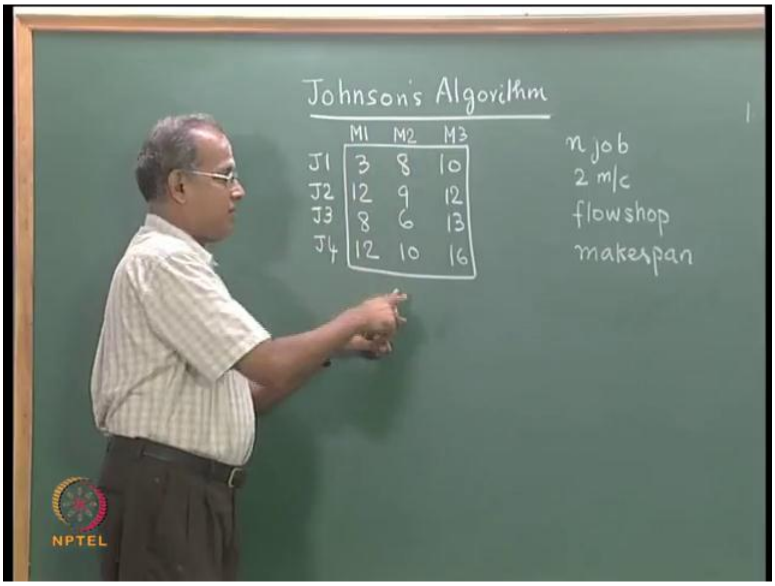
Operation and supply chain management
Prof. G. Srinivasan
Department of Management Studies
Indian Institute of Technology, Madras

Lecture - 26
Flow Shop Scheduling – Three Machines, Johnson's Algorithm and Branch and Bound Algorithm

(Refer Slide Time: 00:11)



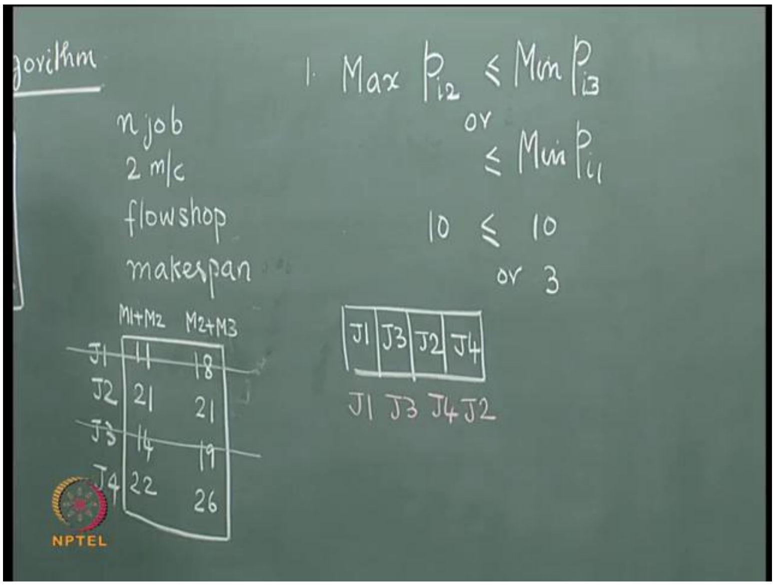
Johnson's Algorithm

	M1	M2	M3
J1	3	8	10
J2	12	9	12
J3	8	6	13
J4	12	10	16

n job
2 m/c
flowshop
makespan

In the previous lecture, we saw the Johnson's algorithm, which could provide an optimum solution to a n job, 2 machine, flow shop, sequencing problem to minimize Makespan. We also showed a numerical illustration to explain the various steps in the Johnson's algorithm. Now, the next obvious question is, can the Johnson's algorithm works for a 3 machine problem, other things being equal, n jobs 3 machines flow shop sequencing problem to minimize makespan. Now, Johnson actually proved that, under certain conditions, the Johnson's algorithm can be extended to solve the 3 machine problem optimally.

(Refer Slide Time: 01:09)



gorithm

n job
2 m/c
flowshop
makespan

	M1+M2	M2+M3
~~J1~~	~~11~~	~~18~~
J2	21	21
~~J3~~	~~14~~	~~19~~
J4	22	26

1. Max $P_{i2} \leq$ Min P_{i3}
or
$\leq$ Min P_{i1}

$10 \leq 10$
or 3

J1	J3	J2	J4

J1 J3 J4 J2

So, Johnson gave two conditions, which are this. If maximum of the processing times is less than or equal to minimum of the processing time, or less than or equal to minimum of the... Now, this condition says that, the maximum of the processing time of the jobs on machine number 2, which is here, the maximum of the processing time. Now, we are considering a 4 job 3 machine problem flow shop sequencing problem to minimize makespan.

So, the largest of the values for machine 2, which is 10 should be less than or equal to the smallest of the values on machine 3, which happens to be another 10, so 10. Or the largest on machine 2, which is 10 should be less than or equal to the smallest on machine 1, which is 3. Now, out of the two inequalites we observe that, for this particular problem, the first one is satisfied, the second one is not satisfied. But, it is also enough if one of them is satisfied, therefore the Johnson condition holds for this problem.

