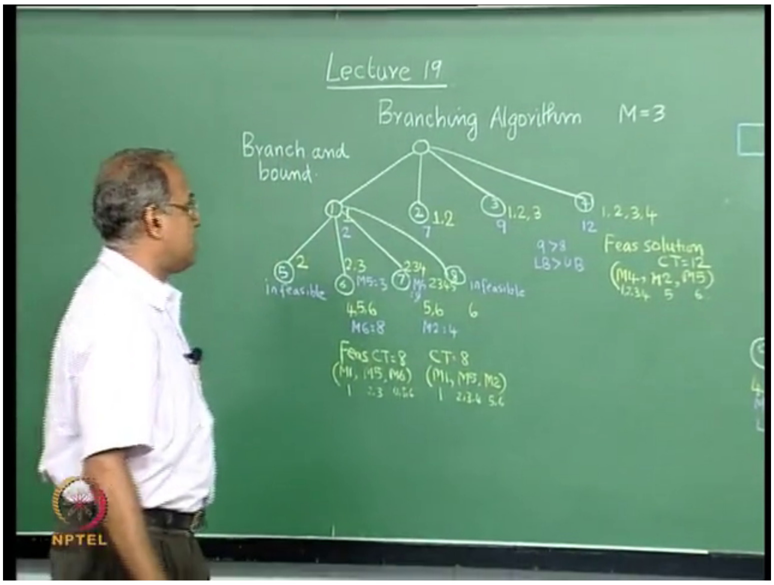
(Refer Slide Time: 01:17)



	1	2	3	4	5	6
1	2			4		4
2		3	1		1	3
3	3			3	2	5
4	3	4	2	3		
5		2	1	5		3
6	4			2	1	5

We will use the same matrix where there are 6 operations that have to be performed there are 6 candidate machines and we want a solution that ensures unidirectional flow. Let us look at a solution that ensures unidirectional flow.

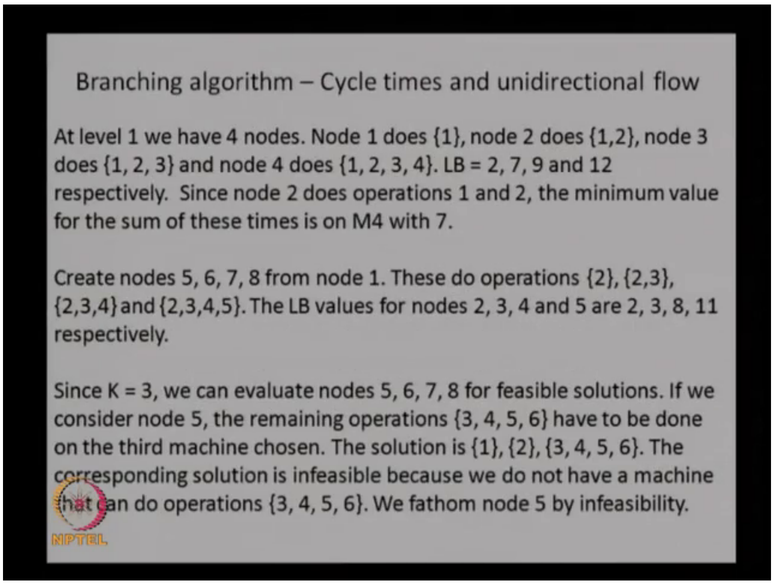
(Refer Slide Time: 01:36)



Considering exactly 3 machines M equal to 3 the earlier integer programming formulation actually gave us a solution with 2 machines, but it did not have any restriction on the cycle time.

Here what we try and do is we try to choose 3 machines. We ensure unidirectional flow, and we also want to find out the minimum cycle time with such an assignment, those are the conditions are the constraints or parameters of the problem since there are 3 machines that we will choose the branching algorithm will have 3 levels each level corresponding to a machine and in each level we allocate operations to the position or to the level.

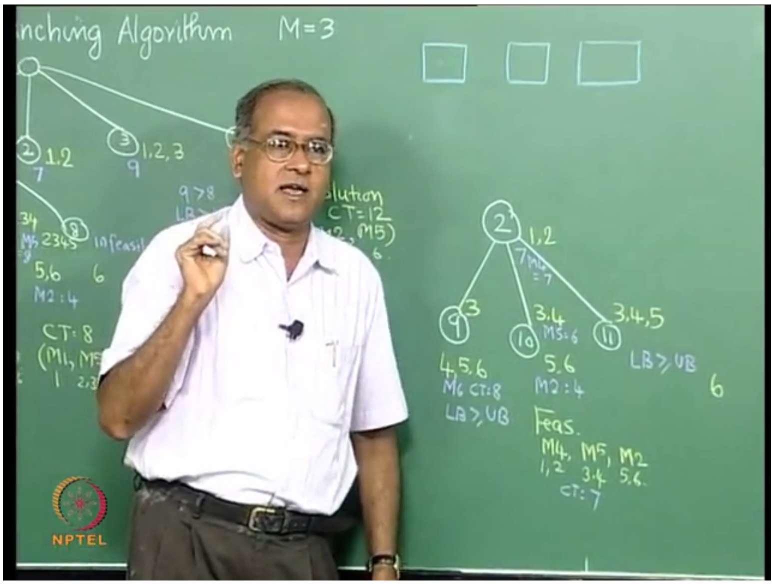
(Refer Slide Time: 02:45)



So, what we will do in the branching algorithm is we would begin with 4 nodes. We will begin with 4 nodes, we will start here we will begin with 4 nodes which we call node 1, node 2, node 3 and node 4, Now node 1 we will assign operation number 1 here, we will assign 1 and 2 here, we will assign 1 2 and 3 and here we will assign 1 2 3 and 4.

Now, since there are 3 machines and 6 operations have to be assigned. Now this is these will represent the set of operations that are assigned at level 1 which means we can do 1 or we can do 1 and 2 or we can do 1 2 and 3 or we can do 1 2 3 and 4 on the first machine it is like there are 3 machines here.

(Refer Slide Time: 04:03)



We stop with 1 2 3 4 here because it is like assigning 1 2 3 four, but the remaining 5 and 6 will have to go to these 2 machines. If we build another node here with 1 2 3 4 5 then this 1 cannot get any operation assigned to it, because 6 will automatically go to this. We would not possibly we do not want to put a fifth note here the fifth node will come much, later now every time we do this we should look at is there a feasible machine that can handle this if there is no feasible machine that can handle this then the node will become infeasible.

