1

Introduction

1.1 Introduction to Sequencing and Scheduling

Scheduling is a term in our everyday vocabulary, although we may not always have a good definition of it in mind. Actually, it's not scheduling that is a common concept in our everyday life; rather it is *schedules*. A schedule is a tangible plan or document, such as a bus schedule or a class schedule. A schedule usually tells us when things are supposed to happen; it shows us a plan for the timing of certain activities and answers the question, "If all goes well, when will a particular event take place?" Suppose we are interested in when dinner will be served or when a bus will depart. In these instances, the event we are interested in is the completion of a particular activity, such as preparing dinner, or the start of a particular activity such as a bus trip. Answers to the "when" question usually come to us with information about timing. Dinner is scheduled to be served at 6:00 p.m., the bus is scheduled to depart at 8:00 a.m., and so on. However, an equally useful answer might be in terms of sequence rather than timing: That is, dinner will be served as soon as the main course is baked, or the bus will depart right after cleaning and maintenance are finished. Thus, the "when" question can be answered by timing or by sequence information obtained from the schedule.

If we take into account that some events are unpredictable, then changes may occur in a schedule. Thus, we may say that the bus leaves at 8:00 a.m. unless it is delayed for cleaning and maintenance, or we may leave the condition implicit and just say that the bus is scheduled to leave at 8:00 a.m. If we make allowances for uncertainty when we schedule cleaning and maintenance, then passengers can trust that the bus will leave at 8:00 a.m. with some confidence. Using a time buffer (or *safety time*) helps us cope with uncertainty.

Principles of Sequencing and Scheduling, Second Edition. Kenneth R. Baker and Dan Trietsch.
© 2019 John Wiley & Sons, Inc. Published 2019 by John Wiley & Sons, Inc.

Intuitively, we think of scheduling as the process of generating the schedule, although we seldom stop to consider what the details of that process might be. In fact, although we think of a schedule as something tangible, the process of scheduling seems intangible, at least until we consider it in some depth. For example, we often approach the problem in two steps: sequencing and scheduling. In the first step, we plan a sequence or decide how to select the next task. In the second step, we plan the start time, and perhaps the completion time, of each task. The determination of safety time is part of the second step.

Preparing a dinner and doing the laundry are good examples of everyday scheduling problems. They involve tasks to be carried out, the tasks are well specified, and particular resources are required – a cook and an oven for dinner preparation and a washer and a dryer for laundry. Scheduling problems in industry have similar elements: they contain a set of tasks to be carried out and a set of resources available to perform those tasks. Given tasks and resources, together with some information about uncertainties, the general problem is to determine the timing of the tasks while recognizing the capability of the resources. This scheduling problem usually arises within a decision-making hierarchy in which it follows some earlier, more basic decisions. Dinner preparation, for example, typically requires a specification of the menu items, recipes for those items, and information on how many portions are needed. In industry, analogous decisions are usually part of the *planning* function. Among other things, the planning function might describe the design of a company's products, the technology available for making and testing the required components, and the volumes that are required. In short, the planning function determines the resources available for production and the tasks to be scheduled.

In the scheduling process, we need to know the type and the amount of each resource so that we can determine when the tasks can feasibly be accomplished. When we specify the tasks and resources, we effectively define the boundary of the scheduling problem. In addition, we describe each task in terms of such information as its resource requirement, its duration, the earliest time at which it may start, and the time at which it is due to complete. If the task duration is uncertain, we may want to suppress that uncertainty when stating the problem. We should also describe any logical constraints (precedence restrictions) that exist among the tasks. For example, in describing the scheduling problem for several loads of laundry, we should specify that each load requires washing to be completed before drying begins.