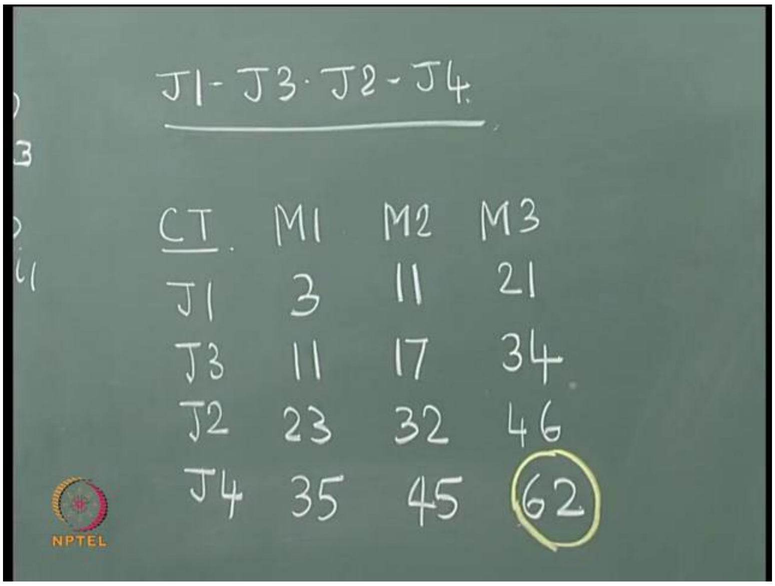
So, when the Johnson condition holds for this problem, how do we get the optimum solution? What we do here is, we create an equivalent 2 machine problem, because the basic Johnson's algorithm works for 2 machines. So, we create an equivalent 2 machine problem with the 4 jobs J 1, J 2, J 3 and J 4 on 2 equivalent machines and these two equivalent machines are called M 1 plus M 2 and M 2 plus M 3. So, the first column here, which is called M 1 plus M 2 is actually the sum of the processing times on machines 1 and 2.

So, it becomes 3 plus 8, 11, 12 plus 9, 21, 8 plus 6, 14, and 12 plus 10, 22, second column of the equivalent Johnson 2 machine problem is M 2 plus M 3 which means, we add the processing times on machines 2 and 3 to get 8 plus 10, 18, 9 plus 12, 21, 13 plus 6, 19 and 10 plus 16, 26. Now, we have an equivalent Johnson problem with 2 machines, so we apply the Johnson's algorithm on this data and try and get the optimum sequence that minimizes makespan for the equivalent 2 machine problem.

So, once again as we have done in the past, we write a small table with 4 positions, which are for the 4 jobs. Take the smallest number available in this, which happens to be 11 and it happens for job 1, therefore come it happens to be for job 1 on this first machine. So, come from the left hand side and place job 1 in the first available position. Now, strike of job 1. Now try to find out the smallest number that is available, which is 14 here, which is for job 3 and on first equivalent machine.

So, come from the left and put J 3 here, now once again remove this, now find out the smallest number, which happens to be 21, which is for J 2, both on M 1 and M 2, so there is tie. So, let us go by this 21, which happens to be on the first machine, therefore J 2 will come here, which automatically means that, the left over job J 4 will come to this side. Now, since there was a tie, the tie could be broken other way, so when we break the tie the other way, the sequence would have been J 1 J 3, choosing this 21 would have put J 2 here and J 4 here. Now, for both the sequences, let us try and find out the makespan with the original data, because the original data is the problem that we are looking at. So, we look at the sequence J 1 J 3 J 2 J 4.

(Refer Slide Time: 06:37)



J1 - J3 - J2 - J4

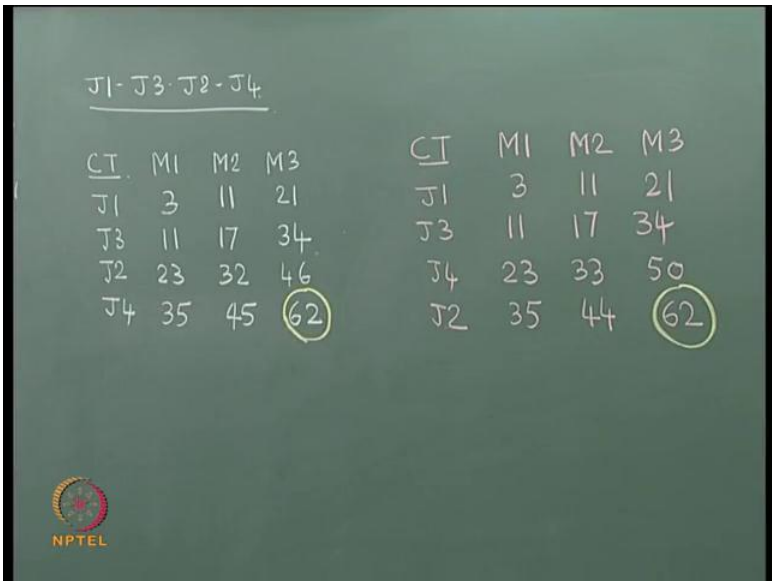
CT	M1	M2	M3
J1	3	11	21
J3	11	17	34
J2	23	32	46
J4	35	45	62