For example, let us go back and do that calculation, now we have assigned operation 1 here, there are several machines that can do operation 1. We are with that now we have assigned operations 1 and 2 here we still have 1 machine which can do operations 1 and 2 which is M 4, this can give us a cycle time of 7 because we have M 4 that can do 1 and 2 the time required is 7.

We have still only 1 case of a machine that can do 1 2 and 3 which is again M 4 and the cycle time is 9 here and we have only 1 case of a machine which can do 1 2 3 and 4 with cycle time equal to 12. Now if we assign only operation 1 to the first position then we could use machines 1 or 3 or 4 or 6 and the minimum cycle time possible is 2, now in addition, if we assign 1 2 3 and 4 to position 1 which is like writing 1 2 3 4 here, automatically M 4 has to go here. Now 5 will have to go to this position and 6 will have to go to this position operation 5 will have to come here, operation 6 will have to come here because they are having 3 machines and we have a restriction that every machine should have at least 1 operation and unidirectional flow has to be maintained for this straight away 5 will have to come here and 6 will have to come here now 5 can be done on machines 2 3 and 6 right now, this is an M 4, it we still have feasible machines 2 3 and 6 we can do 6 can be done on 1 2 3 5 6 we are again, if we look at this there will be a cycle time of 12 on this because this will be M 4.

Now, to do operation 5 we could consider M 2 or M 6 both have cycle time of 1. We could consider 1 and either M 2 or M 6 and for operation number 6 we could consider 1 3 5 and 6, we could consider M 5 with minimum cycle time of 3 M 5, we have 1 feasible solution with cycle time equal to 12 we write here 1 2 3 4 feasible solution cycle time equal to 12 M 4 M 2 M 5 operations 1 2 3 4 5 6 with cycle time equal 12 this has been done. Now, we go to this and now we can branch off here further with here operation 1 is given, at this level I can have operation 2 which would mean at the third level I will have 3 4 5 6, I could have 2 3, I could have 2 3 and I could have 2 3 4 and I could have 2 3 4 5. These are all the possibilities, now this would ensure that 6 will come to the third stage.

So, now I go back the moment I have written this there are only 3 stages. This would simply mean that, this is node number 5, node number 6, node number 7 and node number 8. The moment I assign 1 and 2 I have to do 3 4 5 6 to the third, because there are only 3. So, right here the third one gets fixed automatically this would mean a solution where operation 1 goes to the first position operation, 2 goes to the second position which means 3 4 5 6 will have to go to the third position. This would give me another solution with 3 4 5 6 going to the third position.

Now I look at this table and I realize that there is no single machine that can do 3 4 5 and 6. Therefore, this solution is infeasible not at this stage, but at the next stage because next stage would mean 3 4 5 6 go here. Now I look at 1 2 and 3 which means at the third stage I will have 4 5 6 now I go back and see 4 5 6 would force me to use M 3 here I or M 6. This can be M 3 or M 6 since I am trying to minimize cycle times I try to use M 6 here at the third stage with cycle time equal to 2 plus 1 3 plus 5 8 whereas, if I had used M 3 my cycle time would have been 10.

I go back to 2 and 3 2 and 3 I can use M 2 M 4 and M 5, I choose M 5 with minimum cycle time equal to 3 and then I go to 1 I can use 1 3 4 and 6 I have already used 6. I have 1 3 and 4. I use M 1 here. So, the solution will be again feasible solution with cycle time equal to 8 with M 1 coming here with 2 M 1, M 5, M 6 with 1 2 3 4 5 6. I update the solution from a best solution of 12 to a best solution of 8 now, I go back to this 1 2 3 4 would force 5 and 6 here now I go back for 5 and 6 I can use M 2 or M 3 or M 6. I use M 2 with cycle time equal to 4. I do M 2 with cycle time equal to 4 then I go to 2 3 4 and from this table I realize that machines 4 and 5 can carry out 2 3 and 4 machine 4 has a cycle time of 4 plus 2 plus 3 9 while machine 5 has a cycle time of 2 plus 1 3 plus 5 8. I would choose M 5 here to do this is equal to 8 and then I have 1.

Now I have operation 1 can be carried out in 1 3 4 and 6, I can use any of them 1 3 4 and 6. So, I would use M 1 there, again feasible solution with cycle time equal to 8 I choose M 1 in the first stage M 5, in the second stage and M 2 in the third stage, I would assign 1 2 3 4 5 and 6 again I have a solution with cycle time equal to 8, it does not get better. I can either update to this solution or I can leave with this solution both of them have cycle time equal to 8 if we are going to have other considerations then we could choose one of them depending on other considerations like cost for example.

Right now we are not considering cost of machines in this algorithm. The solution is the same up to this point the solution is the same whether we take this or we take this both give us a cycle time of 8 and use 3 machines now we go back to this one there 1 2 3 4 5 which means automatically 6 has to come to this one. Now, I go back 6, I can do it on 1 2 3 5 and 6, but I realize somewhere right here then I have to do 2 3 4 5 on a single machine I realize that I do not have a machine that can do 2 3 4 and 5 on the same machine therefore, this becomes infeasible.

Now, I already have a solution with cycle time equal to 8, now if I branch from this node I have already allocated 1 2 and 3 operations at the first stage and I have got this 9 from the table which says M 4 is the only machine that can do operations 1 2 and 3 with minimum cycle time or only cycle time equal to 9, if we proceed from here and if we assign the remaining operations 4 5 and 6 to 2 different machines, the cycle time can still be 9 or more it cannot become less than 9 because I already have in this stage the minimum possible cycle time as 9, proceeding from this particular node cannot give us a solution with cycle time 7 we already have feasible solutions with cycle time 8 now the moment we have feasible solutions with cycle time 8 we would now like to explore whether the cycle time can actually reduce further, now this already has a minimum cycle time of 9 and proceeding from here cannot reduce the cycle time further therefore, we fathom this node by considering that this 9 is greater than 8.

This 9 represents a kind of a lower bound is greater than upper bound it is a minimization problem every feasible solution is an upper bound. When this lower bound is greater than upper bound we fathom, now the only note that remains here is 1 and 2 from which we have to branch. Let me just write this 1 and 2 separately and branch from there. That it does not get very congested here. We would write this is node number 2 that I have here and node number 2 has operations 1 and 2 given here with a minimum cycle time of 7.

