International Series in
Operations Research & Management Science

P. Simin Pulat
Subhash C. Sarin
Reha Uzsoy *Editors*

Essays in Production, Project Planning and Scheduling

A Festschrift in Honor of Salah Elmaghraby

Springer

International Series in Operations Research & Management Science

Volume: 200

Series Editor
Frederick S. Hillier
Stanford University, CA, USA

For further volumes:
http://www.springer.com/series/6161

P. Simin Pulat • Subhash C. Sarin • Reha Uzsoy
Editors

Essays in Production, Project Planning and Scheduling

A Festschrift in Honor of Salah Elmaghraby

Springer

Editors
P. Simin Pulat
College of Engineering
The University of Oklahoma
Norman
Oklahoma
USA

Reha Uzsoy
Dept of Industrial & Systems Engineering
North Carolina State University
Raleigh
North Carolina
USA

Subhash C. Sarin
Dept of Industrial & Systems Engineering
Virginia Tech
Blacksburg
Virginia
USA

ISSN 0884-8289
ISBN 978-1-4614-9055-5
DOI 10.1007/978-1-4614-9056-2
Springer New York Dordrecht Heidelberg London

ISSN 2214-7934 (electronic)
ISBN 978-1-4614-9056-2 (eBook)

Library of Congress Control Number: 2013954994

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

This festschrift is devoted to recognize the career of a man who not only witnessed the growth of operations research from its inception, but also contributed significantly to this growth. Dr. Salah E. Elmaghraby received his doctorate degree from Cornell University in 1958, and since then, his scholarly contributions have enriched the fields of production planning and scheduling and project scheduling. This collection of papers is contributed in his honor by his students, colleagues, and acquaintances. It offers a tribute to the inspiration received from his work, and from his guidance and advice over the years, and recognizes the legacy of his many contributions.

Dr. Elmaghraby is a pioneer in the area of project scheduling (in particular, project planning and control through network models, for which he coined the term 'activity networks'). In his initial work in this area, he developed an algebra based on signal flow graphs and semi-Markov processes for analyzing generalized activity networks involving activities with probabilistic durations. This work led to the development of what was later known as the Graphical Evaluation and Review Technique (GERT), and GERT simulation models. He has made fundamental contributions in determining criticality indices for activities, in developing methodologies for project compression and time/cost analysis, and in the use of stochastic and chance-constrained programming and Petri Nets for the analysis of activity networks. These contributions have been brought together in a seminal book in this area entitled, "Activity Networks: Project Planning and Control by Network Models" published by John Wiley, and a monograph on "Some Network Models in Management Science" published by Springer-Verlag. Dr. Elmaghraby also wrote one of the first books on production planning entitled, "The Design of Production Systems."

His fundamental contributions to the economic lot scheduling problem (ELSP) and economic manufacturing quantity (EMQ) analysis are also widely cited. This work presented a novel methodology using a combination of a dynamic programming-based model, integer programming, and a method to circumvent infeasibility. He later extended this work to include learning and forgetting effects, and to the computation of power-of-two policies. Dr. Elmaghraby's extensive work on a wide range of deterministic and stochastic sequencing and scheduling problems, arising in different machine environments, has resulted in many landmark contributions which have advanced this field of study and have strengthened its knowledge

base. It has offered novel ideas and effective methodologies relying on mathematical rigor for the solution of these problems.

Dr. Elmaghraby is one of the rare individuals who have excelled both as a researcher and an administrator. He was appointed as University Professor and Director of the Graduate Program of Operations Research at North Carolina State University in his early 40's, and over the years, he directed that program with aplomb without losing any of his scholarly productivity. That program flourished for all these years under his leadership, providing a world-class education to its students. His superb guidance and leadership by example in bringing quality in everything that he does has been a defining force that has shaped the careers of his students. It is, therefore, not surprising that, among his numerous awards, Dr. Elmaghraby has been recognized with the Frank and Lillian Gilbreth Award, the highest and most esteemed honor bestowed by The Institute of Industrial Engineers on individuals who have distinguished themselves through contributions to the welfare of mankind in the field of industrial engineering.

This volume brings together 14 contributions, which can be viewed under the following three main themes: operations research and its application in production planning, project scheduling, and production scheduling, inspired by, and in many cases based on, Dr. Elmaghraby's work in these areas. The first five chapters are devoted to the first theme, followed by four chapters each devoted to the other two, respectively. An additional chapter is devoted to the vulnerability of multimodal freight systems.

In the first chapter, "Ubiquitous OR in Production Systems", Leon McGinnis puts forth an argument for a paradigm shift in OR education, from the traditional emphasis on teaching of standalone 'artisan' type tools (where each model is developed to address a specific problem), to a reusable platform that enables their broader and deeper penetration in a domain. This argument is made in view of the advent of new computer technologies, and for applications to production systems that are well understood.

In the second chapter entitled "Integrated Production Planning and Pricing Decisions in Congestion-Prone Capacitated Production Systems," Upasani and Uzsoy address a production planning problem when the customer demand is sensitive to delivery lead times. Since the lead times are known to increase nonlinearly with the utilization of capacitated resources, a large reduction in price may increase demand to the extent that it can no longer be satisfied in a timely manner by available capacity, thereby negatively impacting customer satisfaction and future sales. They present an integrated model for dynamic pricing and production planning for a single product under workload-dependent lead times, and study interactions among pricing, sales, and lead times. Their investigation reveals a different behavior of the integrated model from a conventional model that ignores the congestive effect on resources because of price variations.

A "Refined EM Method for Solving Linearly Constrained Optimization Problems" is presented by Yu and Fang in the third chapter. They extend the original Electromagnetism-like Mechanism (EM) that has been widely used for solving global

optimization problems with box-constrained variables to solving optimization problems with linear constraints, and call it a 'Refined EM Method.' The EM method is a stochastic search method that uses a functional evaluation at each step, and does not require any special information or structure about the objective function. The proposed method explicitly considers linear constraints in an efficient manner to direct sample points to attractive regions of the feasible domain. Results of a computational investigation are also presented that show the proposed method to outperform known methods and to converge rapidly to global optimal solutions.

In "The Price of Anarchy for a Network of Queues in Heavy Traffic," Shaler Stidham investigates the price of anarchy in a congestive network of facilities in which the cost functions at the facilities follow the characteristics of the waiting-time function for a queue with infinite waiting room. Similar to a network of parallel *M/M/1* queues, Stidham develops an analytical expression for the price of anarchy for the *GI/GI/1* network.

In the fifth chapter entitled, "A Comparative Study of Procedures for the Multinomial Selection Problem," Tollefson, Goldsman, Kleywegt, and Tovey address the multinomial selection problem originally formulated by Bechhofer, Elmaghraby, and Morse (1959), that of determining the number of trials needed to select the best among a given number of alternatives. The aim is to minimize the expected number of trials required while exceeding a lower bound on the probability of making the correct selection. The authors present a comparative study on the performances of various methods that have been proposed for this problem over the years.

The sixth chapter is entitled, "Vulnerability of Multimodal Freight Systems." In this chapter, Aydin and Pulat explore the vulnerability of multimodal freight transportation infrastructure in the face of extreme disruptive events. The freight transportation system constitutes a backbone of global economy. This study, motivated by recent hurricane-related events encountered in the USA, examines the concepts of vulnerability, reliability, resilience, and risk, and the relationship among them, for the freight transportation infrastructure, and provides valuable insights on how vulnerable and resilient the transportation infrastructure is to extreme disruptive events.

The following two chapters address stochastic project scheduling problems. In, "Scheduling and Financial Planning in Stochastic Activity Networks," Dodin and Elimam analyze the impact of stochastic variations in the renewable and nonrenewable resources required by each activity of the project, on project cost and duration. An analytical approach is used to determine the probability density functions of the project cost and duration. A linear programming model is used to distribute the resulting project budget over its activities and to minimize the project duration. Willy Herroelen presents "A Risk Integrated Methodology for Project Planning Under Uncertainty" in the eight chapter. A two-phase methodology is presented in the face of the risk of resource breakdown and variability of activity durations. In the first phase, the number of regular renewable resources to be allocated to the project is determined, and in phase two, first a resource-feasible proactive schedule is constructed, after which resource and time buffers are inserted to protect it against disruptions.

The schedule is then tested by simulating stochastic disruptions and by appropriately repairing it if it becomes infeasible. This approach provides an implementable schedule along with a workable reactive schedule procedure that can be invoked in case it becomes infeasible despite the protection built in it.

In the ninth chapter, entitled, "Dynamic Resource Constrained Multi-Project Scheduling Problem with Earliness/Tardiness Costs," Pamay, Bulbul, and Ulusoy address the problem of scheduling a new arriving project against a set of known renewable resources when a number of projects are already in process. The due dates and earliness/tardiness penalties of the activities of the existing project are known while the due date of the new project is to be determined, which is accounted for by assigning a penalty cost per unit time the new project spends in the system. A heuristic method is proposed to solve large-sized problems, and its efficacy is demonstrated.

"A Multi-Mode Resource-Constrained Project Scheduling Problem Including Multi-Skill Labor" is discussed by Santos and Tereso in the tenth chapter. Each activity of the project may require only one unit of a resource type, which can be utilized at any of its specified levels (called modes) that dictates its operating cost and duration. The processing time of an activity is given by the maximum of the durations that result from the different resources allocated to that activity. The objective is to determine the operating mode of a resource for each activity so as to minimize the total cost incurred, given a due date as well as a bonus for earliness and penalty cost for tardiness. A filtered beam method is proposed for the solution of this problem, and results of its performance are presented.

The last four chapters address production scheduling problems. Allaoui and Artiba consider "Hybrid Flow Shop Scheduling with Availability Constraints" in the eleventh chapter. They assume that a machine is not continuously available, and instead, is subjected to at most one preventive maintenance in a specified time window. The jobs are non-resumable, and the objective is to minimize the makespan. For a special case of this problem, with one machine at each stage (the traditional two-machine flow shop problem), a dynamic programming-based method is presented to determine an optimal schedule, while for the hybrid flow shop with one machine at the first stage and m machines at the second stage, a branch-and-bound procedure is proposed that exploits an effective lower bound.

In the twelfth chapter entitled, "A Probabilistic Characterization of Allocation Performance in a Worker-Constrained Job Shop," Lobo, Thoney, Hodgson, King, and Wilson address a job shop scheduling problem in the presence of dual resource constraints pertaining to limited availabilities of both machines and workers. The objective is to minimize maximum lateness. For a given allocation of workers to the machines, they estimate a distribution of the difference between the maximum lateness achievable and a lower bound on maximum lateness. Both heuristic methods for worker allocation and schedule generation as well as a lower bound on maximum lateness that are used for this investigation are presented in an earlier paper.

McFadden and Yano address a problem on "A Mine Planning Above and Below Ground: Generating a Set of Pareto-Optimal Schedules Considering Risk and Return" in chapter thirteen. They assume the availability of different methods for

mining minerals with each method leading to a different profit and risk. They employ a methodology based on a longest-path network framework to determine mining plans that give the k best values of expected profit, and integrate it with various measures of risk to construct a set of Pareto-optimal solutions. The various measures of risk considered include variance, probability of achieving a specified profit target, and conditional value-at-risk. The methodology is illustrated using a simple example with conditional value-at-risk as the risk measure.

In chapter fourteen entitled, "Multiple-Lot Lot Streaming in a Two- stage Assembly System," Yao and Sarin apply lot streaming to a two-stage assembly shop in which the first stage consists of m parallel machines and the second stage consists of one assembly machine. Each lot consists of items of a unique product type. A lot-attached set up time is incurred at the machines at both the stages. For a given number of sublots of each lot, the problem is to determine sublot sizes and the sequence in which to process the lots at both the stages so as to minimize the makespan. Although the problem of scheduling in such a machine environment has been addressed in the literature, the application of lot streaming to this problem is new. Some structural properties for the problem are presented, and a branch-and-bound-based method is applied for its solution. The efficacy of this method is also demonstrated through computational investigation.

We hope that the contributions in this volume serve to extend the body of knowledge in the wide range of research areas to which Professor Elmaghraby has contributed, which we believe is the most appropriate recognition for an outstanding scholar and administrator. The fields of Industrial Engineering and Operations Research will remain deeply in his debt for many years to come.

Contents

Biography

Salah E. Elmaghraby earned a Bachelor's degree in Mechanical Engineering from Cairo University in 1948, a Master of Science degree in Industrial Engineering from Ohio State University in 1955 and a PhD from Cornell University in 1958. He is University Professor Emeritus at the Edward P. Fitts Department of Industrial and Systems Engineering at North Carolina State University, where he has been a professor of Operations Research and Industrial Engineering since 1967. He established the interdisciplinary Graduate Program in Operations Research and was its Director from 1970 to 1989. Previously, he was Associate Professor at Yale University; Research Leader at the Western Electric Engineering Research Center in Princeton, NJ; and Visiting Professor at Cornell University, the Katholieke Universiteit Leuven (Belgium) and the FUCAM (Belgium), the Claude Bernard Université Lyon I (France), and the Nagoya Institute of Technology (Japan). He has 12 years of industrial experience, including eight abroad in Egypt, Kuwait (where he was Principal Scientist and Project Leader for 2 years) and Europe (the U.K., Belgium and Hungary where he was Inspecting Engineer for the Egyptian Railways for 5 years). He has served as reviewer for many US and European journals; was Regional Editor (the Americas) for the International Journal of Production Economics and was the founder and editor-in-chief of the Journal of Operations and Logistics, 2004–2011.

