PRINCIPLES OF SEQUENCING AND SCHEDULING

Kenneth R. Baker
Tuck School of Business
Dartmouth College
Hanover, New Hampshire

Dan Trietsch
College of Engineering
American University of Armenia
Yerevan, Armenia

WILEY

A JOHN WILEY & SONS, INC. PUBLICATION

PRINCIPLES OF SEQUENCING AND SCHEDULING

PRINCIPLES OF SEQUENCING AND SCHEDULING

Kenneth R. Baker
Tuck School of Business
Dartmouth College
Hanover, New Hampshire

Dan Trietsch
College of Engineering
American University of Armenia
Yerevan, Armenia

WILEY

A JOHN WILEY & SONS, INC. PUBLICATION

Copyright © 2009 by John Wiley & Sons, Inc. All rights reserved.

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 750-4470, or on the web at www.copyright.com. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at http://www.wiley.com/go/permissions.

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services or for technical support, please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic formats. For more information about Wiley products, visit our web site at www.wiley.com.

Library of Congress Cataloging-in-Publication Data:

Baker, Kenneth R., 1943 –
Principles of sequencing and scheduling / Kenneth R. Baker, Dan Trietsch.
p. cm.
Includes bibliographical references and index.
ISBN 978-0-470-39165-5 (cloth)
1. Production scheduling. I. Trietsch, Dan. II. Title.
TS157.5.B35 2009
658.5′3–dc22 2008041829

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

CONTENTS

PREFACE

This textbook provides an introduction to the concepts, methods, and results of scheduling theory. It is written for graduate students and advanced undergraduates who are studying scheduling, as well as for practitioners who are interested in the knowledge base on which modern scheduling applications have been built. The coverage assumes no background in scheduling, and for stochastic scheduling topics, we assume only a familiarity with basic probability concepts. Among other things, our first appendix summarizes the important properties of the probability distributions we use.

We view scheduling theory as practical theory, and we have made sure to emphasize the practical aspects of our topic coverage. Thus, we provide algorithms that implement some of the solution concepts we describe, and we cover the use of spreadsheet models to calculate solutions to scheduling problems. Especially when tackling stochastic scheduling problems, we must balance the need for tractability and the need for realism. Thus, we stress heuristics and simulation-based approaches when optimization methods and analytic tools fall short. We also provide many examples in the text along with computational exercises among our end-of-chapter problems.

Coverage of the Text

The material in this book can support a variety of course designs. An introductory-level course covering only deterministic scheduling can draw from Chapters 1–5, 8–10, 12–14, 16, and 17. A one-quarter course that covers both deterministic and stochastic topics can use Chapters 1–11 and possibly 15. Our own experience suggests that the entire book can support a two-quarter sequence, especially with supplementary material we provide on the Internet.

The book contains three appendices. The first reviews the salient properties of well-known probability distributions, as background for our coverage of stochastic models. It also covers some specialized topics on which some of our advanced coverage is based. The second appendix includes background derivations related to the "critical ratio rule," which arises frequently in safe scheduling models. Our third appendix is an introduction to the formulation of sequencing models as integer programs, which represents a long-neglected subject that ought to be revisited in the research literature.

Our coverage is substantial compared to other scheduling textbooks, but it is not encyclopedic. Our goal is to enable the reader to delve into the research literature (or in some cases, the consulting literature) with enough background to appreciate the contributions of state-of-the-art papers.

For the reader who is interested in a more comprehensive link to the research literature than our text covers, we provide a set of Web-based Research Notes. The Research Notes represent unique material that expands the book's coverage and builds an intellectual bridge to the research literature on sequencing and scheduling. In organizing the text, we wanted to proceed from simple to complex and to maintain technological order. As much as possible, each new result is based only on previous coverage. As a secondary guiding principle, the text minimizes any discussion of connections between models, thus keeping the structure simple. Scheduling theory did not develop along these same lines, however, so research-oriented readers may wish to look at the bigger picture without adhering to these principles with the same fidelity. One purpose of our Research Notes is to offer such a picture. Another purpose is to provide some historical background. We also mention open research questions that we believe should be addressed by future research. Occasionally, we provide more depth on topics that are not sufficiently central to justify inclusion in the text itself. Finally, for readers who will be reading research papers directly from the source, we occasionally need to discuss topics that aren't crucial to the text but that arise frequently in the literature.