Along with resources and tasks, a scheduling problem contains an objective function. Ideally, the objective function should consist of all costs that depend on scheduling decisions. In practice, however, such costs are often difficult to measure or even to completely identify. The major operating costs – and the most readily identifiable – are determined by the planning function, while scheduling-related costs are difficult to isolate and often tend to appear fixed. Nevertheless, three types of decision-making goals seem to be prevalent in scheduling: *turnaround*, *timeliness*, and *throughput*. Turnaround measures the time required to complete a task. Timeliness measures the conformance of a particular task's completion to a given deadline. Throughput measures the amount of work completed during a fixed period of time. The first two goals need further elaboration, because although we can speak of turnaround or timeliness for a given task, scheduling problems require a performance measure for the entire set of tasks in a schedule. Throughput, in contrast, is already a measure that applies to the entire set. As we develop the subject of scheduling in the following chapters, we will elaborate on the specific objective functions that make these three goals operational.

We describe a scheduling problem by providing information about tasks, resources, and an objective function. However, finding a solution is often a fairly complex matter, and formal problem-solving approaches are helpful. Formal models help us first to understand the scheduling problem and then to find a good solution systematically. For example, one of the simplest and most widely used models is the *Gantt chart*, which is an analog representation of a schedule. In its basic form, the Gantt chart displays resource allocation over time, with specific resources shown along the vertical axis and a time scale shown along the horizontal axis. The basic Gantt chart assumes that processing times are known with certainty, as in Figure 1.1.

A chart such as Figure 1.1 helps us to visualize a schedule and its detailed elements because resources and tasks show up clearly. With a Gantt chart, we can discover information about a given schedule by analyzing geometric relationships. In addition, we can rearrange tasks on the chart to obtain comparative information about alternative schedules. In this way, the Gantt chart serves as an aid for measuring performance and comparing schedules as well as for visualizing the problem in the first place. In this book, we will examine graphical, algebraic, spreadsheet, and simulation models, in addition to the Gantt

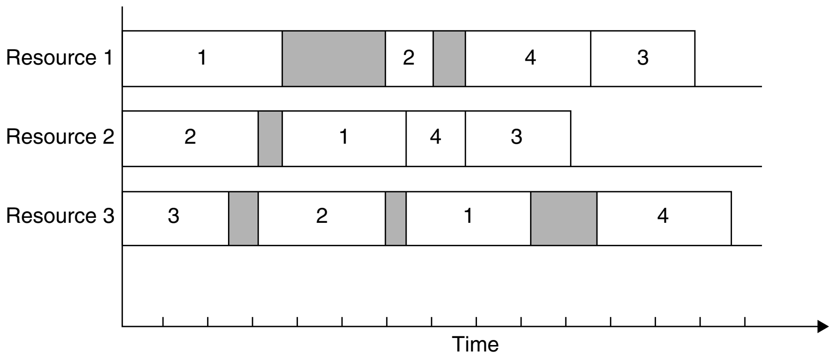

Figure 1.1 A Gantt chart.

chart, all of which help us analyze and compare schedules. In essence, models help us formalize the otherwise intangible process we call scheduling.

Many of the early developments in the field of scheduling were motivated by problems arising in manufacturing. Therefore, it was natural to employ the vocabulary of manufacturing when describing scheduling problems. Now, although scheduling work is of considerable significance in many nonmanufacturing areas, the terminology of manufacturing is still frequently used. Thus, resources are usually called *machines* and tasks are called *jobs*. Sometimes, jobs may consist of several elementary tasks called *operations*. The environment of the scheduling problem is called the *job shop*, or simply, the *shop*. For example, if we encounter a scheduling problem faced by underwriters processing insurance policies, we could describe the situation generically as an insurance "shop" that involves the processing of policy "jobs" by underwriter "machines."

1.2 Scheduling Theory

Scheduling theory is concerned primarily with mathematical models that relate to the process of scheduling. The development of useful models, which leads in turn to solution techniques and practical insights, has been the continuing interface between theory and practice. The theoretical perspective is also largely a quantitative approach, one that attempts to capture problem structure in mathematical form. In particular, this quantitative approach begins with a description of resources and tasks and translates decision-making goals into an explicit objective function.

