5

Basics of time-dependent scheduling

This chapter completes the first, the introductory, part of the book. In this chapter, we indicate the place of time-dependent scheduling in the general framework of the scheduling theory, formulate the problems of time-dependent scheduling in a more formal way and introduce the terminology used throughout the whole book.

Chapter 5 is composed of five sections. In Sect. 5.1, we present a comparison of the scheduling theory and time-dependent scheduling. In Sect. 5.2, we give the formulation of a generic time-dependent scheduling problem, which is the basis for all time-dependent scheduling problems considered in the book. In Sect. 5.3, we introduce the terminology and notation used in the book for describing time-dependent scheduling problems. In Sect. 5.4, we discuss applications of time-dependent scheduling. The chapter is completed with bibliographic notes in Sect. 5.5.

5.1 The scheduling theory vs. time-dependent scheduling

In Chap. 4, we briefly described the basics of the scheduling theory. The following are the most important assumptions of this theory:

(A1) at every moment of time, each job (operation) can be processed by at most one machine and each machine can process at most one job (operation);
(A2) processing speeds of machines may be different but during the execution of jobs (operations) the speeds do not change in time;
(A3) the processing times of jobs (operations) are fixed and known in advance.

Throughout this book, the scheduling theory with assumptions (A1)–(A3) will be called the *classic scheduling theory*, as opposed to the *non-classic scheduling theory*, where at least one of these assumptions has been changed.

In the period of almost 60 years that elapsed since the classic scheduling theory was formulated, numerous practical problems have appeared, which could not be solved in the framework of this theory. The main reason for that was a certain restrictiveness of assumptions (A1)–(A3). For example, a machine may have a variable speed of processing due to the changing state of this machine, job processing times may increase due to job deterioration, etc. In order to overcome these difficulties and to adapt the theory to cover new problems, assumptions (A1)-(A3) were repeatedly modified. This, in turn, led to new research directions in the scheduling theory, such as scheduling multiprocessor tasks, scheduling on machines with variable processing speed and scheduling jobs with variable processing times. For the completeness of further presentation, we will now shortly describe each of these directions.

5.1.1 Scheduling multiprocessor tasks

In this case, assumption (A1) has been modified: the same operations (called *tasks*) may be performed at the same time by two or more different machines (processors).

The applications of scheduling multiprocessor tasks concern reliable computing in fault-tolerant systems, which are able to detect errors and recover the status of the systems from before an error. Examples of fault-tolerant systems are aircraft control systems, in which the same tasks are executed by two or more machines simultaneously in order to increase the safety of the systems. Other applications of scheduling multiprocessor tasks concern modelling the work of parallel computers, problems of dynamic bandwidth allocation in communication systems and loom scheduling in textile industry.

5.1.2 Scheduling on machines with variable processing speeds

In this case, assumption (A2) has been modified: the machines have variable processing speeds, i.e., the speeds change in time.

There are three main approaches to the phenomenon of the variable processing speeds. In the first approach, it is assumed that the speed is described by a differential equation and depends on a continuous resource. Alternatively, the speed is described by a continuous (the second approach) or a discrete (the third approach) function. In both cases, the speed depends on a resource that is either continuous or discrete.

Scheduling with continuous resources has applications in such production environments in which jobs are executed on machines driven by a common power source, for example, common mixing machines or refueling terminals. Scheduling with discrete resources is applied in modern manufacturing systems, in which jobs to be executed need machines as well as other resources such as robots or automated guided vehicles.

5.1.3 Scheduling jobs with variable processing times

In this case, assumption (A3) has been modified: the processing times of jobs are variable and can change in time.

The variability of job processing times can be modelled in different ways. For example, one can assume that the processing time of a job is a fuzzy number, a function of a continuous resource, a function of the job waiting time, a function of the position of the job in a schedule or is varying in some interval between a certain minimum and maximum value.

Scheduling with variable job processing times has numerous applications, e.g., in the modelling of the forging process in steel plants, manufacturing of preheated parts in plastic molding or in silverware production, finance management and scheduling maintenance or learning activities.

