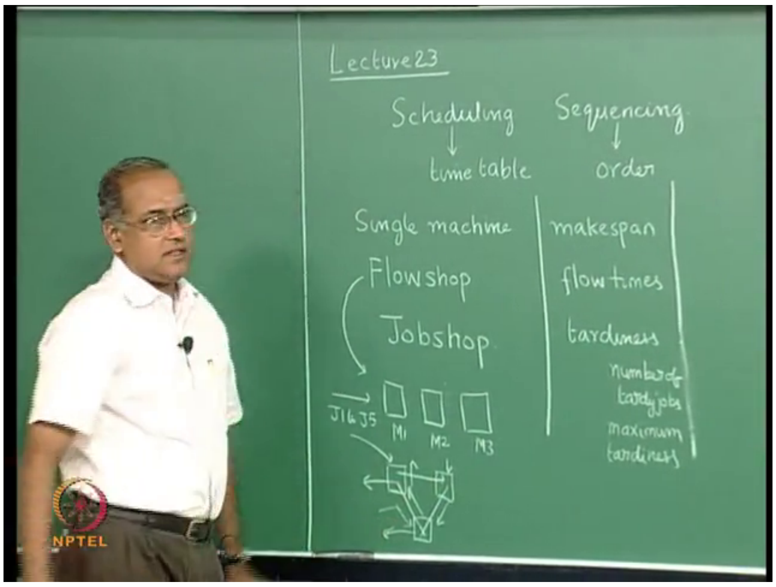
Manufacturing Systems Management

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

Lecture - 23
Cell scheduling and sequencing

In this lecture we continue the discussion on Cell Scheduling and Sequencing. Scheduling and sequencing is extremely important when a set of jobs have to be processed on a set of machines.

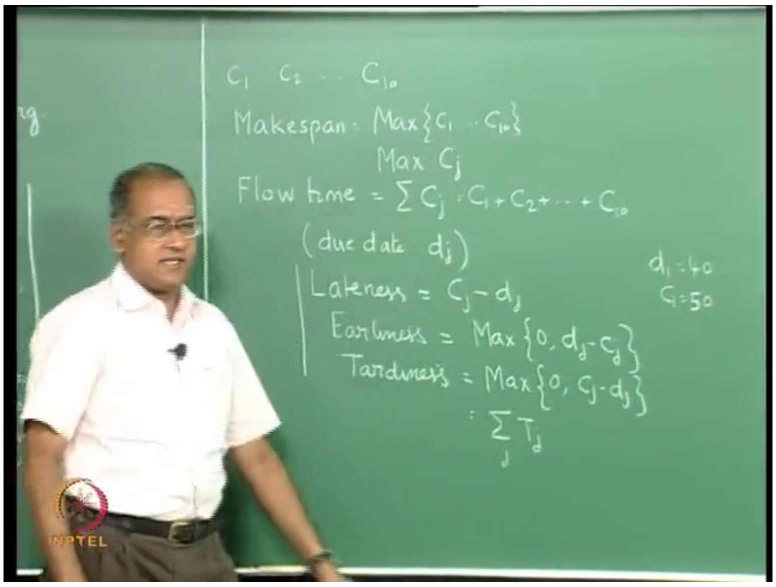
(Refer Slide Time: 00:19)



While sequencing tells us the order in which these jobs have to be processed on the machines. Scheduling gives us a timetable which goes into the details of at what time each job gets into each machine scheduling and sequencing is a very well researched area it is also used extensively almost every day in every manufacturing organization. Traditional literature on scheduling and sequencing considers scheduling and sequencing under what is called single machine problems flow shop scheduling and what is called job shop scheduling. In the traditional scheduling and sequencing literature we also use several objectives such as makespan, flow times, tardiness and many more.

Given a certain number of jobs that have to be processed on a set of machines the time at which all the jobs are completed so that all of them together can leave the shop is called the makespan. The makespan is essentially the span of time required to make all the parts in the shop. So, if you are given a set of jobs makespan is the time at which all jobs are completed. Flow times are essentially they represent the time at which these jobs the times these jobs spend in the system. So, if there are some 10 jobs and all the jobs are available at the beginning which means at time equal to 0 and each of these jobs is completed at time C 1 C 2 etcetera C 10 if they are completed at this time then the makespan is the time at which all of them are completed, so makespan is the maximum of the completion times C 1 to C 10 or in general the maximum of C j where C j is the time of completion of job j.

(Refer Slide Time: 03:17)


$C_1 \quad C_2 \cdots C_{10}$
Makespan = $\mathrm{Max}\{C_1 \cdots C_{10}\}$
$\mathrm{Max}\ C_j$
Flow time = $\sum C_j = C_1 + C_2 + \cdots + C_{10}$
(due date d_j)
Lateness = $C_j - d_j$
Earliness = $\mathrm{Max}\{0, d_j - C_j\}$
Tardiness = $\mathrm{Max}\{0, C_j - d_j\}$
$= \sum_j T_j$
$d_1 = 40$
$C_1 = 50$

Flow time is the time at which times for which the parts or products stay in the system. So, if all of them have come in at time equal to 0 and if there are 10 jobs or parts processed in the system then the first 1 stays for a period C 1 between 0 and C 1 the second 1 stays for a period C 2 between 0 and C 2 and so on and the flow time will be sigma C j equal to C 1 plus C 2 plus C 10.

The third objective that is often used is called tardiness sometimes it may not be possible to deliver all the products in time. So, each product or part has what is called a due date which is given by d j for job J. If the time at which the job is completed exceeds the due date then the job is late if the time at which the job is completed is before the due date the job is early normally we represent early finish as early and late finish as late, but here what we do is we define a generic term called lateness. So, lateness will be equal to C j minus d j if C j is a time of completion of job J and d j is the due date the lateness is given by C j minus d j. So, when the job is completed ahead of the due date C j minus d j will be negative and this is called earliness. Earliness is when the job is completed before the due date and therefore, earliness will be equal to maximum of.