We categorize the major scheduling models by specifying the resource configuration and the nature of the tasks. For instance, a model may contain one machine or several machines. If it contains one machine, jobs are likely to be single-stage activities, whereas multiple machine models usually involve jobs with multiple stages. In either case, machines may be available in unit amounts or in parallel. In addition, if the set of jobs available for scheduling does not change over time, the system is called *static*, in contrast to cases in which new jobs appear over time, where the system is called *dynamic*. Traditionally, static models have proven to be more tractable than dynamic models and have been studied more extensively. Although dynamic models would appear to be more important for practical application, static models often capture the essence of dynamic systems, and the analysis of static problems frequently uncovers valuable insights and sound heuristic principles that are useful in dynamic situations. Finally, when conditions are assumed to be known with certainty, the model is called *deterministic*. On the other hand, when we recognize uncertainty with explicit probability distributions, the model is called *stochastic*.

Two kinds of *feasibility* constraints are commonly found in scheduling problems. First, limits exist on the capacity of machines, and second, technological restrictions exist on the order in which some jobs can be performed. A *solution* to a scheduling problem is any feasible resolution of these two types of constraints, so that "solving" a scheduling problem amounts to answering two kinds of questions:

- Which resources should be allocated to perform each task?
- When should each task be performed?

In other words, a scheduling problem gives rise to allocation questions and sequencing questions. From the start, the scheduling literature has relied on mathematical models to help answer such questions. In more recent developments, referred to as *safe scheduling*, the models use safety time to mitigate disruptions due to uncertainty.

Traditionally, many scheduling problems have been viewed as problems in optimization subject to constraints – specifically, problems in allocation and sequencing. Sometimes, scheduling is purely allocation (e.g. choosing the product mix with limited resources), and in such cases mathematical programming models are usually appropriate for determining optimal decisions. These general techniques are described in many available textbooks and are not emphasized in our coverage. At other times, scheduling is purely sequencing. In these cases, the problems are unique to scheduling theory and account for much of our emphasis in the chapters that follow.

The theory of scheduling also includes a variety of methodologies. Indeed, the scheduling field has become a focal point for the development, application, and evaluation of combinatorial techniques, simulation procedures, and heuristic solution approaches. The selection of an appropriate method depends mainly on the nature of the model and the choice of objective function. In some cases, it makes sense to consider alternative methods. For this reason, it is important to study methodologies as well as models.

A useful perspective on the relation of scheduling problems and their solution techniques comes from developments in a branch of computer science known as *complexity theory*. The notion of complexity refers to the computing effort required by a solution algorithm. Computing effort is described by order-of-magnitude notation. For example, suppose we use a particular algorithm to solve a problem of size n. (Technically, n denotes the amount of information needed to specify the problem.) The number of computations required by the algorithm is typically bounded from above by a function of n. If the order of magnitude of this function is polynomial as n gets large, then we say the algorithm is *polynomial*. For instance, if the function has order of magnitude n^2, denoted $O(n^2)$, then the algorithm is polynomial. On the other hand, if the function is $O(2^n)$, then the algorithm is nonpolynomial (in this case, exponential). Other things being equal, we prefer to use a polynomial algorithm because as n grows large, polynomial algorithms are ultimately faster.

A class of problems called *NP-complete* problems includes many well-known and difficult combinatorial problems. These problems are equivalent in the sense that if one of them can be solved by a polynomial algorithm, then so can the others. However, many years of research by mathematicians and computer scientists have not yielded a polynomial algorithm for any problem in this class, and the conjecture is that no such algorithm exists. Optimization problems as difficult as these, or even more difficult, are called *NP-hard* problems. The usefulness of this concept, which applies to many scheduling problems, is that if we are faced with the need to solve large versions of an NP-hard problem, we know in advance that we may not be able to find optimal solutions with available techniques. We might be better off to use a *heuristic* solution procedure that has a more modest computational requirement but does not guarantee optimality. NP-hard instances exist for which it would take less time to actually perform the work in the shop (using any reasonable sequence) than to solve the problem optimally on the fastest available computer. Therefore, the reliance on heuristics is often the rule in practice, rather than the exception. Finally, some solution procedures involve simulation. Although simulation is inherently imprecise, it can produce nearly optimal solutions that are completely satisfactory for practical purposes. In that respect, simulation is conceptually similar to the use of heuristics.

