

CHAPTER – II

SEQUENCING AND SCHEDULING -AN OVERVIEW

CHAPTER II

SEQUENCING AND SCHEDULING -AN OVERVIEW

Scheduling is defined as the allocation of resources over time to perform a collection of tasks¹. This rather general definition of the term does convey two different meanings that are important in understanding the purpose.

First, scheduling is a decision-making function: it is the process of determining a schedule. In this sense, much of what we learn about scheduling can apply to other kinds of decision making and therefore has general practical value.

Second, scheduling is a body of theory: it is a collection of principles, models, techniques, and logical conclusions that provide insight into the scheduling function.

The vital elements in scheduling models are resources and tasks. The set of tasks available for scheduling does not change over time, the system is called static², in contrast to cases in which new tasks arise over time, where the system is called dynamic³.

2.1 SEQUENCING AND SCHEDULING

Scheduling is the allocation of resources over time to perform a collection of tasks and it is a decision making function. The practical problem of allocating resources over time to perform a collection of tasks arises in a variety of situations. In most cases, however, scheduling does not become a concern until some fundamental planning problems are resolved, and it must be recognized that scheduling decisions are of secondary importance to a broader set of managerial decisions. The scheduling process most often

arises in a situation where resource availability fixed by the long- term commitments of a prior-planning horizon.

Sequencing is the order of processing a set of tasks over available resources. Scheduling involves sequencing⁴ task of allocating as well as the determination of process commencement and completion times i.e., time-tabling. Sequencing problems occur whenever there is a choice to the order in which a group of tasks can be performed. The shop supervisor or scheduler can deal with sequencing problems in a variety of ways. The simplest approach is to ignore the problem and accomplish the tasks in any random order. The most frequently used approach is to schedule heuristically according to predetermined "rules of thumb". In certain cases, scientifically derived scheduling procedures can be used to optimize the scheduling objectives.

2.2 THE TYPICAL OBJECTIVES OF SCHEDULING

. This section addresses some of the objectives of job shop scheduling environment.

2.2.1 Maximization of the Utilization of Resources :

For a finite set of tasks, the utilization of the resources is inversely proportional to the time required to accomplish all the tasks. This time is known as the makespan or maximum flow time of the schedule in a static scheduling system. In finite scheduling problems, resource utilization is improved by scheduling a set of tasks so as to reduce makespan⁵.

2.2.2 Minimization of the Work-in-process Inventory :

The reduction of the average number of tasks waiting in a queue while the resources is busy with other tasks. If the jobs spend less time in the system, the inventory is reduced. In scheduling terms, this leads to minimizing the average flow time (mean flow time) or the average of the times the jobs spend in the system. The makespan of a scheduling is constant, the sequence that reduces mean flow time also reduces mean in-process inventory⁶.

2.2.3 Minimization of Tardiness :

In many situations, some or all of the tasks have due dates and a penalty is incurred if a task is finished after that due date. The possible objectives relating to tardiness are the minimization of maximum tardiness, the minimization of the number of tasks that are tardy, or the minimization of mean tardiness⁷. Sometimes, completion of the jobs ahead of the due dates is also undesirable. For such scheduling problems, the objective is to minimize a penalty function of earliness and tardiness. Here, it can minimize either the number of early/tardy jobs or minimize the maximum earliness/tardiness. If all the jobs have a common due date, this is usually achieved by minimizing the squared deviation of the job completion times about the due date. If the analyst can fix the due date, the problem becomes one of minimizing the variance of completion times.

2.3 SCHEDULING IN MANUFACTURING SYSTEMS

Scheduling can be defined as a matching of needed activities with limited resources to maximize customer satisfaction, maximize shop utilization and minimize operating costs. Scheduling decisions including⁸

- Sequencing
- Timing / Release
- Routing

Typical scheduling environment⁹, in order of similarity and volume of related jobs, are: Job shop, open job shop, batch shop, flow shop, batch/flow shop, manufacturing cell, assembly line, transfer line, and flexible transfer line.

2.4 THE JOB SHOP SCHEDULING PROCESS

The setting in which shop scheduling is done is described below: Each job represents one or more tasks that must be performed in a prescribed order and can be assigned to one or more machines, which have the same or different processing routes. Scheduling consists of two stages, viz. loading and sequencing. The first stage involves assigning jobs to machine tools. If there is a choice of machines; considerations such as quality, setup costs, preventive maintenance, and operation availability may dictate the job assignment. All else being equal, a good rule would be to assign a job to the machine with the least workload. The second stage of job scheduling is to sequence the tasks on the machines, which have multiple tasks to perform.