Professor Elmaghraby is a recipient of numerous awards and honors, including the Frank and Lillian Gilbreth Industrial Engineering Award (IIE, 2003), the Alexander Quarles Holladay Medal for Excellence (NCSU, 2000), the Kuwait Foundation for the Advancement of Science Distinguished Award (1990), the R. J. Reynolds

Distinguished Award in Research and Education (College of Engineering, NCSU, 1987), the Operations Research Division Award (IIE, 1980), and the David F. Baker Distinguished Research Award (IIE, 1970). He obtained an Honorary Doctorate from the Université Claude Bernard Lyon I (France, 1998). He was elected Fellow of the Institute of Industrial Engineers in 1986 and Fellow of the Institute for Operations Research and Management Sciences (INFORMS) in 2004.

Professor Elmaghraby has written four books, among them the seminal production management text "The design of production systems" (Reinhold 1966) and the pioneering activity networks textbook "Activity networks" (Wiley 1977). He edited/co-edited three books, contributed chapters in nine books, and authored/co-authored over 118 scientific papers.

He initiated the research in generalized activity networks by developing an algebra for the analysis of networks in which activities may be undertaken probabilistically. By providing the theoretical foundations, he paved the way for what later became the GERT model (Graphical Evaluation and Review Technique) and the special purpose GERTS simulation models.

Professor Elmaghraby developed numerous deterministic and stochastic algorithms for scheduling and sequencing problems involving single and parallel machines, flow jobs, and job shops. Most noteworthy and of fundamental impact, however, is his work in the domain of activity networks. He pioneered in the analysis of probabilistic and generalized activity networks, the analysis of activity networks under generalized precedence relations, network representation problems and methodologies for criticality and sensitivity analysis. He made fundamental contributions in the use of stochastic and chance-constrained programming and Petri nets, and published seminal papers on project compression and time/cost trade-off analysis, project bidding, project risk management, complexity issues and test set generation.

Over the years, Professor Elmaghraby has supervised over 60 doctoral and master's students in the USA and abroad, and inspired an extensive population of researchers over the world. At the age of 84, he still continues his research in the field of project planning and control.

Chapter 1
Introduction: For Daddy

Wedad J. Elmaghraby and Karima N. Radwan

It is hard to write a brief introduction for a man whom you have viewed most of your life as "part-God". It is a bit awkward to step back and try to describe him to others. This is our attempt to do so—to express our love and respect for, quite simply, the most beautiful man we know, and one we were so fortunate enough to have as our father.

Since our father's academic history is clear, we would like to share with you a little bit about his life *before* operations research (OR) entered into his life, and then conclude with a few stories about him that, we believe, clearly illustrate the true scholar and gentleman he is.

Before Operations Research Our father was born in 1927 in Egypt—he was the second son out of four children. He lived his early life in Alexandria, briefly fleeing to Rosetta in World War II (WWII) to escape from Rommel and his army (always the engineer, even as a child, he built himself a radio with crystals to hear all the news of the day in WWII). From the stories we heard growing up—it was clear that our father always had an inquisitive mind and a strong aptitude for studies. When he finished elementary school, he ranked first in his national exams. One of his best friends was the son of a Basha (a high ranking military officer) in Egypt and he, unfortunately, failed his exams. When his friend retook the exams, he managed to pass the second time around. Proud of his son's success, the Basha went out and bought his son a shiny new bike. Our father was excited by this development and shared this with his own father. He told his father that, since he not only passed his exam, but came out *first* amongst his peers, he should not only receive a new bike, but one with all the bells and whistles that were available on the market. His father, who was a high school teacher, told him that he was proud of his son for doing well, but he was not going to buy him anything. The reward is learning and achieving something, and *that* is something that stays with you forever.

W. J. Elmaghraby (✉)
Robert H. Smith School of Business, University of Maryland, College Park, USA
e-mail: welmaghr@rhsmith.umd.edu

K. N. Radwan
Northern Virginia Community College, Annandale, USA
e-mail: kelmaghr@hotmail.com

P. S. Pulat et al. (eds.), *Essays in Production, Project Planning and Scheduling,*
International Series in Operations Research & Management Science 200,
DOI 10.1007/978-1-4614-9056-2_1,

Our father graduated from high school at the young age of 15 and went to study Mechanical Engineering at the prestigious Cairo University. His first job upon graduation (at the age of 20) was with the Coca-Cola bottling plant in Cairo. His job was to help oversee production at the plant. It was an enviable position as an engineer, and gave him a place of rank within the hierarchical Egyptian society. One day he was advised by some of the other engineers to eat his lunch in his private office, and not in full view of the factory workers. They feared that eating in front of the manual workers would make them jealous and would then bring the evil eye upon him. Always a man of science, our father listened to their advice and then promptly moved his desk to the center of the factory floor to dispel any myths about evil eyes.

Although the job at Coca-Cola was prestigious and paid very well, after a short time, our father did not feel that he was being sufficiently challenged. He applied for and was awarded a position working for the Egyptian Railroads authority in 1949. They posted him in the UK to serve as a quality control inspector. At the time, Egypt was purchasing locomotives from abroad and would send engineers to the respective producing countries to inspect the production processes. Our father recalls that he was sent there with a few other engineers who were the "sons of important men". While the other young men, excited by their new found freedom away from home, enjoyed their days in England in ways we might imagine young men would, our father spent his days in factory floors, taking notes of absolutely everything and sending back reports to Egypt. His supervisor was surprised by our father's diligence and asked why he did not "relax" and enjoy his posting abroad. Our father's response was that he *was* enjoying himself—learning about locomotives, their design and all of the science that went into their production! His reports back home continued in a steady manner, and more than once he stopped a shipment of parts back to Egypt because he did not feel that the work was done well.

When we ask our father about his time there, he says that it was interesting, but that he never felt happy in the grey, smoggy weather of England. His supervisor took pity on him and heeded his request for a sunnier climate. He was transferred to Hungary in 1952. While in Hungary, he saw the effects of the communist revolution in that country. He attended some of the most beautiful operas and symphonies for prices next to nothing, but he also saw the demise of the social elite. His doorman was a Count who had only an elementary school education and therefore was not qualified to do anything other than the most menial of tasks. While the uneducated social elite was thrown down the economic ladder, he saw that doctors, engineers, and scientists, who had been well-educated before the revolution, still continued in their professions. He says that it was then that he truly understood—your mind is your most valuable asset, and no one can *ever* take away your education.

While his family preferred for him to return to Egypt, our father's quest for learning drew him to the USA. While working for the Egyptian Railroad Authority, he had managed to save enough money for a voyage to the USA and *one* year of study. Not deterred, he went to Ohio State where he managed to complete both his course work and write a Masters thesis in one year. Finally, he was accepted into the PhD program at Cornell University's Mechanical Engineering department. The Operations Research and Industrial Engineering (ORIE) department did not exist at the time!

After Operations Research Our father's love of learning and striving for excellence is palpable and infectious. Possibly, that explains his jump from an Assistant Prof. at Yale straight to being a University Prof. at NCSU. But more than academic titles, we believe that it is his commitment to his students and colleagues and the "institution" of learning that distinguishes him. When we he was brought to NCSU, he was charged with building an Operations Research department. Part of this is building the infrastructure—the class lists, the faculty roster, the departmental policies, etc. But more than this, what our father did was build a *community*. We remember having to attend the OR picnics every spring and fall at one of the local parks in Raleigh, where faculty and students would barbecue and play volleyball together. Then there would be the dinners that my mother would host for all of the PhD students once each semester. The students would confess that they would not eat all day for they knew (or had been told) what feasts awaited them in the Elmaghraby household! Finally, there was the steady stream of seminar speakers who were picked up by our father from the airport and brought to our home to join us for dinner. At the time, we did not know that this was unusual—going "above and beyond" the call of duty. For us, this was the reality of life—building and sustaining the OR department was part a huge part of our father's life, and hence a part of ours.

Over the years, the networks of students and colleagues our father has built continues strong. Meetings with new PhD students still punctuate his days, occurring at cafes, in the office, and even at our parents' home, when a research problem just could not wait until the doctor's ordered "2 days of rest" were over. With the "old" PhD students (now themselves established Associate and Full Professors), he still searches out opportunities to go visit them for several weeks at a time, wherever they may be—China, Taiwan, Belgium, France, Morocco, etc. To put this into context, keep in mind that our father is now 84, and his last secondment to China was last year. While we are sometimes annoyed that his commitment to his students takes both him and my mother away from us sometimes for an entire semester (for certain, our mom would not stay in Raleigh while our dad travels the world—they must go together!). We understand that he cannot stop, for he loves what he does.

While it is true that the OR department was socially a large part of our lives, we were lucky enough that our father left most of his talk about "work" in the office. While we were never given lectures about Activity Networks or Dynamic Optimization, we knew that if we asked for some help with our math, our father was probably going to start by describing the origins of the number zero, or the beauty of π. No topic was safe from our father's love of math. Once when Karima asked what the best age was for getting married, our father replied that it was a nonlinear function. While we laugh about these stories now, we (and many of his students) know that we were fortunate enough to have been touched by his view of life and learning. This desire to learn what is new is what prompted him to buy us a Commodore computer back in 1982 and encourage my sister and I to learn how to use it. When we asked why, he would reply, "Because, this is the way of the future. If you do not learn it, you will be left behind." He would always encourage us, and everyone around him, to look forward with an open and inquisitive mind.

We would not want to conclude and have anyone think that our father's love of learning was unidimensional, directed only toward math and engineering. While it is true that Wedad went into IEOR (it is the truth when we say that this was not because of deep discussion over the topic with our dad; actually, Wedad never much listened to what our parents had to say and specifically avoided talking about anything serious like school), Karima decided to pursue cooking and the Classics. When Karima entered into the University of North Carolina at Chapel Hill and declared that she wanted to be a Classics major, the Egyptian community in Raleigh was perplexed. "Why is she doing this? She is a smart girl." they would ask of our parents. My father's response was always the same—"This is what she loves." When Karima decided that she wanted to go to cooking school in France, and the snide remarks surfaced—"Why send her to France - my wife can teach her how to cook and it won't cost you anything." Our dad would smile and say "This is what she wants to do. She is going to study with the best". It is that kind of open-mindedness and appreciation of all subjects and jobs that makes him a true scholar and a wonderful father.

We would like to conclude with a few favorite sayings of our father:

> There are no dull subjects—only dull people. Education—it is the one thing they can never take away from you. I need to go study for my next exam. Don't be a jack of all trades and a master of none. Do what you love and never work a day in your life.

A final note from Wedad I was fortunate enough to go to Cornell for my undergraduate education in ORIE, being taught by some of my father's former professors and colleagues, and earn my PhD in IEOR (University of California, Berkeley). From the very beginning, I would occasionally be approached and asked "Are you related to *the* Salah Elmaghraby?" During the first 10 years or so, not knowing much about my father and the magnitude of his contribution to OR, I would say "Yes - I'm his daughter" and then be surprised when the person would gush out many accolades about my father, want to shake my hand, etc. While personally I thought that my dad was special because he was *my* dad, I did not quite understand why anyone else would be excited about knowing him or having met his daughter. It has been a couple of decades since this started to happen, and I now know how very unique my father is and why all the fuss. Simply put, my father sincerely loves to connect with other scholars, is excited by new ideas from a variety of fields, shares his own selflessly with others, and works tirelessly to accomplish the next goal, whether that be helping a student find a job, working on a paper, submitting a new grant (yes, he still submits grants!), writing a book, studying for an exam in a new class he is auditing (he was still auditing statistics and math classes as a Full Prof.), or hosting an unknown colleague from abroad coming to visit him merely because the person asked of him to do so. He gives of himself to others, and because this is rare, it is noted and appreciated.

For some unexplained reason having to do with the gravitational pull of our offices, I often find that people do not make an effort to attend a seminar in another department, let alone another university. It was not so with my dad. I can recall that when I was visiting Duke, the junior faculty there commented to me that they were surprised to see my father at some of their seminars. They should not have been. If

you know him, you know that a drive of 30 min is something he is happy to do in order to learn what is new. I try to take this lesson to heart and make the effort to do the same. He has set a very *very* tough act for me to try follow. I console myself with the fact that there are few "Salah Elmaghraby" in this world—and I am just lucky enough to have had him as a role model.

Chapter 2
Ubiquitous Operations Research in Production Systems

Leon F. McGinnis

Introduction

The contemporary education of an operations research (OR) professional is structured around an artisanal model of OR practice. We teach the artistic techniques of the discipline, i.e., the "fundamental methods" of mathematics and mathematical applications, computational methods and tools, and "genres" of application domains, such as production, logistics, or health care delivery. We teach the creative part of the art of OR, i.e., "modeling"—if at all—as a "studio" course; we demonstrate for the budding OR artisan what it means "to model," pose them challenges and critique their work, in the hope that they will acquire that essential esthetic appreciation that characterizes the master OR artisan. The paradigm we teach is the hand-crafted, purpose-built model of a specific problem. We send our graduates out into the world to work as OR professionals have worked for the past 70 years, albeit with an ever-growing and improving technical toolkit. In practice, our graduates are sometimes fortunate enough to work in teams with both domain experts and IT experts to build large scale persistent OR models. These kinds of models are intended to be used routinely over time, and must accommodate changing instance data. In contemporary practice, OR professionals have access to very powerful analysis modeling tools, to IT tools that can harvest data and conform it to our models, to solvers that benefit from 40 years of algorithmic and computational research, and to computing platforms that accommodate gigabyte databases and teraflop computations.

Over the past three decades, this marriage of OR and IT has enabled our profession to accomplish some amazing feats in logistics, finance, medical decision making, and in almost all walks of modern life. One could argue, however, that the penetration of OR in production systems decision making is a fraction of what it could and should be, based on the proven results. Successful applications are not replicated nearly as often as they could be, in large part because of the time and cost for replicating them.