We will have occasion to refer to the computational complexity of certain algorithms. We will also mention that certain problems are known to be NP-hard. This is relevant information for classifying many of the problems we introduce, but the details of complexity theory are beyond the scope of our main coverage. For a thorough introduction to the subject, see Garey and Johnson (1979).

1.3 Philosophy and Coverage of the Book

Scheduling now represents a body of knowledge about models, techniques, and insights related to actual systems. If we think of scheduling as including pure allocation problems, the formal development of models and optimization techniques for modern scheduling theory probably began in the years preceding World War II. Formal articles on properties of specialized sequencing problems gained recognition in the 1950s, and textbooks on the subject date from the 1960s. An early collection of relevant papers is Muth and Thompson (1963), and the seminal work in the field is Conway, Maxwell, and Miller (1967). Articles and textbooks, not to mention the demand for solving scheduling problems in government and industry, stimulated even more books in the field during the 1970s and 1980s. The better-known examples are Coffman (1976) and French (1982), in addition to the first precursor of this volume, Baker (1974). Eventually, additional perspectives were compiled by Morton and Pentico (1993), Blazewicz et al. (1993), Pinedo (1995), Brucker (1995), Leung (2002), and T'Kindt and Billaut (2002). Now the field of deterministic scheduling is well developed, and there is a growing literature on stochastic models, including safe scheduling.

With this perspective as background, we can think of scheduling knowledge as a tree. Around 1970, it was possible to write a textbook on scheduling that would introduce a student to this body of knowledge and, in the process, examine nearly every leaf. In a reasonable length text, it was possible to tell the student "everything you always wanted to know" about scheduling. But over the last three decades, the tree has grown considerably. Writing a scheduling text and writing a scheduling encyclopedia are no longer similar tasks.

This material is a text. The philosophy here is that a broad introduction to scheduling knowledge is important, but it is no longer crucial to study every leaf on the tree. A student who prepares by examining the trunk and the major branches will be capable of studying relevant leaves thereafter. This book addresses the trunk and the major branches: it emphasizes basic knowledge that will prepare the reader to delve into more advanced sources with a firm sense of the scope of the field and the major findings within it. Thus, our first objective is to provide a sound basis in deterministic scheduling, because it is the foundation of all scheduling models. As such, the book can be thought of as a new edition of its precursors, Baker (1974) and (2005). But we also have a new objective: to present the emerging theory of safe scheduling (Baker and Trietsch 2007) and to anticipate the future directions in which it may develop. There are growing concerns after half a century of intensive development that scheduling theory has not yet delivered its full promise. One reason for this shortcoming could be the fact that most scheduling models do not address safety time. For this reason, we believe that our second objective is an important one.

Our pedagogical approach is to build from specific to general. In the early chapters, we begin with basic models and their analysis. That knowledge forms the foundation on which we can build a broader coverage in later chapters, without always repeating the details. The priority is on developing insight through the use of specific models and logical analyses. In the early chapters we concentrate on deterministic scheduling problems, along with a number of optimal and heuristic solution techniques. That foundation is followed by a chapter introducing stochastic scheduling and another chapter with our initial coverage of safe scheduling. Thereafter, we address safe scheduling issues as extensions of the deterministic models, in the spirit of building from the specific to the general.

We approach the topic of scheduling with a mathematical style. We rely on mathematics in order to be precise, but our coverage does not pursue the mathematics of scheduling as an end in itself. Some of the results are presented as theorems and justified with formal proofs. The idea of using theorems is not so much to emphasize mathematics as it is simply to draw attention to key results. The use of formal proofs is intended to reinforce the importance of logical analysis in solving scheduling problems. Similarly, certain results are presented in the form of algorithms. Here, again, the use of algorithms is not an end in itself but rather a way to reinforce the logic of the analysis. Scheduling is not mainly about mathematics nor is it mainly about algorithms, but we use such devices to develop systematic knowledge and understanding about the solution of scheduling problems.

