INTRODUCTION

The short-term schedules show an optimal order (sequence) and time in which jobs are processed as well as show timetables for jobs, equipment, people, materials, facilities and all other resources that are needed to support the production plan. The schedules should use resources efficiently to give low costs and high utilisations. Other purpose of scheduling are, minimising customers wait time for a product, meeting promised delivery dates, keeping stock levels low, giving preferred working pattern, minimising waiting time of patients in a hospital for different types of tests and so on.

The general scheduling or sequencing problem may be described as: Let there be n jobs to be performed one at a time on each of m machines. The sequence (order) of the machines in which each job should be performed is given. The actual or expected time required by the jobs on each of the machines is also given. The general sequencing problem, therefore, is to find the sequence out of $(n!)^m$ possible sequences which minimise the total elapsed time between the start of the job in the first machine and the completion of the last job on the last machine.

In particular, if n = 3 and m= 3, then total number of possible sequences will be $(3!)^3 = 216$. Theoretically, it may be possible to find optimum sequence but it will require a big computational time. Thus, one should adopt sequencing technique.

To find optimum sequence we first calculate the total elapsed time for each of the possible sequences. As stated earlier, even if values of m and n are very small, it is difficult to get the desired sequence with total minimum elapsed time. However, due to certain rules designed by Johnson, the task of determining an optimum sequence has become quite easy.

NOTATIONS, TERMINOLOGY AND ASSUMPTIONS

Notations

t_{ij} = Processing time (time required) for job i on machine j.

T = Total elapsed time for processing all the jobs. This includes idle time, if any.

I_{ij} = Idle time on machine j from the end of job (j - 1) to the start of job i.

Terminology

- **Number of Machines:** The number of machines refer to the number of service facilities through which a job must pass before it is assumed to be completed.
- **Processing Time:** It is the time required by a job on each machine.
- **Processing Order:** It refers to the order (sequence) in which machines are required for completing the job.
- **Idle Time on a Machine:** It is the time for which a machine does not have a job to process, i.e idle time from the end of job (i-1) to the start of job i.
- **Total Elapsed Time:** It is the time interval between starting the first job and completing the last job including the idle time (if any) in a particular order by the given set of machines.
- **No Passing Rule:** It refers to the rule of maintaining the order in which jobs are to be processed on given machines. For example, if n jobs are to be processed on two

machines M_1 and M_2 in the order M_1 M_2, then each job should go to machine M_1 first and then to M_2.

Assumptions

1. The processing time on different machines are exactly known and are independent of the order of the jobs in which they are to be processed.
2. The time taken by the job in moving from one machine to another is negligible.
3. Once a job has begun on a machine, it must be completed before another job can begin on the same machine.
4. All jobs are known and are ready for processing before the period under consideration begins.
5. Only one job can be processed on a given machine at a time.
6. Machines to be used are of different types.
7. The order of completion of jobs are independent of the sequence of jobs.

PROCESSING n JOBS THROUGH TWO MACHINES

Let there be n jobs, each of which is to be processed through two machines, M_1 and M_2 in the order M_1 M_2, i.e. each job has to be passed through the same sequence of operations. In other words, a job is assigned on machine M_1 first and after it has been completely processed on machine M_1, it is assigned to machine M_2. If the machine M_2 is not free at the moment for processing the same job, then the job has to wait in waiting line for its turn on machine M_2, i.e. passing is not allowed.

Since passing is not allowed, therefore, machine M_1 will remain busy in processing all the n jobs one-by-one which machine M_2 may remain idle time of the second machine. This can be achieved only by determining sequence of n jobs which are to be processed on two machines M_1 and M_2. The procedure suggested by Johnson for determining the optimal sequence can be summarised as follows:

The Algorithm

Step 1 List the jobs along with their processing times on each machine in a table as shown below:

Processing Time on Machine	Job Number 1	 2	 3	 n
M_1	t_{11}	t_{12}	t_{13}	t_{1n}
M_2	t_{21}	t_{22}	t_{23}	t_{2n}

Step 2 Examine the columns for processing times on machines M_1 and M_2, and find the smallest processing time in each column, i.e find out, min. (t_{1j}, t_{2j}) for all j.