L. F. McGinnis (✉)
H. Milton Stewart School of Industrial and Systems Engineering,
The Georgia Institute of Technology, Atlanta, GA30332-0205 USA
e-mail: leon.mcginnis@gatech.edu

P. S. Pulat et al. (eds.), *Essays in Production, Project Planning and Scheduling,*
International Series in Operations Research & Management Science 200,
DOI 10.1007/978-1-4614-9056-2_2,

There is an emerging need, and a burgeoning opportunity, to "industrialize" OR in production systems. To industrialize OR in production decision making would make a broad range of "standard" OR applications available to the masses of decision makers whose decisions could be significantly improved through more and better OR analysis—much faster and cheaper than is possible today with the conventional approach to model development. The rapid growth of "business analytics" could be viewed as one manifestation of this need and opportunity (see, e.g., Kiron, Schockly et al. 2011) for a recent survey). One contemporary emphasis in business analytics can be viewed as the "industrialization" of statistical methods and tools to enable managers to understand and exploit transactional data without the direct involvement of statistics or IT experts. There is a similar opportunity to industrialize OR methods and tools to enable better decision making for production systems design, planning, and control.

The purpose of this chapter is to explore this concept, and in particular, to argue that methods and tools from computing and software engineering could be used to make OR applications ubiquitous in production systems. Such a transformation would have profound impacts on both the decision makers, who would gain access to these OR tools and methods, and the operations researchers, who develop, implement, and maintain production system decision support systems.

The chapter starts with perhaps the simplest possible example of an OR application in production in order to begin to frame the issues, of which knowledge capture and knowledge management are paramount. This section suggests that there are multiple categories of models that are important for OR applications in production systems. Next comes a very high level introduction to the basic concepts of "model-driven architecture (MDA)," an approach to software engineering that may not be widely familiar to the OR community. The following two sections describe how MDA concepts can be used to capture important knowledge, i.e., models, and to automate the transformation of models of one kind into models of another kind. The implications of these capabilities are explored briefly, two fundamental intellectual challenges are identified, and the chapter closes with some concluding thoughts.

No doubt, there are those in the OR community who will question the wisdom of providing powerful OR analyses to non-OR experts. That question is not the focus of this chapter and, in any event, will be answered by the non-OR experts who will decide for themselves whether or not access to powerful OR analyses will be valuable to them. Rather, the focus here is on the technologies already available to enable the industrialization of OR for particular domains of application.

OR and Production Knowledge

The native tongue of OR is mathematics. At any OR conference, in any session, on any topic, the focus of attention is almost invariably on the mathematical formulation of "the problem" and on the subsequent (mathematical or computational) analysis of

that formulation. A corollary to this phenomenon is that, almost invariably, the original problem stakeholders—those who must make actual decisions about designing or operating the system being modeled—do not speak mathematics with sufficient fluency to truly understand what is being presented. The stakeholders have their own language which is specific to the domain of the problem—a semantic model of the domain that allows them to organize information about what they observe, and communicate efficiently among themselves regarding the problems in their domain.

As an illustration, consider one of the most basic OR modeling examples. In the terms of the stakeholder, the problem is described as follows. A firm has warehouses in 10 cities, each containing a known inventory of a popular product. The firm has orders from 50 customers, scattered around the country, and must decide how to allocate the available inventories to the customer orders in hand. A reasonable way to make the allocation is to seek the largest net profit, considering the price to be charged to each customer, the cost to deliver the product to the customer, and the cost of the product in the warehouse.

The OR instructor, presenting this problem in an introductory course, will draw a network (perhaps even pointing out that it is a directed bipartite graph) to illustrate the connections between warehouses and customers. Then, perhaps implicitly, the instructor will make some associations, which often is referred to as "representing the problem mathematically":

Warehouse index, $i = 1, \ldots, 10$
Cost per unit in the warehouse, $c_i, i = 1, \ldots, 10$
Supply at the warehouse, $s_i, I, \ldots, 10$
Customer index, $j = 1, \ldots, 50$
Customer demand, $d_j, j = 1, \ldots, 50$
Price to customer, $p_j, j = 1, \ldots, 50$
Transport cost per unit between warehouse and customer, $t_{ij}, i = 1, \ldots, 10$, $j = 1, \ldots, 50$
Shipment from warehouse i to customer j, $x_{ij}, i = 1, \ldots, 10$, $j = 1, \ldots, 50$

Finally, the instructor will write out "the problem" using the usual linear programming (LP) formulation of the classical transportation problem as shown in Fig. 2.1. From this point forward, the discussion will be focused on this formulation, this mathematical statement of an analysis which is intended to indicate what the best decisions would be, i.e., the optimal values of the flow variables.

Once students are comfortable with the mathematical formulation, the discussion will then turn to how to actually solve the problem. At this point, students are introduced to a modeling language, which will allow them to prepare the input necessary for some open source or commercial solver. For example, AMPL ("A Mathematical Programming Language," http://www.ampl.com/) might be used to create a computational model of the form shown in Fig. 2.2.

Typically, the decision maker will not directly comprehend the models illustrated in either Fig. 2.1 or 2.2, although in this simple case, the OR analyst can make a direct translation to the domain semantics. The decision variables correspond to allocations, the constraints correspond to conservation relationships, etc. In more complex scenarios, such a translation may not be so easy.

Fig. 2.1 Transportation problem formulation

$$\max P = \sum_{i=1}^{10}\sum_{j=1}^{50}\left[p_j - c_i - t_{ij}\right]x_{ij}$$

$$s.t. \sum_{j=1}^{50} x_{ij} \leq s_i \; \forall \, i$$

$$\sum_{i=1}^{10} x_{ij} \geq d_j \; \forall \, j$$

$$x_{ij} \geq 0 \; \forall \, i, j$$

This simple example illustrates a fundamental aspect of OR-based decision support, namely that there are three important, related but distinct kinds of knowledge involved. The first is *domain knowledge,* which is common to the stakeholders in the domain (though sometimes tacit rather than explicit) and which has its own semantics (warehouse, customer, product, shipment, etc). The second is *analysis knowledge,* or knowledge of a particular analysis, which could be used to support a particular decision in the domain (the LP formulation of the transportation problem) which has its own (mathematical) semantics and syntax, along with, perhaps, knowledge of a particular computational modeling language, and even a particular solver. The third is the *modeling knowledge* that enables the translation of a problem from its domain semantics into the semantics and syntax of a particular OR analysis, considering the limitations of the analytic model. Each category of knowledge is essential for a successful OR decision support project, and each presents its own challenges for knowledge capture and reuse.

Domain knowledge is rarely formalized; in fact it is a common problem to find that different companies in the same industry will use different terms for the same concept, or the same term for different concepts. The standards that have been developed tend to be either very generic and high level (like the supply chain operations reference (SCOR) model for supply chains (Huan et al., 2004)) or focused on information technology (like Business Process Model and Notation (BPMN, http://www.bpmn.org/) or ISA-95 (http://www.isa-95.com/). There have been some research publications on the use of ontologies, e.g., in material handling (Libert and ten Hompel 2011), manufacturing (Jiang et al., 2010), production (Chungoora et al., 2011), but to date, there is not a commonly used, agreed-upon production system ontology. Thus, domain knowledge in production systems remains largely ad hoc, making it difficult to reuse, to teach, or to learn.

This stands in sharp contrast to analysis knowledge, which ultimately is expressed in very precise and canonical mathematical forms and in analysis-specific modeling. This knowledge typically is gained through the student's exposure to the canonical mathematical formulations and particular modeling languages, and by their creating formulations and using the modeling languages for homework and projects;

```
set SOURCE;   # sources
set DEST;   # destinations

param supply {SOURCE} >= 0;   # amounts available at sources
param demand {DEST} >= 0;   # amounts required at destinations

check: sum {i in SOURCE} supply[i] = sum {j in DEST} demand[j];

param cost {SOURCE,DEST} >= 0;   # shipment costs per unit
var Trans {SOURCE,DEST} >= 0;    # units to be shipped

minimizetotal_cost:
sum {i in SOURCE, j in DEST} cost[i,j] * Trans[i,j];

subject to Supply {i in SOURCE}:
sum {j in DEST} Trans[i,j] = supply[i];

subject to Demand {j in DEST}:
sum {i in SOURCE} Trans[i,j] = demand[j];
```

Fig. 2.2 AMPL model for transportation problem formulation

it is refined and deepened through practice in application. Analysis methods are largely mathematical and thus, by their nature, somewhat formalized. The corresponding modeling languages make it relatively easy to create, archive, teach, and learn particular modeling applications and "tricks."

This difference between domain knowledge and analysis knowledge leads to what might be called a "semantic gap" that is a key issue in the practice of OR in production systems. The OR models and OR methods invariably rely on the semantics of mathematics and particular mathematical methods and may be influenced by the analysis modeling language and even the solver to be used, while the stakeholders invariably rely on the semantics of their domain and frequently find themselves incapable of directly evaluating the fidelity between the model developed by the OR analyst and the domain problem as they understand it.

Thus, the contemporary practice of OR in production systems requires the OR analyst or team to bridge this gap by using, and often creating, "modeling knowledge" to translate between the (natural) language of the stakeholders and the (formal) language of OR. The translation from "problem" to "formulation" tends to require significant investment of time for both analysts and stakeholders, is subject to interpretation errors, and is usually static, i.e., the resulting models may not accommodate changes in the modeled system. The translation from analytic results back to the stakeholder decision space also is largely the responsibility of the analysts, and likewise may be subject to interpretation errors. The test of analysis model fidelity often is simply

whether or not the analysis results "make sense" when viewed in light of the prior experience of the domain stakeholders.

It is safe to say that this modeling knowledge is the least codified of the three kinds of knowledge needed for OR-based decision support of production systems decision making. In fact, OR faculty have struggled, almost from the emergence of OR as a discipline, to discover an effective way for students to learn "how to model," which almost always means "how to extract a mathematical model of a process or decision from a somewhat ambiguous domain-specific problem description."

In the simple transportation problem illustration given above, the semantic gap is small and, one would hope, presents no great challenge to either the domain stakeholder or the OR analyst. Likewise, the modeling process itself seems straightforward, once illustrated. In more complex scenarios, the semantic gap becomes a larger problem, as does the challenge of modeling. For example, the creation of large scale optimization or simulation models to support the design and management of global logistics systems involves translating relatively arcane considerations, such as local content requirements, or export/import duties into precise mathematical relationships. Similarly, the development of large scale optimization models to design radiation therapies also involves translating what may be known with some ambiguity about the effects of radiation into a precise mathematical structure.

One contemporary approach to bridging the semantic gap is to create "parametric" analysis models which can accommodate any instance data conforming to the parametric definitions. For our simple example, this would give the decision maker the ability to specify the warehouses and customers, perhaps extracting the supplies, demands, and transport costs from appropriate data sources. This is an important step toward ubiquitous OR, but it obscures rather than resolves the semantic gap. Bridging the semantic gap still requires tacit knowledge that is not captured in a form that is transferable, reusable, teachable, and deployable. Moreover, the domain knowledge is encoded in the specification of the parametric data for the optimization formulation. In this form, the specification of the domain knowledge will be of limited value in supporting other relevant decision support models, such as simulation or risk analysis.

Effectively managing and exploiting these three kinds of knowledge—domain, analysis, and modeling—is the key to achieving a broader and deeper penetration of OR in production system decision making. This knowledge management problem has two fundamental challenges:

- How to capture each kind of knowledge in a form that is transferable, reusable, teachable, and deployable
- How to make the three kinds of knowledge interoperable, i.e., how to use modeling knowledge to support the transformation from instances of domain models—created using domain knowledge—to instances of analysis models in an appropriate computational form

In this regard, there is much to be learned from the experience of the software engineering community about knowledge representation and model transformation.

Model-driven Architecture

Until recently, software development has also been a largely artisanal activity. The image of the "hacker" is iconic in modern society—the idiosyncratic individual who can understand the nature of the needed computation, and craft an elegant code to make it possible. The limitation of the hacker model is the realized mismatch between the supply of hackers and the demands for software in modern society. The response of the software engineering community has been to "industrialize" the production of well-understood software applications (see, e.g., the evolution of BPMN (White 2006)). This industrialization is being accomplished by an evolving suite of theories, tools, and methods that permit individuals with less than "true hacker" credentials to create satisfactory implementations of the needed software. The essential nature of these tools and methods is that they capture both domain and software engineering knowledge in a form that is transferable, reusable, teachable, and deployable. The resulting "industrialization" of the artisanal software process is aptly captured in the term "software factories" (anonymous 2012a).

This movement in software engineering has been called "model-driven architecture" (MDA) (http://www.omg.org/mda/) or "model-driven engineering" (see, e.g., Meyers and Vangheluwe 2011). The fundamental enablers of MDA are formal modeling languages and model transformation theories and tools. The Unified Modeling Language (UML) (http://www.uml.org/) has evolved over the past 20 years to dominate modeling in the software engineering process. Emerging tools like the Object Management Group's (OMG) Query/View/Transformation (QVT) standard (http://en.wikipedia.org/wiki/QVT) enable the computational transformation of a model created with one language (syntax and semantics) to a model expressed in a different language. For example, the source model could be a UML-based description of a business process, and the target model could be the Java code necessary to provide the computational implementation of the business process.

Within systems engineering there is a growing community of researchers and practitioners who are adapting the tools and methods of MDA to systems engineering, calling it "model- based systems engineering" or MBSE (Ramos, et al., 2011). The language used most often in this community is OMG's Systems Modeling Language (OMG SysML^{TM}), which is an extension of UML to expand its modeling capabilities beyond software systems to address hardware, people, requirements, and parametric relationships (http://omgsysml.org/). A great deal of effort is being directed to understanding how to use SysML to model large scale, complex systems, incorporating multiple (discipline-specific) views, and integrating multiple analysis tools (Peak et al., 2009).

