Sequencing

Introduction :

The selection of an appropriate order in which to service waiting customers (or jobs) is called **sequencing**. Here the effectiveness measure which may be time, cost, mileage etc. is a function of the order or sequence in which a series of tasks are performed (or processed). There are the problem in which we have n tasks to be processed on some or all of m different machines in which the total effectiveness depends on the order of processing. There are $(n!)^m$ possible sequences for the problem. This is a lengthy and long process. To overcome such situations an effective method of the sequencing can be adopted.

A Sequencing Problem :

A general sequencing problem can be defined as:
Let there are n – jobs (1,2,3,.......,n) which have to be processed one at a time at each of m machines A, B, C,........... . The order of the machines for each job, in which it should go to the machines is given. The time required by the jobs on each of the machines is also given. Then the problem is to find the sequence, out of $(n!)^m$ sequences, which optimizes (minimizes) the total time elapsed from the start of the first job to the completion of last job.

Mathematically

If A_i, B_i,..........., etc. are the times for the job i on machines A, B, C.......and T is the total times elapsed from the start of the first job to the completion of the last job to, then the problem of sequencing is to find for each machine a sequence, (i_1, i_2, ...,i_n), where (i_1, i_2, ...,i_n) is a permutation of the integers (1, 2,...,n), which minimizes the T.

There are following four cases of sequencing:

1. n jobs to be processed on two machines A and B, all jobs to be processed in the order AB.

2. n jobs to be processed on three machines A, B and C, all jobs to be processed in the order ABC.

3. n jobs to be processed on m machines.

4. Two jobs to be processed on m machines. Each job to be processed through the machines in a prescribed order which is not necessarily the same for both jobs.

General Assumptions:

We shall solve the sequencing problem under the following assumptions:

1. The processing times of jobs on machines are independent of their processing order.

2. Each job, once started on a machine, must be performed upto the completion on that machine.

3. No machine may process more than one job at a time.

4. The time taken by the jobs in going from one machine to another negligible.

5. There is only one machine of each type.

6. All jobs are known and are ready to start

Sequencing Decision Problem for n – jobs on two machines:

Johnson's Method

Here we consider the problem of processing n – jobs 1, 2, ..., n on two machines A and B. under the following assumptions:

1. Each job is processed in order AB.
2. A_i = Processing time of i – th job on machine A. (i = 1, 2, ..., n).
3. B_i = Processing time of i – th job on machine B. (i = 1, 2, ..., n).

The problem is to find the sequence of jobs to be processed on two machines so that, the total time (T) elapsed from the start of the first job to the completion of the last job is minimized.

The procedure for the solution of the above problem, was developed by **Johnson and Bellman**. The method is based on minimizing the idle time for second machine **The Johnson's procedure for determining an optimal sequence** is as follows:

1. Select the smallest processing time in the lists $A_1, A_2, \ldots\ldots\ldots, A_n$ (processing times of jobs on machine A) and $B_1, B_2, \ldots\ldots\ldots\ldots, B_n$ (processing times of jobs on machine B). If there is a tie then either of these smallest processing times may be chosen.

2. If the smallest processing time is A_r (in the list $A_1, A_2, \ldots\ldots\ldots, A_n$) then do the r^{th} job first. If it is B_s (in the list $B_1, B_2, \ldots\ldots\ldots\ldots, B_n$) then do the s^{th} job last.

3. Delete the times of the job already assigned from the two lists of A_i's and B_i's i.e. if r^{th} job is assigned previously, then delete A_r and B_r both and if s^{th} job is assigned previously then delete A_s and B_s both.

4. After assigning one job we are left with $(n - 1)$ jobs. Repeat step 1 to 3 on the processing times of the remaining $(n - 1)$ jobs.

5. Continue in this way until all the jobs have been ordered. In this way we get an optimal sequence of jobs. Finally the minimum time T elapsed can be found by forming a chart.