Step 3(a) If the smallest processing time is on machine M_1, then schedule the job as early as possible without moving jobs already schedules, i.e place the job in the first available position in the sequence. If the processing time is on machine M_2, then schedule the job as late as possible without moving any jobs already scheduled, i.e. place the job in the last available position in the sequence.

b. If there is a tie in selecting the minimum of all the processing times, then there may be three situations:

a. Minimum among all processing times is same for the machine i.e. min (t_{1j}, t_{2j}) = t_{2k} = t_{2r}, then process the kth job first and the rth job last.
b. If the tie for the minimum occurs among processing times t_{1j} on machine M_1 only, then select the job corresponding to the smallest job subscript first.
c. If the tie for the minimum occurs among processing times t_{2j} on machine M_2, then select the job corresponding to the largest job corresponding to the largest job subscript last.

Step 4 Remove the assigned jobs from the table. If the table is empty, stop and go to Step 5. Otherwise, got to Step 2.

Step 5 Calculate idle time for machines M_1 and M_2:
a. Idle time for machine M_1 = (Total Elapsed Time) - (Time when the last job in a sequence finishes on machine M_1)
b. Idle time for machine M_2 = Time at which the first job in a sequence finishes on machine M_1 + $\sum_{j=2}^{n}$ {(Time when the jth job in a sequence starts on machine M_2) - (Time when the (j - 1)th job in a sequence finishes on machine M_2)}.

Step 6 The total elapsed time to process all jobs through two machines is given by
Total Elapsed time = Time when the nth job in a sequence finishes on Machine M_2.
= $\sum_{j=1}^{n} M2j + \sum_{j=1}^{n} I2j$
where M_{2j} = Time required for processing jth job on machine M_2.
I_{2j} = Time for which machine M_2 remains idle after processing (j - 1)th job and before starting work in jth job.
EXAMPLE 1
Find the sequence that minimises the total elapsed time required to complete the following tasks on two machines:

Task	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**
Machine I	**2**	**5**	**4**	**9**	**6**	**8**	**7**	**5**	**4**
Machine II	**6**	**8**	**7**	**4**	**3**	**9**	**3**	**8**	**11**

Solution:
The smallest processing time between the two machines is 2 which corresponds to task A on Machine I. Thus, task A is scheduled as early as possible to give the sequence as shown below:

A								

After the task A has been set for processing first, we are left with 8 tasks and their processing times as given below:

Task	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**
Machine I	**5**	**4**	**9**	**6**	**8**	**7**	**5**	**4**
Machine II	**8**	**7**	**4**	**3**	**9**	**3**	**8**	**11**

The minimum processing time in this reduced problem is 3 which corresponds to task E and G both on machine II. Since the corresponding processing time of task E on machine I is less than the corresponding processing time of task G on machine I, therefore, task E will be scheduled in the last and task G shall be scheduled before it. Tasks E and G will not be considered further. Thus, current partial sequence of scheduling tasks becomes:

A							G	E

A set of processing times now gets reduced to:

Task	**B**	**C**	**D**	**F**	**H**	**I**
Machine I	**5**	**4**	**9**	**8**	**5**	**4**
Machine II	**8**	**7**	**4**	**9**	**8**	**11**

The smallest processing time in this reduced problem is 4, which corresponds to task C and I on machine I and to task D on machine II. Thus task C will be placed in the second sequence cell and task I in the third sequence cell and task D in the sequence cell before task G. The entries of the partial sequence are now:

A	C	I				D	G	E

The set of processing time now gets reduced as follows:

Task	**B**	**F**	**H**
Machine I	**5**	**8**	**5**
Machine II	**8**	**9**	**8**

the smallest processing time in this reduced problem is 5, which corresponds to tasks B and H both on machine I. Since the corresponding processing times of B and h on machine II is same, therefore, either of these two tasks can be placed in fourth and fifth sequence cells. Thus, it indicates an alternative optimal sequence. the optimal sequences are, therefore, given below:

A	C	I	B	H	F	D	G	E

A	C	I	H	B	F	D	G	E

The minimum elapsed time for machines I and II is calculated as shown in Table 1.

