Chapter 2

Deterministic Models: Preliminaries

2.1 Framework and Notation ........................................... 13
2.2 Examples ........................................................................ 20
2.3 Classes of Schedules ...................................................... 21
2.4 Complexity Hierarchy ..................................................... 26

Over the last fifty years a considerable amount of research effort has been focused on deterministic scheduling. The number and variety of models considered is astounding. During this time a notation has evolved that succinctly captures the structure of many (but for sure not all) deterministic models that have been considered in the literature.

The first section in this chapter presents an adapted version of this notation. The second section contains a number of examples and describes some of the shortcomings of the framework and notation. The third section describes several classes of schedules. A class of schedules is typically characterized by the freedom the scheduler has in the decision-making process. The last section discusses the complexity of the scheduling problems introduced in the first section. This last section can be used, together with Appendixes D and E, to classify scheduling problems according to their complexity.

2.1 Framework and Notation

In all the scheduling problems considered the number of jobs and the number of machines are assumed to be finite. The number of jobs is denoted by $n$ and the number of machines by $m$. Usually, the subscript $j$ refers to a job while the subscript $i$ refers to a machine. If a job requires a number of processing steps or operations, then the pair $(i,j)$ refers to the processing step or operation of job $j$ on machine $i$. The following pieces of data are associated with job $j$. 

**Processing time** ($p_{ij}$) The $p_{ij}$ represents the processing time of job $j$ on machine $i$. The subscript $i$ is omitted if the processing time of job $j$ does not depend on the machine or if job $j$ is only to be processed on one given machine.

**Release date** ($r_j$) The release date $r_j$ of job $j$ may also be referred to as the ready date. It is the time the job arrives at the system, i.e., the earliest time at which job $j$ can start its processing.

**Due date** ($d_j$) The due date $d_j$ of job $j$ represents the committed shipping or completion date (i.e., the date the job is promised to the customer). Completion of a job after its due date is allowed, but then a penalty is incurred. When a due date must be met it is referred to as a deadline and denoted by $\bar{d}_j$.

**Weight** ($w_j$) The weight $w_j$ of job $j$ is basically a priority factor, denoting the importance of job $j$ relative to the other jobs in the system. For example, this weight may represent the actual cost of keeping the job in the system. This cost could be a holding or inventory cost; it also could represent the amount of value already added to the job.

A scheduling problem is described by a triplet $\alpha | \beta | \gamma$. The $\alpha$ field describes the machine environment and contains just one entry. The $\beta$ field provides details of processing characteristics and constraints and may contain no entry at all, a single entry, or multiple entries. The $\gamma$ field describes the objective to be minimized and often contains a single entry.

The possible machine environments specified in the $\alpha$ field are:

**Single machine** (1) The case of a single machine is the simplest of all possible machine environments and is a special case of all other more complicated machine environments.

**Identical machines in parallel** ($Pm$) There are $m$ identical machines in parallel. Job $j$ requires a single operation and may be processed on any one of the $m$ machines or on any one that belongs to a given subset. If job $j$ cannot be processed on just any machine, but only on one belonging to a specific subset $M_j$, then the entry $M_j$ appears in the $\beta$ field.

**Machines in parallel with different speeds** ($Qm$) There are $m$ machines in parallel with different speeds. The speed of machine $i$ is denoted by $v_i$. The time $p_{ij}$ that job $j$ spends on machine $i$ is equal to $p_j/v_i$ (assuming job $j$ receives all its processing from machine $i$). This environment is referred to as uniform machines. If all machines have the same speed, i.e., $v_i = 1$ for all $i$ and $p_{ij} = p_j$, then the environment is identical to the previous one.

**Unrelated machines in parallel** ($Rm$) This environment is a further generalization of the previous one. There are $m$ different machines in parallel. Machine $i$ can process job $j$ at speed $v_{ij}$. The time $p_{ij}$ that job $j$ spends on machine $i$ is equal to $p_j/v_{ij}$ (again assuming job $j$ receives all its processing from machine $i$). If the speeds of the machines are independent of the jobs, i.e., $v_{ij} = v_i$ for all $i$ and $j$, then the environment is identical to the previous one.