CL: Six jobs to be processed first over machine 1 and then over machine 2. The order of the completion of jobs has no significance. The following table gives the machine times in hours for 6 jobs on the two machines.

Job No. i	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
Time on machine 1(A_i) Time on machine 2(B_i)	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

Find the sequence of the jobs that minimizes the total elapsed time to complete the jobs. Find the minimum time by using Gantt's chart or by other method.

Solution:

3	1	5	6	2	4

To find the minimum elapsed time

Job	Machine 1		Machine 2		Idle time of machine 2
	Time in	Time out	Time in	Time out	
3	0	4	4	12	4
1	4	9	12	19	
5	9	17	19	28	
6	17	23	28	33	
2	23	32	33	37	
4	32	39	39	42*	2

The total time elapsed is 42 hours and the idle time for machine 1 is 42 – 39 = 3 hrs and for machine 2 is 6 hrs.

CL: Five jobs to be processed first over machine 1 and then over machine 2. The order of the completion of jobs has no significance. The following table gives the machine times in hours for 5 jobs on the two machines.

Job No. i	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
Time on machine 1(A_i) Time on machine 2(B_i)	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

Find the sequence of the jobs that minimizes the total elapsed time to complete the jobs. Find the minimum time by using Gantt's chart or by other method.

Solution:

2	4	3	5	1

To find the minimum elapsed time

Job	Machine 1		Machine 2		Idle time of machine 2
	Time in	Time out	Time in	Time out	
2	0	1	1	7	1
4	1	4	7	15	
3	4	13	15	22	
5	13	23	23	27	1
1	23	28	28	30	1

The total time elapsed is 30 hours and the idle time for machine 1 is 30 – 28 = 2 hrs and for machine 2 is 3 hrs.

Sequencing Decision Problem for n – jobs on three machines:

Modified Johnson's Method

Here we consider the problem of processing n – jobs 1, 2, ..., n on three machines A, B and C. under the following assumptions:

1. Each job is processed in order ABC.
2. A_i = Processing time of i – th job on machine A. (i = 1, 2, ..., n).
3. B_i = Processing time of i – th job on machine B. (i = 1, 2, ..., n).
4. C_i = Processing time of i – th job on machine C. (i = 1, 2, ..., n).

The problem is to find the sequence of jobs to be processed on three machines so that, the total time (T) elapsed from the start of the first job to the completion of the last job is minimized.

The previous method is can be extended to the cases in which either or both of the following condition hold.

1. The smallest processing time for machine A $\geq$ the largest processing time for machine B.

2. The smallest processing time for machine C $\geq$ the largest processing time for machine B.

This problem is replaced as the previous problem if we are able to convert three machines in to two fictitious machines G and H. Then the processing time for two machines G_i and H_i is given by

$G_i = A_i + B_i$

$H_i = B_i + C_i$

Now find the optimal sequence of jobs in the order GH on these two machines G and H by the previous method. These resulting optimal sequences on two machines G and H will also be optimal sequence for three machines A, B and C.

CL: Five jobs to be processed over machines A, B and C. The order of the completion of jobs is ABC. The following table gives the machine times in hours for 5 jobs on the three machines.

Job No. i	1	2	3	4	5
Time on machine A (A_i)	5	7	6	9	5
Time on machine B (B_i)	2	1	4	5	3
Time on machine C (C_i)	3	7	5	6	7

Find the sequence of the jobs that minimizes the total elapsed time to complete the jobs. Find the minimum time by using Gantt's chart or by other method.