The approaches and experiences of MBSE present the OR community with two tantalizing opportunities. The first opportunity arises in situations where much is already known about using OR to answer particular kinds of questions in a particular domain, e.g., cycle time estimation in electronics manufacturing, production scheduling in aircraft assembly, or vehicle routing in package delivery. The opportunity is to package that knowledge together with a formal semantic model of the domain, and

deliver to domain stakeholders the capability to describe their problem—in its own terms, which they already understand—and get immediate and transparent access to appropriate OR analyses, without the direct intervention of an OR analyst. Simply put, the opportunity is to capture what we already know, and make it transferable, reusable, teachable, and deployable. Given the enormous collective repertoire of models and analyses, this is an opportunity to increase the reach and penetration of OR manyfold. Moreover, if both domain and OR knowledge are captured in formal semantics, they become much more easily taught and learned.

The second opportunity is to leverage the first opportunity to accelerate the creation of new and valuable OR-based knowledge, and its conversion to a transferable, reusable, teachable, and deployable form. If they are based on formal languages, domain-specific semantics can be elaborated to account for newly recognized problem domain elements or factors. New OR analyses, or enhancements to existing analyses could be more rapidly deployed by elaborating an existing infrastructure of domain specific languages and integrated OR analyses.

Formal Language and Knowledge Capture

The goal of capturing knowledge in a form that that is transferable, reusable, teachable, and deployable requires making knowledge explicit. Over the past 20 years, there has been a great deal of interest in methods to accomplish this, particularly in the context of information systems and the Internet. For example Vernadet (2007) has suggested the construction of ontologies as a way to achieve information systems interoperability through the use of metadata repositories. In the computing community, "ontology" usually implies the formal definition of classes representing concepts in a domain, properties of the classes representing features and attributes of the concept, and possibly restrictions on the properties (Dieng 2000). The ontology, together with instances of its classes, will constitute a "knowledge base." In this form, a knowledge base is machine readable, and can be manipulated using software.

There are many computational tools for authoring, editing, and visualizing ontologies (see, e.g., the techwiki page http://techwiki.openstructs.org/index.php/Ontology_Tools). However, these tools tend to be somewhat arcane and are often not easily accessible by application domain experts. A different strategy developed in the software engineering community and currently gaining traction in the systems engineering community is to create domain-specific languages (DSLs) that conform to a domain-specific ontology and thus are easier for domain experts to understand and use.

The language most commonly used by software engineers in the design of software applications is UML (http://www.uml.org/). UML is a graphical, object-oriented modeling language based on 13 diagram types which provide semantics for modeling application architecture, structure, and behavior, as well as business process flows, database, and message structure. A standards-based implementation of UML will include capabilities for elaborating the semantics, e.g., by further refining the

definition of generic objects or by adding new diagram types. For example, a generic object named "class" might be used to define new objects that are special kinds of "class," such as "machine_tool" or "transport_vehicle." These new objects might then be used by a domain expert to describe a particular application.

In 2007, OMG published a standard for a new modeling language, OMG SysML™, which is based on a subset of UML, and adds new diagram types specifically to support the modeling of complex systems incorporating software, hardware, and people (http://omgsysml.org/). A derivative of UML, SysML also is object-oriented and graphical. SysML supports the modeling of systems from multiple perspectives in a unified manner (Peak et al., 2009). It is a very expressive language for system modeling because it integrates the representation of structure (classes and the multiple kinds of relationships among them) and behavior (activities, state machines, and the sequence and timing of interactions among blocks).

Despite the relatively recent emergence of SysML, there have been a number of examples of its use in manufacturing (Huang et al., 2008; Batarseh et al., 2012), and supply chains (Thiers and McGinnis 2011; Ehm et al., 2011). Modeling an electronics assembly operation is described in Batarseh and McGinnis (2012), where the goal is to significantly reduce the time and cost of developing simulation models used to support production program planning.

In the system studied in Batarseh and McGinnis (2012), the assembly process starts with populated circuit card assemblies, to which hardware, such as connectors, will be assembled, and conformal coatings will be applied. The cards are then assembled into a chassis and additional coatings may be applied. Because the products have very high reliability requirements and may operate in extreme conditions, a large amount of testing is required, leading to significant amounts of rework. SysML was used to capture the semantics of the production process. Figure 2.3 summarizes the result. It illustrates the use of the "stereotype" facility of SysML to define new modeling concepts, e.g., refining "class" to specify a set of resource types, each with its own particular set of attributes. Specific instances of each resource type can be defined and stored in a library for ease of reuse. The stereotype facility also was used to define "part" and "final product" so that bills of materials could be created, and production schedules or requirements could be associated with final products. Finally, the types of processes required to produce a product were specified as stereotypes of the SysML "call action" object, and each different process type was given a set of appropriate attributes.

The domain expert would use these stereotyped objects, and perhaps libraries of their instances, to create both a bill of materials and a process plan for each subassembly and final assembly. A simple bill of materials is illustrated in Fig. 2.4 and a simple process plan in Fig. 2.5. These examples illustrate how the expressiveness of SysML can be exploited to create a graphical DSL that is easily accessible by the domain experts.

In this approach, two kinds of domain knowledge are captured in two distinct phases. First, the generic knowledge, the domain semantics, is captured using the stereotyping facility of SysML. This requires collaboration between domain experts and SysML modeling experts. In the second phase, the "use phase," the domain

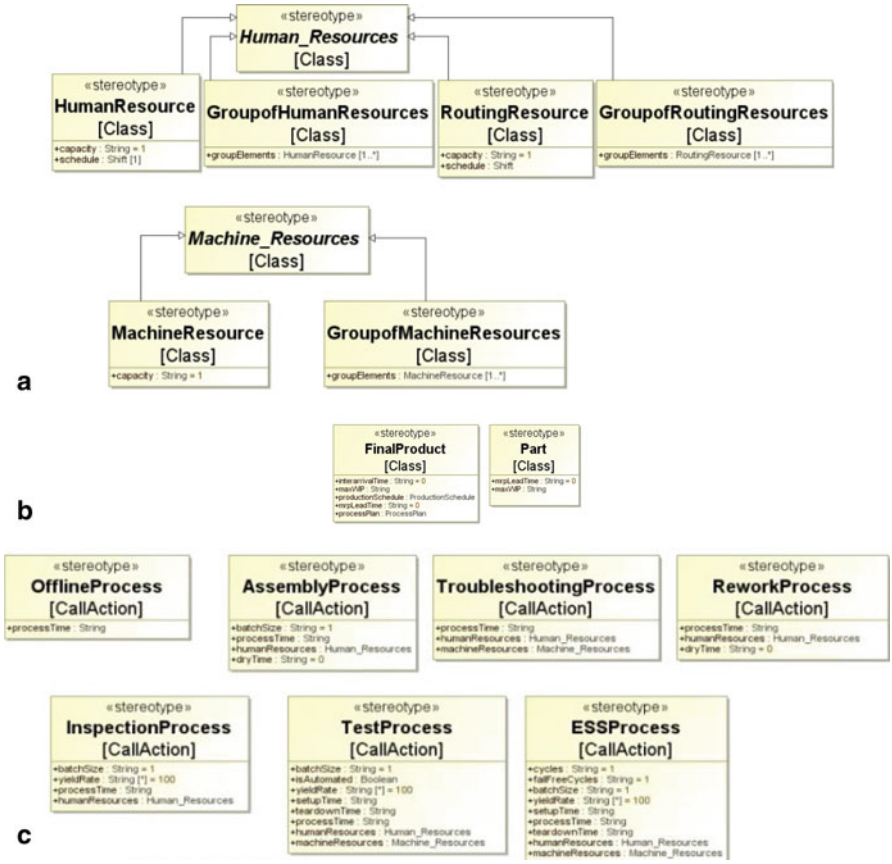


Fig. 2.3 Example of domain specific language semantics. **a** Resource semantics. **b** Product semantics. **c** Manufacturing processes semantics

specific language is used to capture knowledge of a particular application. One might reasonably ask, "how is this different from the usual OR study approach, where the OR analyst team works with domain experts to create the OR model?"

The difference, in fact, is quite significant. In the conventional approach, the knowledge captured about the domain is encoded in the OR model, severely limiting the opportunity to reuse this knowledge or to share it with other analysts. In particular, it makes it very difficult to reuse the knowledge for a different kind of analysis. For example, if the initial analysis used an optimization model, e.g., to establish capacity levels, a subsequent model using simulation, e.g., to size work-in-process buffers, would not be able to reuse the knowledge in a straightforward manner. With a DSL, reusable knowledge is captured both in the language itself, and possibly in every use of the language, as new information is added to libraries of similar objects.

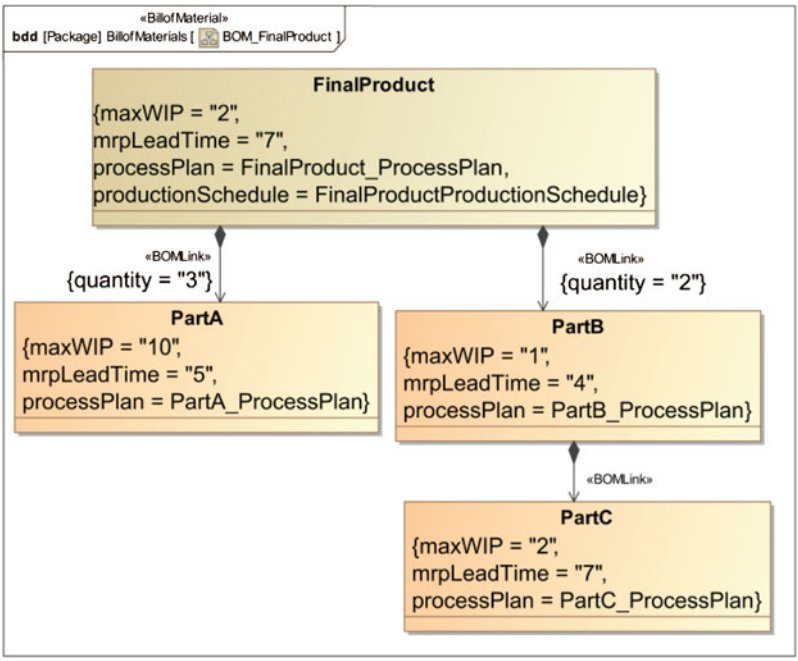


Fig. 2.4 Example bill of materials

Using SysML to create domain specific languages for well-understood problems appears to be a very tractable strategy. To understand how the other two kinds of knowledge—analysis knowledge and modeling knowledge—would be captured, it is important to understand some other aspects of the MDA approach

Meta-object Facility and Model Transformation

OMG has developed the Meta-Object Facility (MOF), "as an extensible model-driven integration framework for defining, manipulating and integrating metadata and data in a platform-independent manner" (http://www.omg.org/technology/documents/modeling_-spec_catalog.htm#MOF). In the MOF context, models expressed in a MOF-conforming language are simply data, to be authored, edited, viewed, manipulated, and exchanged between software systems. Metadata are "data about data," which can provide information about the structure of the data, and also important information about the data themselves, such as when they were created, by whom, etc.

MOF can be described in terms of both languages and models. The MOF architecture consists of four levels, with the highest level, M3 representing the most abstract language or model, and the lowest level, M0, representing an instance of a model,