If the job is completed before the due date then d j minus C j is positive and that represents the extent of earliness, if the job is completed after the due date then d j minus C j will be negative and this one is not early, so the earliness will be 0. Now, when the job is completed after the due date then the job is called tardy and tardiness is equal to maximum of 0 and C j minus d j. So, if the job is early then the job is completed ahead or before the due date.

So, it has an earliness of d j minus C j. For example, if the due date is 40 and if the completion time is 30 then the earliness is 10. So, earliness is maximum of 0 and 40 minus 30 10, so earliness is 10 and tardiness is maximum of 0 and minus 10, so tardiness is 0. If it is completed at time equal to 50 then it is late or tardy. Now, as far as earliness is concerned it is maximum of 0 comma minus 10. So, earliness is 0. So, it is not early tardiness is maximum of 0 and 10. So, tardiness will be 10.

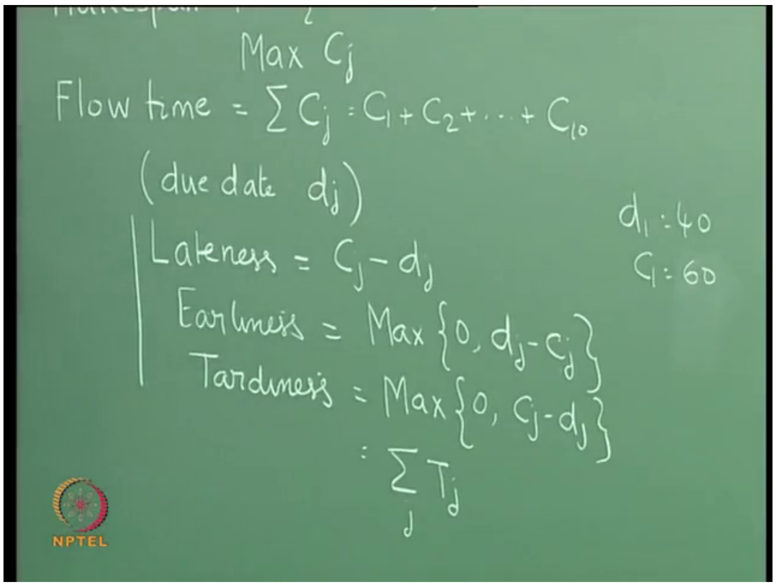
So, these are 3 normal objectives in scheduling and sequencing. So, we minimize the makespan we minimize the sum of flow times which is given by sigma C j and we minimize the sum of tardiness across all jobs. So, this will be sigma summed over J, T j sum of tardiness. Sometimes we also have other objectives such as number of tardy jobs, number of tardy jobs and maximum tardiness to be minimized maximum tardiness is also to be minimized.

Now, these shops the sequencing and scheduling problems from conventional literature address these objectives in the context of single machine scheduling, flow shop scheduling, job shop scheduling and other types of jobs assembly scheduling open shop scheduling and so on. A single machine problem is one where there is only one machine and jobs come get work done on that machine an example would be a single press that is available in a shop or a single heat treatment facility that is available in the shop.

A flow shop is a shop or an ors or a manufacturing system where the machines are arranged in a particular order and all the jobs pass through all the machines in the same order. So, if there are 3 machines we call them as M 1 M 2 and M 3 and if there are 5 jobs J 1 to J 5 all the jobs J 1 2 J 5 will visit M 1 first followed by visiting M 2 and then they will visit M 3. So, the question is what is the order in which these jobs have to be sequenced such that, any one of these objectives or given objective is minimized. In a job shop we have these machines or departments differently and each job will have a unique order of wizard and so on. For example, job 1 may go to M 1 and then M 2 and then M 3 maybe come back to M 1 and exit, another job may to start with M 2 go to M 3 and then exit, a third job may start with M 3 go to M 1 and exit and so a job shop is where each job has a pre specified visit or a pre specified order of visit of the machines. All the jobs need not visit all the machines a job can skip a machine there can even be revisits a job that visit M 1 can go back to M 2 and again come back to M 1. So, all these are allowed in the context of job shop scheduling.

Now, we also want to see the relevance of some of these in the context of cellular manufacturing and also in the context of other manufacturing methodologies. In order to do that let us visit these 3 terms again which is lateness, earliness and tardiness. In a conventional manufacturing system each job has a certain due date d j and a completion time C j. Now, we have defined earliness as the quantity or the amount of time at which the job finishes ahead of its due date. For example, if the due date is 40 and the completion time is 30 we will now say that the earliness is 10.

(Refer Slide Time: 13:03)



Max C_j

Flow time $= \sum C_j = C_1 + C_2 + \dots + C_{10}$

(due date d_j)

Lateness $= C_j - d_j$

Earliness $= \text{Max}\{0, d_j - C_j\}$

Tardiness $= \text{Max}\{0, C_j - d_j\}$

$= \sum_j T_j$

$d_1 : 40$

$C_1 : 60$

Now, if the due date is 40 and the completion time is say 60 then it is tardy with tardiness equal to 20. Tardiness means we are unable to finish the job in time and send it to the customer so the customer would suffer, if the customer is a person external to the organization then several other plans and schedules of the customer will get affected. If the customer is internal to the organization which means from one shop the job is moving to another shop even within the organization many other this can be can get affected if the job finishes later than the due date. So, tardiness is an important objective and we want to bring down tardiness to 0 or to minimize tardiness.

Now, let us look at earliness. Now, earliness is a situation where the job is completed ahead of the schedule, so due date maybe 40 completion time maybe 30, so earliness is 10. Early when the job is completed early we can do either of these 2 things one is to dispatch this job immediately to the customer or to the next stage in the system. So, that certain amount of scheduling it can help scheduling in the next stage. Alternately the finished job can remain in the shop till the due date is reached after which it can be shipped because the customer would expect this to be shipped at d 1 equal to 40 whether the customer is an internal customer or an external customer the customer would like this to be shipped at d equal to 40.

