UNIT 11 PLANNING AND CONTROL FOR JOB SHOP PRODUCTION

Objectives

After completion of this unit, you should be able to:

- understand the nature of job production
- appreciate the variety of problems that may arise in job shops
- sequence a number of jobs in a static job shop with a single machine to accomplish a number of objectives
- sequence a number of jobs in a static two-machine flow shop to minimise the total completion time
- develop a schedule to minimise time of completion (graphically) of two jobs requiring processing on a number *of* machines in any specified order.
- become aware of the use of different priority despatching rules in practical job shops.

Structure

11.1 INTRODUCTION

Unlike mass production systems where a limited number of items are produced continuously without any changes, the job shop characterises a situation where a variety of jobs is handled, each job being different from the previous one. Job shops differ from batch production in the following manner. In batch production there is generally a continuous demand for products but owing to a higher rate of production as compared to the rate of demand, production is undertaken, in batches. Thus batch production caters to a predictable variety of jobs. In the job *shop,* however, both the nature and demand of jobs is unpredictable. Moreover, each order or job is unique, requiring its own makeup of operations and times for processing through a number of machines or facilities in a factory.

A job shop typically consists of general purpose machines clubbed into different departments. Each job, governed by its unique technological requirements, demands processing on machines in a certain order. Because of the variety of tasks, the job shop becomes *a* complex queueing system: a job leaves one machine and proceeds *on* its route to another for the next operation, only to find other jobs already waiting for the machine to complete its current task, so that a queue of jobs in front of that machine is formed; alternatively, a machine may finish its task and be ready to take the next job, but no jobs are available, so that the machine becomes idle. Planning for the job shop essentially involves deciding the order or priority for jobs waiting to be processed at each machine to achieve the desired objectives.

11.2 VARIETY OF PROBLEMS IN JOB PRODUCTION

A typical formulation of the job shop scheduling problem is: given n jobs to be processed through m machines, each job having a predetermined sequence of operations and processing times, in what order should the jobs be loaded on the machines so as to optimise certain performance criteria?

A typical list of performance criteria to be optimised is:

1 Total processing time or makespan
2 Mean flow time (or mean time in the job shop)
3 Idle time of machines
4 Mean lateness of jobs (lateness of a job is defined as the difference between the actual completion time of the job and its due date)
5 Mean earliness of job (if a job is completed before its due date, then its lateness value is negative and it is referred to as earliness instead)
6 Mean tardiness of jobs (if a job is completed after its due date, then its lateness value is positive, and it is referred to as tardiness instead)
7 Number of tardy jobs
8 Mean queue time
9 Mean number of jobs in the system.

Moreover the solution procedure depends on the following factors:

1 The number of jobs to be scheduled
2 The number of machines in the machine.shop
3 Type of manufacturing facility (flow shop or job shop)
4 Manner in which jobs arrive at the facility (static or dynamic)
5 Criterion by which scheduling alternatives are to be evaluated

If the number of jobs (n) and the number of machines (m) increase, the scheduling problem becomes more complex. In fact, no exact or optimal solutions exist for sequencing problems with large n and m. Simulation and heuristic algorithms seem to be the solution techniques for real life scheduling problems.

We initiate the discussion on job shops with some very simple one machine and two machine cases for which exact solutions exist. Many practical situations would fit into these categories as you would see. Next, a graphical procedure for scheduling of 2 jobs to minimise the total time of processing is proposed. And, finally, some scheduling rules and results from simulation studies are presented.

11.3 n JOBS ONE MACHINE CASE

The case when a number of jobs is to be processed on a single facility occurs quite often in practice such as a number of cars to be serviced at a service station, number of patients to be treated by a doctor, number of job requests from different terminal users to be executed by a computer, a number of different jobs to be machined on a lathe, or a number of broken down machines to be repaired by a service mechanic.

In all such cases, we, as human beings, are used to the first come first served concept. This seems to be the almost universally accepted code for handling queues at milk booths, post-offices, banks and almost anywhere we go. Perhaps the validity of this rule stems from the democratic notion that all individuals are equally important and it is `just' or `proper' to assign priorities on seniority. In fact we all know how unpleasant scenes and heart-burnings can result in real life if this chosen rule is violated, for instance, when a newcomer tries to enter a queue at •a vantage • position or a senior is superseded in a job promotion.

We shall see that if the objective is not to give a sense of satisfaction and justice to waiting jobs (when, for instance, the jobs lack the human attributes of ego and pride) it is possible to consider a number of other rules which would yield more favourable results in terms of objectives like 'the mean flow time, average in-process inventory of jobs, the mean lateness, mean waiting time of jobs, the number of tardy jobs and the total processing time of jobs. It is in this framework that the following scheduling rules should be considered:

We shall consider a static job shop in which a number of jobs, n, with known processing times require processing on a single machine. The job shop is static in that new job arrivals do not disturb the processing of these n jobs. It may be assumed that new job arrivals wait for being considered in the next batch of jobs after the processing of the current n jobs is accomplished.