Task Sequence	Machine I		Machine II	
	Time In	Time Out	Time In	Time Out
A	0	2	2	8
C	2	6	8	15
I	6	10	15	26
B	10	15	26	34
H	15	20	34	42
F	20	28	42	51
D	28	37	51	55
G	37	44	55	58
E	44	50	58	61

In table 1, the minimum elapsed time, i.e time from start of task A to completion of last task E is 61 hours. During this time the machine I remains idle for 61 - 50 = 11 hours. The idle time for machine II is equal to the time at which the first task A in the sequence finishes on machine I plus the last task E in the sequence starts on machine II minus the last but one task G finishes on machine II, i.e 2 + 58 - 58 = 2 hours.

PROCESSING n JOBS THROUGH THREE MACHINES

Johnson provides an extension of his procedure to the case in which there are three instead of two machines. Each job is to be processed through three machines M_1, M_2 and M_3. The list of jobs with their processing times is given below. An optimal solution to this problem can be obtained if either or both of the following conditions hold good.

Processing time on Machine	Job Number			
	1	2	3	4
M_1 M_2 M_3	t_{11} t_{21} t_{31}	t_{12} t_{22} t_{32}	t_{13} t_{23} t_{33}	t_{1n} t_{2n} t_{3n}

1. The minimum processing time on machine M_1 is at least as great as the maximum processing time on machine M_2, that is,

$$\min t_{1j} \geq \max t_{1j}, \qquad j = 1, 2, 3, ...n$$

2. The minimum processing time on machine M_3 is at least as great as the maximum processing time on machine M_2, that is

$$\min t_{3j} \geq \max t_{2j}, \qquad j = 1, 2, 3, ...n$$

If either or both the above conditions hold good, then the steps of the algorithm can be summarised in the following steps:

THE ALGORITHM

Step 1: Examine processing times of given jobs on all three machines and if either one or both the above conditions hold, then go to step 2, otherwise the algorithm fails.

Step 2: Introduce two fictitious machines, say G and H with corresponding processing times given by

i. $t_{Gj} = t_{1j} + t_{2j}, \; j = 1, 2, 3, ...n.$

that is, processing time on machine G is the sum of the processing times on machines M_1 and M_2, and

ii. $t_{Hj} = t_{2j} + t_{3j}, \; j = 1, 2, 3, ...n.$

that is, processing time on machine H is the sum of the processing times on machines M_2 and M_3.

Step 3: Determine the optimal sequence of jobs for this n-job, two machine equivalent sequencing problem with the prescribed ordering GH in the same way as discussed earlier.

EXAMPLE

Find the sequence that minimises the total time required in performing the following jobs on three machines in the order ABC. Processing times (in hours) are given in the following table:

Job	1	2	3	4	5
Machine A	8	10	6	7	11
Machine B	5	6	2	3	4
Machine C	4	9	8	6	5

Solution: Here, min (t_{Aj}) = 6; Min (t_{Cj}) = 4; max (t_{Bj}) = 6. Since min (t_{Aj}) ≥ (t_{Bj}) for all j is satisfied, the given problem can be converted into a problem of 5 jobs and two machines. The processing time on two fictitious machines G and H can be determined by the following relationships:

$t_{Gj} = t_{Aj} + t_{Bj}$, j = 1, 2, 3, ...n.

and $t_{Hj} = t_{Bj} + t_{Cj}$, j = 1, 2, 3, ...n.

The processing times for the new problem are given below:

Job	1	2	3	4	5
Machine G Machine H	8 + 5 = 13 5 + 4 = 9	10 + 6 = 16 6 + 9 = 15	6 + 2 = 8 2 + 8 = 10	7 + 3 = 10 3 + 6 = 9	11 + 4 = 15 4 + 5 = 9

When the algorithm described for n jobs on two machines is applied to this problem, the optimal sequence so obtained is given by

3	2	5	1	4

The total minimum elapsed time is given in Table 1:

Job Sequence	Machine A		Machine B		Machine C	
	Time In	Time Out	Time In	Time Out	Time In	Time Out
3	0	6	6	8	8	16
2	6	16	16	22	22	31
5	16	27	27	31	31	36
1	27	35	35	40	40	44
4	35	42	42	45	45	51

Table 1 indicates that the minimum total elapsed time is 51 hours. The idle time for machines A, B and C is 9 (=51 - 42) hours, 6 (= 51 - 45) hours and 9 (= 8 - 0) + (45 - 44) hours, respectively.