The time-dependent scheduling problems that we will consider in this book are scheduling problems with variable job processing times.

5.2 Formulation of time-dependent scheduling problems

As we said in Sect. 5.1, in time-dependent scheduling problems, the processing time of each job is variable. The general form of the job processing time is as follows.

In parallel-machine time-dependent scheduling problems, the processing time of each job depends on the starting time of the job, i.e.,

$$p_j(S_j) = g_j(S_j), \tag{5.1}$$

where g_j are arbitrary non-negative functions of $S_j \geq 0$ for $1 \leq j \leq n$.

In dedicated-machine time-dependent scheduling problems, the processing time of each operation is in the form of

$$p_{i,j}(S_{i,j}) = g_{i,j}(S_{i,j}), \tag{5.2}$$

where $g_{i,j}$ are arbitrary non-negative functions of $S_{i,j} \geq 0$ for $1 \leq i \leq n_j$ and $1 \leq j \leq n$.

These two forms of presentation, (5.1) and (5.2), are rarely used, since they do not give us any information about the way in which the processing times are changing.

The second way of describing the time-dependent processing time of a job,

$$p_j(S_j) = a_j + f_j(S_j), \tag{5.3}$$

where $a_j \geq 0$ and functions f_j are arbitrary non-negative functions of $S_j \geq 0$ for $1 \leq j \leq n$, is more often encountered. Similarly, the following form of the processing time of an operation,

$$p_{i,j}(S_{i,j}) = a_{i,j} + f_{i,j}(S_{i,j}), \tag{5.4}$$

where $a_{i,j} \geq 0$ and $f_{i,j}$ are arbitrary non-negative functions of $S_{i,j} \geq 0$ for $1 \leq i \leq n_j$ and $1 \leq j \leq n$, is more common than the form (5.2). The main reason for that is the fact that in (5.3) and (5.4), we indicate the constant part a_j ($a_{i,j}$) and the variable part $f_j(S_j)$ ($f_{i,j}(S_{i,j})$) of the job (operation) processing time.

The constant part of a job (operation) processing time, a_j ($a_{i,j}$), will be called the *basic processing time*.

Remark 5.1. The assumption that functions $g_j(S_j)$ and $f_j(S_j)$ ($g_{i,j}(S_{i,j})$ and $f_{i,j}(S_{i,j})$) are non-negative for non-negative arguments is essential and from now on, unless otherwise stated, we will consider it to be satisfied.

Remark 5.2. Since the forms (5.3) and (5.4) of job processing times give us more information, in further considerations, we will mainly use the functions $f_j(S_j)$ ($f_{i,j}(S_{i,j})$).

Remark 5.3. Since the starting time S_j is the variable on which the processing time p_j depends, we will write $p_j(t)$ and $f_j(t)$ instead of $p_j(S_j)$ and $f_j(S_j)$, respectively. Similarly, we will write $p_{i,j}(t)$ and $f_{i,j}(t)$ instead of $p_{i,j}(S_{i,j})$ and $f_{i,j}(S_{i,j})$, respectively.

Remark 5.4. A few authors (Cheng and Sun [45], Lee [192], Lee et al. [195], Toksarı and Güner [268], Wang [281], Wang and Cheng [282, 290]) considered time-dependent scheduling problems with the so-called *learning effect* (cf. Bachman and Janiak [12], Biskup [24]). Since, in this case, job processing times are functions of both the starting time of the job and the job position in the schedule, the problems of this type will be not studied in the book.

Other parameters which describe a time-dependent scheduling problem, such as the parameters of a set of jobs (machines) or the applied optimality criterion, are as those in the classical scheduling (cf. Chap. 4).

Example 5.5. Assume that the set $\mathcal{J}$ is composed of 3 jobs, $\mathcal{J} = \{J_1, J_2, J_3\}$, such that $p_1 = 1 + 3t$, $p_2 = 2 + t$ and $p_3 = 3 + 2t$, there are no ready times and deadlines, and all jobs have unit weights.