And we look only at completion times on M 1 M 2 M 3, J 1 J 3 J 2 and J 4, so J 1 goes first, comes out of machine 1 at time equal to 3, requires 3 minutes. At time equal to 3, it goes to machine 2, it requires a further 8 minutes processing, so comes out at time equal to 11. Once again enter M 3 at 11 and comes out at 21, because it requires a further 10 minutes on M 3. J 3 on M 1, J 3 starts at time equal to 3, requires another 8, so 3 plus 8, 11. Exactly at 11, M 2 is free, so it takes another 6 units, 11 plus 6, 17.

Now, it has come out of M 2, but M 3 is free only at time J equal to 21, therefore it visits M 3 at time equal to 21 and requires another 13 time units, so 21 plus 13 is 34. Now, J 2 starts at time equal to 11, finish at time equal to 23. Now, at time equal to 23, M 2 is available, so requires another 9 units on M 2, so 23 plus 9, 32, but has to wait for M 3, because M 3 is available at 34, so 34 plus 12 is 46. Now, J 4 enters, so M 1 is available at time equal to 23, so J 4 requires another 12 units, so 23 plus 12, 35.

At 35, M 2 is free, it requires another 10 time units, so finishes at 45. Now it has to wait for M 3, because M 3 is available only at 46. So, enters or visits M 3 at 46, takes another 16 time units and comes out at time equal to 62. So, the makespan for this sequence will be 62 we have, we also have an alternate sequence, which is J 1 J 3 J 4 and J 2.

(Refer Slide Time: 09:18)



J1 - J3 - J2 - J4

CT	M1	M2	M3
J1	3	11	21
J3	11	17	34
J2	23	32	46
J4	35	45	62

CT	M1	M2	M3
J1	3	11	21
J3	11	17	34
J4	23	33	50
J2	35	44	62

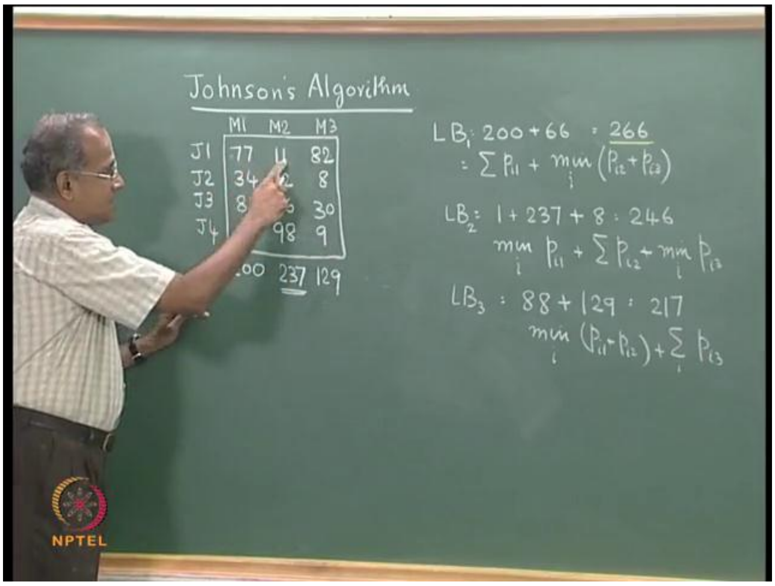
So, let us try and compute the makespan for the other sequence J 1 J 3 J 4 and J 2, so J 1 J 3 J 4 and J 2. Since J 1 and J 3 are the first two jobs, the completion times will be the same up to this, so we can write 3, 11, 21, 11, 17, 34, so J 4 enters at time equal to 11 on M 1 and finishes at 23. Now, at 23, M 2 is available, so takes another 10 units, so finishes at 33 and has to wait till 34, because M 3 is available only at 34. So, it visits M 3 at 34 requires another 16 units, so 34 plus 16 is 50, comes out at 50.

Now, J 2 enters M 1 at time equal to 23, so 23 plus 12, 35, it comes out, so at 35, M 2 is available, so takes another 9 units, 35 plus 9 is 44, but has to wait till 50. So, at 50, it visits M 3, takes another 12 units and comes out at time equal to 62. So once again the makespan for the second sequence is also 62. So, there are two alternate solutions, both of which are optimum with the same makespan of 62. So, this is how the Johnson's algorithm for the 2 machine problem can be extended to 3 machine case, provided the data satisfies these conditions.

So obviously, the next question that would come is, what happens if this data does not satisfy the condition, is there another way to try and get to the optimum solution? Now, what happens is, the moment we do not satisfy this condition, when we realize that the Johnson's extension does not give the optimum solutions to the problem or optimum makespan to the problem.