Now this 1 and 2 with minimum cycle time of 7 came by using M 4, M 4 is the only machine which can do 1 and 2 together and give us a cycle time of 7. Now from here we have to branch we have to branch to the next stage where I assign machine operation 3 here. So, I call this node number 9 I call this node number 9 with operation 3 here and I assign node number 10 with 3 4 and assign node number 11. This will have 3 4 and this will have 3 4 5.

Now, we look at each one of this now this I put 1 2 3, the third machine should have 4 5 and 6, now I go back to this table and I realize that M 3 is the only one that M 3 and M 6 can do 4 5 and 6, M 3 takes time equal to 10, M 6 takes time equal to 8. I choose M 6 with cycle time equal to 8 and I can actually stop here because I do not have to go back and allocate a machine here and a machine here because I already have cycle time equal to 8 here.

Then I do the machine allocation and compute the 3 individual cycle times and choose the maximum of the 3 that can only be 8 or more, the lower bound here is 8 and I already have feasible solutions with 8 and using the same principle that lower bound is greater than or equal to upper bound I fathom this node, because if I proceed I can only get a solution with 8 or more. Now, I look at node 10 this is going to put operations 5 and 6 on the third machine I go back to this table and I realize that operations 5 and 6 can be carried out on M 2, M 3 and M 6 at present I choose M 2. I choose M 2 with cycle time equal to 4 M 2 with cycle time equal to 4.

Now, I go back to operations 3 and 4 operations 3 and 4 can be done by M 4 and M 5. Now already M 4 and M 5, I go back I assign M 4 has a cycle time of 5, M 5 has a cycle time of 6, but I tentatively assigned M 4 with minimum cycle time of 5, but then I go back to 1 and 2 and I realize M 4 is the only one that I can do with 1 and 2. Therefore, this 7 is going to absorb M 4. This 7 is going to absorb M 4 here this would force me not to use M 4 here, but to use M 5 here and M 5 for operations 3 and 4 has a cycle time equal to 6 M 4 has cycle time equal to 7. This gives me another feasible solution another feasible solution with M 4, M 5 and M 2 which will do operations 1 2 3 4 5 6 with cycle time equal to 7 now cycle time equal to 7 is smaller than cycle time equal to 8. So, the solution gets updated.

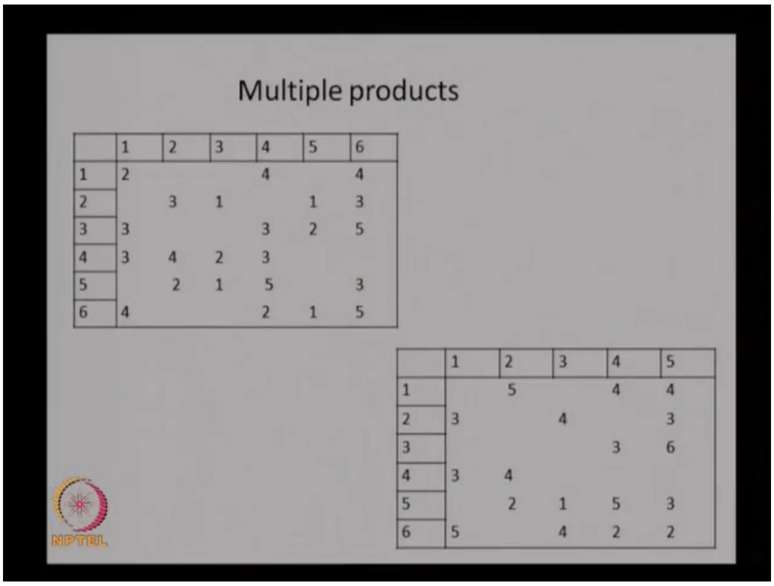
Now, the moment the solution gets updated I do not have to do this because this already has a cycle time of 7,even I f I proceed here and if I find a feasible solution it is cycle time cannot be less than 7 it can only be 7 or more, here I do not proceed I say lower bound greater than or equal to upper bound because upper bound is 7 now feasible solution is an upper bound from here already I know it is 7 I cannot get something less than 7 so I stop if I do not follow this logic of lower bound less than or greater than equal to upper bound and fathoming the node I will go back and see this is going to force me to use 6 here and then I go back and find out is there a machine that can do 3 4 and 5 together there is no such machine.

This would have been fathom by infeasibility after some computational effort and search, now right here I can fathom this. So, there is no node the best solution is the one that has cycled M equal to 7 with 3 machines M 4, M 5 and M 2 with operations one 2 3 4 and 5 6 done. This algorithm is actually a branch and bound algorithm; because we have used the idea that when the lower bound is greater than or equal to upper bound I can fathom the node. A node gets fathomed by infeasibility by a feasible solution and by and when lower bound is greater than equal to upper bound there is no more nodes available to proceed the algorithm terminates.

This branch and bound algorithm is very useful to solve this kind of a problem we can have additional restrictions brought in like budget restrictions on the machines and on it can be done the only place where we have to be a little careful is that at earlier places when here we have only 1 machine, we wrote 7 somewhere we could have alternatives say at this stage when we had alternatives we chose the minimum cycle time as a lower bound, but we did not assign a machine there we should not do that like what happened here if we had assigned 4 M 4 here then we will not have anything else to do here it might throw up an infeasibility which is not correct. Only when we reach this stage we have to go back and then we have to make sure that do we have enough machines which can provide a feasible solution. This portion alone makes the branch and bound a little more involved with more search and so on, very technically if we look at this problem this itself is a machine selection problem which is a sub problem then we have alternate machines. The sub problem of machine selection right now we used it a heuristically by picking the minimum cycle time because this problem instance is a small instance.

For large problem instances we should have we should be solving a small knapsack problem inside which will try and give us the best selection of machines. Still we can use this branch and bound algorithm provided we understand all these aspects of the branch and bound algorithm. The optimum solution considering 3 positions and 3 machines and unidirectional flow is given by this solution with 1 2 3 4 and 5 6 getting allocated with cycle time equal to 7.