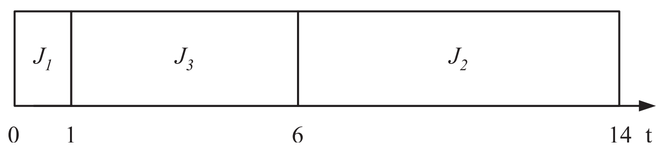

Fig. 5.1: The optimal schedule for Example 5.5

For this set of jobs, there exist the following semi-active schedules (cf. Definition 4.16): $\sigma^1 = (1, 2, 3)$, $\sigma^2 = (1, 3, 2)$, $\sigma^3 = (2, 1, 3)$, $\sigma^4 = (2, 3, 1)$, $\sigma^5 = (3, 1, 2)$ and $\sigma^6 = (3, 2, 1)$. The optimal schedule for the $C_{\max}$ criterion is schedule σ^2, $C_{\max}(\sigma^2) = 14$. The schedule is presented in Fig. 5.1. ♦

Example 5.6. In time-dependent scheduling problems the processing time of the same job may be different in different schedules. For example, consider schedules σ^1 and σ^5 from Example 5.5. The processing time of job J_2 in schedule σ^1 is equal to 4, while in schedule σ^5 it is equal to 41.

Algorithms that solve time-dependent scheduling problems will be called *time-dependent scheduling algorithms*. In this book, we will consider mainly offline time-dependent scheduling algorithms. Examples of online and semi-online time-dependent scheduling algorithms will be given in Chap. 9.

5.3 Terminology and notation

As we said in Sect. 5.2, in any time-dependent scheduling problem, job processing times are described by the functions $f_j(t)$ and $f_{i,j}(t)$, which appear in (5.3) and (5.4), respectively. The form of these functions is related to the problem we consider. For example, if we know nothing about the properties of these functions, then we deal with the *alteration* of job processing time: the processing time of a job varies in time in an unknown way. If we know something more, e.g., whether these functions are monotonic, then two cases are worth considering:

1° $f_j(t)$ and $f_{i,j}(t)$ are increasing (or non-decreasing);
2° $f_j(t)$ and $f_{i,j}(t)$ are decreasing (non-increasing).

The first case is more often encountered in the literature and, as it seems, it is easier to study. The case when job processing times are described by increasing (non-decreasing) functions will be called *deteriorating processing times*: while waiting for processing, the jobs deteriorate and as a result the processing time of each job increases in time.

The second case may cause some problems already at the stage of problem formulation, since we have to make some additional assumptions to avoid the case of negative job processing times. The case when job processing times are described by decreasing (non-increasing) functions will be called *shortening processing times*: unlike the previous case, jobs grow shorter and the processing time of each job is reduced in time.

Remark 5.7. Regardless of the type of functions that we have chosen to describe job processing times in our problem, we still deal with *deterministic* scheduling, since all parameters of the problem are assumed to be known in advance. This objection is important, since *stochastic* scheduling problems with deteriorating jobs are also considered (see, e.g., Glazebrook [112]).

Generally, the time-scheduling problems considered in this book will be denoted using the $\alpha|\beta|\gamma$ notation (see Sect. 4.4 for details). Each problem will be denoted by $\alpha_1\alpha_2|p_j(t) = a_j + f_j(t)|\varphi$ or $\alpha_1\alpha_2|p_{i,j}(t) = a_{i,j} + f_{i,j}(t)|\varphi$, where $\alpha_1\alpha_2$, $f_j(t)$ and φ denote the machine environment, the form of the variable part of job processing time and the criterion function, respectively.

Remark 5.8. We will use the $\alpha|\beta|\gamma$ notation if it will yield a simple notation for the considered scheduling problem. In some cases, however, we will resign from the notation in favour of the description by words if the descriptive approach will be more readable.