Now, today let us take the case where the job is completed early and the customer is not willing to accept it at time equal to 30, but wants to take it at time equal to 40 the reason is the customer might have scheduled some activity for this at time equal to 40. So, if this job is finished at time equal to 30 and we wish to send it to the customer the customer is not ready to take it because this would add 10 minutes inventory on the customer side. Now, sometimes it can be 10 minutes inventory, sometimes it can be 10 hours inventory, sometimes it could be 10 days of inventory. So, the customer is not interested in increasing the inventory at his or her end.

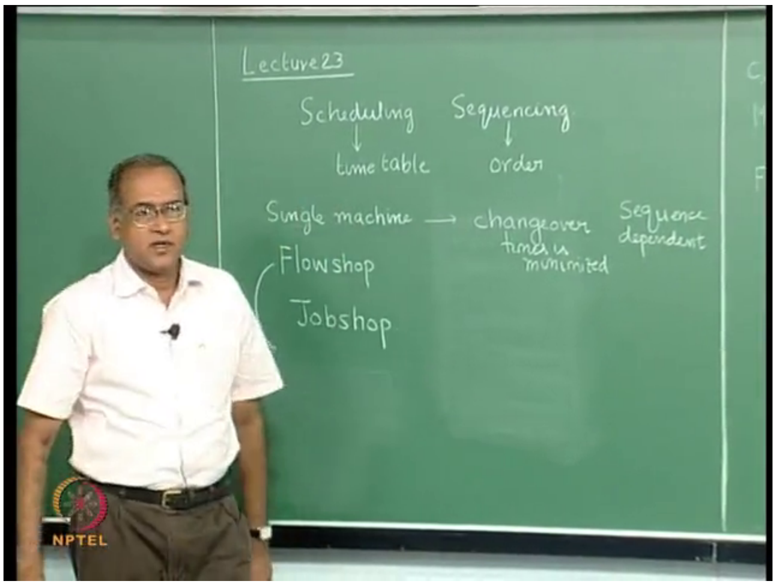
From the beginning of this lecture series we have understood and we have stressed the importance of reducing the inventory. So, when the job is finished early the customer is not willing to take it immediately because the customers inventory will increase therefore, this has to be with the factory for an additional 10 time units which will only add to the inventory cost of the fact. So, in the present context of manufacturing it becomes important that we do not have too much of earliness also because earliness would add to the inventory cost in the organization. So, sequencing and scheduling sometimes tries to minimize both earliness and tardiness or a weighted combination of earliness and tardiness understanding fully well that even though earliness is also not desirable tardiness is still more undesirable and therefore, tardiness will have a higher weightage than earliness. But if we want to take a closer look at these objectives in the context of newer manufacturing methodologies then earliness becomes an important addition to the objective.

We also have to understand that when we write these objectives particularly these 3. Now, these two are called internal objectives, objectives that are internal to the organization. So, if I am a shop producing something makespan is the time at which I can complete everything for the day. Flow time in some sense represents the amount of time spent by each part inside the factory and therefore, represents the extent of work in progress inventory that I have. So, minimizing makespan would mean that I could minimize the time at which all the jobs are completed so that all the jobs can be dispatched and the next set of jobs can be taken up.

Flow time is an extent of the amount of inventory that is held in the system and by minimizing flow times we minimize the work in progress inventory inside the plant. So, these two present internal objectives to the organization. Now, tardiness and its associated objectives are external to the organization. So, tardiness is essentially concerned with meeting the customer due dates and if there is tardiness then we are not meeting the due date specified by the customer. So, tardiness is an external measure as far as sequencing and scheduling is concerned and with modern day manufacturing concerned more and more with the customer and being able to satisfy the customer's needs and importance.

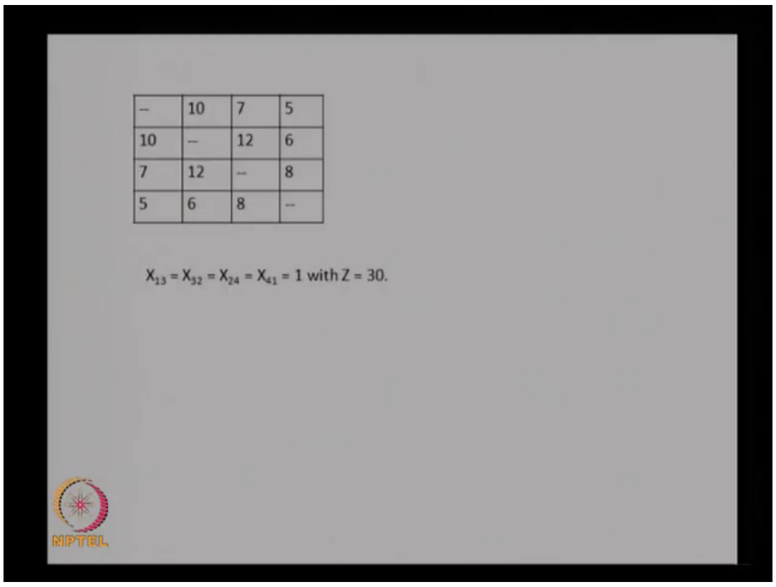
External objectives now become far more important than internal objectives, but if we view it in the context of minimizing inventory which is an important objective in modern day manufacturing all the objectives become important flow times become important because they represent the amount of inventory held in the system tardiness become important in the context of being able to meet the due date and being able to meet the customer requirement.

(Refer Slide Time: 19:49)



Now, in single machine sequencing problems the problem essentially deals with trying to sequence a set of jobs such that changeover times is minimized, changeover times is minimized. We also know that when jobs are done one after another on the machine there is a set up time or a changeover time and if we are doing certain number of jobs in this example, if we are doing 4 jobs, if we are doing 4 jobs. Now, in whatever order these 4 jobs are done the time the processing time spent would still be p 1 plus p 2 plus p 3 plus p 4 which are the processing times on a single machine.