PROCESSING n JOBS THROUGH m MACHINES

Let there be n jobs, each of which is to be processed through m machines, say M_1, M_2, ...M_m in the order M_1, M_2, ...M_m. The optimal solution to this problem can be obtained if either or both of the following conditions hold good.

(a) Min $\{t_{1j}\}$ ≥ Max $\{t_{1j}\}$; j = 2, 3, ... m - 1

and or (b) Min $\{t_{mj}\}$ ≥ Max $\{t_{ij}\}$; j = 2, 3, ... m - 1

that is, the minimum processing time on machines M_1 and M_m is as great as the maximum processing time on any of the remaining (m - 1) machines.

If either or both these conditions hold good, then the steps of the algorithm can be summarised in the following steps:

Step 1: Find, Min $\{t_{1j}\}$, Min $\{t_{mj}\}$ and max $\{t_{ij}\}$and verify above conditions. If either or both the conditions mentioned above hold, then go to step 2. Otherwise the algorithm fails.

Step 2: Convert m-machine problem into 2-machine problem by introducing two fictitious machines, say

(i) $t_{Gj} = t_{1j} + t_{2j} + t_{3j} + ...+ t_{m-1j}$ j = 1, 2, 3, ...n.

i.e. processing time of n-jobs on machine G is the sum of the processing times on Machines M_1, M_2 ......M_{m-1j}

(ii) $t_{Hj} = t_{2j} + t_{3j} + t_{4j} + ...+ t_{mj}$ j = 1, 2, 3, ...n.

i.e. processing time of n-jobs on machine H is the sum of the processing times on Machines M_1, M_2 ......M_{mj}.

Step 3: The new processing times so obtained can now be used for solving n-job, two machines equivalent sequencing problem with the prescribed ordering HG in the same way as

$t_{2j} + t_{3j} + ... + t_{m-1j} = k$ (constant)

for all j = 1, 2, 3, ... m - 1, then the optimal sequence can be obtained for n-jobs and two machines M_1 and M_m in the order M_1 M_m as usual.

2. If $t_{1j} = t_{mj}$ and $t_{Gj} = t_{Hj}$, for all j = 1, 2, 3, ...n, then total number of optimal sequences will be n and total minimum elapsed time in these cases would also be the same.

3. The method described above for solving n-jobs and m-machines sequencing problem is not a general method. It is applicable only to certain problems where the minimum cost (or time) of processing the jobs through first and/or last machine is more than or equal to the cost (or time) of processing the jobs through remaining machines.

EXAMPLES

1. Find an optimal sequence for the following sequencing problems of four jobs and five machines when passing is not allowed of which processing time (in hours) is given below:

Job	Machines				
	M_1	M_2	M_3	M_4	M_5
A	7	5	2	3	9
B	6	6	4	5	10
C	5	4	5	6	8
D	8	3	3	2	6

Also find the total elapsed time.

Solution: Here,

Min $(t_{M1,j}) = 5 = t_{M1,C}$

Min $(t_{M5,j}) = 6 = t_{M5,D}$

and Max $\{t_{M2,j}, t_{M3,j}, t_{M4,j}\} = \{6, 5, 6\}$ respectively.

Since the condition of Min $(t_{M5,j}) \geq$ Max $\{t_{M2,j}, t_{M3,j}, t_{M4,j}\}$ is satisfied, therefore the given problem can be converted into a four jobs and two machines problem as G and H. The processing times of four jobs denoted by t_{Gj} and t_{Hj} on G and H, respectively are as follows:

Job	A	B	C	D
Machine G	17	21	20	16
Machine H	19	25	23	14

where $t_{Gj} = \sum_{i=1}^{m-1} tij$ and $t_{Hj} = \sum_{i=2}^{m} tij$.

Now using the optimal sequence algorithm, the following optimal sequence can be obtained.

A	C	B	D

The total elapsed time corresponding to the optimal sequence can be calculated as shown in Table 1, using the individual processing times given in the original problem.

Table 1 shows that the minimum total elapsed time is 51 hours. The idle time for machines M_1, M_2, M_3, M_4 and M_5 is 25, 33, 37 and 18 hrs respectively.