2.4.1 Scheduling n Tasks on One Processor

The most elementary scheduling problem occurs whenever a set of tasks is waiting to be accomplished and only one processor is available. The processing times and due date of each job are known and are independent of the sequence in which the tasks are run. The scheduling problem in this situation is one of deciding the order in which tasks are to be accomplished. The choice of sequence will fix the completion time of each task. The

makespan, which is the time needed to complete the whole set of tasks, is constant and it is independent of the sequence. If it assume that all tasks are available when the schedule is started, the flow time of each task equals its completion time. Although the makespan is a constant for all the sequences, the mean flow time, mean lateness, and mean tardiness they can be minimized by selecting a good sequence. It is well known that the Shortest Processing Time (SPT) rule minimizes flow times¹⁰ in these systems.

2.4.2 Scheduling n Tasks on m Processors

In the case of m parallel processors, each job visits only one of the processors. In the case of m processors in series, each task must visit each processor in the same order. In the case of m identical processors, the problem is to select both the processor to be used and the sequence for the tasks on each processor. If the objective is to minimize mean flow time, a simple variation of the SPT scheduling rule can be used. Scheduling situations with two or more identical processors in parallel can be handled by first ordering all tasks using some appropriate single-processor rule and then allocating the tasks to the processor with the least scheduled time.

