CHAPTER 1

DETERMINISTIC MACHINE SCHEDULING

We define the basic machine scheduling model, mainly with the intention to introduce some of the notation that will be used throughout the thesis.

1.1 PRELIMINARIES

The machine scheduling problem as considered in this thesis are characterized by (possibly a subset of) the following collection of data. There are *n activities*, or *jobs* $V = \{1, 2, \dots, n\}$. Each job j has a *processing time* $p_j \geqslant 0$, which is the time required to process the job. The jobs have to be processed on *m parallel machines*. Any machine can process only one job at a time, and any job can be scheduled only on one machine at a time. The machines are available continuously from time 0 on. The jobs have to be processed *non-preemptively*, that is, once the processing of a job j has been started, it must be processed continuously for p_j time units. A job $j \in V$ may have a *release date* $r_j \geqslant 0$, which is the earliest time when the processing of the job may start. Finally, there may be *precedence constraints* between the jobs, given as an acyclic, directed graph $G = (V, A)$, where the nodes are given by V, the set of jobs, and the directed arcs $A \subset V \times V$ are the precedence constraints. The intended meaning of a precedence constraint $(i, j) \in A$ is that job j must not be started before job i has been completed. A *schedule* $S = (S_1, S_2, \dots, S_n)$ is an assignment of non-negative *start times* S_j to the jobs $j \in V$ such that

- release dates are respected: $S_j \geqslant r_j$ for all $j \in V$,
- precedence constraints are respected: $S_j \geqslant S_i + p_i$ for all $(i, j) \in A$,
- at any time $t \geqslant 0$ there are no more than m jobs in process:

$$|\{ j \in V \mid S_j \leqslant t < S_j + p_j \}| \leqslant m .$$

Since each job may be processed on any of the machines, the assignment of jobs to machines is irrelevant for the model with identical, parallel machines. Notice that the limited number of available machines constitutes a scarce resource, since no more than m jobs can be processed at any time. For resource constrained scheduling problems as considered in Chapter 5, these constraints will be further generalized. The *completion time* of a job j in a schedule will be denoted by C_j. Notice that $C_j = S_j + p_j$. We speak of a *partial schedule* if non-negative start times are assigned only to jobs from a subset $W \subseteq V$ and if they define a schedule on the subset W. The following definition will be required regularly.

Definition 1.1.1 (availability). *For a given (partial) schedule, a job j is called available at a given time t, if all predecessors of j (with respect to the precedence constraints) have been scheduled and already completed by time t, and if additionally $t \geqslant r_j$.*

Finally, an *objective function* is specified which is to be minimized. It is assumed that it is a function of the completion times of the jobs. The goal is to find a schedule which minimizes the given objective function. We will mainly concentrate on the so-called *makespan*, which is the completion time of the latest job $C_{\max} = \max_{j \in V} C_j$, and the *total weighted completion time* $\sum_{j \in V} w_j C_j$, where w_j is a non-negative *weight* which is thought of as a measure for the job's importance. If all weights w_j are equal to 1, the latter objective function is also called the *total completion time* $\sum C_j$.

In some textbooks on scheduling theory, the above representation of the precedence constraints is called *activity-on-node*, in contrast to the *activity-on-arc* representation, where the arcs of a digraph are associated to the jobs. See, e.g., Elmaghraby (1977) for more details.

1.2 THE $\alpha \,|\, \beta \,|\, \gamma$-NOTATION

Since the number of different problem types that have been considered in the area of scheduling is enormous, it is convenient to use the standard classification scheme by Graham, Lawler, Lenstra, and Rinnooy Kan (1979). A problem is referred to in the three-field notation $\alpha \,|\, \beta \,|\, \gamma$, with the following intended meaning.

- The field α specifies the machine environment. For instance, $\alpha = \mathrm{P}$ denotes the model with identical, parallel machines as described before, $\alpha = \mathrm{Q}$ denotes the problem where machines have different speeds s_k, and the processing time of job j on machine k is p_j / s_k, and $\alpha = 1$ is used for problems with a single machine.

- The field β contains the job characteristics. It can be empty, which implies the default of non-preemptive, independent jobs. Possible entries are, among many others, r_j if release dates are present, *prec* for precedence constrained jobs, or *pmtn* for preemptive jobs (that is, the processing of any job may be interrupted any time and resumed later on any machine).

- The field γ denotes the objective function. It is generally a function of the completion times of the jobs. For the total weighted completion time we write $\gamma = \sum w_j C_j$. For the makespan $\gamma = C_{\max}$. The objective function is called *regular* if it is a component wise non-decreasing function $\mathbb{R}_+^n \to \mathbb{R}_+$ (which is the case for both $\sum w_j C_j$ and $C_{\max}$).

As an example, $\mathrm{P}|r_j, prec|\sum w_j C_j$ is the problem to minimize the total weighted completion time of precedence constrained jobs with release dates on parallel, identical machines. Whenever we only want to specify the machine and job characteristics, but not a particular objective function, we use the notation

$$\alpha\,|\,\beta\,|\,*\,.$$

Generalizations of this three-field notation have been suggested, particularly in order to capture also resource constrained project scheduling problems and other scheduling models that cannot be represented in the original three-field notation of Graham, Lawler, Lenstra, and Rinnooy Kan (1979). Currently, however, none of these generalizations seems to prevail. Let us therefore refer to Blazewicz, Lenstra, and Rinnooy Kan (1983), Brucker (1998), Brucker, Drexl, Möhring, Neumann, and Pesch (1999), Herroelen, Demeulemeester, and De Reyck (1999) for further details on the classification of scheduling problems.