(Refer Slide Time: 26:39)

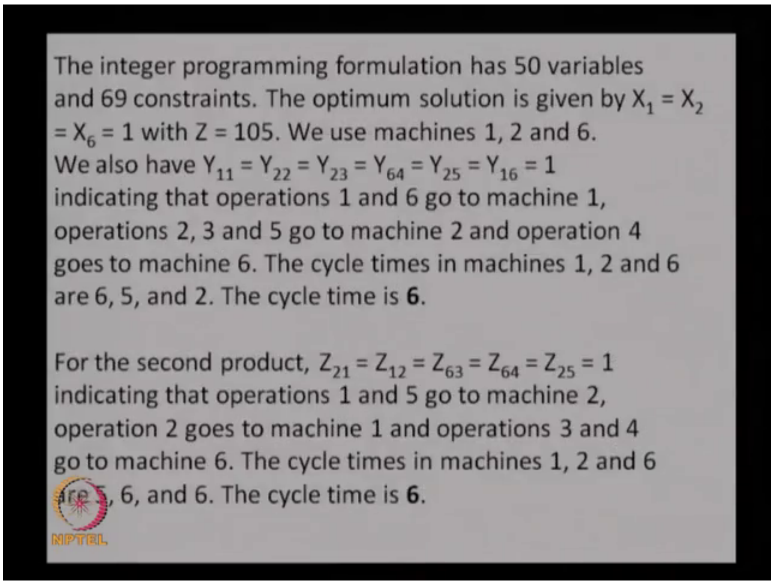


	1	2	3	4	5	6
1	2			4		4
2		3	1		1	3
3	3			3	2	5
4	3	4	2	3		
5		2	1	5		3
6	4			2	1	5

	1	2	3	4	5
1		5		4	4
2	3		4		3
3				3	6
4	3	4			
5		2	1	5	3
6	5		4	2	2

Now the next model that we will see is when we have multiple products, far we have seen that the cell makes only one product now what happens when the cell makes more than 1 products, we choose or use these 2 matrices.

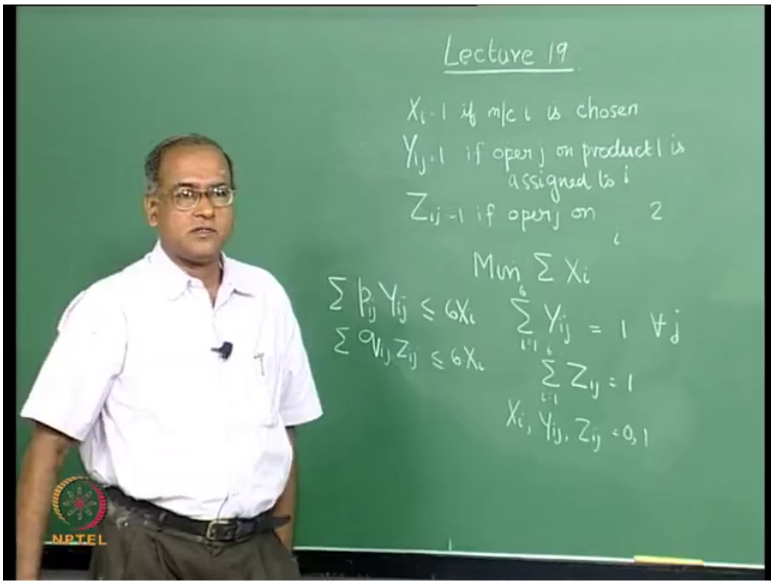
(Refer Slide Time: 26:54)



To represent multiple products from this we can say that there are 2 products each product has first product has 6 operations second product has 5 operations the cycle times are given there now we want to choose minimum number of machines such that we assign each of these operations to these machines.

And we have a cycle time restriction on this the formulation is a logical extension of one of the earlier formulations where we had a restriction on the cycle time. And we minimize the number of machines, we now have a second product that also comes in Xi equal to 1 if machine I is chosen Yi j equal to 1 if operation j on product 1 is assigned to machine I Zi j equal to 1 if operation j on product 2 is assigned to machine I.

(Refer Slide Time: 27:57)

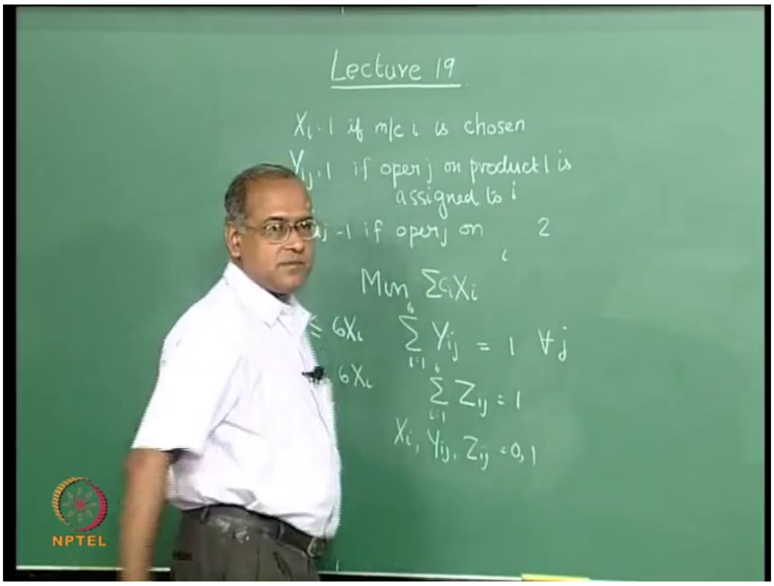


The objective function will try to minimize the number of machines considering the first product each operation of the first product should be assigned to only 1 machine for every j considering the first product each operation is assigned to only 1 machine, considering the second product each operation is again assigned to only 1 machine. Now for the first product Pi j Yi j sigma is less than or equal to 6 times Xi, if we take the first product and if we take for example, the operations assigned to machine 3 then 3 Y 3 1 plus 3 Y 3 4 plus 2 Y 3 5 plus 5 Y 3 6 should be less than or equal to 6 X 3 only when X 3 is chosen the 6 units of time is available and whatever is assigned should be less than or equal to 6.

In a similar manner sigma q_{ij} z_{ij} is again less than or equal to 6 Xi we are not saying that put together the cycle time is 6 cycle time is 6 for each product both the products are not going to be produced at the same time in the cell at the same time the cell is going to produce only one of the products and whichever one it is making the cycle time should be less than or equal to 6. This will again ensure that if we take the second product and if we are looking at machine number 5 then we will have a constraint which is 2 Y 5 2 plus 1 2 Z 5 2 plus 1 Z 5 3 plus 5 Z 5 4 plus 3 Z 5 5 is less than or equal to 6.