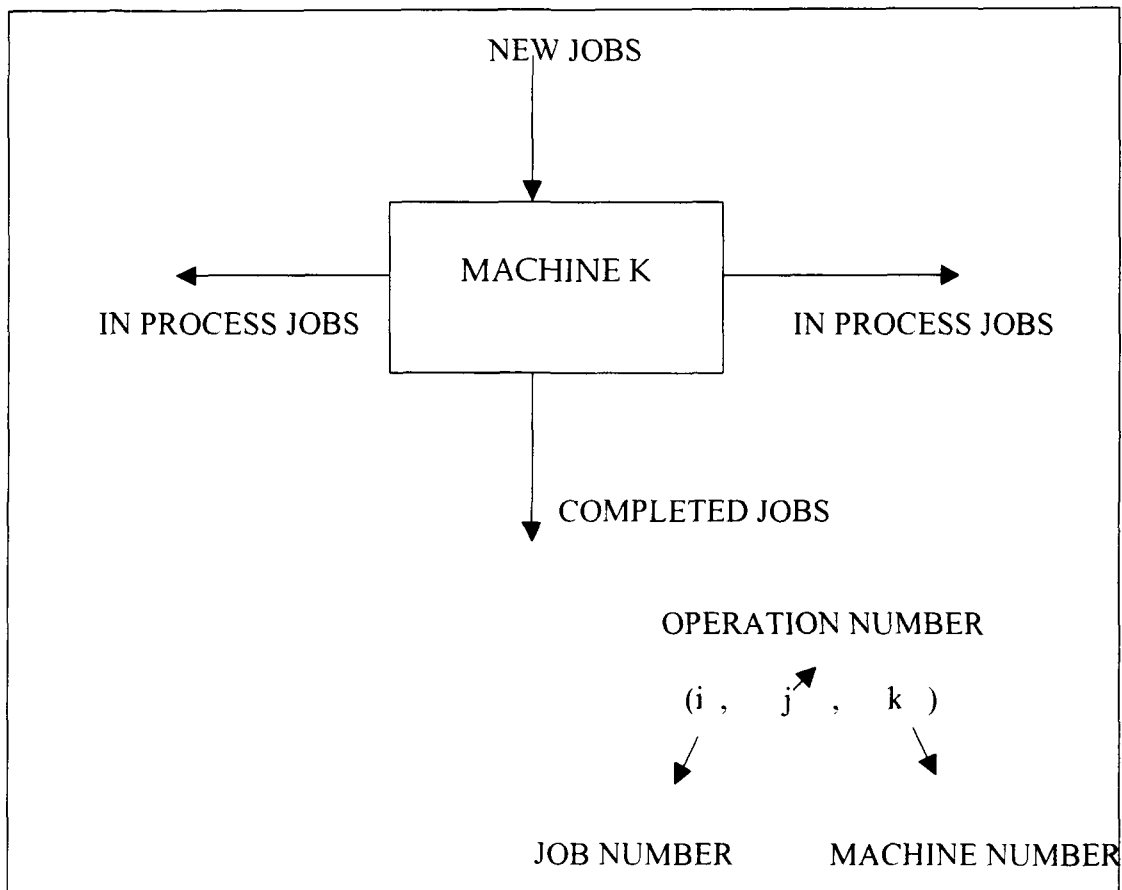
2.3 CLASSICAL JOB SHOP MODEL

The classical Job Shop Scheduling (JSS) problem can be stated as follows. There are m different machines and n different jobs to be scheduled. Each job is composed of a set of operations and order on each machine is prespecified. The required machine and the fixed processing time characterize each operation. In the job shop case it is more appropriate to describe an operation with a triplet (i, j, k) in order to denote the fact that

operation j of job i requires machine k . Workflow at a typical machine¹¹ in a job shop is shown in the Figure 2.1

Figure: 2.1

Work flow at a typical machine in a job shop



Source: Secondary Data

2.4 TYPES OF SCHEDULES

In principle, there are infinite numbers of feasible schedules for any job shop problem, because an arbitrary amount of idle time can be inserted at any machine between adjacent pair of operations. It should be clear, however, that once the operation sequence is specified for each machine, its kind of idle time couldn't be helpful for any regular measure

of performance. Rather, it is desirable that the operations can be processed as compactly as possible. Superfluous idle time exists in a schedule if some operations can be started earlier in time without altering the operation sequences on any machine. Adjusting the start time of some operation in this way is equivalent to moving an operation block to the left on the Gantt chart¹² while preserving the operation sequences. This type of adjustment is called a local left-shift.

Given an operation sequence for each machine, there is only one schedule in which no local shift can be made. The set of all schedules in which no local shift can be made is called the set of semi active schedules and is equivalent to the set of all schedules that contains none of the superfluous idle time. This set dominates the set of all schedules, which means that it is sufficient to consider only semi active schedules to optimize any regular measure of performance.

The number of semi active schedules is at least finite, although it may well be quite large. The exact number is usually difficult to determine. For the classical job shop scheduling problem, in which each job has exactly one operation on each machine, each machine must perform n operations. The number of possible sequences is therefore $n!$ for each machine. If the sequences on each machine were entirely independent, there would be $(n!)^m$ semi active schedules. However, the effect of the precedence structure and machine routing for each job is usually to render some of these combinations of sequences are infeasible.

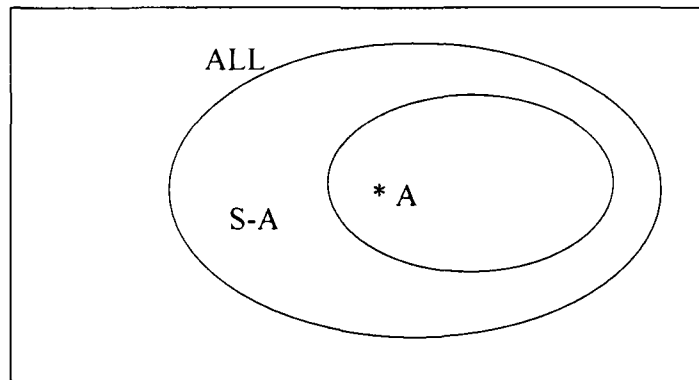
Just as the set of semi active schedules dominates the set of all schedules. In optimizing any regular measure of performance it is sufficient to consider only active schedules¹³. Evidently the number of active schedules is a function of both the routing and

the processing times in a given problem, whereas the number of semi active schedules is a function of only the routings.

While one semi active schedule corresponds to each feasible combination of machine sequences, as discussed above, many semi active schedules can often be compacted into the same active schedule through a series of left-shifts. Several different active schedules can be constructed by a series of left-shifts starting with a given semi active schedule. The roles of semi active and active schedules are shown in the Figure. 2.2