(Refer Slide Time: 20:20)



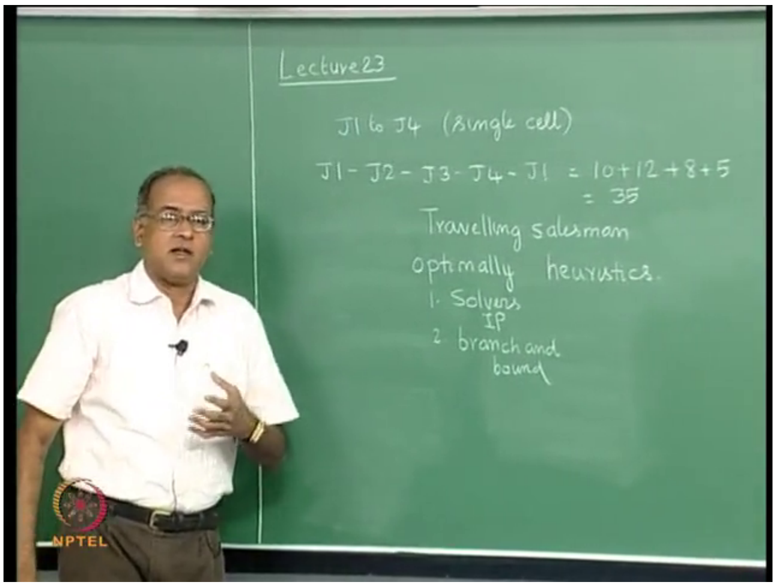
--	10	7	5
10	--	12	6
7	12	--	8
5	6	8	--

$X_{13} = X_{32} = X_{24} = X_{41} = 1$ with $Z = 30$.

But we have to changeover from one job to another and there is a changeover time that is associated with the process of changing over and that changeover depends on the sequence in which these jobs are produced. So, it is called sequence dependent changeover times. So, in the context of cellular manufacturing if we have a situation where let us say there is a manufacturing cell and there are 4 parts or jobs it makes and then each of these is done in batches and there is a changeover time from one part to another or one product to another. In such a context we have to find out the sequence in which these parts have to be produced such that the sum of changeover times is minimized.

So, the first problem that we will consider is a well known problem from the sequencing scheduling literature the context comes from the fact that we are looking at a single cell which make certain number of products or parts and there is a changeover time that is given which is sequence dependent. So, we look at this example where we consider 4 parts on a single cell. So, just to be consistent with the scheduling literature let us call these 4 as J 1 to J 4. So, there are 4 jobs or parts J 1 to J 4 and a single cell and we have to sequence them such that the sum of changeover times is minimized.

(Refer Slide Time: 22:25)



So, the changeover time matrix is shown in that slide and if we follow a sequence J 1 followed by J 2 followed by J 3 followed by J 4 and we come back to J 1 if we do that then the changeover times from the matrix is J 1 to J 2 is 10, 1 to 2 is 10, 2 to 3 is 12, 3 to 4 is 8, 4 to 1 is 5. So, 20 to 30 plus 5, 35 is the changeover time.

Now, we have to look at something in this case there is a single manufacturing cell there are 4 parts that are going to be made and we assume that we are going to follow this sequence which means in this shift or in this time period all these jobs are going to be done and more importantly we will go back and start the next one again with J 1. So, the sequence would be J 1, J 2, J 3, J 4 followed again by J 1. So, this problem is very similar to what is called the Travelling salesman problem where a salesmen starts from the shop visits a set of customers once and only once and comes back to the starting point or the shop.

So, given the distance matrix amongst all these points the shop and several customers what is the root or what is the order of visit of the customers such that the total distance travelled by the salesperson is minimized. Now, the changeover time matrix that we have here can be equated to the distance matrix in a travelling salesman problem and in the travelling salesman problem even though if there are 4 by 4 if we look at an equivalent travelling salesman problem it is like saying I start from the shop the salesman begins from a shop visits 3 other customers in some order and then comes back to the shop.

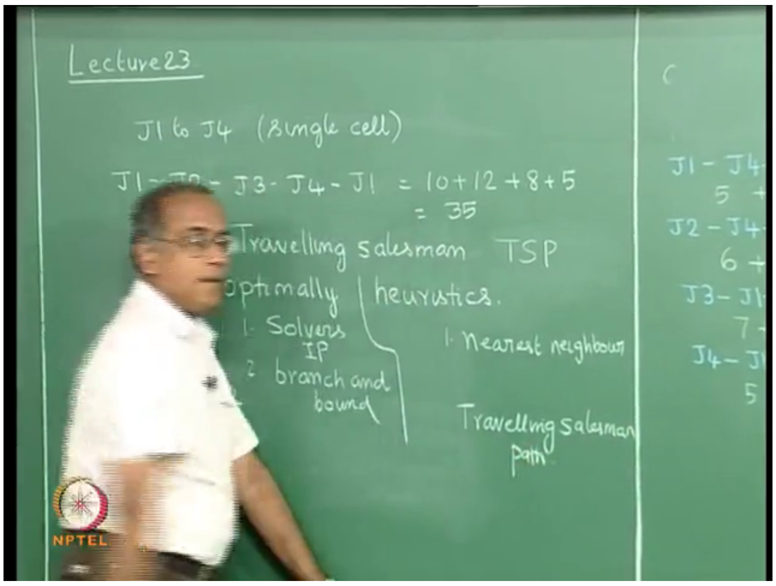
Now, we can also quickly understand that it actually does not matter whether the person starts from the shop visits and comes back or start from any one of the customers and moves around for example, J 1 J 2 J 3 J 4 and back to J 1 is 35, if we take J 2 J 3 J 4 J 1 and back to J 2 it is also 35. So, it does not matter where we start. So, this problem is similar to the travelling salesman problem. It is called the travelling salesman circuit problem because you come back to the starting point. So, if there are 4 parts in this there a typical sequence will look like 1 2 3 4 1 it could be 2 4 3 1 2, but there will be 4 links or 4 legs to it 1 to 2, 2 to 3, 3 to 4 and 4 to 1. So, in a 4 city travelling salesman problem or a problem with 4 jobs and sequence dependent changeover times we will have 4 parts to the solution which comes to 35.