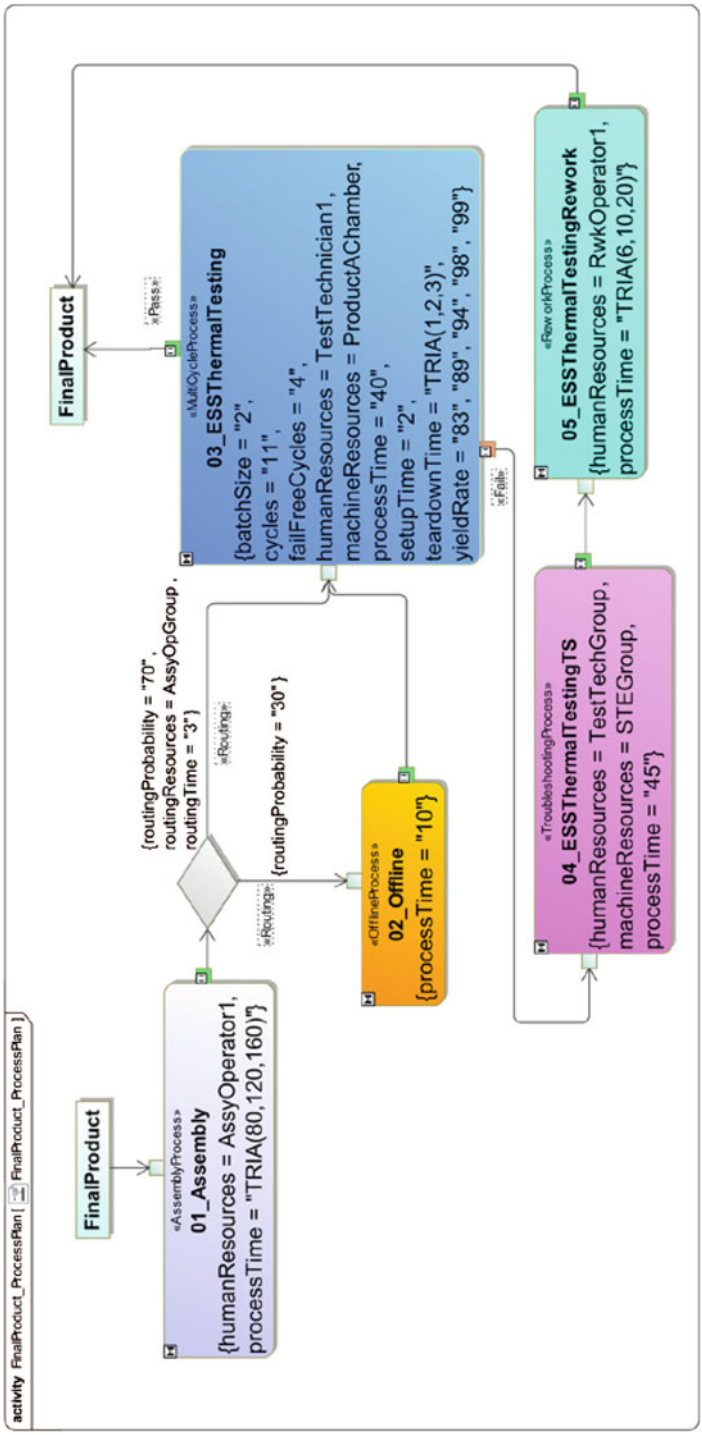


Fig. 2.5 Example process plan

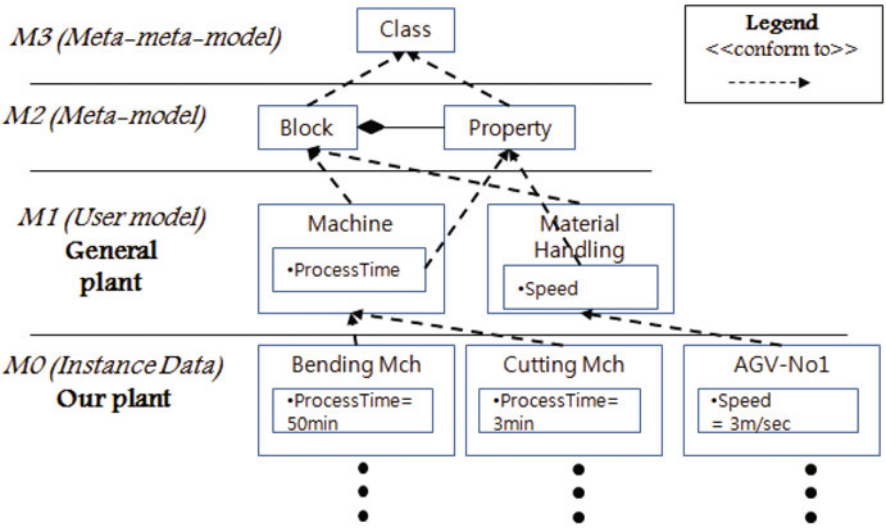


Fig. 2.6 Example of OMG modeling hierarchy

or a specific "expression" in some language. At level M3 is the meta-language (or meta-meta-model) which is used to define languages; often this meta-language also is referred to as "MOF." In MOF, this meta-language is used to define a number of specific languages, such as UML (for software system design), Common Warehouse Metamodel (CWM, for data warehousing) and SysML (for systems modeling), among others (see http://www.omg.org/technology/documents-/index.htm for a list of OMG technologies). Figure 2.6 from Kwon (2011) illustrates the OMG modeling hierarchy in the context of a DSL for production.

In Fig. 2.6, M3 contains the fundamental modeling constructs of the meta-language, e.g., the concept of "class." M2 corresponds to a specific language, such as SysML; in SysML the meta-language is used to refine the concept "class" by creating two new concepts, "block" and "property," where a property is a "part of" a block. The "part of" relationship used in M2 also is defined using the meta-language, although this is not shown in the figure. In the M1 level of the hierarchy, the M2 language, e.g., SysML, is used to describe a particular domain, by defining categories of "block" which have domain specific semantics, e.g., "machine" and "material handling," and each of these new kinds of blocks has particular kinds of properties. It is at the M1 level that a "language" of production is created, and thus it could be said that SysML is the "meta-language" for this domain-specific "production language." Finally, at the M0 level, a description of a specific factory contains instances of the machine and material-handling blocks, representing particular machines and material-handling resources in the particular factory. The "part of" relationship between a block and its properties is shown explicitly in M2, but implicitly in M1 and M0 by containing the properties within the owning block. Note that in Fig. 2.6, each level is characterized in terms of "models," where M0 corresponds to an

"instance model" and M3 corresponds to a "meta-meta-model." It is generally assumed that four levels of modeling hierarchy are sufficient, where the top two levels are "standard" languages (or models, if one prefers) and the bottom two levels are the application of those standards to a particular problem domain. For the example of electronics assembly presented in Figs. 2.3, 2.4, and 2.5, Fig. 2.3 would correspond to a "user model" or DSL at M1, and the specific model constructed with that DSL, shown in Figs. 2.4 and 2.5, would correspond to M0.

The OMG modeling hierarchy is a powerful approach for capturing domain semantics in a way that is accessible by the domain experts because the domain specific language—the "user model" at M1 in Fig. 2.6—can employ the semantics that are familiar to the domain expert. At the same time, because the user model conforms to the meta-model, which conforms to the meta-meta model, the instance models created with this DSL are easily manipulated using appropriate software tools for model transformation.

In fact, this is the true power of the MDA approach—given two languages, both conforming to the MOF hierarchy (i.e., both conforming to the meta-language), and both capable of expressing a view of a particular system, then, under certain conditions, it is possible to define a mapping between the two languages, and use that mapping to transform an instance model in one language to an instance model in another language. The classic example in MDA is the description of a business process stated using BPMN, see http://www.omg.org/spec/BPMN/2.0/) and the transformation from BPMN to, say, Java to create the source code for the application software required to implement the business process.

In adapting these concepts to production systems decision support, the goal is to translate an instance of a production system model, expressed in a DSL derived from SysML, into an instance of an analysis model, expressed in some appropriate modeling language. For this to be possible, the information contained in the source and target meta-models must be sufficient to allow the definition of a set of rules for mapping from the source meta-model to the target meta-model that, when applied to the source instance model, will translate it into the desired target instance model. In other words, between the source and target meta-model and the mapping rules, all the knowledge needed to create a target instance model is captured in a formal way.

To support this idea of model transformation, OMG has specified a set of languages, referred to collectively as QVT (see http://en.wikipedia.org/wiki/QVT for a good overview) for creating and executing mappings between MOF-compliant models. The essence of the model transformation process is illustrated in Fig. 2.7, which identifies seven distinct models. In the electronics assembly example given earlier, the source model, which conforms to a source meta-model, which conforms to the meta-meta-model, would be the instance model created using the DSL (a customization of SysML), which conforms to MOF. The target model might be, e.g., a simulation model, which conforms to its meta-model, which conforms to MOF. The sixth model is the meta-model for transformation rules, which also conforms to MOF. The final model is the model specifying the particular transformation rules, which conforms to its meta-model and which references the source and target meta-models.

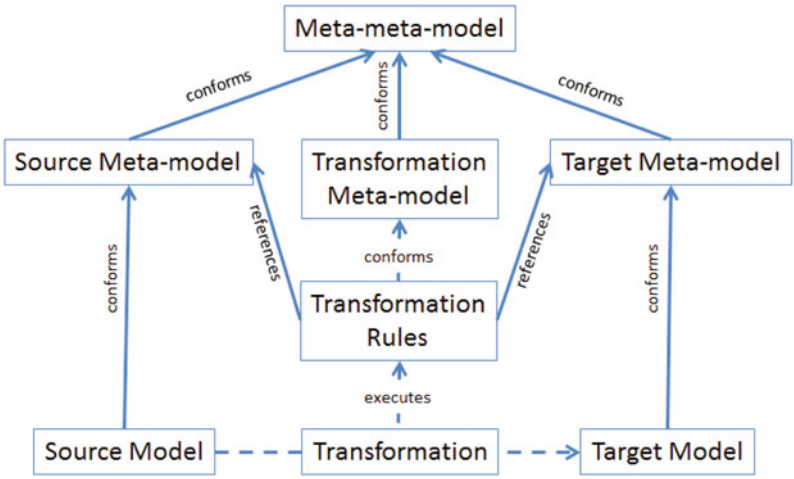


Fig. 2.7 Model transformation

Of course, to implement the process illustrated in Fig. 2.7, a computational tool (labeled "Transformation" in Fig. 2.7) is required, which will take as input the source model, the source meta-model, the transformation rules model, and the target meta-model, and using these inputs will create the target model. This is an area of active development, but there are available open-source tools, such as the Atlas Transformation Language (ATL) (http://www.eclipse.org/atl/).

The study described in Batarseh and McGinnis (2012) demonstrates that the MDA approach can be adapted to support OR modeling in production systems. In their study, the target model was an Arena™ simulation. The Access™ database model export/import facility of Arena was used as a proxy for Arena, and MOF was used to create a meta-model for the corresponding data schema. A transformation script was developed, which enabled the transformation of production system models created with the DSL into Access databases, which then were imported into Arena for analysis. The process was extensively tested in an industry setting, and the impact on "typical" simulation studies has been a reduction from about 200 person hours for developing and running simulations in the conventional approach to about 20 person hours using the DSL and model transformation approach.

Ubiquity of Models and Modeling

Models and modeling are ubiquitous in any application of OR. In a particular application, there will be models of the question to be answered or the problem to be solved, models of the analysis that supports answering the question or solving the problem, and models of the computation needed to support the analysis. As discussed above, there can be models of the relationships between models. Each of

these models may be explicit or implicit. An example of an explicit model is the mathematical formulation given in Fig. 2.1, or the SysML-based model of a process plan in Fig. 2.5. The semantics of the domain is often an implicit model, or at best, partially explicit, e.g., through the use of a list of terminology. The model of the relationship between the domain semantic model and the explicit analysis model is almost always completely implicit, i.e., it remains the personal knowledge of the analyst/modeler. The knowledge contained in implicit models is very difficult to share and impossible to archive. In MDA or MBSE, the implicit knowledge that is critical in creating solutions is made explicit, whether the solutions are Java codes for implementing business processes, or OR-based decision support models.

MDA and MBSE go beyond simply making modeling knowledge explicit, which could be done using documents. MDA and MBSE make the explicit modeling knowledge *formal,* in the sense that it is computer readable, but also conforms to a formal syntax and semantics, so that it can be algorithmically manipulated. Capturing modeling knowledge explicitly and formally is the key to making OR ubiquitous, i.e., making OR-based decision support available, on-demand, to domain stakeholders and decision makers. This is because doing so means that the formerly labor-intensive task of using implicit knowledge to translate between implicitly known domain models and explicit formal analysis models can be replaced by a much simpler process of explicitly describing the domain problem and automating the creation of the corresponding analysis model using explicit modeling knowledge.

In some ways, the application of MDA and MBSE to OR-based decision support in production may be the next phase in the natural evolution of the field. If analysis modeling languages like AMPL are seen as corresponding to third-generation programming languages, then the integration of a production DSL, model transformation, and target analysis model solver could be seen as corresponding to a fourth-generation programming language (see http://en.wikipedia.org/wiki/Fourth-generation_programming_language and the links there for a discussion of programming language generations).

Implications

The adaptation of MDA/MBSE in the deployment of OR models to support production system decision makers has significant implications for the curriculum of OR and production systems, for the way OR-based decision support is deployed in routine applications, and for the nature of research addressing decision support in production systems.

Today, the typical curriculum content addressing OR in production systems comes in two primary forms. Analysis content addresses the canonical analysis formulations and analysis methodologies, e.g., linear optimization, the simplex method, and a modeling language/solver like AMPL/CPLEX, or Monte Carlo sampling, discrete event simulation, and modelers/solvers like Arena or AnyLogic. Domain content for applications in significant areas of practice, such as supply chain engineering,

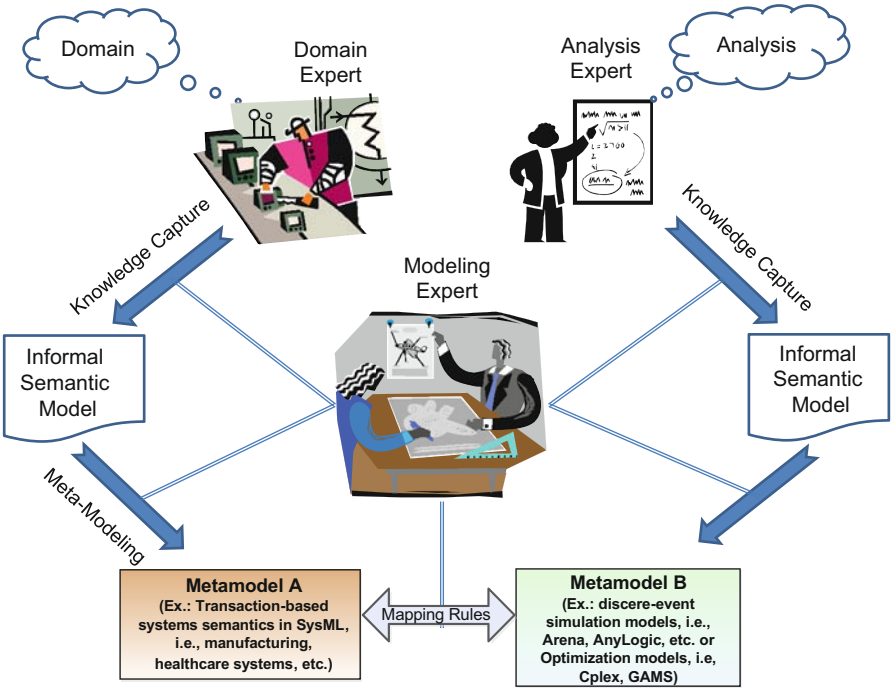


Fig. 2.8 Future deployment—off-line activities

humanitarian logistics, finance, or health care delivery is addressed informally by defining terms, often through examples, and perhaps presenting mini-case studies. If we recognize modeling per se as a category of knowledge that can be captured and deployed in routine applications, the curriculum will need to change to reflect the tools and methods required and the growing archive of modeling knowledge.

Faculty and students who choose the path of modeling as their area of expertise will need to become conversant with formal languages and model transformation theories, as well as with tools for creating and deploying DSLs and model transformations. Just as today we see deep mathematical results contributing to the advance of the field, in the future we will see deep theoretical results from linguistics and computer science enhancing our ability to create and deploy powerful solutions.