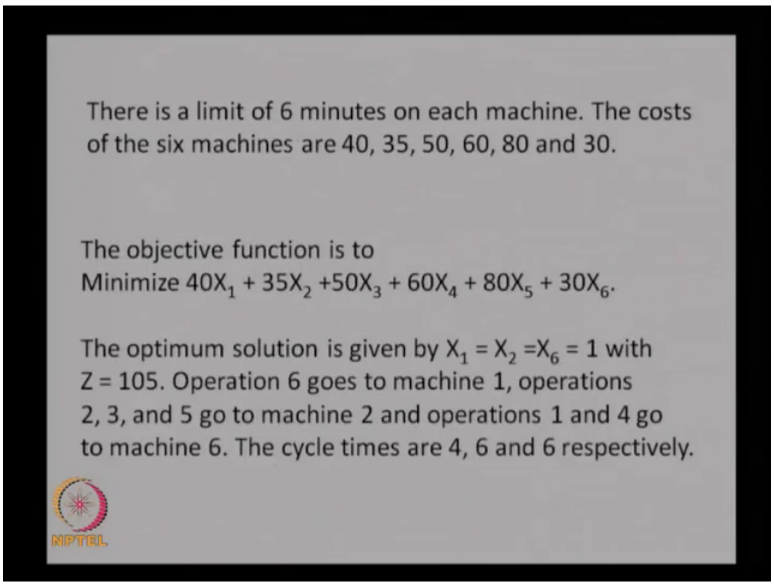
This will again ensure that the same thing happens then we have X_i Y_{ij} Z_{ij} equal to binary 0 1, now this will be the formulation considering multiple products minimizing the number of machines and subject to a cycle time restriction. If we finally, look at the solution which has oh we also try to minimize the cost in this case. So, the objective is not to minimize sigma Xi, but the objective is to minimize CiXi the cost of the machines are given in the earlier as in the earlier model.

(Refer Slide Time: 32:15)



We finally, look at X 1, X 2 and X 6. Let us go back, X 1, X 2 and X 6.

(Refer Slide Time: 32:29)

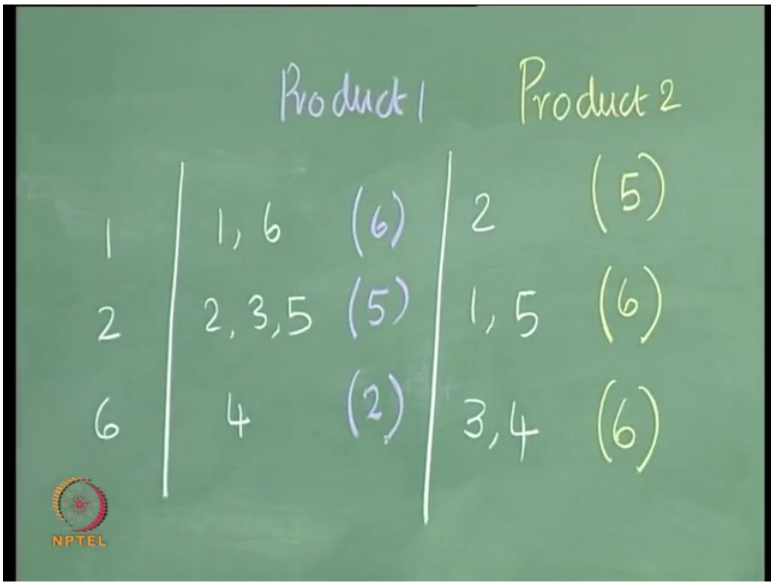


X 1, X 2 and X 6 will have a cost of 40 plus 35, 75 plus 30 100 and 5.

100 and 5, we use machines 1 2 and 6 are chosen machines 1 2 and 6 are chosen, now for the first product we have said that 1 and 6 to 3 and 5 and 4. 1 and 6 2 3 and 5 and 4 and

for the second product 1 and 5 go to machine 2 operation 2 goes to machine 1 3 4 and 3 4 goes here.

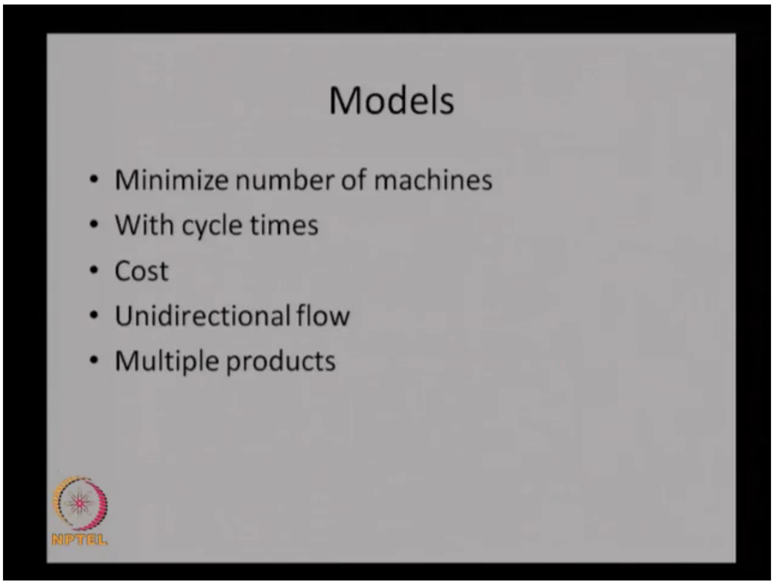
(Refer Slide Time: 32:54)



The cycle times will be 1 and 6 would give us a cycle time of on machine 1 operations one and 6 will give a cycle time of 6 2 3 and 5 will give a cycle time of 5 4 on machine number 6 would give a cycle time of 2 for the first product. All cycle times are within 6 cycle times here for the second product machine 1 operation 2 has a cycle time of 5 machine 2 operations 1 and 5 have cycle time of 6 machine 6 with 3 and 4 also have cycle time of 6. So, machines 1 2 and 6 are chosen, this is product 1 and this is product 2.