Historical Background

This book is an updated version of Baker's text, so some historical background is appropriate at the outset. *Introduction to Sequencing and Scheduling* (ISS) was published by John Wiley & Sons in 1973 and became the dominant textbook in scheduling theory. A generation of instructors and graduate students relied on that book as the key source of information for advanced work in sequencing and scheduling. Later books stayed abreast of developments in the field, but as references in journal articles would indicate, most of those books were never treated as fundamental to the study of scheduling.

Sales of ISS slowed by 1980, and Wiley eventually gave up the copyright. Although they found a publishing house interested in buying the title, Baker took back the copyright. For several years, he provided generous photocopying privileges to instructors who were still interested in using the material, even though some of it had become outdated. Finally, in the early 1990s, Baker revised the book. The sequel was

Elements of Sequencing and Scheduling (ESS), self-published in 1992 and expanded in 1995. Less encyclopedic than its predecessor, ESS was rewritten to be readable and accessible to the student while still providing an intellectual springboard to the field of scheduling theory. Without advertising sales reps, and without any association with a textbook publishing house, ESS sold several hundred copies in paperback through 2007. Another generation of advanced undergraduate and graduate students used the book in courses, while other graduate students were simply assigned the book as required reading for independent studies or qualifying exams. Current research articles in scheduling continue to cite ISS and/or ESS as the source of basic knowledge on which today's research is being built.

Perhaps the most important topic not covered in ESS was stochastic scheduling. With the exception of the chapter on the job shop simulations, almost all the coverage in ESS dealt with deterministic models. In the last 15 years, research has focused as much on stochastic models as on deterministic models, and stochastic scheduling has become a significant part of the field. But traditional approaches to stochastic scheduling have their limitations, and new approaches are currently being developed. One important line of work introduces the notion of safe scheduling, an approach pioneered by Trietsch and others, more recently extended in joint work by Baker and Trietsch. This book updates the coverage of ESS and adds coverage of safe scheduling as well as traditional stochastic scheduling. Because the new material comes from active researchers, the book surpasses competing texts in terms of its timeliness. And because the book retains the readability of its earlier versions, it should be the textbook of choice for instructors of scheduling courses. Finally, its title reinforces the experiences of two generations of students and scholars, providing a thread that establishes this volume as the latest update of a classic text.

Acknowledgments

We wish to acknowledge Lilit Mazmanyan of the American University of Armenia for her assistance with many detailed aspects of the book's preparation. We also wish to acknowledge a set of reviewers who provided guidance to our editors as well as anonymous comments and suggestions to us. This set includes Edwin Cheng (The Hong Kong Polytechnic University), Zhi-Long Chen (University of Maryland), Chung-Yee Lee (Hong Kong University of Science and Technology), Michael Magazine (University of Cincinnati), Stephen Powell (Dartmouth College), and Scott Webster (Syracuse University).

Kenneth R. Baker
Dan Trietsch

Hanover, New Hampshire
Yerevan, Armenia

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 the term 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 pm, the bus is scheduled to depart at 8:00 am, 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. Even then, the schedule is useful: by letting passengers know when the bus is due to leave, we help them plan their own schedules. Thus, we may say that the

Principles of Sequencing and Scheduling By Kenneth R. Baker and Dan Trietsch
Copyright © 2009 John Wiley & Sons, Inc.

bus leaves at 8:00 am 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 am. If we make allowances for uncertainty when we schedule cleaning and maintenance, then passengers can trust that the bus will leave at 8:00 am with some confidence. In turn, they may schedule their own time buffer when planning their arrival at the station. Using a time buffer (or *safety time*) helps us cope with uncertainty.

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 quite intangible, until we consider it in some depth. 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 or 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, a washer and a dryer for laundry. Scheduling problems in industry have a similar structure: 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 problem usually arises within a decision-making hierarchy in which scheduling 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 will be needed. In industry, analogous decisions are usually said to be 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 parts, and the volumes to be produced. 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 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. In general, the task duration is uncertain, but we may want to suppress that uncertainty when stating the problem. We should also describe any technological constraints (precedence restrictions) that exist among the tasks. Information about resources and tasks defines a scheduling problem. 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. 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

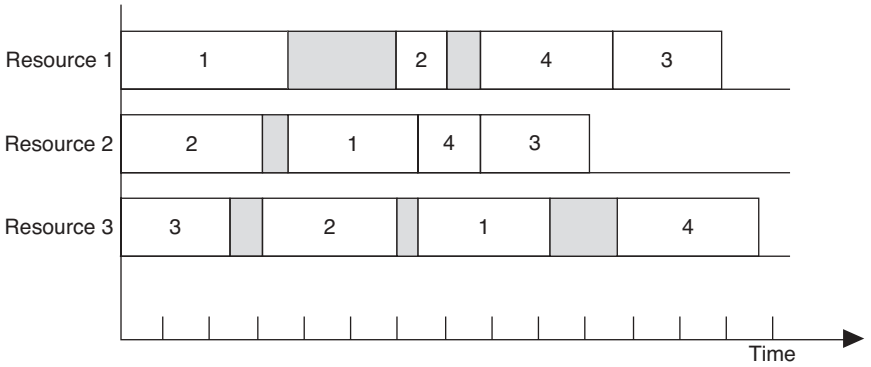

FIGURE 1.1 A Gantt chart.