The following definitions are in order

n = number of jobs

t_i = processing time of job i

W_i= waiting time before processing for job i

F_i= flow time of job i = W, + t

C_i= completion time of job

d_i= due date of job i

L_i= lateness of job i= C_1-d_1

$E_{i=}$earliness of job i= max = max (0, - L_i)

T_i= tardiness of job= max (0, L_i)

N_T = number of tardy jobs

Since the total time to complete the jobs (makespan, as it is often called) is fixed and equal to sum of processing times for all jobs in all possible sequences in this n machine one job case, other optimality criteria, e.g. minimising the mean flow time or some function of lateness of jobs is resorted to.

Shortest Processing Time (SPT) Rule

Sequencing the jobs in a way that the job with least processing time is picked up first followed by the one with the next smallest processing time and so on is referred to as SPT sequencing and achieves the following objectives simultaneously:

i) minimising mean lateness
ii) minimising mean flow time
iii) minimising mean waiting time
iv) minimising the mean number of tasks waiting as in-process inventory.

A variation of the SPT rule is the weighted-scheduling rule (WSPT) which is used when the importance 'of the tasks vary. A, weight W_i is assigned to each job, a larger value indicating greater importance. Then, by dividing the processing time by the weighting factor the tendency is to move the more important task to an earlier position in the sequence. The weighted mean flow time is given by:

$$\text{WMFT}=\frac{\sum_{i=1}^{n} W_i F_i}{\sum_{i=1}^{n} W_i}$$

The WSPT rule for minimising weighted mean flow time sequences the n jobs such that

$$\frac{t_{[1]}}{w_{[1]}} \leq \frac{t_{[1]}}{w_{[1]}} \leq \text{-------------------------} \leq \frac{t_{[1]}}{w_{[1]}}$$

where the number in square brackets defines the position of the job in the optimal sequence.

The. Proofs of the preceding statements concerning SPT and WSPT are available in literature (see for instance, Baker). We shall only illustrate the application of these ' rules to an example problem.

Example 1

Consider the 8 jobs with processing times, due dates and importance weights as shown in Table 1.

The SPT sequence is 4-8-1-3-7-2-5-6 resulting in completion of these jobs at times 3,6,11,17,24,32,42 and 56 respectively. The mean flow time is thus

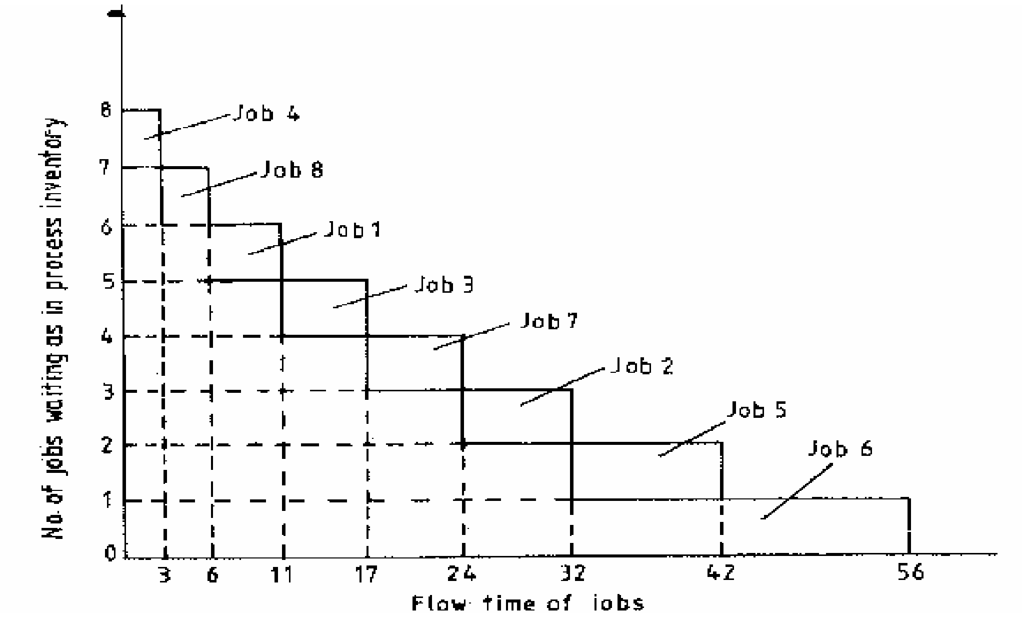
$$\frac{3+6+11+17+24+32+42+56}{8}=\frac{191}{8}=23.875 \quad \text{hours}$$