Figures 2.8 and 2.9 illustrate key aspects of how OR-based decision support systems will be deployed in the future for routine applications. Off-line, as a foundational activity, OR modeling experts will collaborate with domain experts to capture knowledge about the domain, first as informal semantic models, perhaps using SysML, and then as meta-models. This process can be iterative, and it can proceed by first capturing a basic description of the domain and subsequently elaborating the description, adding new aspects of the domain as they become recognized as important and valuable to include.

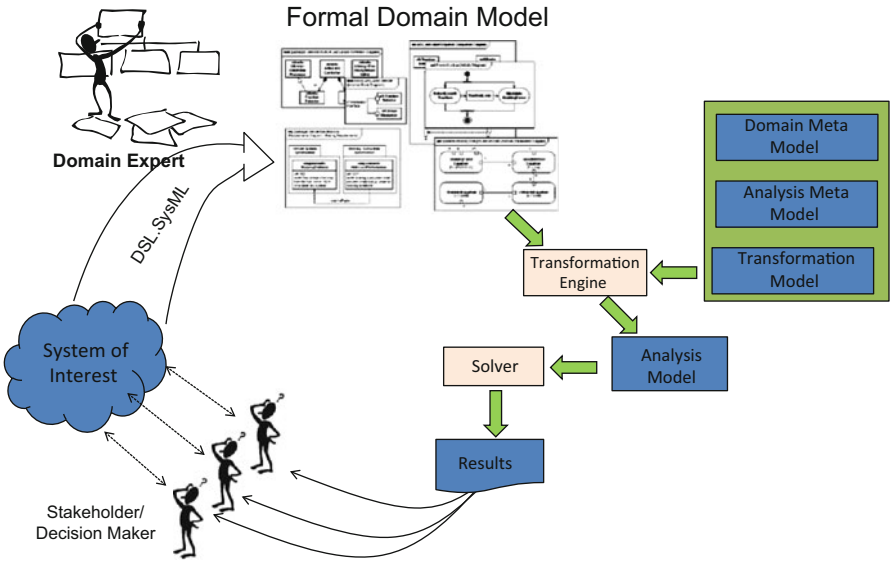


Fig. 2.9 Future deployment—on-line

Similarly, modeling experts will work with analysis experts to capture knowledge about the kinds of analyses that would be valuable to domain stakeholders as they make important decisions. This knowledge also might be captured initially using SysML and then captured formally as meta-models. Again, this knowledge capture can be iterative, continuously improving the range and scope of the analyses available to the domain stakeholders. Finally, the modeling expert will work to create the mapping rules relating the domain meta-model to each of the relevant analysis meta-models. This work also may require collaboration with both domain and analysis experts.

Perhaps most important is that the process in Fig. 2.8 is not a one-time process with a single result. Rather, the knowledge captured in this process can be continuously refined and extended, thus continuously expanding the scope of "routine applications."

Figure 2.9 illustrates how MDA/MBSE would impact the actual use of OR-based decision support. In general, there will be multiple stakeholders/decision makers for any production system. Using the DSL for the production system, a domain expert (who also could be a decision maker) will create the formal model that reflects the problem aspects important to the collection of decision makers. The knowledge captured in the off-line activities of Fig. 2.8 will then be used to generate specific analysis models which provide information or guidance to the decision makers.

The process described in Fig. 2.9 will require not only capabilities for generating instances of appropriate decision support models, but also user interface and data validation capabilities. The rapidly developing field of "analytics" will provide many of the necessary data validation capabilities.

Note that the product of the activities illustrated in Fig. 2.8 is a new archive of knowledge, much of which could be integrated into the traditional curriculum. Furthermore, as tools become available for performing the activities of Fig. 2.9, these tools also could be integrated into the curriculum, much as are the contemporary analysis modeling tools like AMPL or Arena. As the knowledge and tools are integrated into the curriculum, much more realistic domain problems, such as global supply chains, distribution networks, etc., also can be integrated into the curriculum, giving students much more realistic case problems, and the opportunity to gain broader insights than is currently practical.

Finally, there are implications for research on OR-based decision support in production systems. Creating the canonical model for a domain of practice, such as supply chains, finance, health care delivery, or manufacturing, is a task whose difficulty can hardly be overstated. Such a canonical model must address at least three related aspects of the domain:

- Structure, i.e., the relevant resources and actors (including the external environment or boundary conditions), and the relationships among them
- Behavior, i.e., the ways in which the states of structural components can change and how structural components interact
- Control, i.e., how stakeholders in the domain can or should attempt to achieve a particular trajectory of state changes

Moreover, the canonical model should accommodate the (frequently conflicting) viewpoints of the key domain stakeholders, and should enable the specification of instance models containing all the source information that would be needed to populate the intended target decision support models.

These canonical models will only result from great creativity on the part of teams of researchers, applying knowledge of both the domain and the relevant decision support analyses, and using appropriate modeling languages. This represents a kind of research which is very different from what one might find today in the journals that publish production systems research, but which is clearly of great archival value to the field.

In a similar way, creating the analysis meta-models and the transformation rules also presents daunting challenges. Many decision support models share a "core formulation," on which variations are developed, and it would seem to be desirable to have a "core meta-model" for the associated analysis, which could be further refined for the variations. Contemporary research, on the other hand, tends to treat each formulation as a distinct entity, without reliance on any other formulation, so there is considerable intellectual work simply to establish an appropriate modeling framework within which the core meta-model and its variations could be constructed.

Just as there may be families of decision support meta-models, there may be corresponding families of transformation rule models. In fact, a major research opportunity is simply to better understand the model transformation process in this context, and to begin to "engineer" transformation solutions.

Two Fundamental Intellectual Challenges

The famous statistician George E. P. Box wrote that "essentially, all models are wrong, but some are useful" (Box and Draper, 2012). Among the most fundamental questions in science and engineering are those whose answers improve the repertoire of useful models we have at our disposal for helping us to understand both natural and man-made phenomena and to aid us in harnessing these phenomena for useful purposes. The history of particle physics aptly illustrates the process of asking and answering fundamental questions: Prout's concept of the proton (Prout 1815) was "proven" by Rutherford's discovery in 1917 (Rutherford 1919); Gell-Mann and Zweig independently conjectured that the proton was really made up from other particles (http://en.wikipedia.org/wiki/Quark), and those particles were subsequently observed at the Stanford Linear Accelerator (Bloom et al., 1969). The models of protons, quarks, and all the other subatomic particles are part of a larger search for fundamental knowledge about the physical universe. New models emerged from the investigation of older models, or as very different alternative explanations of phenomena. Importantly, the models in particle physics are formal models whose semantics are well documented and universally used within the research community.

The kind of deep knowledge of the physical universe represented by models in particle physics is essential to the invention, development, and application of new materials and processes that enable our modern way of life, from the biology and chemistry of food crops, to the synthesis of materials for clothing and shelter, to the production and distribution of energy. All these materials and processes result from understanding and manipulating physical processes.

Production systems, of course, depend also upon deep knowledge of the physical universe. But production systems are, themselves, an artificial construct, in the sense that their configuration and the rules by which they operate, while conforming to the laws of physics, cannot be explained purely in terms of physical phenomena—they also have a significant artificial component, which results from the decisions made by their stakeholders.

In order for OR to become ubiquitous in the support of production system decision making, it is necessary that our knowledge of production systems becomes formalized, in much the same way that the knowledge of particle physics has become formalized. So a fundamental question is simply this: "What do we know about production systems qua production systems, and how do we know it?" This is a question about the models in which we encode what we know about production systems, and today it would be a very difficult question to answer because there is not a common semantic model that is used by researchers and practitioners in the field of production systems. The development, dissemination, maintenance, and use of such a common semantic model collectively represent a fundamental challenge. One might think of this as the "science" of production systems decision support.

It is not enough to create a common semantic model of what is known about production systems. In order for OR to become ubiquitous in production system decision support; this knowledge of production systems must also be made actionable.

A central component in making this knowledge actionable is combining it with modeling knowledge in order to automate the creation of decision support analysis models. This is the essence of the second fundamental challenge, i.e., discovering an effective strategy for combining semantic knowledge of the domain, semantic knowledge of the analysis, and modeling knowledge of the relationships between domain knowledge and analysis knowledge. One might think of this as the "engineering" of production systems decision support.

Conclusion

The continuing growth of the field of OR in general, and in production systems in particular, depends on the discovery of new knowledge—knowledge about domains of practice, knowledge about forms of analysis that support decision making, and knowledge about the translation between the domain instance and the analysis instance. This chapter has been about evolving developments that hold the promise of capturing that knowledge in a form that makes it transferable, reusable, teachable, and deployable. The potential impact of these developments is at least as great as the impact of the computing revolution, which brought large-scale OR analyses to the desktop of the OR practitioner. Capturing the benefits will require operations researchers to embrace these new knowledge capture and exploitation tools with the same enthusiasm that they embraced computation in the mid-1970s.

Acknowledgments While I alone am responsible for what I write, the ideas expressed in this chapter have been strongly influenced by my involvement with PDES, Inc. as a Board Member representing Georgia Tech's Manufacturing Research Center, by my collaborations with Dr. Chris Paredis and Dr. Russell Peak in the Model-Based Systems Engineering Center at Georgia Tech, and by my work on MBSE with a number of for user and current graduate students at Georgia Tech, including Dr. Edward Huang, Dr. Ky Sang Kwon, and Mr. George Thiers, as well as by working with two excellent postdoctoral fellows, Dr. Volkan Ustun and Dr. Ola Batarseh. This work has been supported by a variety of sponsors, including the Gwaltney Professorship, Lockheed Martin, Rockwell Collins, General Electric Energy Systems, Boeing, and DARPA.

References

Anonymous. (2011a). Artisan. http://www.merriam-webster.com/dictionary/artisan. Accessed 26 Dec 2011.

Anonymous. (2012a). Software factory. http://en.wikipedia.org/wiki/Software_factory. Accessed 4 Jan 2012.

Batarseh, O., McGinnis, L., & Lorenz, J. (2012). MBSE supports manufacturing system design, The 22nd Annual INCOSE International Symposium Proceedings, Rome, Italy.

Bloom, E. D., et al. (1969). High-energy inelastic e-p scattering at 6° and 10°. *Physical Review Letters, 23*(16), 930–934.

Box, G. E. P., & Deaper, N. R. (1987). *Empirical model-building and response surfaces* (p. 424). Wiley.

Chungoora, N., Cutting-Decelle, A.-F., Young, R. I. M., Gunendran, G., Usman, Z., Harding, J. A., & Case, K. (2011). Towards the ontology-based consolidation of production-centric standards, *International Journal of Production Research, 51*(2), 327–345.

Dieng, R. (2000). Knowledge management and the internet. *Intelligent Systems and their Applications, IEEE, 15*(3), 14-17 (May/Jun 2000).

Ehm, H., Heilmayer, S., Ponsignon, T., & Russland, T. (2011). A discussion of object-oriented process modeling approaches for discrete manufacturing on the example of the semiconductor industry. In Proceedings of the 2011 Winter Simulation Conference, S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, eds.

Estefan, Jeff A. (2007). Survey of model-based systems engineering (MBSE) Methodologies. http://www.omgsysml.org/MBSE_Methodology_Survey_RevA.pdf.

Huan, S., Sheoran, S., & Wang, G. (2004). A review and analysis of supply chain operations reference (SCOR) model. *Supply Chain Management, 9*(1), 23–29.

Huang, E., Ky S. K., & McGinnis, L. F. (2008). Toward on-demand wafer fab simulation using formal structure and behavior models. Proceedings of the 2008 Winter Simulation Conference, S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler eds.

Huang, E., Ramamurthy, R., & McGinnis, L. F. (2008). System and simulation modeling using SysML. Proceedings of the 2007 Winter Simulation Conference, S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler eds.

Jiang, Y., Peng, G., & Liu, W. (2010). Research on ontology-based integration of product knowledge for collaborative manufacturing. *The International Journal of Advanced Manufacturing Technology, 49*(9), 1209–1221.

Kiron, D., Schockly, R., et al. (2011). Analytics: The widening divide. *MITSloan Management Review, 53*(2):3–20.

Lavrischeva, K. M. (2011). Theory and practice of software factories. *Cybernetics and Systems Analysis, 47*(6)*,* 961–972.

Libert, S., & ten Hompel, M. (2011). Ontology-based communication for the decentralized material flow control of a conveyor facility. *Logistics Research, 3*(1), 29–36.

Meyers, B., & Hans, V. (2011, 1 December). A framework for evolution of modelling languages. *Science of Computer Programming, 76*(12), 12.

Peak, R., Paredis, C., McGinnis, L., Friedenthal, S., & Burkhart, R. (2009). Integrating system design with simulation and analysis using SysML. *INCOSE Insight Special Edition on MBSE, 12*(4), 40–43.

Prout, W. (1815). On the relation between the specific gravities of bodies in their gaseous state and the weights of their atoms. *Annals of Philosophy, 6,* 321–330.

Ramos, A. L. F., Vasconcelos, J., & Barcelo, J. (2011). Model-Based Systems Engineering: An Emerging Approach for Modern Systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42*(1), 101–111.

Rutherford, E. (1919). Collision of alpha particles with light atoms; an anomalous effect in nitrogen. *The Philosophical Magazine, 37*(222), 537–587. London: Taylor and Francis.

White, S. A. (2006). Introduction to BPMN. http://www.bpmn.org/Documents/OMG_BPMN_Tutorial.pdf. Accessed 28 Dec 2011.