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 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 encountered 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 with the translation of decision-making goals into an explicit 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 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, 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 proved 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, there are limits on the capacity of machines, and second, there are technological restrictions 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 decisions and sequencing decisions. From the start, the scheduling literature has relied on mathematical models

for these two kinds of decision problems. In more recent developments, referred to as *safe scheduling*, the models recognize service levels as well. Safe scheduling may also involve the decision to accept a job or reject it in the first place, so that when we make commitments to customers, we can be confident that their jobs will finish within the time allowed. An alternative approach to safe scheduling minimizes the expected economic cost of a schedule, including the cost of tardiness and the cost of safety time. Instead of specifying a service level in advance, this approach determines economic service levels as part of the solution.

The need to account for safety time also has important implications for sequencing decisions. As an example of the economic approach to safe scheduling, consider a hub airport that serves several cities (Trietsch, 1993). Instead of providing direct flights for all pairs of cities, incoming flights from each city are directed to the hub, and passengers then take outgoing flights to their destinations. Flights are interrelated because they feed each other with passengers. Ideally, all incoming flights should arrive at about the same time and all outgoing flights should leave at about the same time. In practice, however, sufficient time gaps must be maintained between aircraft when landing or taking off, so both sequencing decisions and timing decisions are necessary. In sequencing, we must account for the fact that different incoming flights have different variances: in general, higher variance implies the need for more safety time, so flights with high variance should be scheduled to arrive earlier. Thus, sequencing decisions may be quite different from those obtained by deterministic models. Furthermore, sequencing in this case is not necessarily about the final order in which aircraft will arrive but about the best plan from which to deviate later, when we correct for various random events, including stochastic departure delays at the originating airports of the incoming flights, emergencies (such as low fuel) forcing the need to expedite some landings in favor of others, and so on. Very tight schedules are likely to lead to higher disruption costs but loose schedules have higher safety time cost. The challenge is to schedule all incoming and outgoing flights so as to minimize the total expected time cost of passengers and equipment plus the disruption cost that occurs if feeding flights are late or if aircraft are forced to wait too long for permission to land.

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 procedures, simulation techniques, 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 techniques. 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 has 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). All these focused on deterministic models, and the few stochastic models they covered did not include safety time. Eventually, additional perspectives were compiled by Morton and Pentico (1993), focusing on heuristic methods, and by Pinedo (2001), addressing stochastic models. Now the field of deterministic scheduling is well developed, and there is a growing literature on stochastic scheduling, but work on *safe* stochastic scheduling is more recent—with few contributions until the last decade or so (Baker and Trietsch, 2007).

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 Baker (2005). But we also have a new objective: to present the emerging theory of safe scheduling 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 17 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 the basic project scheduling model. Chapter 17 discusses the resource-constrained project scheduling model, and Chapter 18 extends safe scheduling considerations to project scheduling.

REFERENCES

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

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

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

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

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

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

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

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

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

Pinedo, M. (2001). *Scheduling: Theory, Algorithms, and Systems*, Prentice Hall, Upper Saddle River, NJ.

Trietsch, D. (1993). Scheduling flights at hub airports, *Transportation Research, Part B (Methodology)* **27B**, 133–150.

2

SINGLE-MACHINE SEQUENCING

2.1 INTRODUCTION

The pure sequencing problem is a specialized scheduling problem in which an ordering of the jobs completely determines a schedule. Moreover, the simplest pure sequencing problem is one in which there is a single resource, or machine, and all processing times are deterministic. As simple as it is, however, the one-machine case is still very important. The single-machine problem illustrates a variety of scheduling topics in a tractable model. It provides a context in which to investigate many different performance measures and several solution techniques. It is therefore a building block in the development of a comprehensive understanding of scheduling concepts. In order to completely understand the behavior of a complex system, it is vital to understand its parts, and quite often the single-machine problem appears as a part of a larger scheduling problem. Sometimes, it may even be possible to solve the imbedded single-machine problem independently and then to incorporate the result into the larger problem. For example, in multiple-operation processes, a bottleneck stage may exist, and the treatment of the bottleneck by itself with single-machine analysis may determine the properties of the entire schedule. At other times, the level at which decisions must be made may dictate that resources should be treated in the aggregate, as if jobs were coming to a single facility.