Table 1

Data for Eight Job One Machine Example

Task	Processing	Due date	Importance ti/w$_i$	
i	Time ti	di	Weight w$_i$	
1		15	1	5.0
2	8	10	2	4.0
3	6	15	3	2.0
4	3	25	1	3.0
5	10	20	2	5.0
6	14	40	3	4.7
7	7	45	2	3.5
8	3	50	1	3.0

Figure I: SPT Sequence for Example 1



This sequence is shown graphically in Figure I from which the number of tasks waiting as in-process inventory is seen to be 8 during time 0-3, 7 during 3-6, 6 during 6-11, 5 during 1J-17, 4 during 17-24, 3 during 24-32, 2 during 32-42 and 1 during 42-56. Thus the average in-process inventory

$$=\frac{(8\times3)+(7\times3)+(6\times5)+(5\times6)+(4\times7)+(3\times8)+(2\times10)+(1\times14)}{56}$$

$$=\frac{191}{56}=3.41\,jobs$$

In fact Figure I clearly shows how mean flow times and average inventory are related. The former is obtained by summing horizontal strips in the figure to obtain the total area under the curve which is then divided by the number of jobs. The mean inventory is obtained by summing vertical strips for the total area under the curve, which is then divided by the total processing time of all jobs.

Activity A

In the above example I compute the waiting times for each job and the mean waiting time under the SPT rule.

Activity B

In the above example 1 compute the lateness of each job and the mean lateness under the SPT rule.

Activity C

Try any other (non SPT) sequence for the jobs in the above example 1 and compute

i) mean flow time

ii) average in-process inventory of jobs

iii) mean waiting time

iv) mean lateness.

Compare these values with those of the SPT sequence considered above and convince yourself that the SPT sequence does in fact minimise all the above criteria.

Example I (Continued)

If the importance weights, W_i were to be considered, the WSPT could be used to minimise the weighted mean flow time to yield the sequence 3-4-8-7-2-6-1-5. This results by first choosing the job with minimum ti/wi ratio in Table I and so on.

The respective flow times of jobs in this sequence are 6,9,12,19,27,41,46 and 56. The mean flow time equals 27.0 hours and the weighted mean flow time is given by

$$=\frac{(3\times6)+(1\times9)+(1\times12)+(2\times19)+(2\times27)+(3\times41)+(1\times46)+(2\times56)}{3+1+1+2+2+3+1+2}$$

$$=\frac{412}{15}=24.47\,hours.$$

Processing with Due Dates

The SPT rule considered above minimises the mean lateness of jobs. The Earliest Due Date (EDD) rule, where jobs are sequenced in the order of nondecreasing due dates of jobs minimises the maximum job lateness as well as the maximum job tardiness. Unfortunately, the EDD rule tends to make more tasks tardy and increases the mean tardiness.

Considering the earlier data of Table 1, the FDD sequence is 2-1-3-5-4-6-7-8 which yields completion times of these jobs as 8,13,19,29,32,46,53,56 with respective lateness values as -2, -2, 4,9,7,6,8,6. The mean lateness is 4.5 hours while the maximum lateness is 9 hours.

From your solution to Activity B you would have noticed that the mean and maximum lateness figures for the SPT sequence were -3.625 hours and 22 hours respectively. Also the number of tasks actually late was 4. We thus see that in this example the EDD sequence has in fact reduced the maximum lateness to 9 hours (compared to the corresponding figure of 22 hours for the SPT rule) but the mean lateness has gone up to 4.5 from -3.625 and the number of late jobs has gone up to 6 from 4.

Minimising the Number of Tardy Jobs

Suppose there was a penalty cost associated with a job if it was tardy, which was independent of how tardy the job was. In this situation, the objective would be to minimise the number of tardy jobs. The EDD rule gives the desired schedule only if it results in zero or one tardy task. If more than one tardy task results, the following algorithm due to Hodgson will yield the desired objective.

Step 1: Order all tasks by the EDD rule; if zero or one tasks are tardy (positive lateness), then stop. Otherwise go to

Step 2: Starting at the beginning of the, EDD sequence and working towards the end, identify the first tardy task. If no further tasks are tardy, go to step 4; otherwise go to step 3.

Step 3: Suppose that the tardy task is in the ith position in the sequence. Examine the first i tasks in the sequence and identify the one with the longest processing time. Remove the task and set it aside. Revise the completion time of the other tasks to reflect this removal, and return to step 2.

Step 4 place all those tasks were set aside in any order at the end of the sequence.