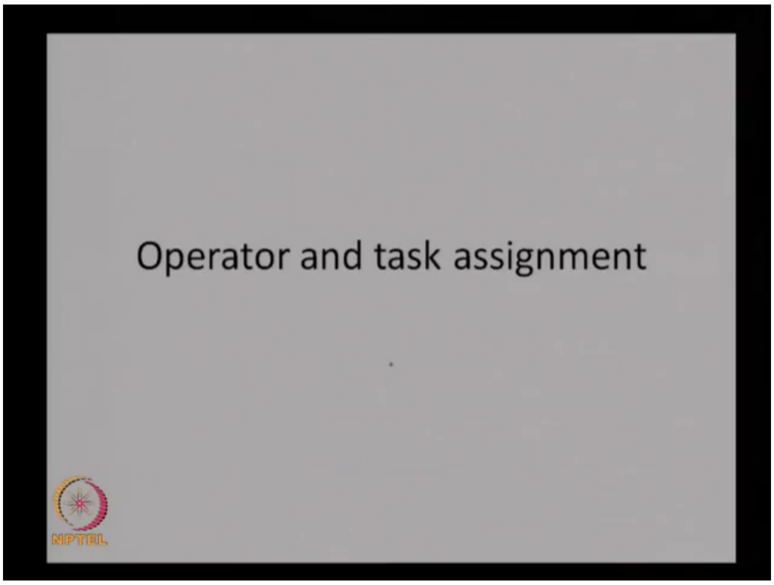
Now, the only thing in this solution is we have not considered unidirectional flow. We could consider unidirectional flow and build similar models we have we have already seen a model with unidirectional flow now we should look at another model for Z as well including along with Y we should also have a Z then we will have additional constraints and solve this problem considering unidirectional flow in this. Similarly we can also do a branching algorithm for multiple products the way we have demonstrated this. These are some of the models that we have seen which would look at product based cellular manufacturing.

(Refer Slide Time: 35:26)



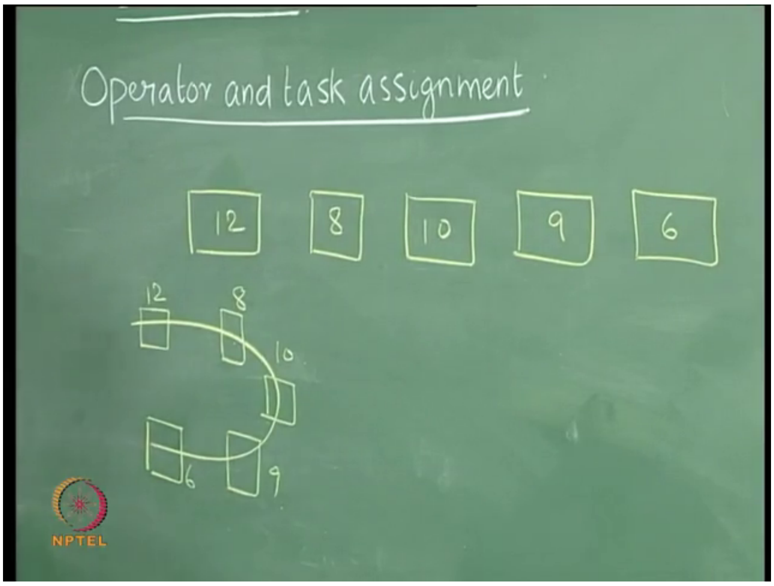
To sum it up we have seen several models all of these are shown here the minimize number of machines to create a cell that can carry out all the operations with cycle time restrictions with cost restrictions with unidirectional flow and considering multiple products several combinations of these can also lead to intermediate models, but all these can be solved optimally we have right now shown some integer programming formulations for these we have shown 1 branch and bound algorithm for these we have not looked at heuristic solutions for these several heuristic solutions are also available which can be used to get solutions very quickly.

(Refer Slide Time: 36:24)



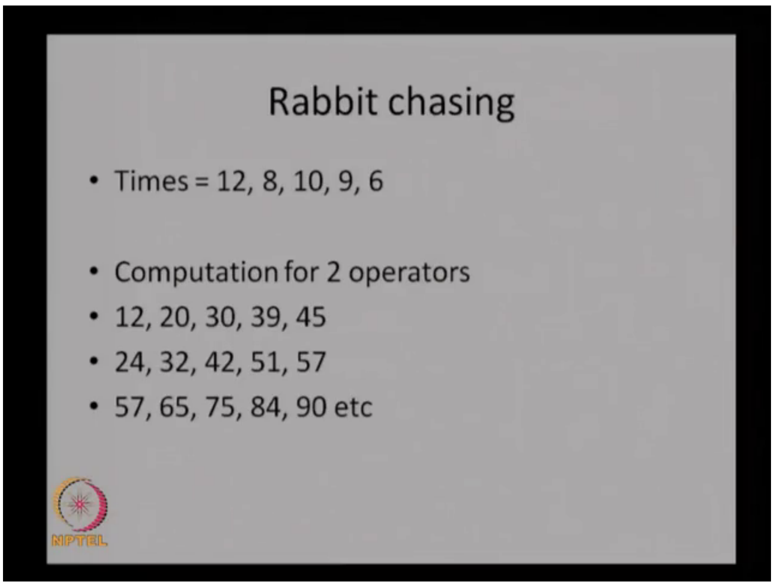
Now, we look at one more aspect of cellular manufacturing which is called operator and task assignment. Let us go back to operator and task assignment.

(Refer Slide Time: 36:48)



So, far we have seen assigning machines to cells and parts to cells assigning machines to cells created machine cells, signing part to cells created part families, now we are going to see how we assign operators to cells. Now that comes under the topic called operator and task assignment.

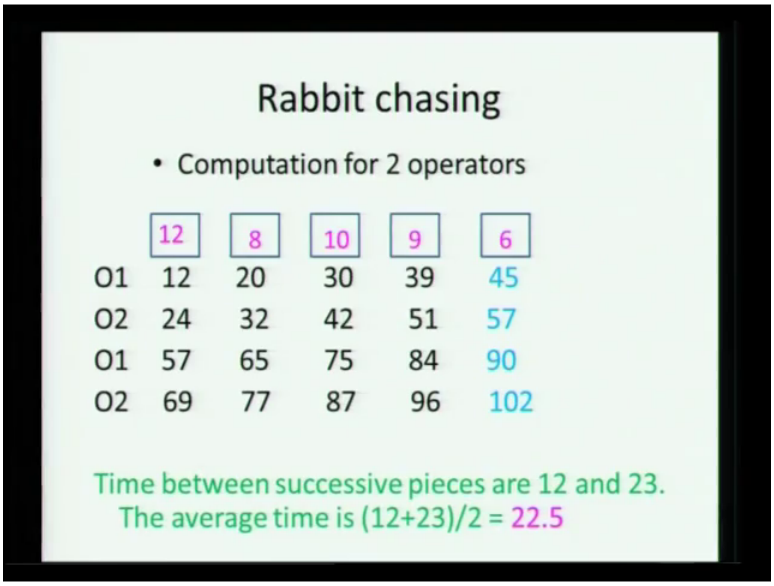
(Refer Slide Time: 37:25)