So, the question is given a sequence J 1 J 2 J 3 J 4 and back to J 1 it will take 35 units of time for total changeover what is the sequence that will minimize the sum of changeovers which is the same as what is the sequence the travelling salesman should take such that he or she visits every city once and only once, the equivalent here is you make every product once and only once and then you come back to the starting point which means you come back to the product with which we started.

So, this problem is a well known optimization problem which is a difficult problem to solve optimally within reasonable time as the number of jobs or number of cities increases. So, many times in the travelling salesman problem we do not try to solve it optimally beyond a certain number particularly when you are you have to make scheduling sequencing decisions we sometimes use what are called heuristics. On the one hand the travelling salesman problem or other problem of minimizing the sum of changeover times can be solved optimally to get the optimal or exact solution to the problem, these optimal solutions can also be solved using solvers as integer programming problems, they can also be solved using branch and bound algorithms. Now, the branch using the branch and bound algorithm we can solve a slightly larger problem within the same amount of time that is taken by solving the integer programming and these two things give us the best or the optimal solution.

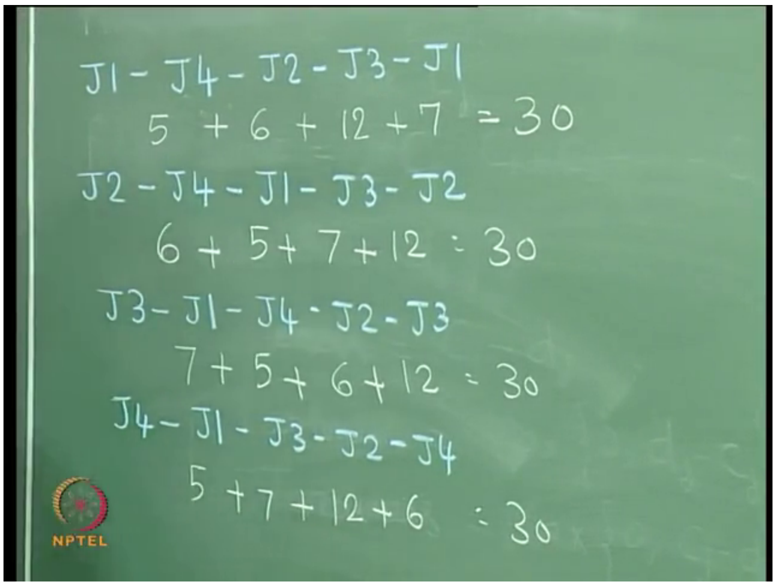
If we are unable to use these optimal solutions or if you are if we are in need of solutions quickly solutions that take very less computational time, but can give good solutions then we use heuristics. Several heuristics are available the most popular one is called the nearest neighbor heuristic is called the nearest neighbor heuristic and let us try the nearest neighbor heuristic on the given data that we have.

(Refer Slide Time: 28:56)



So, if we do a nearest neighbor heuristic. Now, we start with J 1 and find out its nearest neighbor from J 1.

(Refer Slide Time: 29:28)



J1 - J4 - J2 - J3 - J1

5 + 6 + 12 + 7 = 30

J2 - J4 - J1 - J3 - J2

6 + 5 + 7 + 12 = 30

J3 - J1 - J4 - J2 - J3

7 + 5 + 6 + 12 = 30

J4 - J1 - J3 - J2 - J4

5 + 7 + 12 + 6 = 30

So, if we look at this table we observe that the nearest neighbor for J 1 is J 4 which has a changeover time of 5. So, from J 1 we can changeover to J 4 which has the minimum changeover time of 5. Now, from J 4 we look at the 4th row and we realize that once again J 1 is the 1 that has the minimum changeover, but J 1 is already there in the sequence and J 1 can come only at the last one again. So, we look at the next smallest 1 which is J 2 with time equal to 6. Now, from J 2 we go back and see that the smallest is J 4, but J 4 is already here the next smallest is J 1 which is already here. So, the only one that is available is J 3 and from J 3 we come back to J 1. So, the changeover times associated with this is J 1 to J 4 is 5, J 4 to J 2 is 6, J 2 to J 3 is 12 and J 3 to J 1 is 7, so 5 plus 11, 11 plus 12, 23, 23 plus 7, 30.

Now, if we do a nearest neighbor with J 2 as the first one then we go to this table and see which one is the best. So, from J 2 the next smallest is J 4 and from J 4 we go back and realize that J 1 is the next then we have J 3 and then J 2 and the time taken is to J 2 to J 4 is 6, J 4 to J 1 is 5, J 1 to J 3 is 7, J 3 to J 12 which is the same 30.

In fact, we observe that this sequence is the mirror image of the other sequence. This is 1 4 2 3 1 this is 1 3 2 4 1. So, it is the mirror image of that sequence and this matrix is a symmetric matrix where we assume that time to changeover from J 1 to J 2 is the same as time to changeover from J 2 to J 1.

While in a normal travelling salesman problem if these represent distance as that the salesperson has to travel it is not a bad assumption to have that d ij equal to d ji, distance between 2 points is the same whether you measure from 1 to 2 or from 2 to 1. But even there in practice if they represent distance to go distance between 1 place and another for someone to travel considering many other things such as one ways and so on distance between 0.1 to 0.2 may not be the same as the distance between 0.2 to 0.1. We have assumed a symmetric case here which also assumes that time to changeover from product 1 to product 2 is the same as the time to changeover from product 2 to product 1.

Now, such a matrix or a tsp is called a symmetric tsp or cyclic sometimes and sometimes when they are not symmetric which means T 1 2 could be 10, d 21 could be 12. Quite often the example given is for example, if we are having similar looking products, but we have to paint them with different colors changeover time from white to black could be different from changeover time from black to white. So, the matrix need not be symmetric.