Let us now consider the example problem of Table 1 in the light of Hodgson's algorithm. The EDD rule used in step 1 results in the sequence 2-1-3-5-4-6-7-8 with six tardy jobs. Thus we move to step 2 and 3 as shown below:

Task is	2	1	3	5	4	6	7	8
Processing time ti	8	5	6	10	3	14	7	3
Completion time Ci	8	13	19	29	32	46	53	56
Due date d;	10	15	15	20	25	40	45	50
Lateness Li	-2	-2	4	9	7	6	8	6

Task 3 is the first tardy task. Task 2 has the longest processing time of the first three tasks, thus it is set aside yielding the following revised table:

Task i	1	3	5	4	6	7	8
t_i	5	6	10	3	14	7	3
C_i	5	11	21	24	38	45	48
d_i	15	15	20	25	40	45	50
$L_{.i}$	-10	-4	1	-1	-2	0	-2

Task 5 is the first tardy task and of tasks 1,3 and 5 task 5 is the longest. Thus it is set aside, yielding the following:

Task i		1	3	4	6	7	8
t_i	t_i	5	6	3	14	7	3
C_i	C_i	5	11	14	28	35	38
d_i	d_i	15	15	25	40	45	50
L_i	L_i	-10	-4	-11	-12	-10	-12

No more tasks are tardy. Thus the first part of the sequence is 1-3-4-6-7-8 and the last part consists of tasks 2 and 5 in any order. Adding these two in SPT order we obtain the sequence 1-3-4-6-7-8-2-5 for which the completion times of respective jobs are 5,11,14,28,35,38,46 and 56 while the lateness values are -10, -4, -11, -12, -10. -12, 36.36, respectively. Notice that only two tasks are now late, though the mean lateness is 1.625 hours (compared to minimum of -3.625 for SPT rule) and the maximum lateness has gone up to 36 (as compared to the minimum of 9 for the EDD rule).

Minimising the Mean Tardiness

If the penalty for tardiness is the same for all tasks and linear with respect to how tardy the task is, then obviously the objective would be to minimise the mean tardiness. Unfortunately, there is no simple scheduling rule, even for n tasks and one machine case which minimises mean tardiness. Baker has shown that in the following simple situations the EDD and SPT rules also minimise the mean tardiness.

1 If the EDD rule produces zero or one tardy job , then it minimises mean tardiness.

2 If all tasks have the same due date, or if the SPT rule results in all tasks being tardy, then the SPT rule minimises mean tardiness.

A heuristic rule that tends to minimise mean tardiness is the shortest SLACK time rule. Notice that this rule does not necessarily give optimum results, but is yields good solutions in practical cases. SLACK time for job i is defined as its due, date minus its processing time.

For the data in Table 1 the slack times for jobs could be computed as shown below:

Task i	Processing time ti	Due date di	Slack time di - ti
1	5	15	10
2	8	10	2
3	6	15	9
4	3	25	22
5	10	20	10
6	14	40	26
7	7	45	38
8	3	50	47

The resulting sequence is, therefore, 2-3-1-5-4-6-7-8 which yields completion times of 8,14,19,29,32,46,53 and 56 and tardiness values of 0, 0, 4.9,7,6,8 and 6, respectively. The SLACK rules thus yields a mean tardiness of 5.0 hours.

Another algorithm for minimising the mean tardiness is due to Wilkerson and Irwin. The interested reader may refer to the literature for more details.

It is instructive at this stage to compare the performance of the various scheduling rules discussed for processing n tasks on one processor. For the problem data of Table I although all rules yield a total completion time or makespan of 56 hours, their performance with respect to other objectives is summarised in Table 2.

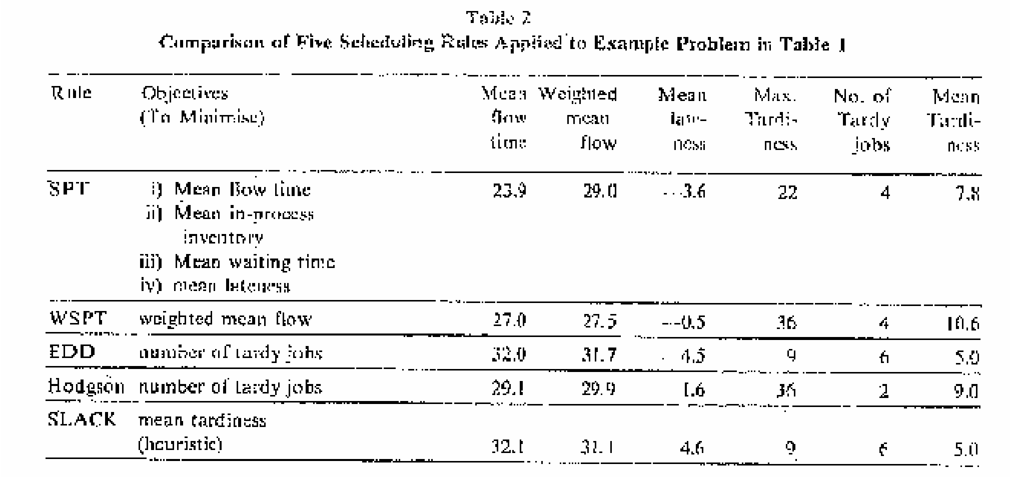


Table 2
Comparison of Five Scheduling Rules Applied to Example Problem in Table 1