First thing that we do is what is called rabbit chasing. Let us assume that there is a cell now most of these come for product based cells they can also be extended to the regular cells. Let us say there is a cell that makes 1 product there are 5 machines and the task times on these 5 machines are task time on these 5 machines are 12 8 10 9 and 6. These need not be 5 operations whose times at 12 8 10 9 6 this can even be the outcome of any of these optimization problems or the branching algorithm where multiple operations are carried out on a machine, but the total of the operation times are 12 8 10 9 and 6. Now let us look at what happens when now there is this is one kind of layout, another kind of layout that we can think of is that is like this, this is 12 this is 8 this is 10 this is 9 and this is 6 these are the 5 machines.

Now, what do we do by rabbit chasing now suppose we have only 2 operators and this first operator comes takes a piece from here does the first operation then goes to the second, goes to the third, fourth, fifth and comes out meanwhile. Second operator follows the first operator and continues. Now once the first operator has finished this operator will leave this part here and then he or she will take another part and do this operation, we will now trace the output of this cell when we have this happening.

(Refer Slide Time: 39:46)



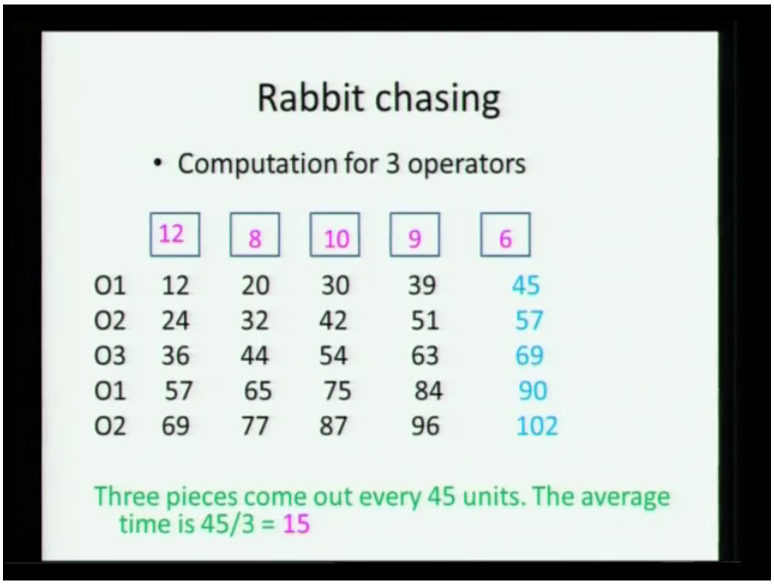
The first operator starts at time equal to 0 and finishes at time equal to 12 which is shown here then the operator moves to the second machine and finishes at 20 which is 12 plus 8 and then moves to the third machine finishes at 30 20 plus 10 is 30 fourth machine finishes at 39 30 plus 9 is 39 and then takes 6 time units to finish at 45. Meanwhile the second operator would start the work. At time equal to 12 the first machine is free the second operator is also free. So, the second operator starts work at time equal to 12 finishes at 24, but by 20 the second machine is free because the first operator has already moved. So, second operator starts at 24 and finishes at 32.

Again the machine number 3 is free at 30 itself. Second operator starts at 32 and finishes at 42 again machine number 4 is free at 39 it itself. So, he starts at 42 and finishes at 51 and finally, finishes at 57.For this calculation we need to take the time which is the maximum of these 2 and add the corresponding time of 9 to get this 51. The second operator finishes a piece at time equal to 57 and the second piece comes out at time equal to 57.

Now the first operator is once again available to come to M 1 at time equal to 45. Again starts at 45 and finishes at 57 by which time M 2 is free and the whole process continues. So, operator 1 again finishes at time equal to 57 65 65 plus 10 75 75 plus 9 84 and 84 plus 69 and the third piece comes out at time equal to 90, operator 2 had finished at 57 and comes back to the starting point now machine number 1 is also free at time equal to 57, operator 1 starts at 57 finishes at 69 and from 69 finishes the second machine at 77 87 96 and 102.

We observe that operator 1 has completed pieces at 45 90 again 135 and so, on while operator 2 will complete pieces at 57 102 147 and so on, time between successive pieces will be 12 and 23 respectively as we see from this table and therefore, the average output or the average time taken is 12 plus 23 by 2 which is 45 by 2 which is 22.5. So, a simpler way of saying is the total is 45 there are 2 operators. So, on an average a piece will come out at time 45 by 2 equal to 22.5 time units.

(Refer Slide Time: 43:17)



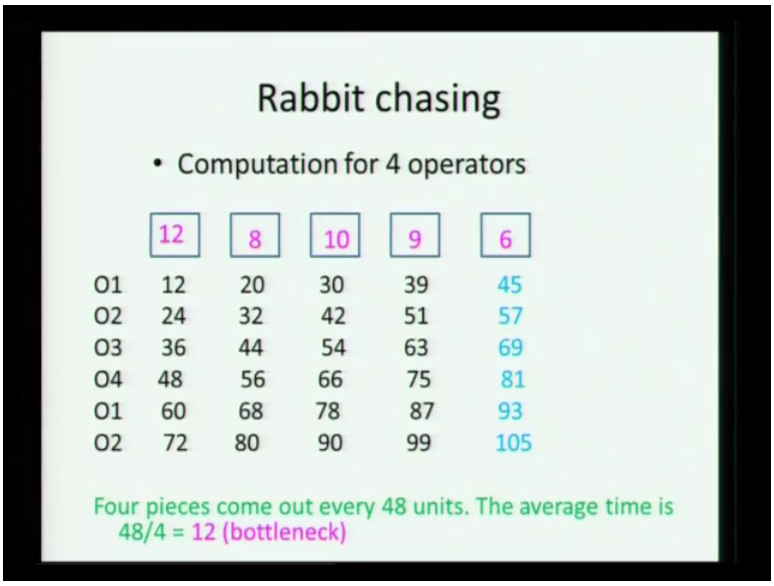
We now see the same computation, but with 3 operators. So, as I mentioned the first 2 operators would finish at time equal to 45 and 57 respectively now the third operator starts the third operator can start at time equal to 24 because M 1 is free at time equal to 24, the third operator starts at 24 and finishes the work at 36 by which time M 2 is free. So, starts at 36 and finishes at 44 once again M 3 is free at 42. This person can start at 44 take 10 more time units finish at 54 similarly finish here at 63 and finish the piece at time equal to 69.