So, we have to resort, if we really want to find the optimum then we have to resort to what are called the Branch and Bound algorithms, which are computationally a little more demanding, but nevertheless give us optimal solution to the problem. So, we look at another illustration, through which we explain the Branch and Bound algorithm to minimize the makespan on multiple processes in a flow shop.

(Refer Slide Time: 12:47)



Johnson's Algorithm

	M1	M2	M3
J1	77	[illegible]	82
J2	34	[illegible]	8
J3	[illegible]	[illegible]	30
J4	[illegible]	98	9
	[illegible]	237	129

$LB_1 : 200 + 66 : \underline{266}$

$: \sum P_{i1} + \min_i (P_{i2} + P_{i3})$

$LB_2 : 1 + 237 + 8 : 246$

$\min_i P_{i1} + \sum P_{i2} + \min_i P_{i3}$

$LB_3 : 88 + 129 : 217$

$\min_i (P_{i1} + P_{i2}) + \sum_i P_{i3}$

Now, let us consider this numerical example and try and see, whether first it satisfies the Johnson's conditions and if so use it, if not try and look at a Branch and Bound algorithm to solve. Now, the Johnson condition says, maximum processing time on machine 2, which happens to be 98, should be less than or equal to the minimum of this. So, minimum of this happens to be 1, minimum of this happens to be 8, so the Johnson condition is clearly not satisfied.

So, we cannot apply this to try and get the optimum solutions or the optimum makespan to the problem. Now, let us ask yourselves a few more questions, now by looking at this data, can we try and estimate the makespan, try and have what is called a lower estimate to the makespan or a lower bound to the makespan. Now, what is the lower bound to the makespan, now let us add the processing times on M 1, M 2 and M 3. So, this is 200, if we add on M 1, on M 2 is 237 and 129 on M 3, so sum of processing times on 3 machines are 200, 237 and 129.

Now, the maximum of these three is 237 and we could say definitely, that the makespan cannot be smaller than 237. Because, all the jobs have to go through machine 2 and the total work load on machine 2 itself is 237, therefore the makespan can only be more than, greater than or equal to 237. So, the maximum of the sum of processing times on the machines give us a lower bound to the makespan, so we could say, lower bound is 237.

Now, we can actually make these lower bound slightly better or we can tighten the lower bound. How do we do that? Now, if we take the first machine, now the total load or total processing on the first machine itself is 200. But then all the jobs are completed only when the last job entering machine 1 also finishes on machine 2 and also finishes on machine 3. So, the machine 1 alone would require 200, so there will be an additional thing that will go into the makespan, because the last of the jobs entering M 1 has to finish M 2 and M 3.

So, the minimum possible increase after this 200 is the minimum of the sum of processing times on M 2 plus M 3. So now, we realize, that this is 11 plus 82 is 93, this is 92 plus 8 is 100, this is 66, this is 107. So, the minimum increase that will happen has to be 66, because one of these four jobs has to be the last job on the machine. And if job 1 happens to be the last job then there is an additional 93.

If job 2 happens to be the last job there is an additional 100, if job 3 happens to be the last job there is an additional 66, job 4 happens to be the last there is an additional 107, so definitely the makespan will go up by 66. So, now the lower bound on the makespan is 200 plus 66, which is sigma P i 1, sum of the processing times on machine 1 plus minimum over i minimum over the jobs P i 2 plus P i 3, sum of processing times on machines 2 and machine 3, so this becomes 266.

So, one could say that, the makespan has to be 266 or more, now this is the lower bound that is computed based on machine 1. Now, let us compute a lower bound based on machine 2. Now, machine 2 by itself has a load of 237, but the first job entering the flow shop cannot start processing on machine 2, unless it has finished machine 1. So, the earliest time machine 2 can begin is the minimum of these numbers, so the earliest machine 2 can begin is 1.

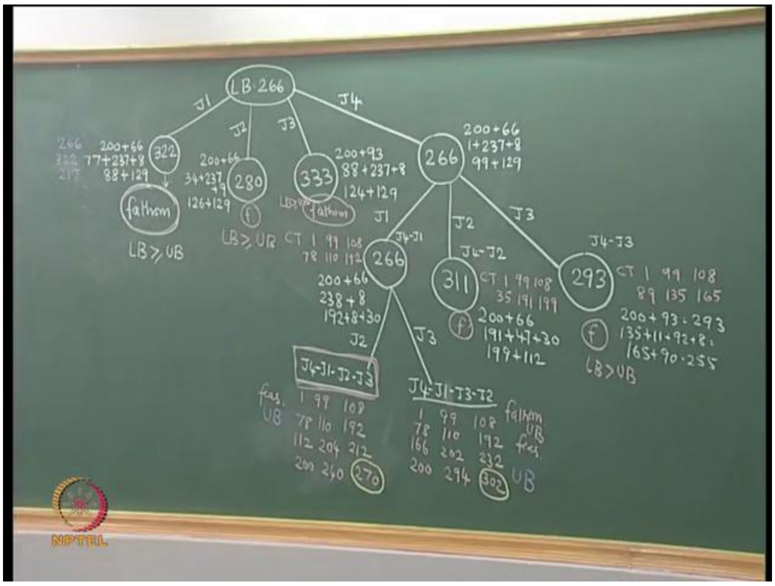
So, lower bound is again 1, which is the earliest machine 2 can begin plus machine 2 requires at least 237 time units. So, earliest machine 2 can finish everything is 1 plus 237 and it is still not over, because the last job on machine 2 has to visit machine 3. So, there is an additional component with the last job on machine 2 visiting machine 3. And therefore, at least a minimum of another 8 will have to be incurred on machine 3 before all the jobs are complete, so 1 plus 237 plus 8, which is 246.