Rule	Objectives (To Minimise)	Mean flow time	Weighted mean flow	Mean late-ness	Max. Tardi-ness	No. of Tardy jobs	Mean Tardi-ness
SPT	i) Mean flow time ii) Mean in-process inventory iii) Mean waiting time iv) mean lateness	23.9	29.0	--3.6	22	4	7.8
WSPT	weighted mean flow	27.0	27.5	--0.5	36	4	10.6
EDD	number of tardy jobs	32.0	31.7	4.5	9	6	5.0
Hodgson	number of tardy jobs	29.1	29.9	1.6	36	2	9.0
SLACK	mean tardiness (heuristic)	32.1	31.1	4.6	9	6	5.0

11.4 n JOBS TWO MACHINES CASE

Consider the situation where n jobs must be processed by two machines, 'vi followed by M2. The processing times of all jobs on MI and M2 are known and deterministic. It is required to find the sequence which minimises the time to complete all jobs.

Johnson developed an algorithm that can be used to obtain an optimal sequence in such a case. Johnson's algorithm is summarised below:

1 List all processing times of all jobs on machine MI and M2.

2 Scan through all processing times for all jobs. Locate the minimum processing time.

3 If the minimum processing time is on M I, place the corresponding job first (as early as possible) in the sequence. If it is on M2, place the corresponding job last (as late as possible) in the sequence.

4 Eliminate the assigned jobs (already placed in the sequence as a result of step 3) and repeat steps 2 and 3 until all jobs are sequenced.

Any ties in processing times while applying step 3 are broken arbitrarily because they cannot affect the minimum elapsed time to complete all jobs.

Example 2

A small workshop undertaking repair of damaged cars has one denter and one painter. Presently there are six cars needing repair. The following estimates in hours of time needed for denting and painting on the cars are available

Car	1	2	3	4	5	6.
Denting time (hrs.)	4	7	3	12	11	9
Painting time (hrs.)	11	7	10	8	10	13

What is the sequence that completes all the jobs in minimum time? What is the corresponding schedule of Jobs?

Solution

This situation conforms to the 2 machine flow shop configuration where the denter constitutes M 1 and the painter M2, and all cars must first be processed on M I and then on M2. Thus we can use Johnson's rule to determine the optimal sequence,
The minimum processing time of 3 occurs on M I for job 3. Thus job 3 is placed first resulting in the partial sequence

3					

Deleting job 3, the next lowest processing time of 4 occurs for job I on M I. Thus job 1 is placed in the first available placed yielding the partial sequence

3	1				

After deleting both jobs 3 and 1, the minimum processing time of 7 occurs for job 2 on both M I and M2. There is thus a tie; we may, therefore, elect to place job 2 either in the first or last available place. This would yield the alternative partial sequences

3	1	2			

and

3	1				2

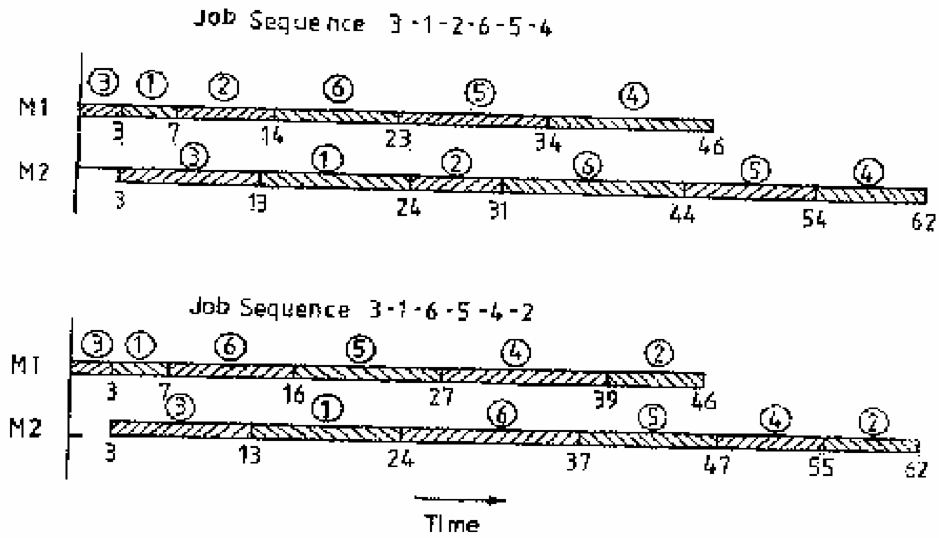
Continuing in this fashion the two alternative optimal sequences are:

3	1	2	6	5	4

and

3	1	6	5	4	2

Figure II: Johnson's Rule for Example 2



Bar charts showing the schedule for these alternative sequences are drawn in Figure 11. Notice that the completion time is 62 hours for all the jobs. Also note that the idle time for the denter is towards the end whereas that for the painter is towards the beginning of the schedule. In this particular case the painter (M2) is continuously busy from time 3 to 62. In certain instances there may be idle times in between if a job has not yet finished processing on MI.

Under certain restrictive assumptions Johnson's rule may be extended to the n job 3 machine flow shop situation. The interested rea^{P}er may refer to any of the references, Baker or Elsayed and Boucher for details.

11.5 TWO JOBS, M MACHINES CASE

When there are two jobs each having its own processing sequence on m machines, the optimal sequence of these jobs on the m machines to minimise the total processing time can be obtained by using a graphical procedure as outlined below:

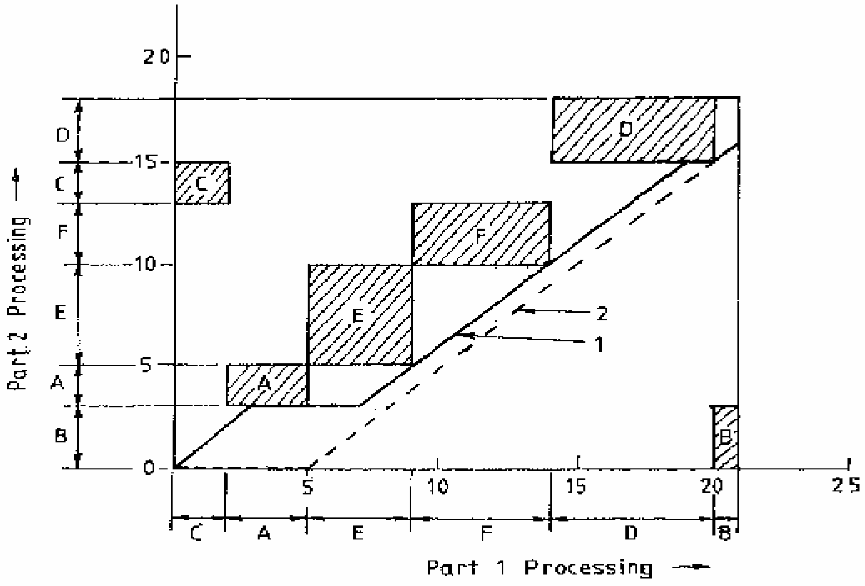
1 Construct a two-dimensional graph where the x axis represents the processing time and sequence of job 1 on the m machines while the y axis represents those of job 2 (Use the same scale for both x and y).

2 Shade the areas where a machine would be occupied by the two jobs at the same time.

3 The processing of both jobs can be represented by a continuous path which consists of horizontal, vertical, and 45 degree diagonal segments.

The path starts at the lower left corner and stops at the upper right corner, while avoiding the shaded areas in the graph. In other words, the path is not allowed to pass through shaded areas which correspond to operating both jobs concurrently on the same machine. Since a diagonal segment implies that both jobs are being processed on different machines at the same time, a feasible path that maximises diagonal movement will minimise the total processing time. The schedule is determined by trial and error. Usually, only a few lines may be drawn before an optimal path is found.

Figure 11: Processing Two Jobs on Six Machines for Example 3



Example 3

Two major parts P_I and P_2 for a product require processing through six machine centres. The technological sequence of these parts on the six machines and the manufacturing times on each machine are:

P1	Machine sequence:	C	-A	-E	-F	-D	-B
	Time(hours):	2	3	4	5	6	1
P2	Machine sequence:	B	-A	-E	-F	-C	-D
	Time(hours):	3	2	5	3	2	3

What would be the optimal scheduling to minimise the total processing time for these two parts?

Solution

Constructing a two dimensional graph where the horizontal axis represents P_I and the vertical axis represents P, we can shade the areas where both parts use the.same machine as shown in Figure III. Lines 1 and 2 maximise the diagonal travel from the bottom left corner to the upper right corner. The total processing time is 24 hours.