The remainder of this book consists of 18 chapters. Chapter 2 introduces the basic single-machine model, deals with static sequencing problems under the most simplifying set of assumptions, and examines a variety of scheduling criteria. By the end of Chapter 2, we will have encountered some reasonably challenging sequencing problems, enough to motivate the study of general-purpose optimization methodologies in Chapter 3 and heuristic methods in Chapter 4. In Chapter 5, the discussion examines a variation of the single-machine model that has been the subject of intensive study and that also happens to be highly relevant for safe scheduling. Chapter 6 introduces stochastic models, and in Chapter 7, we introduce the most basic safe scheduling models. In Chapter 8, we relax several of the elementary assumptions and analyze the problem structures that result.

The second section of the book deals with models containing several machines. Chapter 9 examines the scheduling of single-stage jobs with parallel machines, and Chapters 10 and 11 examine the flow shop model, which involves multistage jobs and machines in series. Chapter 12 takes a look at the details of workflow in the flow shop. Chapter 13 treats the case where it is more economical to batch jobs into groups, or families, and sequence among groups and within groups in two separate steps. Chapter 14 is an overview of the most widely known scheduling model, the job shop, which also contains multistage jobs but which does not have the serial structure of the flow shop. Chapter 15 discusses simulation results for job shops. To a large extent, the understanding of models, techniques, and insights, which we develop in the preceding chapters, is integrated in the study of the job shop. Similarly, the knowledge developed in studying this material builds the integrative view necessary for success in further research and application in the field of scheduling.

In the third section of the book, we focus on nonmanufacturing applications of scheduling. Chapter 16 covers basic project scheduling, and Chapter 17 discusses the added complications of resource constraints. Chapter 18 shows how to obtain reliable data with which to feed project scheduling models. Finally, Chapter 19 extends project scheduling to include safe scheduling considerations. Two technical appendices support our coverage of stochastic models.

Bibliography

Baker, K.R. (1974). *Introduction to Sequencing and Scheduling*. Hoboken, NJ: Wiley.

Baker, K.R. (2005). *Elements of Sequencing and Scheduling*. Hanover, NH: Tuck School of Business.

Baker, K.R. and Trietsch, D. (2007). *Safe Scheduling, Chapter 5 in Tutorials in Operations Research* (ed. T. Klastorin), 79–101. INFORMS.

Blazewicz, J., Ecker, K., Schmidt, G., and Welgarz, J. (1993). *Scheduling in Computer and Manufacturing Systems*. Berlin: Springer.

Brucker, P. (1995). *Scheduling Algorithms*. Berlin: Springer.

Coffman, E.G. (1976). *Computer and Job-Shop Scheduling Theory*. Hoboken, NJ: Wiley.

Conway, R.W., Maxwell, W.L., and Miller, L.W. (1967). *Theory of Scheduling*. Reading, MA: Addison-Wesley.

French, S. (1982). *Sequencing and Scheduling*. Chichester: Ellis Horwood, Ltd.

Garey, M.R. and Johnson, D.S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco: Freeman.

Leung, J.-T. (2002). *Handbook of Scheduling*. Boca Raton, FL: Chapman and Hall/CRC.

Morton, T.E. and Pentico, D.W. (1993). *Heuristic Scheduling Systems*. Hoboken, NJ: Wiley.

Muth, J.F. and Thompson, G.L. (1963). *Industrial Scheduling*. Englewood Cliffs, NJ: Prentice Hall.

Pinedo, M. (2016). *Scheduling: Theory, Algorithms, and Systems*, 5e. Upper Saddle River, NJ: Prentice Hall.

T'Kindt, V. and Billaut, J.-C. (2002). *Multicriteria Scheduling: Theory, Models, and Algorithms*. Berlin: Springer.