So, to write this algebraically, we will say, minimum over i minimum over jobs P i 1 plus sigma P i 2 plus minimum over i P i 3, is another lower estimate of the makespan, so the makespan cannot be less than 246, it has to be 246 or more. Now, lower bound based on machine 3, let us call these as LB 1, LB 2 and LB 3, lower bound based on machine 3 is this. Machine 3 by itself has a time of 129, but machine 3 cannot begin till at least one job has finished on M 1 and M 2.

So, the earliest machine 3 can begin is the minimum of the sum of these two, so this is 88, this is 126, this is 124 and this is 99. So, minimum, the earliest M 3 can begin is 88 if J 1 happens to be the first job. So, earliest it can begin is 88 and it requires at least 129 then the last job is over, so the earliest based on M 3 would be 88 plus 129, so 217. Now, to write it in our notation, this will be minimum over i P i 1 plus P i 2 plus sigma P i 3 over i.

So now, we have calculated three lower bounds, one based on each machine, so based on M 1, we could say the 266 is a lower bound to the makespan, based on M 2 246 and based on M 3 217. Now, out of the three, we could say now that, the makespan can only be the 246 or more, so the actual lower bound is the maximum of the three lower bounds that we have. And then we say that, the lower bound right now is 266, for a makespan is 266, which is the biggest of the three numbers.

(Refer Slide Time: 22:03)



So, we start what is called branch and bound tree by saying that, the lower bound is 266, now what we do is, we try and branch in this tree by trying to fix each of the jobs as the first job in the sequence. So, there are four jobs, so there are four possible branches, the four possible branches will have J 1 or job 1 as the first job, job 2 as the second, job 3 as the third, four jobs J 1 J 2 J 3 J 4, each as first job in the sequence. So, this sequence will have J 1 as the first job, this sequence will have J 2 as the first job and so on.

So now, we try and find out the lower bound for each of these four loads of the branch and bound tree. So, let us first compute this, so for this, we actually compute three numbers, once again based on these three things which means that, if we fix J 1 as the first job then on M 1, all the processing will be over at time equal to 200 and this lower bound will take the minimum of P i 2 plus P i 3 over i. Since J 1 is the first job, J 1 cannot be the last job, because a very basic idea of this lower bound is that, the total processing time on M 1 plus the last jobs processing time on M 2 plus M 3.

So, since the J 1 is fixed as the first job, J 1 cannot be the last job, therefore to compute this minimum, we would only look at the minimum of this (Refer Slide Time: 23:59) this and this, we will leave out this 93. So, minimum of this, this, this, still remains as 66, so 200 plus 66 is the first computation. Now, for the second computation, J 1 is the first job, so J 1 takes 77 time units on machine 1. So, M 2 can start only at time equal to 77, so this will become 77 plus sum of processing times on machine 2, which is 237 plus the last on M 3.

We will leave out J 1, because J 1 is already the first job, so J 1 cannot be the last job, so the last job can be only J 2 or J 3 or J 4. So, out of J 2 or J 3 or J 4, we take the minimum, so out of 8, 30 and 39, we take the minimum, which again happens to be 8. So 77 plus 237 plus 8. Now on M 3, J 1 is the first, so earliest this can begin is 77 plus 11, which is 88 and M 3 by itself has a time of 129. So, the maximum of these three numbers will now be the lower bound if job 1 is fixed as the first job.

And we calculate that, so this is 266, this is 312, 237, 245, 252, 322, so this is 322 and this will be 217. So, the maximum of the three values is 322, therefore we say that, if we start with J 1, the lower bound is 322. Now, let us do this computations with J 2 as the first job, now in J 2 is the first job, on M 1 the total time required is 200. Now, J 2 cannot be the last job, because it is the first job, so the last can only be J 1 or J 3 or J 4. So the minimum is 93, again 66 and 107, so this will remain as 200 plus 66.

Now, the lower bound based on M 2, J 2 is the first job, so earliest M 2 can begin is 34 plus 200 and 37, which is the sum of processing times on this. So, 34 plus 237, since J 2 is the first job, now it cannot be the last job, so the last job can only be the 1, 3 or 4. So, take the minimum of 1, 3 or 4, which happens to be 9, so we have plus another 9. Now, if J 2 happens to be the first job, the earliest M 3 can begin is the sum of 34 plus 92, which is 126 plus another 129, so 126 plus 129.