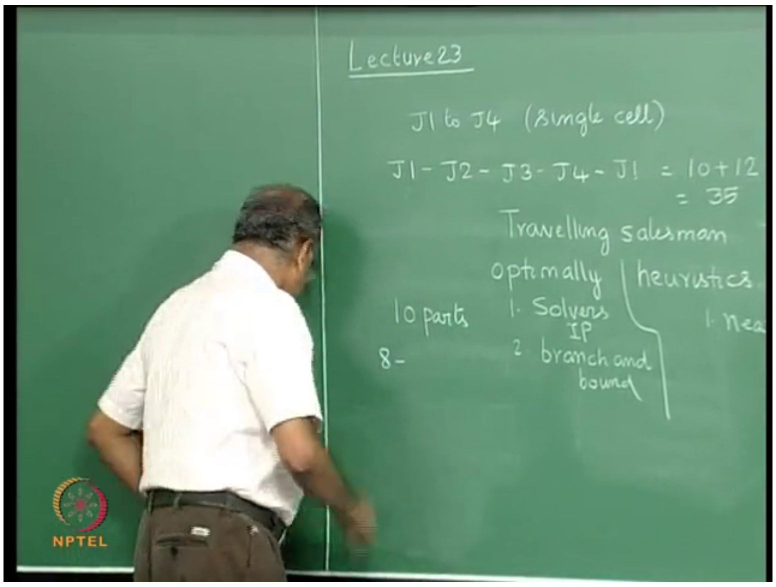
So, they are called asymmetric travelling salesman problems. But the same rule can be applied to asymmetric problem as well except that we may not get mirror image solutions like we got here. So, if we start with J 3 as the first one, J 3 the smallest 1 is J 1 and from J 1 we will move to J 4 and then to J 2 and then to J 3 and in this case the solution again will be J 3 to J 1 is 7, J 1 to J 4 is 5, J 2 to J 4 is 6, J to 2 J 3 is 12. So, we will again get 30. So, the last one is we could start with J 4. So, from J 4 we moved to J 1. Now, from J 1 we moved to J 3 and then to J 2 and then to J 4. So, once again we will get J 4 to J 1 is 5, J 1 to J 3 is 7, J 2 to J 3 is 12, J 2 to J 4 is 6 we get 30.

In this particular example all the 4 possible solutions under the nearest neighborhood have actually given us the same 30. In different examples we could have different values coming here and we could choose the best out of these to be the solution to the travelling salesman problem. Now, the nearest neighborhood method does not always guarantee the best or the optimal solution for multiple reasons the most important reason is that if we take this sequence J 1 to J 4 and then we write J 2 then automatically J 3 has to come here because we do not have any other choice. So, if J 2 to J 3 was a very large number then this sequence would end up giving us a very large number here. So, since we do not have a choice with respect to the last or last, but one term in the sequence this cannot guarantee the best or the optimal solution.

Nevertheless this is a good method that people use for the simple reason it gives good solutions and this is perhaps the most intuitive method one can think of. So, we start at some point go to the nearest point keep going up to the nearest points from each one keep visiting another city or place that we have not visited which is closest and then come back which is equivalent of saying start with one product go to the next product which has minimum changeover and do this till you exhaust all the products and then you come back to the starting one.

So, when we come back to the starting one it is called travelling salesman problem or a travelling salesman circuit problem and often called as TSP travelling salesman problem. Sometimes we would like to have 4 let us say these 4 parts have to be produced in this shift and we could even sometimes have the setup for what was produced in the last shift. So, we could have a situation where we know which one we are beginning.

(Refer Slide Time: 37:18)



Lecture 23

J1 to J4 (single cell)

J1 - J2 - J3 - J4 - J1 = 10 + 12
= 35

Travelling salesman

optimally | heuristics

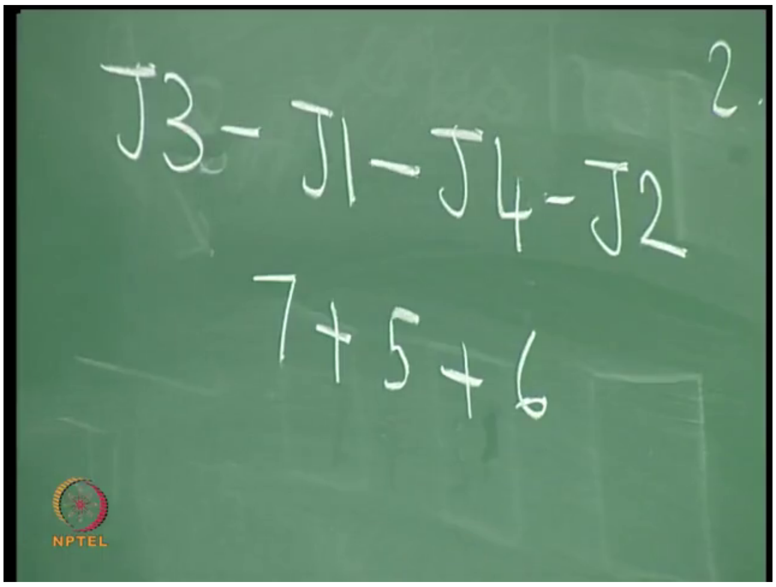
10 parts 1. Solvers IP

8 - 2 branch and bound

1. Nea

Suppose instead of 4 we have 10 parts that have to be produced in this shift. Now, we know already this is set up for one of the 10 which was the last produced item in the previous shift. So, let us say we start with 8 then what we want to do is we start with 8 we have to produce all of them we can produce which means we have to write the remaining ones or to make it simpler we come back to these 4 let us say the same for have to be produced. Now, let us say that the last one produced in the previous shift was J 3. So, we are we need not make another changeover. So, we could start today with J 3. Now, the question is in what order do we produce the rest of them, so what order are we going to produce the rest of them.