Solution:

Here Min A_i = 5 and Max B_i = 5 Min C_i = 3

Since Min $A_i \geq$ Max B_i

Therefore we can consider two fictitious machines G and H and their processing times as follows:

Jobs	Processing Times	
	$G_i = A_i + B_i$	$H_i = B_i + C_i$
1	7	5
2	8	8
3	10	9
4	14	11
5	8	10

Following are three possible sequences can be obtained:

2	5	4	3	1

5	4	3	2	1

5	2	4	3	1

To find the minimum elapsed time for the first sequence:

Job	Machine A		Machine B		Machine C		Idle time of Machine B	Idle time of Machine C
	Time in	Time out	Time in	Time out	Time in	Time out		
2	0	7	7	8	8	15	7	8
5	7	12	12	15	15	22	4	
4	12	21	21	26	26	32	6	4
3	21	27	27	31	32	37	1	
1	27	32	32	34	37	40*	1+6	

Thus the minimum time elapsed is 40 hrs. The time may be verified for the rest of two sequences. Idle time for machines A, B and C are 8, 25 and 12 hrs:

CL: Five jobs to be processed over machines A, B and C. The order of the completion of jobs is ABC. The following table gives the machine times in hours for 5 jobs on the three machines.

Job No. i	1	2	3	4	5
Time on machine A (A_i)	4	9	8	6	5
Time on machine B (B_i)	5	6	2	3	4
Time on machine C (C_i)	8	10	6	7	11

Find the sequence of the jobs that minimizes the total elapsed time to complete the jobs. Find the minimum time by using Gantt's chart or by other method.

Sequencing Decision Problem for n – jobs on m - machines:

Now we consider the problem of processing n jobs 1, 2, ………, n on m machines M_1, M_2, ………, M_m under the following assumptions.

1. Each job is performed in order M_1, M_2, ………, M_m .
2. M_{ij} = Processing of i – th job on j – th machine (i = 1, 2, ……….., n and j = 1, 2, …………, m)

The problem is to find sequence of jobs to be performed on the machines in the order M_1, M_2, ………, M_m so that the total time elapsed from the start of the first job to the completion of the last job is minimized. There is no general method available for the solution of above problem. The previously discussed method can be extended as follows.

(i) Min $M_{i1} \geq$ Max M_{ij} for j = 2, 3, ……. , (m – 1)

(ii) Min $M_{im} \geq$ Max M_{ij} for j = 2, 3, ……. , (m – 1)

This method is to replace the problem with an equivalent problem involving n jobs on two machines. If G and H denote the two machines (fictitious) then the processing times M_{iG} and M_{iH} of i^{th} ($i = 1, 2, \ldots, n$) job on these two machines are given by

$$M_{iG} = M_{i1} + M_{i2} + \ldots + M_{i(m-1)}$$
$$M_{iH} = M_{i2} + M_{i3} + \ldots + M_{im}$$

Now find the optimal sequence of n – jobs in order GH on the two fictitious machines G and H by previously discussed method. The resulting optimal sequence on two machines G and H will also be the optimal sequence on m machines M_1, M_2, ..., M_m .

CL: Solve the following sequencing problem when passing time is not allowed. Processing time in hrs is given below

	Machines				
Items	A	B	C	D	E
I	9	7	4	5	11
II	8	8	6	7	12
III	7	6	7	8	10
IV	10	5	5	4	8

Since, Min $M_{i1} = 7$, Min $M_{i5} = 8$

Max $M_{ij} = 8$, for $j = 2, 3, 4$

The condition Min $M_{i5} \geq$ Max M_{ij} for $j = 2, 3, 4$ is satisfied. Therefore we consider two fictitious machines G and H. The processing times on these two fictitious machines G and H can be given by:

Jobs	$M_{iG} = \sum_{j=1}^{4} M_{ij}$	$M_{iH} = \sum_{j=2}^{5} M_{ij}$
I	25	27
II	29	33
III	28	31
IV	24	22

Proceeding in usual manner we get the following optimal sequence:

I	III	II	IV

Processing two jobs through m-Machines:

Now we shall consider the following problem:

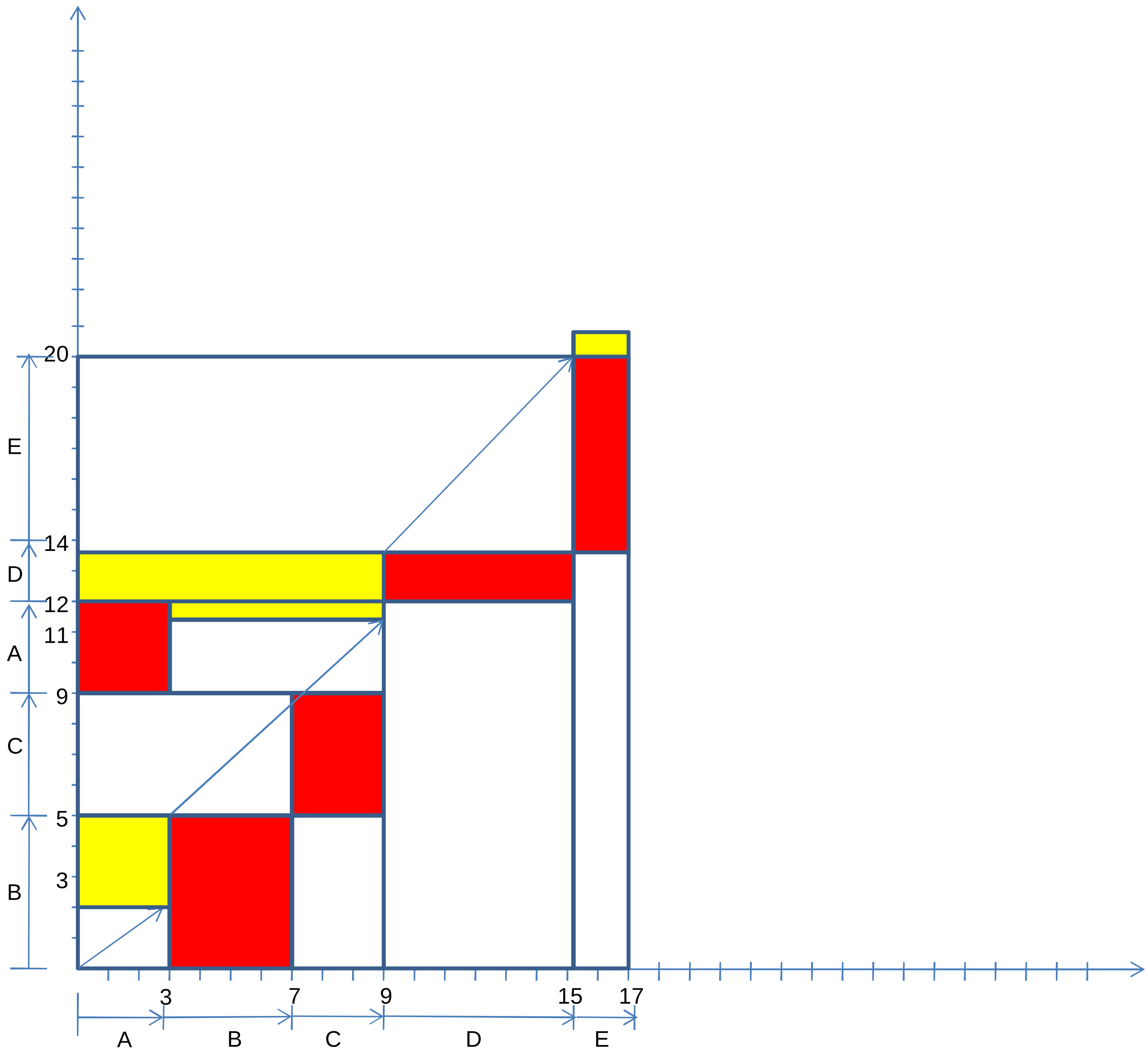
1. There are 2 jobs 1 and 2 to be performed.
2. There are m machines denoted by A, B, C, ..., M.
3. The technological ordering of each of the two jobs 1 and 2 through m machines is given in advance. This ordering need not to be same for jobs.
4. The exact or expected processing times A1, B1, C1, ..., M1 for job 1 and A2, B2, C2, ..., M2 for job 2 on the m machines are given.