The short form of the symbol $\alpha_1\alpha_2|p_j(t) = a_j + f_j(t)|\varphi$ is the symbol $\alpha_1\alpha_2|p_j = a_j + f_j(t)|\varphi$. The short form of the symbol $\alpha_1\alpha_2|p_{i,j}(t) = a_{i,j} + f_{i,j}(t)|\varphi$ is the symbol $\alpha_1\alpha_2|p_{i,j} = a_{i,j} + f_{i,j}(t)|\varphi$. Throughout this book, we will use the short form of the symbols which will denote time-dependent scheduling problems.

If the form of the functions $f_j(t)$ ($f_{i,j}(t)$) is known, we will call the processing times $p_j = a_j + f_j(t)$ ($p_{i,j} = a_{i,j} + f_{i,j}(t)$) by the name of the function. For example, if the functions $f_j(t)$ ($f_{i,j}(t)$) are proportional (linear, polynomial, etc.), the processing times will be called *proportional* (*linear*, *polynomial*, etc.) processing times. If the functions are non-negative (non-positive), the processing times will be called *deteriorating* (*shortening*) processing times.

In a similar way, we will call the processing times of jobs, if non-linear forms of job deterioration are considered. For example, if the functions $f_j(t)$ are step functions or piecewise proportional-step functions, the processing times will be called *step* and *proportional-step* processing times, respectively.

If the same function $f(t)$ is used for all jobs, $f_j(t) = f(t)$ for $1 \leq j \leq n$ or $f_{i,j}(t) = f(t)$ for $1 \leq i \leq n_j$ and $1 \leq j \leq n$, we will speak about *simple deterioration* (*shortening*) of job processing times. In the opposite case, we will speak about *general deterioration* (*shortening*) of job processing times.

Example 5.9.

(a) The symbol $1|p_j = b_j t|C_{\max}$ will denote a single machine scheduling problem with proportional job processing times and the $C_{\max}$ criterion.

(b) The symbol $Pm|p_j = a_j + f(t)|\sum C_j$ will denote a multiple identical machine scheduling problem with simple general deterioration of jobs and the $\sum C_j$ criterion.

(c) The symbol $F2|p_{i,j} = a_{i,j} + b_{i,j}t|L_{\max}$ will denote a two-machine flow shop problem with linear job processing times and the $L_{\max}$ criterion.

(d) The symbol $O3|p_{i,j} = b_{i,j}t, b_{i,3} = b|C_{\max}$ will denote a three-machine open shop problem with proportional job processing times such that all job processing times on machine M_3 are equal to each other, and with the $C_{\max}$ criterion.

(e) The symbol $J2|p_{i,j} = b_{i,j}t|C_{\max}$ will denote a two-machine job shop problem with proportional job processing times and the $C_{\max}$ criterion. ◆

Remark 5.10. In Sect. 6.1, we will extend the $\alpha|\beta|\gamma$ notation to include the symbols describing time-dependent scheduling problems in batch environments and on machines with non-availability periods.

5.4 Applications of time-dependent scheduling

The motivation for research into time-dependent scheduling follows from the existence of many real-life problems which can be formulated in terms of scheduling jobs with time-dependent processing times. Such problems appear in all cases in which any delay in processing causes an increase (a decrease) of the processing times of executed jobs. If job processing times increase, we deal with deteriorating job processing times; if they decrease, we deal with shortening job processing times. In this section, we give a few examples of problems which can be modelled in time-dependent scheduling.

5.4.1 Scheduling problems with deteriorating job processing times

Gupta et al. [129] consider the problem of the *repayment of multiple loans*. We have to repay n loans, $L_1, L_2, \ldots, L_n$. A loan may represent an amount of borrowed cash or a payment to be made for a credit purchase. Loan L_k qualifies for a discount u_k if it is paid on or before a specified time b_k. A penalty at the rate v_k per day is imposed if the loan is not paid by due date $d_k, 1 \leq k \leq n$. The debtor earmarks a constant amount of q dollars per day, $q < v_k$, for repayment of the loans. Cash flows are continuously discounted with discount factor $(1+r)^{-1}$. The aim is to find an optimal repayment schedule that minimizes the present value PV of all cash outflows, $PV := \sum_{k=1}^{n} \frac{A_k}{(1+r)^{T_k}}$, where A_k and T_k denote, respectively, the actual amount paid for loan L_k and the time at which the loan L_k is repaid, $1 \leq k \leq n$. This problem can be modelled as a single-machine scheduling problem with time-dependent job processing times and the PV criterion.