Figure: 2.2

Role of semi active and active schedules



Source: Secondary Data

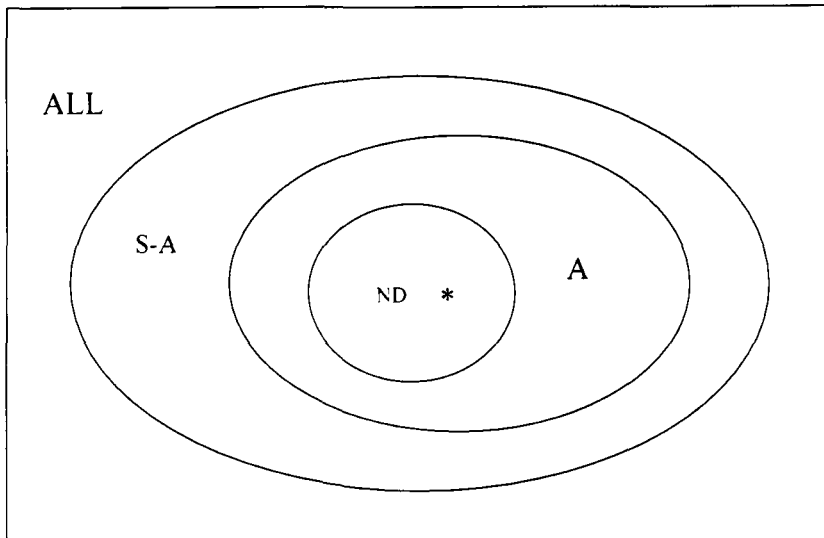
The large rectangle in the Figure. 2.2 represents the roll of semi active and an Active schedule represents the set of all the schedules. The interior of the region labeled S-A represents the finite set of semi active schedules. Wholly contained within that set is the set of active schedules, represented by the region labeled A. The asterisk represents some optimal schedules, placed to indicate that at least one optimum must lie in the active subset.

The number of active schedules still tends to be large, and it is sometimes convenient to focus on an even smaller subset called the non-delay schedules. In a non-delay schedule no machine is kept idle at a time when it could begin processing some operation.

All non-delay schedules are active schedules since no left shifting would be possible. On the other hand many active schedules may not be non-delay schedules. This means that the number of non-delay schedules can be significantly less than the number of active schedules. There is no guarantee that the non-delay subset will contain an optimum. The Figure.2.3 depicts a situation in which at least one optimal schedule, while the Figure.2.4 depicts that no optimal schedule is a non-delay schedule.

Figure:2.3

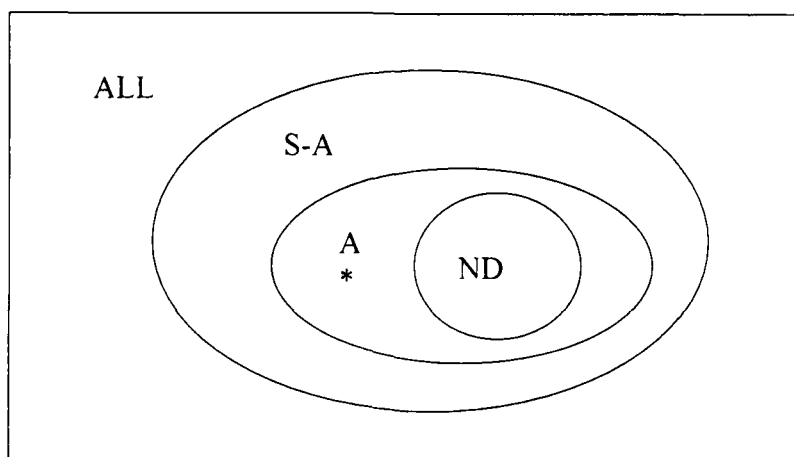
At least one optimal schedule is a non –delay schedule



Source: Secondary Data

Figure:2.4

No optimal schedule is a non delay schedule



Source: Secondary Data

The active schedules are generally the smallest dominated set in the job shop problem. The non-delay schedules are smaller in number but are not dominant. The best non-delay schedules can usually be expected to provide a very good solution, if not an optimum. In this sense, the role of non-delay schedules is similar to that of the permutation schedules in large flow shop problems: although the set is not always dominant, it can usually produce a solution close to the optimum.

2.5 DYNAMIC JOB SHOP SCHEDULING

One basic distinction in scheduling research refers to the nature of the job arrivals in the shop. In a static model jobs arrive simultaneously and are available for to be scheduled at the same instant. Accordingly, their ready or release times are zero. i.e. the total set of jobs is scheduled at time $t = 0$ ("all-at-once-scheduling"). New entering jobs are not admitted to the shop until the preceding scheduling cycle is finished.

A dynamic model allows for a continuous stream of arriving orders in time that are intermittently released to the shop and are included in the current scheduling procedure. Reasonably, the distinction between simultaneous and intermittent job arrivals involves the one between known and fixed job data on one hand and stochastic data, in particular job inter arrival times, on the other hand.