Principles of Sequencing and Scheduling By Kenneth R. Baker and Dan Trietsch
Copyright © 2009 John Wiley & Sons, Inc.

In addition to the limitation to a single machine, the basic problem is characterized by these conditions:

C1. There are n single-operation jobs simultaneously available for processing (at time zero).
C2. Machines can process at most one job at a time.
C3. Setup times for the jobs are independent of job sequence and are included in processing times.
C4. Job descriptors are deterministic and known in advance.
C5. Machines are continuously available (no breakdowns occur).
C6. Machines are never kept idle while work is waiting.
C7. Once an operation begins, it proceeds without interruption.

Under these conditions, there is a one-to-one correspondence between a sequence of the n jobs and a permutation of the job indices $1, 2, \ldots, n$. The total number of distinct solutions to the basic single-machine problem is therefore $n!$, which is the number of different sequences of n elements. Whenever a schedule can be completely characterized by a permutation of integers, it is called a *permutation schedule*, which is a classification that extends beyond single-machine cases. In describing permutation schedules, it is helpful to use brackets to indicate position in sequence. Thus $[5] = 2$ means that the fifth job in sequence is job 2. Similarly, $d_{[1]}$ refers to the due date of the first job in sequence.

After covering some preliminaries in Section 2.2, we review the elementary sequencing results in Section 2.3 for problems containing no due dates, and in Section 2.4 for problems involving due dates. The chapter is organized to show how differences in the choice of a criterion often lead to differences in the optimal schedule. Later, we examine several general-purpose methodologies that can be applied to single-machine problems.

2.2 PRELIMINARIES

In dealing with job attributes for the single-machine model, it is useful to distinguish between information that is known in advance and information that is generated as the result of scheduling decisions. Information that is known in advance serves as *input* to the scheduling process, and we usually use lowercase letters to denote this type of data. Three basic pieces of information that help to describe jobs in the single-machine case are:

Processing time (p_j)	The amount of processing required by job j
Release date (r_j)	The time at which job j is available for processing
Due date (d_j)	The time at which the processing of job j is due to be completed

Under condition C3 the processing time p_j generally includes both direct processing time and facility setup time. The release date can be thought of as an arrival time—the time when job j appears at the processing facility—and in the basic model, the assumption in condition C1 is that $r_j = 0$ for all jobs. Due dates may not be pertinent in certain problems, but meeting them is a common scheduling concern, and the basic model can shed some light on objectives oriented to due dates.

Information that is generated as a result of scheduling decisions represents *output* from the scheduling function, and we usually use capital letters to denote this type of data. Scheduling decisions determine the most fundamental piece of data to be used in evaluating schedules:

Completion time (C_j) The time at which the processing of job j is finished

Quantitative measures for evaluating schedules are usually functions of job completion times. Two important quantities are:

Flowtime (F_j) The time job j spends in the system: $F_j = C_j - r_j$

Lateness (L_j) The amount of time by which the completion time of job j exceeds its due date: $L_j = C_j - d_j$

These two quantities reflect two kinds of service. Flowtime measures the response of the system to individual demands for service and represents the interval a job waits between its arrival and its departure. (This interval is sometimes called the *turnaround* time.) Lateness measures the conformity of the schedule to a given due date and takes on negative values whenever a job is completed early. Negative lateness represents earlier service than requested; positive lateness represents later service than requested. In many situations, distinct penalties are associated with positive lateness, but no benefits are associated with negative lateness. Therefore, it is often helpful to work with a quantity that measures only positive lateness:

Tardiness (T_j) The lateness of job j if it fails to meet its due date, or zero otherwise: $T_j = \max\{0, L_j\}$

Schedules are generally evaluated by aggregate quantities that involve information about all jobs, resulting in one-dimensional *performance measures*. Measures of schedule performance are usually functions of the set of completion times in a schedule. For example, suppose that n jobs are to be scheduled. Aggregate performance measures that might be defined include the following:

$$\text{Total flowtime:} \quad F = \sum_{j=1}^{n} F_j$$

$$\text{Total tardiness:} \quad T = \sum_{j=1}^{n} T_j$$