Table 1 Minimum Elapsed Time

Job Sequence	Machine				
	M_1	M_2	M_3	M_4	M_5
A	0 - 7	7 - 12	12 - 14	14 - 17	16 - 26
B	7 - 12	12 - 18	16- 21	21 - 27	27 - 35

C	12 - 18	18 - 24	24 - 28	28 - 33	35 - 45
D	18 - 26	26 - 29	29 - 32	33 - 35	45 - 51

2. Solve the following sequencing problem giving an optimal solution when passing is not allowed.

Machine	Job				
	A	B	C	D	E
M_1	11	13	9	16	17
M_2	4	3	5	2	6
M_3	6	7	5	8	4
M_4	15	8	13	9	11

Solution: From the data of the problem it is observed that

Min $(t_{M1,j}) = 9 = t_{M1,C}$

Min $(t_{M4,j}) = 8 = t_{M4,B}$

and Max $\{ t_{M2,j}\} = 6 =$, $t_{M2,E;}$ Max $\{ t_{M3,j}\} = 8 =$, $t_{M2,D}$.

Since both the conditions

Min $(t_{M1,j}) \geq$ Max $\{ t_{M2,j;} t_{M3,j}\}$; j = 1, 2, ..., 5

are satisfied, therefore given problem can be converted into a 5-jobs and 2-machine problem as G and H.

Further, it may be noted that, $t_{M2,j} + t_{M3,j} = 10$ (a fixed constant) for all j (j = 1, 2, ..., 5).

Thus the given problem is reduced to a problem of solving 5-jobs through 2-machines M_1 and M_4 in the order M_1 M_4. This means machines M_2 and M_4 will have no effect on the optimality of the sequences.

The processing times of 5 jobs on machine M_1 and M_4 is as follows:

Job	A	B	C	D	E
Machine M_1	11	13	9	16	17
Machine M_4	15	8	13	9	11

Now using the algorithm described earlier, the optimal sequence so obtained as follows:

C	A	E	D	B

The total elapsed time corresponding to the optimal sequence is 83 hours as shown in table 1, using the individual processing times given in the original problem:

Table 1 Minimum Total Elapsed Time

Job Sequence	Machine				
	M_1	M_2	M_3	M_4	
C	0 - 9	9 - 14	14 - 19	19 - 32	
A	9 - 20	20 - 24	24 - 30	32 - 45	
E	29 - 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	

PROCESSING TWO JOBS THROUGH m MACHINES

Let there be two jobs A and B each of which is to processed on m machines say M_1, M_2, ..., M_m in two different orders. The technological ordering of each of the two jobs through m machines is known in advance. Such ordering may not be same for both the jobs. The exact or expected processing times on the given machines are known. Each machine can perform only one job at a time. The objective is to determine an optimal sequence of processing the jobs so as to minimise total elapsed time.

The optimal sequence in this case can be obtained by using graph. The procedure can be illustrated by taking examples.

Example 1: Use the graphical method to minimise the time needed to process the following jobs on the machines shown, i.e. each machine find the job which should be done first. Also calculate the total elapsed time to complete both jobs.

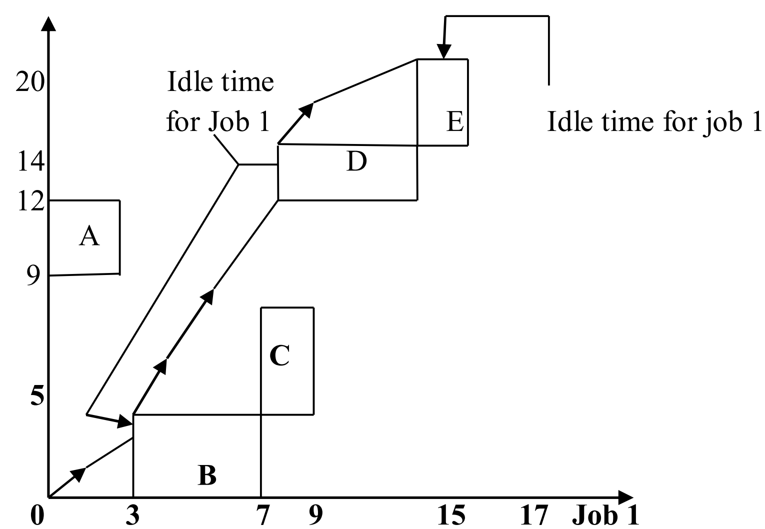
	Machine				
Job 1 {Sequence:	A	B	C	D	E
Time (hrs)	3	4	2	6	2