So, this is 266, this is 237 plus 9 is 246, 280. This is 126 plus 129 is 255, so maximum of the three is 280 and therefore, 280 is the lower bound when J 2 is fixed as the first job. So, let us now do that, if J 3 is fixed as the first job, now when J 3 is fixed as the first job, M 1 by itself will require 200. The last of the jobs cannot be J 3, because J 3 is fixed as the first job. So, the last can only be J 1, J 2 or J 4. So, sum of 11 plus 82 93, 92 plus 8 100, 98 plus 9 107, so minimum is 93 here, so we will get 200 plus 93.

Now, if J 3 happens to be the first job, the earliest M 2 can begin is 88, because J 3 has to finish on M 1 and only then it can enter M 2. So, earliest it can enter is 88 plus another 237, now J 3 cannot be the last job, so we have to take the minimum of this, (Refer Slide Time: 29:16) this and this, which happens to be 8, so 88 plus 237 plus 8. Now, once again if J 3 is the first job, earliest M 3 can begin is after it has finished this and this, which is 88 plus 36, 124 plus another 129.

So, 124 plus 129, this is 293, this is 245, 253, 333 and this is 253, so the highest of them is 333, so the lower bound becomes 333. Now, let us look at J 4 as the first job and complete this. So, M 1 takes a time equal to 200, J 4 now cannot be the last job, last job can only be J 1 J 2 and J 3, so minimum of 93, 100 and 66, so you will get the same 200 plus 66 here. Now, if J 4 is the first job, the earliest M 2 can begin is 1, so 1 plus 237 is 238, once again J 4 cannot be the last job.

So, minimum of this, this and this, which is 8, so 1 plus 237 plus 8, which is 246 and if J 4 is the first job, earliest M 3 can begin is 99, because J 4 has to finish 1 plus 98. So, 99 plus 129, which is 228, so this is 266, this is 246, this is 228, so lower bounds remains as 266. So, what we have observed now after this calculation is that, if we fix job J 1 as the first job, the makespan is going to be 322 or more. If we fix J 2, makespan is going to be 280 or more, if we fix J 3, 333 or more and if we fix J 4, it is 266 or more.

Since we want to minimize the makespan, we proceed in this direction, because this is the smallest number 266. So, there is a possibility that, we get a smaller number as we proceed from here. So, we proceed from this node right now, we still have to evaluate this, this and this, we will see that as we move along. But then we proceed from the node, which has the smallest value of the lower bound. So, from this node we proceed, now we have already fixed job 4 as the first job.

So, we have to now fix the second job, so job 1 is the second job, job 2 can be the second job, job 3 can be the second job. Now, let us evaluate this, if job 4 is the first job and job 1 is the second job, now what we have to do is, now we have what is called a partial sequence, so according to this, the partial sequence is J 4 and then J 1. Now, let us first evaluate the completion times for this partial sequence, which is J 4 and J 1, let us first evaluate the completion times on this.

So, J 4 begins here, this is completion time, so J 4 is 1, it finishes at 1, 1 plus 98, 99 and 99 plus 9 is 108. Now, J 1 is here say, J 1 starts at 1 and finishes at 78, now we are evaluating this for the partial sequence J 4 J 1. Now, at 99, it goes it visits M 2 takes another 11 units, so comes out at 110. So, at 110, M 3 is free, takes another 82 units, comes out at 192, so we keep this partial sequence with us to help us make the lower bound better.

Now, what we do, now we have to find out the three lower bound corresponding to the 3 machines. So, corresponding to M 1, the minimum time is 200, already 4 and 1 are fixed, 2 and 3 are remaining, so 2 and 3, one of them only can be the last job. So, we take the minimum of 100 and 66, so the first lower bound still remains as 200 plus 66. Now, when we look at M 2, the computation will be very interesting, so we would have... Now, at this point, when we consider the partial sequence J 4 J 1, now M 2 is available for the remaining two jobs, which are J 2 and J 3 at time equal to 110.

So, the minimum M 2 can complete all the jobs is 110 plus 92 plus 36, so 110 plus 92 plus 36 is 110 plus 92 is 202, 202 plus 36 is 238. Now, the earliest M 2 can complete is 238, now two jobs are remaining J 2 and J 3, one of them has to be the last job. So, it would require an additional 8, which is the minimum of this, so 238 plus 8. And we will see the same thing happening for M 3 also, now based on these partial sequence, M 3 is available for the remaining jobs J 2 and J 3 at time equal to 192, so at time equal to 192, M 3 can start looking at J 2 and J 3.

So now, from 192, J 2 and J 3 plus 8, 200 plus 30, 230, so 192 plus 8 plus 32, 130, so the lower bound is 266, 246, 130, so lower bound remains as 266. Now, let us show one more computation here with J 4 and J 2 as the two jobs. So we will do for J 4 and J 2. Now, let us quickly do the completion times, so J 4 will be again 199 and 108, coming from this 199 and 108. Now, J 2 enters at 1, finishes at 35, so J 2 finishes at 35, enters at 99 requires another 92, so 99 plus 92 is 191, so this is 191 and M 3 is free at 191 takes another 8 units, so 199 it comes out.