The problem is to minimize the total elapsed time T from the start of the first job on the first machine to the completion of the last job on the last machine.

It is obvious that for each machine there are two sequences of jobs i.e., (1, 2) or (2, 1). Thus, we have to select one sequence out of these two for each machine. Graphical method is a simple method for solution of this method.

CL: Use graphical method to minimize the time needed to process the following jobs on the machine shown, i.e., for each machine find the job which should be done first. Also calculate the total time needed to complete both the jobs.

Job 1		Job 2	
Sequence of Machines	Time	Sequence of machines	Time
A	3	B	5
B	4	C	4
C	2	A	3
D	6	D	2
E	2	E	6



20
E
14
D
12
11
A
9
C
5
3
B
3
7
9
15
17
A
B
C
D
E

The Travelling Salesman (0r Routing) Problem:

This problem is about the travelling salesman who want to visit a certain number of cities. His problem is to select a route that will minimize the total distance travelled (or cost or the time). In visiting each city once and only once and returning to his home city (city from where he has started from beginning).

Formation as Assignment Problem:

This problem can be formulated as an assignment problem. Let c_{ij} is the distance (or cost of time) of going from city i to city j. Then these cost can be arranged in the form of a square matrix.

Here we have two extra restrictions;

(i) We can not go from city i to city i, in tis case we say ($c_{ii} = \infty$).
(ii) No city is visited twice since the tour is completed.

Cost effectiveness matrix for :

To

		A_1	A_2	....	A_i	....	A_n
	A_1	∞	c_{12}	....	c_{1i}	....	c_{1n}
	A_2	c_{21}	∞	....	c_{2i}	....	c_{2n}
From		....	....	∞	....	....	....
		....	....	....	....	....	....
	A_i	c_{i1}	c_{i2}	....	∞	....	c_{in}
		....	....	....	....	....	....
		....	....	....	....	....	....
	A_n	c_{n1}	C_{n2}	....	c_{ni}	....	∞

Solution of a Travelling Salesman Problem:

The problem of travelling salesman can be solved by simple assignment problem. Suppose the solution after solving the assignment problem is like

$$A_1 \to A_5, A_5 \to A_1, A_2 \to A_3, A_3 \to A_4, A_4 \to A_2$$

Here the salesman proceed from city A1 to A5, then come back to A1 without visiting cities A2, A3, A4. If he starts from city A2 then he proceed from A2 to A3, A3 to A4 and then from A4 to A2 without visiting the cities A1 and A5. Thus this is not a feasible solution for the problem, In such cases after solving the problem by assignment problem technique, we use the method of enumeration by assigning the next minimum element of the matrix in place of 0.

Example:

A machine operator processes four types of items on his machine each week, and must choose a sequence from them. The set up cost per change depends on the item presently on the machine and item to be made , according to the following table.

		To			
		A	B	C	D
From	A	∞	4	7	3
	B	4	∞	6	3
	C	7	6	∞	7
	D	3	3	7	∞

If he produces each type of item once and only once each week how should he sequence the items on his machine in order minimize the total set up cost.

	A	B	C	D
A	∞	1	$\cancel{0}$	$\boxed{0}$
B	1	∞	$\boxed{0}$	$\cancel{0}$
C	1	$\boxed{0}$	∞	1
D	$\boxed{0}$	$\cancel{0}$	1	∞

Here we get,

$A \rightarrow D$, $D \rightarrow A$, $B \rightarrow C$, $C \rightarrow B$

But D can not be followed by A until B and C are produced. Now to reach the feasible solution we assign the next minimum element 1 instead of 0.

			X	
	A	B	C	D
A	∞	$\boxed{1}$	0	$\cancel{0}$
B	1	∞	0	$\boxed{0}$
C	1	$\boxed{0}$	∞	1
D	$\boxed{0}$	$\cancel{0}$	1	∞

$$A \rightarrow B, \ B \rightarrow C, \ C \rightarrow D, \ D \rightarrow A$$