	Machine				
Job 2 {Sequence:	A	B	C	D	E
Time (hrs)	5	4	3	2	6

Solution

The solution procedure for solving the above problem can be summarised in the following steps:

1. Draw the set of axes at right angle to each other where x-axis represents the processing time of job 1 on different machines while job 2 remains idle and y-axis represents processing time of job 2 while job 2 remain idle.
2. Mark the processing times for jobs 1 and 2 on x-axis and y-axis, respectively according to the given order of machines as shown in the figure:



Graphical solution of 2-Jobs and m-Machines Sequencing Problem

For example, machine A takes 3 hours for job 1 and 3 hours for job 2. Construct the rectangle for machine A as shown in above Figure. Similarly, construct other rectangles for machines B, C, D and E.

3. Construct various blocks starting from the origin by pairing the same machine until a point marked 'finished' is obtained.

4. Draw a line starting from origin to the point marked 'finish' by moving horizontally, vertically and diagonally along a line which makes an angle of 45° with the horizontal axis. Moving horizontally along this line indicates that first job is under process while second job is idle. Similarly, moving vertically along this line indicates that the second job is under process while first job is idle. The diagonal movement along this line shows that both the jobs are under process simultaneously.

Since simultaneous processing of both jobs on a machine is not possible, therefore, diagonal movement is not allowed. In other words, diagonal movement through rectangle areas is not allowed.

5. An optimal path is one that minimises the idle time for both the jobs. Thus, we must choose the path on which diagonal movement is maximum as shown in the above figure.

6. Total elapsed time is obtained by adding the idle time for either job to the processing time for that job. In this example, the idle time for the chosen path is found to be 5 hrs and 2 hrs for jobs 1 and 2, respectively. The total elapsed time is calculated as follows:

Elapsed time, Job 1 = Processing time of Job 1 + Idle time for Job 1

= 17 + (2 + 3) = 22 hrs.

Elapsed time, Job 2 = Processing time of Job 2 + Idle time for Job 2

= 22 + (17-15) = 24 hours.

EXERCISES

1. Explain the four elements that characterise a sequencing problem.
2. Explain the principal assumptions made while dealing with sequencing problems.
3. Give Johnson's procedure for determining an optimal sequence for processing n items on two machines. Give justification of the rule used in the procedure.
4. What is no passing rule in a sequencing algorithm? Explain the principal assumptions made while dealing with sequencing problems.
5. Give three different examples of sequencing problems from your daily life.
6. We have five jobs, each of which must be processed on the two machines A and B in the order AB. Processing times in hours are given in the table below:

Job	1	2	3	4	5
Machine A	5	1	9	3	10
Machine B	2	6	7	8	4

7. We have 5 jobs each of which must go through the machines A, B and C in the order ABC. Processing times (in hours) is as follows:

Job	1	2	3	4	5
Machine A	5	7	6	9	5
Machine B	2	1	4	5	3
Machine C	3	7	5	6	7

8. What do you understand by the following terms in the context of sequence of jobs:

1. Job arrival pattern 2. Number of machines 3. The flow pattern in the shop
4. the criteria of evaluating the performance of a schedule.

9. Find an optimal sequence for the following sequencing problem of four jobs and five machines (when passing is not allowed) of which processing time (in hrs) is as follows:

Job	1	2	3	4
Machine M_1	6	5	4	7
Machine M_2	4	5	3	2
Machine M_3	1	3	4	2
Machine M_4	2	4	5	1
Machine M_5	8	9	7	5

Also find the total elapsed time.

10. Two jobs are to be processed on four machines A, B, C and D. The technological order for these jobs on machines is as follows:

Job 1: A B C D
Job 2: D B A C

Processing times are given in the following table:

	Machines			
	A	B	C	D
Job 1	4	6	7	3
Job 2	4	7	5	8

Find the optimal sequence of jobs on each of the machines.