

Operations Research
Prof. Kusum Deep
Department of Mathematics
Indian Institute of Technology - Roorkee

Lecture – 34
Processing n Jobs Through m Machines

Good morning students, we are learning the topic of sequencing and scheduling and this is lecture number 34 and this is the fourth case that is processing of n jobs on m machines. You remember we have studied the earlier three cases where we had n jobs on 2 machines, n jobs on 3 machines and then 2 jobs on m machines. This processing of n jobs and m machines is the most generalized case of the sequencing and the scheduling problem.

(Refer Slide Time: 01:16)

Processing n jobs through m machines

There are n jobs identified by 1, 2, 3, ..., n

Each job is to be processed through m machines say

$M_1, M_2, M_3, \dots, M_m$ in the order $M_1 M_2 M_3 \dots M_m$

No passing of jobs is allowed

T_{ij} = time taken by i^{th} machine to complete j^{th} job.

So, let us look at this case more closely, there are n jobs and we are going to identify them by 1, 2, 3 and n and each job has to be processed through m machines. Let us say these m machines are called M_1, M_2 and M_m and the order that has to be adopted is M_1, M_2, M_3 and M_m . No passing of jobs is allowed that is the no passing rule is to be adopted. Let us denote by T_{ij} as the time taken by the i^{th} machine to complete the j^{th} job.

(Refer Slide Time: 02:14)

Step 1:

Find $\min(T_{1j})$ and $\min(T_{mj})$ over all j .

Also find $\max[T_{2j}, T_{3j}, \dots, T_{(m-1)j}]$

$T_{1j} = \text{first}$
 $T_{mj} = \text{last}$ } for all j

Step 2: Now check (over j)

(i) $\min T_{1j} \geq \max T_{ij}$ for $i=2, 3, \dots, m-1$ or

(ii) $\min T_{mj} \geq \max T_{ij}$ for $i=2, 3, \dots, m-1$

|| should be satisfied

(If these are not satisfied, then method fails)

So, the first step that has to be worked out is; we have to check whether the following conditions is there or not, find the minimum of T_{1j} and the minimum of T_{mj} , overall j , also find the maximum of T_{2j} , T_{3j} , T_{m-1j} . Please note that this T_{1j} means the first one and T_{mj} means the last one, in each j , this is for all j and these ones are the intermediate ones, so for this you have to find out the minimum and for this also is the minimum.

But for the intermediate ones you have to find the maximum, this is the step number 1, then comes the step number 2; now check over j , first whether minimum of $T_{1j} \geq$ maximum of T_{ij} for $i = 2, 3$ up to $m - 1$. Again, please note that this is for the intermediate ones, the first one and the last one has to be left out and the second one is minimum of T_{mj} is \geq maximum of T_{ij} for $i = 2, 3$ up to $m - 1$.

Again, this has to be checked for the intermediate ones, the first and the last one have to be left out so, therefore we have to check $T_{1j} \geq$ maximum of T_{ij} , and second is minimum of $T_{mj} \geq$ maximum of T_{ij} ; whereas, this i is going to vary only for the intermediate ones. If these conditions are not satisfied then, the method fails. So it is important that these conditions should be satisfied.

(Refer Slide Time: 05:03)

Step 3: Convert the m machine problem into an equivalent 2 machine problem by introducing two fictitious machines G & H:

$$\rightarrow T_{Gj} = T_{1j} + T_{2j} + \dots + T_{(m-1)j} \quad \text{and}$$

$$\rightarrow T_{Hj} = T_{2j} + T_{3j} + \dots + T_{mj}$$

Now determine the optimal sequence of n jobs through 2 machines by using two machines optimal sequence algorithm.

So, the step 3 says convert the m machine problem into an equivalent 2 machines problem by introducing 2 fictitious machines G and H, you remember that we have the 2 machines case, so in this n machine case also, we will convert it into the case where we have 2 machines G and H and also once you do this, you need to calculate the times. So, the first one will be $T_{Gj} = T_{1j} + T_{2j} + T_{m-1j}$, so this is the first machine.