Chapter 3
Integrated Production Planning and Pricing Decisions in Congestion-Prone Capacitated Production Systems

Abhijit Upasani and Reha Uzsoy

Introduction

The highly capital intensive nature of the semiconductor industry requires its factories to operate at high utilization levels, where small changes in utilization can cause large changes in lead times. Demand for these products over time can be quite uneven, which leads to firms trying to shape their demand by price promotions in order to maintain high factory utilization levels. However, it is well known from queuing models of manufacturing systems (Buzacott and Shanthikumar 1993) that higher resource utilization leads to increasing lead times. This raises the possibility of price reductions becoming counterproductive—an unwise price promotion can create a surge in demand that, after some time, results in a large increase in lead times, missed delivery dates, cancelled orders and lost future business.

To this end, companies will often develop aggregate production plans at the product family level for several months (up to 18 months in the case of one semiconductor manufacturing firm described by Allison et al. (1997)) in order to identify potential capacity bottlenecks and make sure that competitive lead times can be maintained. This plan, based on current order books and marketing forecasts, permits the planning of price promotions as part of the process. Given the long planning horizon, an aggregate planning model focusing on the loading of resources and management of prices over time to achieve maximum profit with competitive lead times would be useful to management. The high utilization levels at which many capital-intensive factories, such as semiconductor wafer fabs, operate renders a planning model that accounts for the nonlinear relationship between resource utilization and lead times desirable, especially when customers are sensitive to both lead times and prices.

R. Uzsoy (✉)
Edward P. Fitts Department of Industrial and Systems Engineering,
300 Daniels Hall, Campus Box 7906, North Carolina State University,
Raleigh, NC 27695–7906, USA
e-mail: ruzsoy@ncsu.edu

A. Upasani
Terra Technology, 20 Glover Avenue, Norwalk, CT 06850, USA
e-mail: abhijit.upasani@gmail.com

P. S. Pulat et al. (eds.), *Essays in Production, Project Planning and Scheduling,*
International Series in Operations Research & Management Science 200,
DOI 10.1007/978-1-4614-9056-2_3,

Most existing pricing-production planning models do not address this problem in its full complexity. In particular, most such models do not consider the effect of workload on queues and lead times, and hence may underestimate the price that should be charged at a given output level. In particular, if high demand results in long lead times due to congestion in the production facility, the assumption that demand can be met within a fixed lead time may result in significant lost sales. Dynamic pricing models based on queuing, on the other hand, generally describe long-run steady-state behavior and do not provide a framework for decisions to be made over time.

The model presented in this chapter is a first step towards addressing these issues. We use clearing functions (CFs, Asmundsson et al. 2009) to capture the nonlinear relationship between resource utilization and lead times. Following the literature, customer behavior is modeled using a demand function that is linear in both price and lead time, with a maximum lead time beyond which no demand will be forthcoming. In each planning period, customers can observe the average flow time associated with the current workload of the production system, and place orders accordingly. Such systems are already in use by semiconductor manufacturing companies such as Taiwan Semiconductor Manufacturing Corporation which provide contract manufacturing services to other firms (www.tsmc.com/english/dedicatedFoundry/services/eFoundry.htm). The model jointly determines the price and the amount of work to be released in each time period, thus determining the average lead time associated with that planning period. The model allows the possibility of production smoothing through the accumulation of finished goods inventories and price promotions.

Our results show that when the demand is sensitive to lead times, the CF model with workload-dependent lead times produces significantly higher profits than a conventional model assuming a fixed lead time. In several scenarios the release plans suggested by the fixed lead time model are unable to satisfy the market demand generated by the associated prices, since they assume that a fixed lead time can be maintained in the face of the high demand created by low prices. In fact, the increased demand resulting from price reductions can only be met with long lead times, which end up reducing demand. Hence a thorough understanding of the effects of pricing on lead times and queues is essential for capacity constrained firms that plan to use dynamic pricing. As suggested by Pekgun et al. (2008), the separation of lead time and pricing considerations between the production and marketing operations is a significant obstacle to this understanding, suggesting the need for more emphasis on this interface in capital-intensive firms operating at high utilization levels.

Literature Review

Our research is related to three different streams of literature: joint pricing and production planning models, models for load-dependent lead-time quotation, and steady state models that study relationships between price and lead times.

Joint pricing and production planning models aim to produce a profit-maximizing combination of prices and production plans. Eliashberg et al. (1991) and Yano and

Gilbert (2003) present detailed reviews of this stream of literature. This literature also includes dynamic pricing models that change prices over time to improve profitability (Swann 2001; Charnsirisakskul et al. 2006; Deng and Yano 2006). Ahn et al. (2007) present an interesting model where demand in a given period depends on prices in preceding periods. Adida and Perakis (2006) consider a continuous time model with a linear demand function and an additive model of uncertainty, and present a robust optimization model. In a subsequent paper (Adida and Perakis 2010) they compare robust and stochastic optimization models for this problem, noting that stochastic optimization models can be sensitive to the probability distributions used. The related area of dynamic pricing focusing on the interface with inventory management is reviewed by Elmaghraby and Keskinocak (2003).

Researchers in this area have used simple, aggregate capacity constraints with limited ability to consider interactions between capacity utilization and lead time. When faced with high demands that saturate capacity constraints in a given period, these models will build inventory in earlier periods, effectively increasing lead times. However, this dynamic does not capture the rapid nonlinear increase in lead times observed at higher utilizations, providing an incomplete picture of system behavior. Recent work (Kefeli et al. 2011) has shown that in the presence of congestion the theoretical output of the system may not be achieved due to the very high work in process inventories required to achieve them, causing these types of capacity constraints to give an optimistic picture of the production system's ability to meet demand. We illustrate this effect in our numerical examples.

Chen and Hall (2010), in contrast, consider the pricing of individual orders on a single machine or a two-machine flow shop to maximize profit under different cost criteria which are determined by the production schedules. They provide exact dynamic programming algorithms and heuristics, and demonstrate that even heuristic solutions to the problem yield significant improvement in profit over the case where prices and schedules are determined independently. Since these models represent capacity at a very fine level of detail, they capture the relationship between utilization and lead times correctly. However, such models do not easily scale up to the longer time periods addressed in this chapter.

The second stream of literature encompasses models that estimate lead times based on the current state of the system and use these lead times for order negotiation. These models recognize that lead times are load dependent and address operational decisions like input control or order selection, price and lead-time quotation, and capacity investment (Donohue 1994; Easton and Moodie 1999; Elhafsi and Rolland 1999; Elhafsi 2000; Charnsirisakskul et al. 2004; Plambeck 2004). While these models allow marketing to make realistic lead-time quotations to be used in price negotiation, they do not capture the relationship between prices, resource utilization and lead times.

A related set of models, classified in the literature as order acceptance models, assume stochastic (usually Poisson) customer arrivals and quote each customer a delivery date based on system status (Dellaert 1991; Duenyas 1995; Duenyas and Hopp 1995). These models assume a certain probability that the customer will actually place an order when quoted the delivery date, thus obtaining an effective arrival

rate for orders. Late orders are penalized and the models aim to minimize the impact of this penalty on revenue, which is fixed for every order.

The last stream of models conducts steady state analyses of relationships between price, lead time and capacity for *M/M*/1 systems (Low 1974; Palaka et al. 1998; So and Song 1998; Boyaci and Ray 2003; Ray and Jewkes 2004). Almost all these models use a demand function that is linear in both price and lead time to represent the market and aim to set prices and lead times subject to a service level constraint under steady state conditions. These models yield useful managerial insights through their characterization of optimal behavior, but their steady state nature does not allow them to be used to develop pricing and production plans over a finite horizon. Liu et al. (2007) study price and lead-time setting in a decentralized supply chain where a supplier specified a wholesale price and a planned delivery time, while the retailer quotes a retail price. Customers are sensitive to both lead time and retail price. They model the behavior of the supplier and retailer as a Stackelberg game and obtain the equilibrium strategy of both actors. Pekgun et al. (2008) developed a steady-state make-to-order (MTO) model that incorporates coordination mechanisms for price and lead-time quotation.

Plambeck (2004) considers capacity setting, price and lead-time quotation, and order sequencing decisions in a MTO system with two customer classes and compares dynamic against static lead-time quotations (similar to our Fixed Lead Time (FLT) model). The key assumption the author makes is that customers belonging to the "patient" class will tolerate long lead times. The author requires this slow-moving portion of the order queue to be so large that the system utilization approaches 100 %, allowing the author to apply heavy-traffic queuing approximations to derive optimal decision policies. Our CF model considers a different problem, that of determining an integrated aggregate plan for factory loading and pricing over discrete time periods in the face of the market's sensitivity to lead times. Our model does not impose a utilization level on the system but instead allows the system to choose its optimal utilization level. Consistent with Plambeck's results, our model also shows that taking the state of the system into account can yield significantly higher profit than a fixed lead-time model.

The joint planning models in the first stream represent aggregate planning decisions in a make-to-stock environment, where a different price is quoted every period, but all orders in the same period observe the same price. These joint planning models fall under the domain of models at the production/marketing interface that also includes models for sales-production coordination mechanisms (Eliashberg and Steinberg 1991; Upasani and Uzsoy 2008). Models in the last two environments focus on a MTO environment where no stocks of finished goods inventory are held and each order can be quoted a separate price or lead time. Detailed reviews of models in the last two streams are found in Chatterjee et al. (2002), Keskinocak and Tayur (2004), and Upasani and Uzsoy (2008).

To summarize the existing literature, joint planning-pricing models have limited ability to capture the effects of utilization on delivery times, whereas steady-state lead-time quotation models do not yield medium-term plans over a finite horizon. Recent developments in production planning models with load-dependent lead times

(Pahl et al. 2005) provide avenues for integrating state-dependent lead times into models of the production-marketing interface. Specifically, we use CFs (Pahl et al. 2005, 2007; Asmundsson et al. 2006; Asmundsson et al. 2009; Missbauer and Uzsoy 2010), which relate the expected throughput of a production system in a planning period to the expected work-in-process (WIP) inventory level over the period.

Clearing Functions

A promising approach to modeling workload-dependent lead times in production planning has been the use of CFs (Karmarkar 1989) that represent the expected output of a resource over a given period of time as a function of the expected WIP inventory level over that period. The term has its origin in work by Graves (1986) that specifies the fraction of the current WIP that can be processed to completion ("cleared") by a resource in a given time period. Karmarkar (1989) and Srinivasan et al. (1988) independently develop nonlinear CFs for production planning models. We shall use the term "WIP" to denote any reasonable measure of the WIP inventory level over a period of time that can be used as a basis for a CF. An extensive review of CFs and their use in production planning models is given by Missbauer and Uzsoy (2010)

To motivate the use of a nonlinear CF, consider a resource that can be modeled as a *G/G/*1 queuing system in steady state. The average number in system, i.e., the expected WIP, is given by Medhi (1991) as

$$w = \frac{(c_a^2 + c_s^2)}{2} \frac{\rho^2}{(1-\rho)} + \rho \tag{3.1}$$

where c_a and c_s denote the coefficients of variation of interarrival and service times, respectively and ρ the utilization of the server. Setting $c = ({c^2}_a + {c^2}_s)/2$ and rearranging (1), we obtain a quadratic in W whose positive root yields the desired ρ value. Solving for ρ with $c > 1$ yields

$$\rho = \frac{\sqrt{(W+1)^2 + 4W(c^2-1)} - (W+1)}{2(c^2-1)} \tag{3.2}$$

which has the desired concave form. When $0 \leq c < 1$, the other root of the quadratic will always give positive values for ρ. When $c = 1$, (3.2) simplifies to $\rho = W/(1+W)$, again of the desired concave form. We see that for a fixed c value, utilization, and hence throughput, increase with WIP but at a declining rate due to variability in service and arrival rates.

Several authors discuss the relationship between throughput and WIP levels in the context of queuing analysis, where the quantities studied are the long-run steady-state expected throughput and WIP levels. Agnew (1976) studies this type of behavior in the context of optimal control policies. Spearman (1991) presents an analytic congestion model for closed production systems with increasing failure rate processing time distributions that describes the relationship between throughput and WIP. Hopp

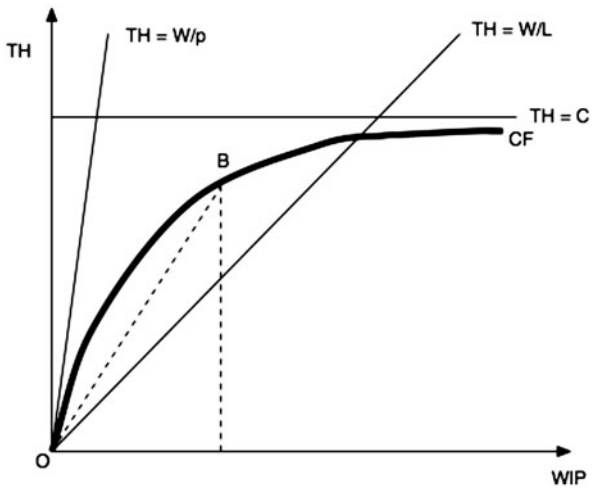


Fig. 3.1 Examples of clearing functions (Karmarkar 1989)

and Spearman (2001) provide a number of illustrations of CFs for a variety of systems. Srinivasan et al. (1988) derive the CF for a closed queuing network with a product form solution. While these approaches are based, as is our analysis above, on steady-state queuing models, a number of researchers have examined the issue of estimating CFs when the underlying queuing system is not in steady state. Asmundsson et al. (2009) show that even under transient conditions the concave shape of the CF will be maintained. Missbauer (2009) and Selçuk (2007) use transient queuing models to derive CFs under somewhat different sets of assumptions.