Now, using this partial sequence, we will now go back and try and find out the lower bound. So, as far as M 1 is concerned, M 1 will finish at 200, jobs 4 and 2 are fixed, so 1 and 3 are available. So, 1 and 3, once again 3 is the minimum with 66, so the first lower bound will be 200 plus 66. Now, M 2 is available at 191 for jobs 1 and 3, so 191 plus 1 and 3, 191, 192, 202, 238, so at 238, all the jobs can be over earliest in M 2. Then once again, it has to go for another minimum over 1 and 3, which is another 30.

So, this will be 191 plus sum of processing times of 1 and 3, which is 11 plus 36, which is 47 plus 30. And as far as M 3 is concerned, M 3 is available at 199 to do 1 and 3, so

199 plus 82 plus 30, which is 112, so 199 plus 112. Now, this is 266, now this is 191 plus 30 is 221, 228, 268, and this is 199 plus 112, 211, plus 100 is 311. So, the lower bound becomes 311 when we do this.

Now let us calculate the other lower bound for J 4 and J 3 also. Now, to do that, we have J 4 and J 3 here, so we will quickly do the completion times 199 and 108, for J 3 it is 88, so 1 plus 88 is 89, starts at 99 requires another 36, so 135, 99 plus 36. At 135, M 3 is free, so takes another 30, 135 plus 30 is 165, so again based on the lower bounds, so as far as M 1 is concerned, it will take 200. Now, 4 and 3 are already fixed, so 1 and 2 are available, so minimum over, this is 93, this is 100, so 200 plus 93, which is 293.

The earliest M 2 is available is 135 for 1 and 2, so 135 plus (Refer Slide Time: 41:05) this plus this is the earliest M 2 is over. Then the last job has to do one more operation, so minimum over 82 and 8, which is 8, so 135 plus 11 plus 92 plus 8 and M 3 available at 165 for 1 and 2. So, it has to do these two plus another 90, so 165 plus 90, 255, this is 293, so this is 100, 235 246, so the lower bound remains at 293. So now, we have all these then we still need to proceed to get to the solution.

So now, the smallest available number is still 266, so we can proceed from this to try and get some more solutions. So, we have already fixed two positions, so we could fix the third position now. So, J 4 and J 1 have been fixed, so now J 2 and J 3 are remaining, so we will say, we fix J 2 as the third job, we fix J 3 as the third job in the other sequence. So, we fix 3 positions now, so J 4 J 1 and J 2, so we can continue in the same way to get the lower bound for J 4 J 1 and J 2.

But, we observed that, there are only four jobs, we have already fix three positions, so the left over job has to automatically take the fourth position. So, instead of computing a lower bound, we compute the makespan for a full sequence, which is J 4 J 1 J 2 and J 3. So, J 4 J 1 J 2 and J 3, the other alternate sequence is J 4 J 1 J 3 and J 2, so the other alternate sequence is J 4 J 1 J 3 and J 2, because here we fix J 3 as the third job, so the left over job, which is J 2 will be the fourth job.

So, branching from here, we are able to get two full sequences, because in all these partial sequence, there is only one more job to go. So, we can evaluate this sequence or we can evaluate this sequence, we can evaluate either of them. So, let us first evaluate this sequence and try to learn a few things then we will come back to this sequence. So, we evaluate the full sequence here, we already have for J 4 and J 1. So, for J 4 and J 1, we already have here, so we write 199, 108, 78, 110, and 192.

Now, we look at J 3, now J 3 enters at 78 takes another 88, so 166. So at 166, M 2 is free, so it takes another 36 – 166, 172, 202. At 202, again M 3 is free, takes another 30, so 232, now J 2 enters at 166 plus 34 is 200, waits till 202 for M 2, takes another 92, so 202 plus 92 is 294. At 294, M 3 is free, so goes to M 3 and comes out at 302, now 302 is the makespan associated with this sequence. Now, we have a feasible solution with makespan equal to 302, 4 1 3 2 is a valid sequence, it is one out of the four factorial possible sequences for this problem.

So, it is feasible and has a makespan of 302, so the moment we have a feasible solution with a makespan of 302, we also understand that, the optimum now can be 302 or less. So, the feasible solution provides an upper bound to the optimum, so it provides an upper bound to the optimum and therefore, the optimum now is 302 or less. Now, with this additional information, we now observe that, if we look at this node from here, with fixing J 1 as the first job, we understand from here, that if we proceed down, the makespan that we will get can only be greater than or equal to 322.

So, we are now not interested in any solution, which has makespan 302 or more, because you want to minimize the makespan and we already have a solution with 302. So, there is no need for us now to proceed from here, because we are only going to get solutions, which are poorer than 302. So, we decide not to proceed from this, so we say that, we fathom this node, so this node is fathomed, because the lower bound here is greater than or equal to the current based upper bound of 302.