The second machine is T_{Hj} , so the times have to be calculated again, T_{2j} , T_{3j} and finally T_{mj} , they have to be all added, please note in the G case (in the first one) last term is missing i.e. the mth term is missing; and in the second case, the first term is missing, so that is how you have to calculate the times for the two fictitious machines G and H.

Now, determine the optimal sequence of n jobs through 2 machines by using the previous case that we have studied that is the 2 machines optimal sequence algorithm so, now the problem has been reduced to the 2 machines case.

(Refer Slide Time: 07:04)

The following points must also be noted:

- (i) If in place of conditions given in step 3, $T_{2j} + T_{3j} + \dots + T_{mj} = C$ (a fixed positive constant) for all $j = 1, 2, \dots, n$, then also algorithm for determining optimal sequence for n jobs and 2 machines M_1 and M_m in the order M_1 and M_m be used.
- (ii) If, in addition to the condition given in step 3, $T_{1j} = T_{mj}$ and $T_{Gj} = T_{Hj}$ for $i = 1, 2, \dots, n$, then a number of optimal sequences will exist. // multiple solutions

Before you do that the following points must be noted, first of all if in place of the conditions given in step 3 that is $T_{2j} + T_{3j} + T_{mj} = C$ (a fixed positive constant) for all $j = 1, 2$ up to n , then also the algorithm for determining the optimal sequence of n jobs and 2 machines M_1 and M_2 in the order M_1 and M_2 can be used. So, here is another interesting condition that is $T_{2j} + T_{3j} + T_{mj} = C$ (a fixed positive constant). But this has to be satisfied for all $j = 1, 2$ up to n then, also the 2 machines case can be used and the second point that has to be noted is; if in addition to the conditions given in step 3, $T_{1j} = T_{mj}$ and $T_{Gj} = T_{Hj}$ for $i = 1, 2$ up to n , then a number of optimum sequences will exist so, this is the case of multiple solutions.

(Refer Slide Time: 08:48)

- (iii) The above procedure is applicable to only such sequencing problems in which minimum time (or cost) of processing the jobs through first and/or the last machine is greater than or equal to the time (or cost) of processing

And the third point says the above procedure is applicable to only such sequencing problems in which the minimum time or of course the cost as may be given in the problem, of processing

the jobs through first and or the last machine is greater than or equal to the time or cost of processing.

(Refer Slide Time: 09:20)

Ex 1: Solve the sequencing problem giving an optimal solution when passing is not allowed.

		Jobs j				
		A	B	C	D	E
Machines i	M ₁	11	13	9	16	16
	M ₂	4	3	5	2	6
	M ₃	6	7	5	8	4
	M ₄	15	8	13	9	11

So, let us look at an example, solve the sequencing problem given in the optimum solution when passing is not allowed, so here you see we have four machines; M₁, M₂, M₃ and M₄ and there are the jobs A, B, C, D and E. Now, what do we find; the table that has been given is the cost coefficients; 11, 13, 9, 16, 16, 4, 3, 5, 2, 6, 6, 7, 5, 8, 4, 15, 8, 13, 9 and 11.

(Refer Slide Time: 10:20)

Sol: Here these conditions are satisfied:

$$\min T_{1j} \geq \max T_{ij}$$

(because $9 > 6$ for $i=2$; $9 > 8$ for $i=3$)

$$\min T_{4j} \geq \max T_{ij}$$

(because $8 > 6$ for $i=2$; $8 = 8$ for $i=3$)

Convert into 5 jobs and 2-machine problem.

Also, $M_{2j} + M_{3j} = 10$, a fixed positive constant
for all $j = 1, 2, \dots, 5$.

Now, in order to solve this problem, we have to check whether the conditions are satisfied, what are those conditions? Minimum of $T_{1j} \geq$ maximum of T_{ij} , why is that so? Let us see minimum of T_{1j} is 9, in the first row here you see it is 9 and the maximum of T_{ij} is 6 for $i = 2$

and 8 for $i = 3$. For $i = 2$, it is 6 and for $i = 3$, it is 8, right, then comes the second condition; minimum of $T_{4j} \geq$ maximum of T_{ij} .