Mosheiov [217] considers the following problem of *scheduling maintenance procedures*. A set of n maintenance procedures P_k, $1 \leq k \leq n$, has to be executed by $m \geq 1$ machines. A maintenance procedure P_k has to take place before a specified deadline d_k. The procedure consists of a series of actions, which last altogether p_k^1 time units. If the procedure does not complete by the deadline, several additional actions are required. The new processing time of procedure P_k is $p_k^2 > p_k^1$ time units. The aim is to find an order of execution of maintenance procedures $P_1, P_2, \ldots, P_n$, which minimizes the maximum completion time of the last executed procedure. This problem can be modelled as a single- or multiple-machine scheduling problem with two-step deteriorating job processing times.

Gawiejnowicz et al. [103] consider the following problem of *scheduling derusting operations*. We are given n items (e.g., parts of devices), which are subject to maintenance (e.g., they should be cleared from rust). This maintenance is performed by a single worker, who can himself determine the sequence of maintenance procedures. All procedures are non-preemptable, i.e., no maintenance procedure can be interrupted once it has started. At the moment $t = 0$, all items need the same amount of time for maintenance, e.g., one unit of time. As time elapses, each item corrodes at a rate that depends on the kind of the material from which the particular item is made. The rate of corrosion for the j-th item is equal to b_j, $1 \leq j \leq n$, and the time needed for the maintenance of each item grows proportionally to the time that elapsed from the moment $t = 0$. The problem is to choose such a sequence of the maintenance procedures that minimizes the total completion time of maintenance of all items. This problem can be modelled as the single-machine time-dependent scheduling problem $1|p_j = 1 + b_j t|\sum C_j$.

Rachaniotis and Pappis [240] consider the problem of *scheduling a single fire-fighting resource* in the case when there are several fires to be controlled. The aim is to find such order of supressing n existing fires that the total damage caused by the fires is minimized. The problem can be modelled as a single machine scheduling problem with time-dependent processing times and the total cost minimization criterion.

5.4.2 Scheduling problems with shortening job processing times

Ho et al. [135] consider the problem of *recognizing aerial threats*. A radar station recognizes some aerial threats approaching the station. The time required to recognize the threats decreases as they get closer. The aim is to find an optimal order of recognizing the threats which minimizes the maximum completion time. This problem can be modelled as a single-machine scheduling problem with shortening job processing times and the $C_{\max}$ criterion.

Kunnathur and Gupta [178] and Ng et al. [226] consider the problem of *producing ingots in a steel mill*. A set of ingots has to be produced in a steel mill. After being heated in a blast furnace, hot liquid metal is poured into steel ladles and next into ingot moulds, where it solidifies. Next, after the ingot stripper process, the ingots are segregated into batches and transported to the soaking pits, where they are preheated up to a certain temperature. Finally, the ingots are hot-rolled on the blooming mill. If the temperature of an ingot, while waiting in a buffer between the furnace and the rolling machine, has dropped below a certain value, then the ingot needs to be reheated to the temperature required for rolling. The reheating time depends on the time spent by the ingot in the buffer. The problem is to find a sequence of preheating the ingots which minimizes the maximum completion time of the last ingot produced. This problem can be modelled as a single machine scheduling problem with shortening job processing times and the $C_{\max}$ criterion.