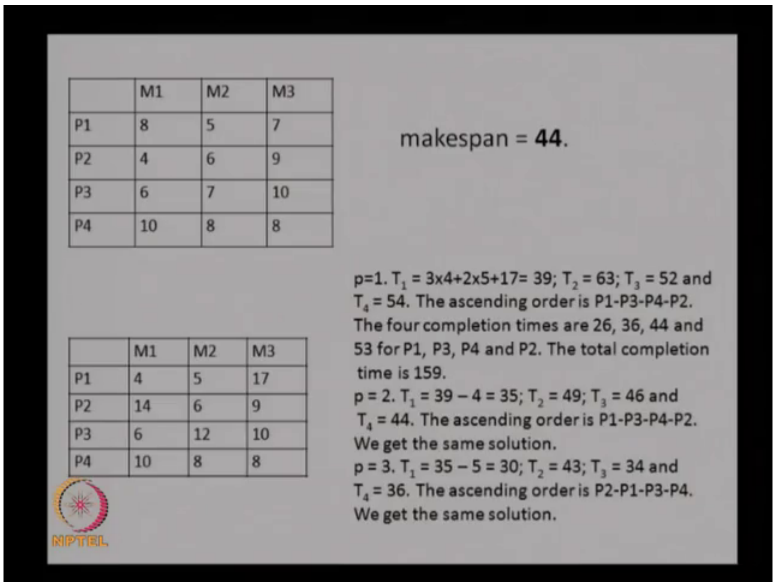
(Refer Slide Time: 38:24)



So, J 3 which means from J 3 we could do a nearest neighbor and then we go back and say from J 3 we will do J 1 based on the nearest neighbor and from J 1 we will go to J 4 based on the nearest neighbor and from J 4 we will go to J 2 based on the nearest neighbor and then we can leave it here. We will say we need not come back to 3 because at the end of this shift we would have produced the second part which we call as J 2 and we will leave it there a next shift can begin with J 2. So, this would give us a changeover time of 7 plus 5 plus 6 we are not coming back to this. This is slightly different from this version of the travelling salesman problem.

Now, the travelling salesman problem also has another version which is called travelling salesman path problem called travelling salesman path problem, but in the path problem the first and the last points are specified. Now, if we make this assumption only the first is specified the last could be anything that is retained and in the next shift we could begin with J 2 and proceed. So, this is how we look at the first issue in sequencing and scheduling. So, the travelling salesmen idea is applicable this is normally used in single machine scheduling where there are several jobs waiting to be processed on a single machine. Now, we use this model when we have several parts that are produced in a single cell and there is a sequence dependent changeover.

(Refer Slide Time: 40:35)



	M1	M2	M3
P1	8	5	7
P2	4	6	9
P3	6	7	10
P4	10	8	8

makespan = **44**.

	M1	M2	M3
P1	4	5	17
P2	14	6	9
P3	6	12	10
P4	10	8	8

p=1. T_1 = 3x4+2x5+17= 39; T_2 = 63; T_3 = 52 and T_4 = 54. The ascending order is P1-P3-P4-P2. The four completion times are 26, 36, 44 and 53 for P1, P3, P4 and P2. The total completion time is 159.

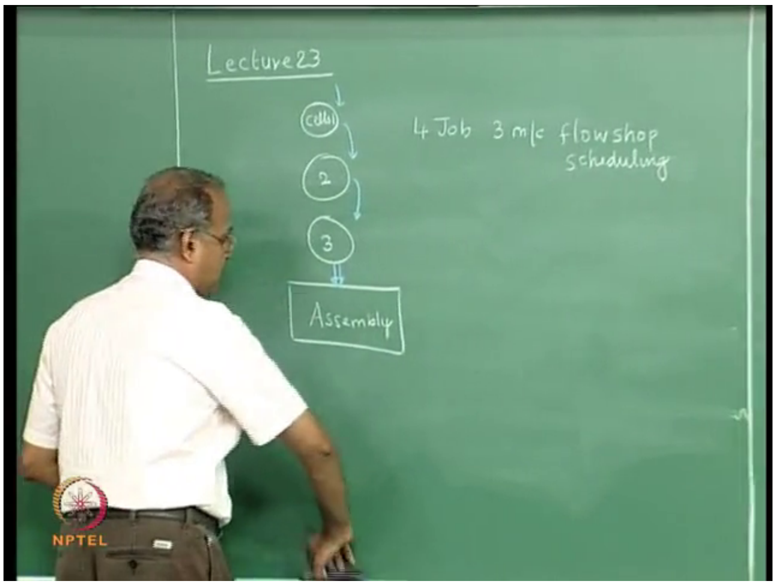
p = 2. T_1 = 39 – 4 = 35; T_2 = 49; T_3 = 46 and T_4 = 44. The ascending order is P1-P3-P4-P2. We get the same solution.

p = 3. T_1 = 35 – 5 = 30; T_2 = 43; T_3 = 34 and T_4 = 36. The ascending order is P2-P1-P3-P4. We get the same solution.

Now, we look at the next model which is called a flow shop scheduling model I will quickly do the flow shop scheduling model and then explain the context in which the flow shop scheduling model is relevant in the context of modern manufacturing.

So, in a flow shop scheduling model we have as shown in the first part of the table we have 4 parts that are produced and there are 3 machines.

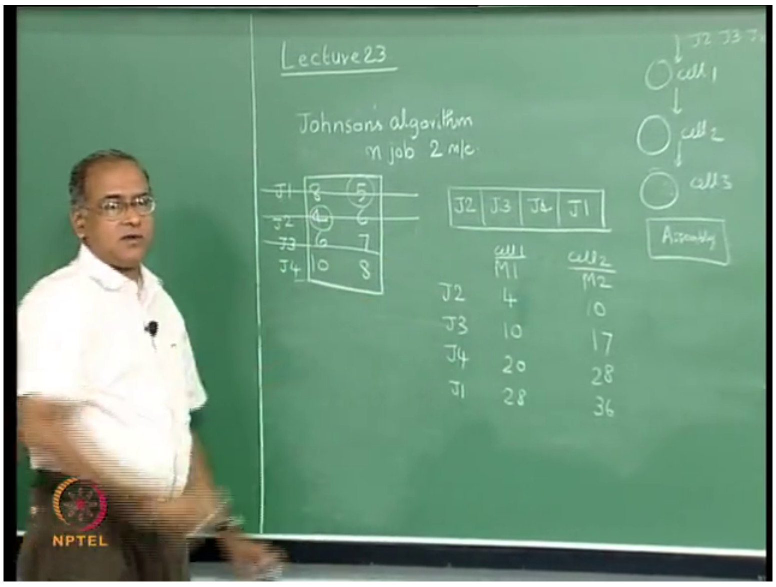
(Refer Slide Time: 41:07)



Now, we will assume in our case that 3 machines are your 3 cells cell 1, cell 2, cell 3 and here is the assembly. So, each of these 4 parts p 1 to p 4 will first start processing on cell 1, then they will visit cell 2, then they visit cell 3 and then they go to assembly let us say we have a cellular manufacturing system which works this way. And if we have that then what is the order in which we will send the 4 parts p 1 p 2 p 3 p 4 such that we minimize the makespan.