Why is that so? Because minimum of T_{4j} is 8, this is ≥ 6 for $i = 2$, and it is ≥ 8 for $i = 3$, it is equal so, here this is equal but since the inequality says that we are allowed to use greater than or equal to so, both the conditions are acceptable and since the conditions are acceptable therefore, we can convert into the 5 jobs and 2 machine case.

Also, in addition to the above conditions, we note the additional conditions that $M_{2j} + M_{3j} = 10$, this is a fixed positive constant for all $j = 1, 2$ up to 5, $M_{2j} + M_{3j} = 10$, $M_{2j} + M_{3j} = 10$.

(Refer Slide Time: 13:10)

So, problem will reduce to an optimal sequence for
5 jobs and 2 machines M_1 and M_4 in the order
 M_1M_4

Optimal sequence $C \rightarrow A \rightarrow E \rightarrow D \rightarrow B$.

The sequencing schedule is :

So, therefore the problem will reduce to an optimum sequence for 5 jobs and 2 machines, M_1 and M_4 in the order M_1M_4 and the optimum sequence can be obtained as before that is the optimum sequence is C, A, E, D and B.

(Refer Slide Time: 13:43)

		Machines			
		M ₁	M ₂	M ₃	M ₄
	C	0-9	9-14	14-19	19-32
Jobs	A	9-20	20-24	24-30	32-45
	E	20-36	36-42	42-46	46-57
	D	36-52	52-54	54-62	62-71
	B	52-65	65-68	68-75	75-83

Total elapsed time is 83 hrs.

And also the sequencing scheduled can be obtained like this so, you have this is the sequence that we got, C, A, E, D and B. The first machine M₁ will begin at 0, the time in will be 0 timeout will be 9 and then it will go to the M₂ machine, so it will be time in will be 9, time out will be 14, for M₃ it will be passing on to 14, so time in will be 14, time out will be 19 and again, the time in for M₄ will be 19 and time out will be 32.

Remember how did we get this 9, from 0 to 9; this is the data that is given in the table, so that is how we know the time in and the time out and like this, you can check for the remaining ones as well, please note for example, the time out for A job on M₁ is 20 and the time in for M₂ is also 20 similarly, time out is 24 for M₂ and time in is 24 for M₃, however this 30 is different and this 32 is different, they are not equal. Do you know the reason for this? The reason for this is that once this 30 has been obtained that is the timeout for this A job has been done for M₃ but this M₄ is still working, so this there is an gap over here that is 32 - 2 that is 30. So, this is the idle time over here, there is a gap over here because the machine is not free, and that is a reason why there is this gap. Similarly, you can also observe that at many places this gap is provided, because the machine is not free and as before we have learnt, we can calculate the elapsed time as 83 because this is the total time required when all the jobs are completed on all the machines.

(Refer Slide Time: 16:39)

Ex 2: There are 4 jobs each of which has to go through the machines M_i , $i = 1, 2, \dots, 6$ in the order $M_1 M_2 \dots M_6$. Processing times are

		Machines					
		M_1	M_2	M_3	M_4	M_5	M_6
	A	20	10	9	4	12	27
Jobs	B	19	8	11	8	10	21
	C	13	7	10	7	9	17
	D	22	6	5	6	10	24

Now, let us look at another example, there in this example, there are 4 jobs, each of which has to go through the machines M_i where i goes from 1, 2 up to 6 in the order M_1, M_2, M_3, M_4, M_5 and M_6 . The processing times for all the jobs on all the machines is given in this table; 20, 10, 9, 4, 12, 27, 19, 8, 11, 8, 10, 21, 13, 7, 10, 7, 9, 17, 22, 6, 5, 6, 10 and 24.

(Refer Slide Time: 17:47)

Determine a sequence of these four jobs which minimizes the total elapsed time T .