5.4.3 Other examples of time-dependent scheduling problems

Shakeri and Logendran [254] consider the following problem of *maximizing satisfaction level* in a multitasking environment. Several plates are spinning on vertical poles. An operator has to ensure all plates spin as smoothly as possible. A value, called the *satisfaction level*, can be assigned to each plate's spinning state. The satisfaction level of a plate is ranging from 0% (i.e., the plate is not spinning) up to 100% (the plate is spinning perfectly). The objective is to maximize the average satisfaction level of all plates over time.

The above problem is applicable to multitasking environments in which we cannot easily determine the completion time of any job. Examples of such environments are the environments of the control of a plane flight parameters, monitoring air traffic or the work of nuclear power plants. A special case of the problem, when a 100% satisfaction level is equivalent to the completion of a job, is a single-machine time-dependent scheduling problem.

Other examples of practical problems which can be modelled in terms of time-dependent scheduling include the control of queues in communication systems in which jobs deteriorate as they wait for processing (Browne and Yechiali [33]), search for an object in worsening weather or growing darkness, performance of medical procedures under deterioration of the patient conditions and repair of machines or vehicles under deteriorating mechanical conditions (Mosheiov [216]).

We refer the reader to the literature (see Alidaee and Womer [6] and Cheng et al. [55]) for more examples of time-dependent scheduling applications.

5.4.4 Scheduling problems with time-dependent parameters

The time dependence may concern not only job processing times but also other parameters of a scheduling problem. For example, Cai et al. [38] consider the following *crackdown scheduling problem*. There are n illicit drug markets, all of which need to be brought down to a negligible level of activity. Each market is eliminated by a procedure consisting in a crackdown phase and a maintenance phase. The crackdown phase utilizes all the available resources until the market is brought down to the desired level. The maintenance phase, which follows after the crackdown phase and uses a significantly smaller amount of resources, maintains the market at this level. The aim is to find an order of elimination of the drug markets that minimizes the total time spent in eliminating all drug markets. The problem can be modelled as a single-machine scheduling problem of minimizing the total cost $\sum f_j$, where f_j are monotonically increasing time-dependent cost functions.

Other examples of scheduling problems in which some parameters are time dependent include multiprocessor tasks scheduling (Bampis and Kononov [16]), scheduling in a contaminated area (Janiak and Kovalyov [147, 148]), multicriteria project sequencing (Klamroth and Wiecek [165]), selection problems (Seegmuller et al. [253]) and scheduling jobs with deteriorating job values (Voutsinas and Pappis [275]).

With these remarks, we end the presentation of the basics of time-dependent scheduling. This chapter also ends the first part of the book. In the next part, we will consider the complexity of time-dependent scheduling problems.

5.5 Bibliographic notes

The problems of scheduling multiprocessor tasks are reviewed in detail by Drozdowski [73, 74] and Lee et al. [190].

The problems of scheduling with continuous resources are discussed by Błażewicz et al. [27, Chap. 12] and Gawiejnowicz [87].

The problems of scheduling on machines with variable speed are considered, e.g., by Dror et al. [72], Gawiejnowicz [88, 89, 90], Meilijson and Tamir [206] and Trick [269].

The variability of the processing time of a job can be modelled in many different ways. The job processing time can be, e.g., a function of the job waiting time (see, e.g., Barketau et al. [17], Finke and Jiang [80], Finke et al. [81], Finke and Oulamara [82], Leung et al. [199], Lin and Cheng [203], Sriskandarajah and Goyal [261]), a function of a continuous resource (see, e.g., Janiak [144, 145]) or a fuzzy number (see, e.g., Słowiński and Hapke [259]).

The processing time of a job may also depend on the position of the job in a schedule (Bachman and Janiak [12], Biskup [24]), the length of machine non-availability period (Lahlou and Dauzère-Pérès [181]) or varies in some interval between a certain minimum and maximum value (see, e.g., Nowicki and Zdrzałka [227], Shakhlevich and Strusevich [255], Vickson [273]).

The problems of time-dependent scheduling are reviewed by Alidaee and Womer [6] and Cheng et al. [55].

Gawiejnowicz [87] discusses time-dependent scheduling problems in the framework of scheduling with discrete and continuous resources.