Hence, it is distinguish a static/deterministic-scheduling problem from a dynamic/stochastic one. Here, the emphasis is on the both cases. Within the subset of dynamic/stochastic models deals with experimental, simulation-based approaches, while ignoring the analytical procedures by means of queuing theory systems. The vast majority of simulation-based dynamic job shop scheduling literature assumes an Exponential

distribution¹⁴ of job arrivals and, processing time is considered as random variable, exponential as well as normal distributions occur. The evident advantage of a dynamic scheduling approach is due to the fact that it allows for an up-to-date decision with respect to meanwhile entering (possibly rush) jobs, by loading a machine at the latest possible moment, namely as that machine gets idle. Contrarily, a static model postpones the urgent job to the subsequent scheduling cycle.

Accordingly, the dynamic property of a simulation model exhibits the obvious disadvantage that each sequencing decisions based on the constrained information horizon given by the set of currently schedulable jobs, which prohibits the definition of an overall optimum: due to the real-time capability of a given sequencing decision, only the selection of the first job of the computed job sequence on that machine is actually performed, while the remainder of the scheduled jobs is rescheduled and possibly revised on the occasion of the subsequent loading moment. Thus, rather than to determine a global optimal sequencing policy, a dynamic job shop scheduling simulation¹⁵ at best is able to provide a heuristic optimum among alternative sequencing strategies by which a given job file is scheduled through the shop in successive simulation runs such a policy that defines a specific sequencing decision each time a machine gets idle, is called a priority rule.

A priority rule allows an idle machine to select its next operation from among those available. Primarily, "available" refers to currently waiting jobs at the corresponding machine; but the availability-property may also be extended to jobs being currently in the queue or on the machine of other workstations before proceeding to the queue in question.

¹ Bruker, C. *Scheduling Algorithms* Springer-verlag, Berlin, 1995.

-
- ² Mahendra, S. Bakshi., Sant Ram Arora. "The sequencing problem" Management Science 16 No. 4 (1969): pp 246-263
- ³ Farn, C.K., Muhlemann, A.P. "The dynamic aspects of a production scheduling problem" International Journal of Production Research 17 No. 1 (1979): pp 15-21.
- ⁴ Kenneth R. Baker., *Introduction to sequencing and scheduling*, John Wiley & Sons, Inc.: 1974.
- ⁵ Hemant Kumar, N.S., Srinivasan, G., "A genetic algorithm for job shop scheduling – A case study" Computers in Industry 31 (1996): pp 155-160.
- ⁶ Michael pinedo, *Scheduling theory, algorithms and systems*, Prentice Hall Englewood Cliffs, New Jersey: 1995.
- ⁷ Douglas A. Elvers., "The sensitivity of the relative effectiveness of job shop dispatching rules with respect to various arrival distributions" AIIE Transactions 6 No. 1 (1974): pp 41-49.
- ⁸ Muhlemann, A.P., Lockett, A.G., and Farn, C.K., "Job shop scheduling heuristics and frequency of scheduling" International Journal of Production Research 20 No.2 (1982): pp 227-241.
- ⁹ Haupt, R., "A survey of priority rule based scheduling" Operations Research Spektrum 11 (1989): pp 3-16.
- ¹⁰ Kannan, V.R., Ghosh, S., "An evaluation of the interaction between dispatching rules and truncation procedures in job shop scheduling" International Journal of Production Research 31 No.7 (1993): pp 1637-1654.
- ¹¹ Simon French, *Sequencing and scheduling- An introduction to the mathematics of the job shop*, Ellis Horwood Ltd., England: 1987.
- ¹² Eilon, S., Cotterill, D.J., "A modified SI rule in job shop scheduling" International Journal of Production Research 7 No. 2 (1968): pp 135-145.
- ¹³ Sun, D., Batta, R., "Scheduling larger job shops: A decomposition approach" International Journal of Production Research 34 No.7 (1996): pp 2019-2033.
- ¹⁴ Michael, H. Bulkin., John, L. Colley., and Harry, W. Steinhoff.JR., "Load forecasting, priority sequencing and simulation in a job shop control system" Management Science 13 No.2 (1996): pp B-28-B51.
- ¹⁵ Doctor, S.R., Cavalier, T.M., Egbelu, P.J., "Scheduling for machining and assembly in a job shop environment" International Journal of Production Research 31 No. 6 (1993): pp 1275-1297.