Sol:

$$\rightarrow \min T_{1j} \geq \max T_{ij} \text{ for } i=2, 3, \dots, m-1$$

$$\min T_{mj} \geq \max T_{ij} \text{ for } i=2, 3, \dots, m-1$$

$$\text{And } \min T_{1j} = 13 \text{ and } \min T_{6j} = 17$$

$$\max T_{2j} = 10$$

$$\max T_{3j} = 11$$

$$\max T_{4j} = 8$$

$$\max T_{5j} = 12$$

Now, with this data we need to determine a sequence of the 4 jobs which minimizes the total elapsed time T , so we have to look at the conditions given in our step number 1. What is the first condition? The first condition is we have to check whether minimum of $T_{1j} \geq$ maximum of T_{ij} , for $i = 2, 3, m - 1$ and as you can see that the second condition is minimum of $T_{mj} \geq$ maximum of T_{ij} , for $i = 2, 3, m - 1$. Now, the minimum of T_{ij} is 13 and the minimum of T_{6j} is 17, let us just look at it.

Minimum of T_{1j} is 13, here it is and the minimum of T_{6j} is 17. Also, maximum of T_{2j} is 10 and maximum of T_{3j} is 11 and also maximum of T_{4j} is 8 and the maximum of T_{5j} is 12, so once we have identified these entries now, let us see whether this condition is satisfied or not and similarly, whether this condition is satisfied or not.

(Refer Slide Time: 20:01)

So the conditions are satisfied.

Converted into 4-job and 2-machine problem.

Choose two fictitious machines G and H so that:

$$T_{Gj} = T_{1j} + T_{2j} + \dots + T_{m-1j} \quad \cancel{\neq}$$

$$T_{Hj} = T_{2j} + T_{3j} + \dots + T_{mj}$$

And we find that the conditions are satisfied and therefore, now we can convert this problem into the 4 job and 2 machine case therefore, we will have two fictitious machines G and H such that we can calculate their times as follows; for the first fictitious machine G, the times can be calculated as $T_{Gj} = T_{1j} + T_{2j} + \dots + T_{m-1j}$ and similarly, for the second machine, we have the times $T_{Hj} = T_{2j} + T_{3j} + \dots + T_{mj}$.

Again, please note that here in the first case, the last entry is missing and in the second case, the first entry is missing.

(Refer Slide Time: 21:13)

The 4-job and 2-machine problem is :

		Jobs			
		A	B	C	D
Machines	G	55	56	46	49
	H	62	58	50	41

An optimal sequence is $C \rightarrow A \rightarrow B \rightarrow D$.

So, the 4 jobs and 2 machine problem is as follows; we have two fictitious machines, G and H and we have the 4 jobs A, B, C, D as follows; 55, 56, 46, 49 and for the G machine 62, 58, 50 and 41 and once you get this two machines case, then you can use the previous case to solve it and the optimum sequence that you get is C, A, B and D; this is the way in which this most generalized situation of n jobs on m machines can be used to solve this generalized problem.

(Refer Slide Time: 22:15)

The sequencing schedule is:

		Machines					
		M ₁	M ₂	M ₃	M ₄	M ₅	M ₆
Jobs	C	0-13	13-20	20-30	30-37	37-46	46-63
	A	13-33	33-43	43-52	52-56	56-68	68-95
	B	33-52	52-60	60-71	71-79	79-89	95-116
	D	52-74	74-80	80-85	91-101	91-101	116-130

Total elapsed time is 130 hrs.

So, the sequencing scheduled is as follows, that is we have the 6 machines; M₁, M₂, M₃, M₄, M₅, M₆ and the 4 jobs A, B, C and D and we will calculate the time in and time out on each of the machines depending upon the data that is given in the problem. So, first one is for the job C, see this is the optimum sequence, look at it, what we have obtained, C, A, B, D, therefore C, A, B, D is the right sequence.

And on the machine 1, time in is 0, timeout is 13, on the machine 2 time in is 13, timeout is 20, on the machine M_3 time in is 20, timeout is 30 and for M_4 ; 30 is the time in, timeout is 37 and like this for M_5 , M_6 . Also, the time in and time out is depending upon the data that has been given. Now, you will observe again the gap that is obtained wherever the time out and the time in of the previous one is not matching. You find that the total elapsed time is 130 hours because the time out of the last machine M_6 on the last job that is D is 130 hours, so like this we can complete the entire sequencing scheduled of all the machines and all the jobs.

So, students with that we come to an end of this topic on sequencing and scheduling of various jobs on various machines, we have seen the four cases and accordingly the algorithms that are required to solve them. Thank you.