Now, operator one starts again at time equal to 45 operator 1 is free, operator one comes back starts at time equal to 45 because this machine is free at 36. So, operator 1 can start at time equal to 45 and finishes at 57 again this machine is free at 44 therefore, operator 1 starts at 57 and finishes at 65 and on finishes this at 75 84 and 90. Now operator 2 had finished the first piece at 57 now operator 1 has finished the second piece on M 1 at 57. Operator to can now start at 57 of course, we assume walking times as negligible in all these computations otherwise it will go up a little bit at the moment for illustration we assume that walking times are negligible.

The person starts at 57 and finishes at 69 here. Again M 2 is free at 65. Operator 2 can start at 69 and finish at 77 again M 3 is free at 75. The person starts at 77 and finishes at 87. Similarly finishes at 96 and again finishes at 102. Now from this calculation we can understand that 3 pieces actually come out in every 45 minutes because 45 and o 1 second piece comes out at 90 similarly 57 102.

So, 3 pieces will come out every 45 time units the average time per piece is 45 by 3 which is 15 time units. The simpler way to say this is total is 45 there are 3 operators. The average time is 45 by 3 which is 15.

(Refer Slide Time: 46:08)



Let me show the computation for 4 operators now. The first 3 operators will perform as shown in the previous slide that is up to this 69. Up to this 69 the 3 operators will perform now let us look at the fourth operator the fourth operator starts the first 1 at time equal to 36 finishes at 48 by which time M 2 is free. Fourth operator finishes at 36 by which time M 3 is free forth operator finishes at 66 again M 4 is free. Fourth operator finishes at 66 plus 9 75 by which time M 5 is free. Fourth operator starts at 75 and finishes at 81.

Now operator 1 comes back to the starting point now operator 1 is free at 45, but the machine is free at 48 only. Therefore, operator 1 comes back for the second piece and can start only at 48 which is larger of this 48 and this 45 starts at 48 and ends at 60 by which time M 2 is free therefore, this is over at 68 again this is free at 66 therefore, this will be over at 68 plus 10 78 again this is free at 75. Therefore, this will be over at 78 plus 98 7 again this is free at 81. Therefore, this is 87 plus 6 which is 93.

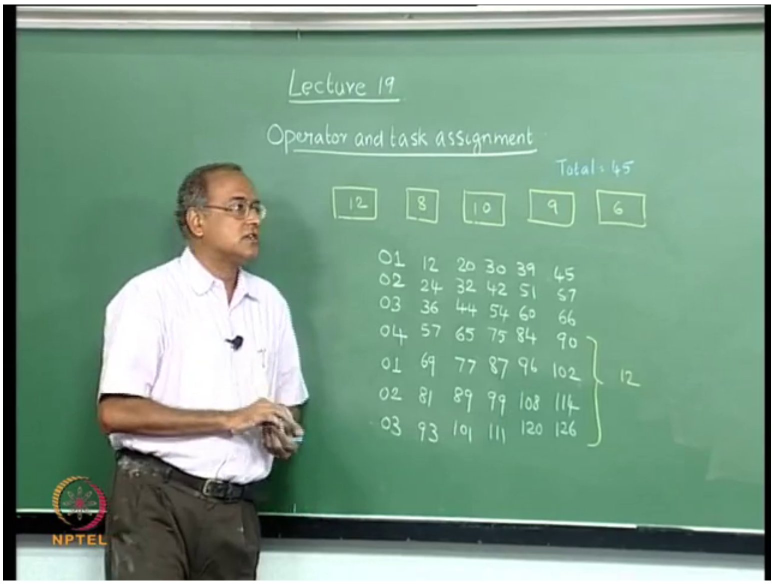
Now, operator 2 comes back to the starting point operator 2 is available at 57, but the machine is available only at 60 therefore, operator to can start only at 60 and finish at 72. So, the completion times for operator 2 will be 72 80 99 and 105. Now we observe that 4 pieces come out for every 48 units that is 45 and 93 the difference is 48 units. 4 pieces will come out at steady state every 48 units the average time is 48 by 4 which is 12 if we go back and make this comparison in the earlier examples of 2 operators we said total is 45 there are 2 operators. So, average is 22.5 here total is 45 3 operators average is 15, but here total is 45 4 operators the average is not 45 by 4, but it is 48 by 4 which is 1. So, what has happened what has happened in this case is when we had 2 operators the average is 22.5 which is more than the maximum of these 5 individual processing times 12 therefore, the average stays at 22.5.

Here the average is 15 which is more than this 12, but when we have 4 operators the average is 11.2, 5 if I take 45 by 4 it is 11.2, 5 which is smaller than 12 now 12 is the maximum among the unit processing times and is called the bottleneck processing time. When the average total time by the number of operators in this case 11.2, 5 becomes smaller than 12 the bottleneck will start dominating and the output will only be every 12 time units and that has happened because if we observe carefully operator 1 is completes the work at 45, but can take up the work only at 48.

So, there is that 3 time unit that gets carried. So, 45 plus 3 will become 48 and 48 divided by 4 will give us 12. So, when the number of operators is such that the total processing time divided by the number of operators becomes less than the bottleneck time the output will only be that of the bottleneck time for 3 operators the total by number was 15 which is more than 12. The output was a piece every 15 time units with 4 operators it is 45 by 12 45 by 4 is less than 12. 12 dominates now at steady state the output will be 12 time units for every piece.

If we do this computation for 5 operators again we can show that the output will be only 12 time units which is the bottleneck time therefore, when we do this rabbit chasing we should first find out the total and we should find out the number of operators then the total by number of operators is more than the bottleneck time the time per piece will be total processing time by number of operators when total processing time by number of operators is less than the bottleneck then the output will be the bottleneck.

(Refer Slide Time: 51:35)



This is the first model that we have seen which uses the idea of rabbit chasing and tells us how we compute the cycle time when operators follow each other subsequent models. We will see in the next lecture.