11.6 SCHEDULING RULES FOR JOB SHOPS

We have seen that the procedures suggested above apply to very simplified situations involving one or two machines or only two jobs. Moreover, the problem treated was one of a static job or flow shop in which all jobs are available at the beginning of the horizon and the schedule once made is not disturbed till all the jobs are completed. Unfortunately, even the general static job shop problem of n jobs and m machines (m >3) with the objective of minimising the total completion time does not have a general exact solution procedure. There are some heuristic techniques that may obtain good sequences or even optimal sequences (though it may not be possible to check for optimality). You may, for instance, refer to Campbell, Dudek and Smith and Stinson and Smith for example of such heuristic procedures.

Practical job shops are much more complicated and may have hundreds of machining centres and thousands of jobs. In such cases resort is made to priority dispatching rules for sequencing jobs at each machine centre. These can apply both in a static and dynamic job environment. In the former case priorities of jobs once assigned are not changed whereas in the latter case priorities are updated as jobs enter or leave the machining centre. Some commonly specified priority dispatching rules are:

1 FCFS: Select the job on a First Come First Served basis.

2 SPT: Select the job with the Shortest Processing Time.

3 EDD: Select the job with the Earliest Due Date.

4 SLACK: Select the job with the minimum static slack (due date minus arrival at machine centre)

5 RANDOM: Select the job at random.

6 LRPT: Select the job with the Least Remaining Processing Time.

7 S/OPR: Select the job with the minimum ratio of job slack time to the of operations remaining.

8 LCFS: Select the job on a Last Come First Served basis.

9 DS: Select the job with least Dynamic Slack (time remaining to due date. less remaining expected flow time).

10 DS/ PT: Select the job with the minimum ratio of Dynamic Slack to Processing Time.

Based on simulation studies of various job shops, several researchers have come to certain conclusions about the behaviour of these rules. Clearly such behaviour is affected by factors like criteria of performance, level of uncertainty and job arrival pattern.

The SPT rule consistently tends to give the lowest mean flow time even for large shops-a property that has been demonstrated in the simple n job one machine case. However, with the SPT rule the variance of flow times tends to be large. In this, regard other rules such as FCFS perform better. A recent survey of dispatching `rules for manufacturing job shop operations has been done by Blackstone, Phillips and Hogg. The interested reader may refer to this for further details.

11.7 PROBLEMS AND PROSPECTS OF JOB PRODUCTION

Most of the problems in job production arise from the variety of jobs arriving and demanding individual processing sequences on the scarce facilities in the job shop. Owing to these factors the job shop is characterised by the following typical problems:

- complicated and unsystematic material flow patterns
- large in-process inventories
- large waiting times for jobs
- large completion times for jobs
- unpredictable problems owing to the large variety of tasks

We have seen that essentially two kinds of research pertaining to job shop scheduling has been done. In the first case there has been an attempt at developing new procedures for scheduling with a variety of objectives. We have seen some examples of these procedures in Sections 11.3, 11.4 and 11.5. Most of these procedures have been developed for the static job or flow shop. Clearly a lot more could be done in this direction. The major difficulty in large job shops is the combinatorial increase in the number of possible sequences. With better bounds and more efficient computational facilities this problem could be probed further.

The second kind of scheduling research has been to test the efficiency of different priority dispatching rules using simulation. Some of these scheduling rules were described in Section 11.6. The advantage of this approach is that dynamic features can be incorporated and evaluated. There are possibly two extremes for scheduling the, new jobs in a dynamic environment: to produce a new schedule each time a new job arrives, or to completely finish the existing schedule before producing a new schedule for the jobs that had arrived in the mean time. Or a solution between the two extremes could be considered. Ignoring computational considerations, one could have an on-line computer system whereby everytime an operation finishes on a machine, this information is fed into the computer. The programme then considers the actual status of the shop at that time and all the relevant data and applies some criterion to select the next operation for that machine. This approach may be computationally very demanding and economically not feasible. Hence less frequent rescheduling may be done. "What is the right frequency of scheduling with its implications on cost and effectiveness?" This has been an important question for researchers. Some studies in this regard are available in Muhlemann, Lockett and Farn.

11.8 SUMMARY

In this unit we have presented the features of job production which essentially involve the production of a variety of jobs in small production quantities with a limited set of facilities. Each job has its unique processing requirement and thus follows its own path through the shop. The variety of problems that may arise in the job shop have been discussed. Depending on the number of jobs to be scheduled, the number of processing facilities, the type of manufacturing facility (flow shop or job shop), the manner in which jobs arrive at the shop (static or dynamic) and the criterion for evaluating the effectiveness of a schedule, different procedures can be used to determine the optimal sequence. Some procedures for the n job one machine case, n job two machine case and the 2 job m machine case have been discussed with simple illustrative examples.

Some priority dispatching rules used in practical job shops have been presented and comments on simulation studies of job shops have been made. The status of research in job shop scheduling is touched upon and some hints for probable future directions of work are given.