Now, this becomes a standard 4 job 3 machine flow shop scheduling problem scheduling or sequencing problem. Now, there are several ways of doing this flow shop scheduling or sequencing problem. Once again the problem is a hard problem to solve optimally, so we would discuss 1 or 2 heuristics to solve this.

(Refer Slide Time: 42:34)



~~J1~~	8	$\bigcirc\!\!\!\!5$
~~J2~~	$\bigcirc\!\!\!\!4$	6
~~J3~~	6	7
J4	10	8

J2	J3	J4	J1

	cell1 M1	cell2 M2
J2	4	10
J3	10	17
J4	20	28
J1	28	36

Now, we will introduce the Johnsons algorithm which is for a n job 2 machine. Now, what we have is 4 job 3 machine or 4 job 3 cell problem. So, if we had a 2 sell problem then you can use the Johnsons algorithm to do that. So, if we have a 3 cell problem we need to modify the Johnsons algorithm.

So, first let us explain the Johnsons algorithm and then look at the modification of the Johnsons algorithm. Suppose we had 4 jobs and only 2 cells instead of 3 and we had this 8 5 4 6 6 7 10 and 8 if there were 2 cells the Johnsons algorithm begins like this. There are 4 jobs or parts, so we may call them as p 1 to p 4 or we may call them as J 1 to J 4. So, in the Johnsons algorithm since there are 4 jobs we have to find the order or sequence in which these jobs have to be processed. So, what we do in the Johnsons algorithm is if we have 4 jobs we look at the smallest of the processing times that are here. So, this is the smallest of the processing times that happens to be for job 2 on the first machine. So, if it is on the first machine right the job number from the left. So, we write job 2 here.

Now, remove this look at the next smallest one this happens to be 5 for the job 1 on the second machine, so if it happens on the second machine come from the end and write J 1 here. So, this is also a removed. The next smallest is here it happens on the first machine. So, come from the left and right J 3 the last one will come here as J 4. So, this is the sequence in which we will do if there are only 2 machines it is. Now, possible to write it either from the left or from the right, if there are 3 machines and for some reason it happens in the middle then we do not know what to do.

Johnsons algorithm is meant only for 2 machines. So, let us look at the Johnson solution. So, if this is the problem that we were looking at if there are only 2 cells instead of 3 that all these 4 parts have to visit then the order in which I will be sending them is J 2 first followed by J 3 followed by J 4 followed by J 1. For example, it is like saying there is a cell 1 cell 2 and then there is a assembly. So, the order will. Now, be J 2, J 3, J 4, J 1. Now, what do I do? I calculate the time at which or the schedule for this. So, I will say M 1 here or cell 1 here cell 2 here which is notionally called as M 2. So, J 2 will enter J 2 will finish at time equal to 4. Now, at time equal to 4 J 2 will come out of this come to cell to take additional 6 and finish at time equal to 10 then I will have J 3 which is the next one coming. So, this one is free at time equal to 4. So, J 3 will take 6 more time units finish at 10 which is exactly the time at which J 2 leaves the cell. So, this becomes free. So, this starts and finishes at 17.

Now, we look at J 4, J 4 can start only a 10 this can start only at 10 because J 3 leaves it at 10. So, 10 it takes another 10 which is 20. Now, when it comes out at 20 this is free because at time equal to 17 this has left. So, this is free, so it can take it at 20 and finish it at 28.

The last one is J 1; J 1 enters at 20 finishes exactly at 28. So, when J 1 finishes here at algorithm 8 this is free at algorithm 8. So, it can take another 8 time units finish at 36. So, the makespan now is 36 and all these 4 jobs can be completed here at time equal to 36 all the 4 jobs are completed they can enter the assembly at times 10, 17, 28 and 36 respectively.

So, this is how you compute the makespan when you have n jobs and 2 machines in the flow shop and in our case n jobs and 2 cells in the flow shop with the time that is given here. But then there could be situations where we will have n jobs and 3 cells as in the example that is given here. So, we will have n jobs and 3 cells and then the assembly. Now, how do we do this? So, essentially we should look at how do we modify the Johnsons algorithm for 3 and then if we are able to do that we will come back and use this. So, the Johnsons algorithm is essentially meant for 2 machines and then if these are the processing times given the Johnson sequence is brought out this way this sequence is called the Johnson sequence and very quickly to recap the Johnson sequence the Johnson sequence works as follows.

Take this matrix, find the smallest number which is there, find out the job corresponding to the smallest number and if the smallest number for that job happens to be on the first machine come from the left and right the job in the first available position. Remove that job and once again find the smallest number, if the smallest number happens to be on the second machine for a particular job then write that job come from the right hand side and write it on the first available position and complete the sequence and once the sequence is completed write down the time table and find out the maxima.

So, we will call all of these as a Johnson algorithm or a Johnson problem. So, when we say solve a Johnson problem it means we assume that there are n jobs, there are 2 machines, we can write this sequence and then we can compute this and write this makespan. So, we have now seen the Johnson problem for 2 machines. Now, for the case that we are actually looking at which has 3 cells then we have to modify the Johnson and try to get the best solution, we will see that in the next lecture.