So, 302 acts as an upper bound, so we fathom a node when the lower bound in that node is greater than or equal to current based upper bound. Now, similarly the 333 is also fathomed for the same reason, that lower bound is greater than or equal to the current based upper bound, but we cannot do that to this 280. Now, this 311 can also be fathom, we normally use fathom or with an f, they mean the same. So, we could put an f and say that, it is fathomed for the same reason, that the lower bound is bigger than the current based upper bound.

We cannot proceed from here, because we have already reached the full sequence, so this node is fathomed by upper bound or by feasibility, because we have a feasible solution there, we cannot proceed any more. So, now we realize that, doing this computation has helped us actually indirectly, because we now do not have to proceed and evaluate sequences from here, we do not have to proceed from here.

But, there are still some nodes that are active which means, we can move from here, we can move from this 293, we can evaluate this. So, we now try and evaluate this. So we already know the sequence for J 4 J 1, so you get 1, 99, 108, 78, 110, and 192. Now, J 2 enters at 78, takes another 34, so 112 is when J 2 finishes here. At 112, machine 2 is available at 110 itself, so takes another 92 units, so 112 plus 92 is 204. At 204, M 3 is available, takes another 8 units, so at 212, this leaves.

Now, J 3 is the last job, J 3 enters M 1 at 112, 112 plus 88 is 200, waits till 204 and then visits M 2 at 204, requires another 36 time units, 204 plus 36 is 240. M 3 available at 240, so visits M 3, takes another 30 units and comes out at 270. Now, the makespan associated with this sequence is 270. Now this 270 is going to help us in many ways. Now, this node as I said, is fathomed by feasibility, this 270 is an upper bound and the moment we have solution with makespan equal to 270, it is better than the current best solution of 302, so we update.

So, we keep the best solution now as 270, now the moment we have a 270 solution, we are not interested in any solution, which are more than 270. So, we now start looking at all these nodes, now this tells us that, if we move from this direction, it is going to be 280 or more. So, we do not have to move from here, because we already have solution of 270, so this is again fathomed with lower bound greater than or equal to upper bound. Only difference is that, the upper bound has been updated, so from 302, it has become 270, therefore you can fathom this.

Now, proceeding here, which is one that is available, 293 is also be fathomed with lower bound greater than upper bound. So now, we are realized, that there is no active node in this network, so we cannot proceed any more from this network, because all the nodes have been fathomed. They have been fathomed either, because they give a feasible solution or they have been fathomed, because the lower bound associated with that load is greater than or equal to the current best upper bound.

So, there is no more node to proceed, the algorithm terminates, the Branch and Bound algorithm terminates and the best feasible solution at this point, which is the solution with 270 is the optimum solution. So, the optimum solution is J 4 J 1 J 2 J 3 with 270 as the optimum makespan. So, this is how the Branch and Bound algorithm works, so we branch by creating sequences and partial sequences, we evaluate lower bounds.

And then as we move down the tree, we evaluate feasible solutions and upper bounds and try to fathom the nodes based on two things in this case, based on a feasible solution and based on the condition that lower bounds is greater than the upper bound. Now, this is called, is like an Implicit Enumeration algorithm, we have actually evaluated all the four factorial, but not explicitly. We have explicitly evaluated only two out of the four factorial, but based on these, the way we worked out the branch and bound tree, we have implicitly evaluated the remaining.

And we have said that, the remaining cannot give a solution, which is better than 270, so it is an implicit enumeration kind of an algorithm, but in the very worst case, we may have to evaluate all the n factorial, which are there. So, this is how the Branch and Bound algorithm works for solving the makespan minimization problem in the flow shop. And other thing that we could have done to cut down the computation is this, somewhere here when we fix two jobs J 4 and J 1, we evaluated the lower bound and we proceed here.

When we fix three jobs, we said there is only one vacant position, so the fourth job will automatically come there and therefore, we got a feasible solution here. In Branch and Bound algorithm, it is customary to actually write here to evaluate the feasible solution. Because, when we fix these two, there are only two more positions remaining with two more jobs remaining and they can be filled only in two ways.

So, right here we could have said, this will give rise to two feasible solutions, which is J 4 J 1 J 2 J 3, J 4 J 1 J 3 J 2 which means, right here we could have evaluated this without doing this and without doing this 266. So, that way you can speed up the Branch and Bound algorithm, when you have only two more positions remaining, you do not evaluate the lower bound, instead you fill it in two ways and evaluate two feasible solutions.

So, this is how the Branch and Bound algorithm works in order to minimize the makespan. This can be extended to general n job m machine based on the idea, we have 3 machines, so we calculated 3 lower bounds and choose the best. If you have m machines, you will choose m such calculations for every node and choose the lower bound. Now, in case, obviously we understand that the Branch and Bounds algorithm is time consuming. And as the problem size increases, there are times the Branch and Bounds algorithm is unable to give the optimum solutions within a reasonable computational effort. So, in such cases, what do we do is the next question that we have to address, we will address that question in the next lecture.