11.9 KEY WORDS

Due Date. The deadline for the completion of a job. There could be some penalty if a job is completed beyond its due date.

Dynamic Arrival Pattern: A situation where jobs arrive intermittently at the shop according to a stochastic process.

Flow Shop: A special type of shop in which jobs proceed sequentially through machines.

Flow Time: The time between the arrival of a job and its completion in the shop. It thus equals processing time plus the time that the task waits before being processed.

Job Shop: A production system consisting of different general purpose facilities used to handle a variety of tasks each with its unique processing requirement.

Lateness: Job completion time minus its due date. A positive lateness indicates that the job is finished after its due date whereas a negative lateness shows that it is completed before its due date.

Makespan: The time to complete all jobs.

Schedule: A time table showing when which job is done.

Sequence: The order in which jobs are done.

Slack: Job's due date minus processing time.

Static Arrival Pattern: A situation where all the jobs to be processed in the shop are available at the beginning of the period, and no new jobs arrive during the processing.

Tardiness: The measure of positive lateness.

11.10 SELF-ASSESSMENT EXERCISES

1 What is the distinctive feature of job production as compared to mass and batch production systems?

2 Give five examples from your daily life of job production systems.

3 For the following data concerning 10 jobs to be processed on one processor determine the minimum mean flow time schedule. What is the minimum mean flow time?

Job i	1	2	3	4	5	6	7	8	9	10
Processing Time ti (hours)	5	6	11	8	12	14	7	10	6	12
Importance weight Wi	3	2	1	3	2	1	3	2		
Due date di (hours)	30	30	40	80	100	70	90	80	40	90

4 For the data of problem 3 develop the minimum weighted mean flow time schedule. What is the weighted mean flow time for this schedule?

5 For the data of problem 3 develop the schedule for minimising maximum lateness? What is the mean flow time for this schedule.

6 For the data of problem 3 find a schedule that minimises the number of tardy jobs.

7 For the data of problem 3 find the schedule which minimises mean tardiness. (Use SLACK rule).

8 Eight jobs must be processed through a two machine flow shop. The processing times of each job on both machines are shown below. Determine the schedule to minimise makespan.

Job	1	2	3	4	5	6	7	8
Processing time on MI	10	12	13	7	8	5	4	3
Processing time on M2	4	9	11	8	7	5	10	2

9 Two jobs are required to go through six machines. The processing times as well as the sequence of each job on the machines are:

Job 1 sequence	A	-B	-D	-C	-F	-E
Job 1 times	5	6	4	7	3	4
Job 2 sequence	B	-A	-D	-C	-E	-F
Job 2 times	7	5	6	8	2	1

Find the makespan of the optimal sequences.

10 Study the paper by Blackstone, Phillips and Hogg and summarise the major features of different priority dispatching rules used in job shops.

11.11 FURTHER READINGS

Baker, K.R:, 1974. Introduction to Sequencing and Scheduling, John Wiley: New York.

Bedworth, D.D. and J.E. Ballev, 1982. Integrated Production Control Systems, John Wiley: New York.

Blackstone, J.H., D.T. Phillips and G.L. Hogg, 1982. "A State of the Art Survey of Dispatching Rules for Manufacturing Job Shop Operations", International Journal of Production Research, Vol. 20, No. 1, (Jan/Feb., 1982).

Campbell, H.G., R.A. Dudek, and M.L. Smith, 1970. "A Heuristic Algorithm for the n Job, m Machine Sequencing Problem", Management Science, Vol. 16, No. 11 (pp_630-637).

Elsayed, E.A., and T.O. Boucher, 1985. Analysis and Control of Production Systems, Prentice Hall, Englewood-Cliffs.

Johnson, S. M,, 1954. "Optimal Two- and Three-stage Production Schedules with Setup Times Included," Naval Research Logistics Quarterly, Vol. I, No. 1, (Pp. 61-68)

Moore, J. M., 1968: "Sequencing n jobs on One Machine to Minimise the Number of Tardy Jobs," Management Science, Vol. 17, No. 1, (September, 1968).

Muhlemann, A.P., A, G. Lockett and C.K. Farn, 1982. "Job Shop Scheduling Heuristics and Frequency of Scheduling", International Journal of Production Research, (Vol. 20, 1982).

Muth, J.F. and G.L. Thompson (ed.), 1963. Industrial Scheduling, Prentice Hall Englewood-Cliffs.

Stinson, J.P. and A.W. Smith, 1982. "A Heuristic Programming Procedure for Sequencing the Static Flow Shop", International Journal of Production Research Vol 20, No. 6.

Wilkerson, L.J. and J.D. Irwin, 1971. "An Improved Method for Scheduling Independent Tasks", AIIE Transactions, Vol. 3, No. 3.