Figure 3.1, derived from Karmarkar (1989) depicts several examples of CFs considered in the literature to date. The horizontal line $TH = C$ corresponds to a fixed upper bound on output over the period, but without a lead-time constraint it implies that production can occur without any WIP in the system if work release and production are synchronized. This is reflected in the independence of output from the WIP level, which may constrain throughput to a level below the upper bound by starving the resource. This approach is implemented in, for example, the Capacitated MRP (MRP-C) approach of Tardif and Spearman (1997) and most LP approaches such as that of Hackman and Leachman (1989), but is supplemented with a fixed lead time that is an exogenous parameter independent of workload. The linear CF of Graves (1986) is represented by the $TH = W/L$ line, which implies a lead time of L periods that can be maintained independently of the WIP level. Note that if WIP and output are measured in the same time units (e.g., hours of work), the slope of the proportional part of the function is $1/L$, where L is the average lead time. However, as seen in Fig. 3.1, this model may suggest infeasible output levels when WIP levels are high. If a fixed lead time is maintained up to a certain maximum output, we have the relationship $TH = min\{W/L, C\}$. In the special case of the Graves CF where the lead time is equal to the average processing time, with no queuing delays at all, we obtain the line $TH = W/p$, where p denotes the average processing time. Assuming that average lead time is equal to the average processing time up to the maximum

output level, it gives the "Best Case" model $TH = min\{W/p, C\}$ described in Chap. 7 of Hopp and Spearman (2001). However, by linking production rate to WIP level, a linear CF differs from the fixed delays used in most LP models, where the output of a production process is simply the input shifted forward in time by the fixed lead time. Orcun et al. (2006) illustrate the differences between these models using system dynamics simulations. For most capacitated production resources subject to congestion, limited capacity leads to a saturating (concave) CF. It is also apparent from the Fig. 3.1 that the CF always lies below the $TH = W/p$ and $TH = C$ lines.

An important issue in using CF models is the question of how long the planning periods should be. If the CFs are derived using steady-state queuing models, the planning period must be long enough that the queues representing the production system can be at least approximately in steady state. Given the long-term, aggregate purpose of this type of model, as outlined in the introduction and the discussion in Allison et al. (1997), the planning buckets are likely to be long enough (e.g., a month) that most production systems with relatively short raw processing times should reach steady state. However, even if this is not the case, current research is exploring means of deriving CFs for systems under transient regimes (for example, Selcuk 2007 and Missbauer 2009), showing that even under transient conditions the concave shape of the CF is maintained.

A number of authors have suggested empirical approaches to estimating CFs, where a functional form with the desired properties is postulated, and then fit to data obtained either from an industrial facility or a simulation model using some form of regression analysis. Karmarkar (1989) suggests a CF of the form

$$X_t = \frac{K_1 W_t}{K_2 + W_t} \tag{3.3}$$

where X_t denotes the output in period t, W_t the WIP at the resource at the start of period t, and K_1 the maximum possible output of the resource in period t. The shape parameter K_2 is estimated by the user. Selçuk et al. (2007) demonstrate the derivation of K_2 for an $M/G/1$ system with bulk arrivals. Srinivasan et al. (1988) suggest an alternative functional form

$$f(W_t) = K_1(1 - e^{kW_t}) \tag{3.4}$$

where k is again a user-estimated shape parameter. Asmundsson et al. (2009) use this latter functional form and give an extensive discussion of various issues in collecting simulation data for fitting this type of CF. Asmundsson et al. (2006) use a visual fit of linear segments to simulation data to develop a CF formulation for a scaled-down semiconductor wafer fabrication facility with unreliable equipment and reentrant flows. Kacar and Uzsoy (2010) and Kacar et al. (2010) use a linear regression approach applied to data collected from a simulation model, with good results. Asmundsson et al. (2009) show that an empirically fitted CF can give good results even under a transient regime. The implication for this research is that it is possible to represent the behavior of a production system with an appropriately fitted

CF. Thus we shall proceed with our model on the assumption this can be done and examine the potential impact on profits of using a model with fixed lead time that does not consider queuing behavior.

A Single Product Dynamic Joint Price-Production Model Incorporating Congestion

We now present a joint price-production model that incorporates CFs and lead-time-dependent demand. We assume a single firm that behaves as a monopolist. The firm sees a linear demand function $D = g(P, L) = Max\{0, M\text{-}aP\text{-}bL\}$, where $a, b \geq 1$ are the price and lead-time sensitivities of demand D with respect to price P and lead time L, respectively. Changes in market conditions are represented by changes in these sensitivities. The intercept M of the demand function represents the maximum possible demand, i.e., the market size.

In a given period t, the firm quotes a price P_t and a delivery time L_t to customers. We assume that the firm quotes a delivery time for orders received in a period equal to the average manufacturing lead time at the start of the period. Since the manufacturing lead time (delivery time) depends on the number of orders waiting, the firm can control the *maximum* delivery time by limiting the number of orders to be processed (per Fig. 3.1). In effect, the firm quotes the delivery time based on the minimum of two values: the average manufacturing lead time, and a guaranteed delivery time L_G by which all orders need to be satisfied, or customers will not place orders. Hence an order received in period t has to be fulfilled by period $t + L_G$.

The firm needs to align its production system with this market preference by mapping L_G on Fig. 3.1 and quoting an average delivery time below the value of L_G. This will, in turn, determine the number of orders that a firm may accept and hold in queue for processing, yielding a target production rate and a target utilization. Thus, the higher the guaranteed delivery time allowed by the market, the higher the utilization at which the firm can operate its resources. From Fig. 3.1, as utilization increases, a large increase in threshold value L_G will allow only a small increase in utilization, since lead time increases rapidly with additional workload at high utilization levels. This guaranteed delivery time assumption is similar to that used by Selcuk et al. (2007) and Spitter et al. (2005a; b) in their supply chain operations planning (SCOP) models, where they assume a planned manufacturing lead time within which an order must finish processing. The idea of a quoted lead time in combination with a maximum lead time is also used by Dellaert (1991) and Duenyas and Hopp (1995) in their models of due-date management with order selection.

Another mechanism by which a firm may control quoted average delivery time is to quote a higher price and thus accept fewer orders. This is possible due to the monopolist assumption and the price and lead-time-dependent nature of demand. Customers may be willing to pay a premium for lower-quoted average delivery times and the relative magnitude of this premium would depend upon their sensitivity to delivery time represented by parameter b_t.

The average delivery time quotation implies that some orders will be ready for delivery earlier than promised. The customer may not always want to take delivery early, in which case the manufacturer has to hold finished goods inventory. We assume that the customer will allow a limited number of orders to be delivered early in the planning horizon and represent this by a parameter ν. Late deliveries are not allowed, though this can be incorporated in a straightforward manner. To further simplify the model, we restrict every order to have a size of one unit. This enables us to eliminate constraints that would otherwise be included to track fulfillments of orders of varying sizes. We define the following notation:

Variables

R_t Order release quantity in period t
W_t WIP inventory at the end of period t
X_t Production quantity in period t
I_t Finished goods inventory (FGI) at end of period t
P_t Price in period t
D_t Sales quantity in period t
Y_t Quantity shipped in period t

Parameters

a_t Price sensitivity of demand in period t
b_t Lead-time sensitivity of demand in period t
h_t Holding cost of finished goods inventory per unit in period t
ω_t Holding cost of WIP inventory per unit in period t
ϕ_t Unit production cost in period t
c_t Order release cost per unit released in period t
ν Maximum units allowed to be shipped before due date over the horizon
K_1 Theoretical maximum production capacity
K_2 Curvature parameter of CF
M Intercept of demand function, i.e., demand when price = lead time = 0
T Length of planning horizon, $t = 1,..,T$
L_G Guaranteed delivery time (in periods)
$f(.)$ CF

Let $\hat{W}_t$ be the estimated average WIP level in a period t. We use the CF form suggested by Karmarkar (1989). From (3.3) we have

$$f\left(\hat{W}_t\right) = \frac{K_1 \hat{W}_t}{K_2 + \hat{W}_t}$$

The production X_t in period t is bounded by the CF in that period.

As mentioned earlier, the demand in period t is expressed by the demand function $D_t = M - a_t P_t - b_t L_t$. By Little's Law, the expected lead time in period t is given

by $L_t = \hat{W}_t / X_t$, expressed in units of periods. By invoking Little's Law we assume that the production system is in steady state within the planning period. Thus, the demand observed in period t is given by

$$D_t = M - a_t P_t - b_t \left(\frac{\hat{W}_t}{X_t} \right)$$

Our CF-based joint price-production planning model is now as follows:

CF model

$$Max \sum_{t=1}^{T} \left[P_t \left(M - a_t P_t - b_t \left(\frac{\hat{W}_t}{X_t} \right) \right) - c_t R_t - \phi_t X_t - h_t I_t - \omega_t W_t \right] \quad (3.5)$$

s.t.

$$\{\lambda_t\} \quad W_t = W_{t-1} - X_t + R_t \qquad \forall t \quad (3.6)$$

$$\{\pi_t\} \quad I_t = I_{t-1} + X_t - Y_t \qquad \forall t \quad (3.7)$$

$$\{\theta_t\} \quad X_t \leq \frac{K_1 \hat{W}_t}{K_2 + \hat{W}_t} \qquad \forall t \quad (3.8)$$

$$\{\mu_t\} \quad M - a_t P_t - b_t \left(\frac{\hat{W}_t}{X_t} \right) \geq 0 \qquad \forall t \quad (3.9)$$

$$\{\sigma_t\} \quad \sum_{\tau=1}^{t} Y_\tau \geq \sum_{\tau=1}^{t-L_G} \left[M - a_\tau p_\tau - b_\tau \left(\frac{\hat{W}_\tau}{X_\tau} \right) \right] \qquad \forall t \quad (3.10)$$

$$\{\rho_t\} \quad \sum_{\tau=1}^{t} Y_\tau \geq \sum_{\tau=1}^{t-L_G} \left[M - a_\tau p_\tau - b_\tau \left(\frac{\hat{W}_\tau}{X_\tau} \right) \right] + \nu \qquad \forall t \quad (3.11)$$

$$\{\chi_t\} \quad \hat{W}_t \leq \frac{1}{2}(W_{t-1} + W_t) \qquad \forall t \quad (3.12)$$

$$P_t, X_t, I_t, W_t, R_t, Y_t, \hat{W}_t \geq 0 \qquad \forall t \quad (3.13)$$

The objective is to maximize total contribution, expressed as the difference between the total revenue in each period and variable operating costs. Equations (3.6) and (3.7) are WIP and finished goods inventory balance constraints. Equation (3.8) represents production capacity using the CF, and constraint (3.9) defines the sales quantity. Constraint (3.10) requires that all orders be shipped within the planned delivery time, but allows orders to be shipped earlier than due, rather than being held as finished goods inventory. Since the customer may impose a limit on the number of orders shipped early over the horizon (given by the parameter ν), we model this preference

in constraint (3.11). We estimate the average WIP level $\hat{W}_t$ within a given period using the WIP levels at the end points of the period using (3.12). All variables are required to be non-negative by (3.13). The Greek letters in curly brackets to the left of each constraint denote its associated Lagrange multipliers. We do not impose a cost on shipping since it would require setting values for another parameter, which we avoid for sake of parsimony in the experimental design. For the same reason, we do not impose a penalty if the average delivery time quotation exceeds the planned delivery time. Instead we reduce sales through our time-dependent demand function. This mechanism is further discussed in the section "Experiments Without Early Delivery Flexibility: $\nu = 0$.

For comparison purposes we now state a joint price-production planning model that assumes a fixed delivery time $L \leq L_G$ which is specified as an exogenous parameter, and hence is denoted as the Fixed Lead Time (FLT) model. The demand observed by this model in period t is expressed as $D_t = M - a_t P_t - b_t L$. The firm must set $L \leq L_G$ to avoid exceeding the target utilization. We assign a Lagrange multiplier for each constraint as was done for the CF model.

FLT model

$$Max \sum_{t=1}^{T} \left[\hat{P}_t \left(M - a_t \hat{P} t - b_t L \right) - c_t \hat{X}_t - h_t \hat{I}_t \right] \tag{3.14}$$

s.t.

$$\{\gamma_t\} \qquad \hat{I}_t = \hat{I}_{t-1} + \hat{X}_{t-L} - \hat{Y}_t \qquad \forall t \tag{3.15}$$

$$\{\delta_t\} \qquad \hat{X}_t \leq K_1 \qquad \forall t \tag{3.16}$$

$$\{\hat{\mu}_t\} \qquad M - a_t \hat{P}_t - L \geq 0 \qquad \forall t \tag{3.17}$$

$$\{\hat{\sigma}_t\} \qquad \sum_{\tau=1}^{t} \hat{Y}_\tau \geq \sum_{\tau=1}^{t-L} (M - a_\tau \hat{P}_\tau - b_\tau L) \qquad \forall t \tag{3.18}$$

$$\{\hat{\rho}_t\} \qquad \sum_{\tau=1}^{t} \hat{Y}_\tau \geq \sum_{\tau=1}^{t-L} (M - a_\tau \hat{P}_\tau - b_\tau L) + \nu \qquad \forall t \tag{3.19}$$

$$\hat{X}_t, \hat{P}_t, \hat{Y}_t, \hat{I}_t \geq 0 \qquad \forall t \tag{3.20}$$

We use the variable $\hat{X}_t$ to denote production initiated in period t. Since there is a fixed production lead time L, production initiated in period t is available to be shipped in period $t + L$. This variable corresponds to the releases variable R_t from the CF model. Hence we incorporate a time lag L in the inventory balance constraint (3.15). Since the FLT model ignores the buildup of queues in the system due to its fixed lead-time assumption, it does not have any WIP variables or WIP balance constraints. This model is consistent with FLT production planning models (Johnson

and Montgomery 1974; Hackman and Leachman 1989; Spitter et al. 2005a; Spitter et al. 2005a) or the joint price-production model of Swann (2001).

In our numerical experiments we use a modified version of the FLT model that facilitates direct comparisons with the CF model. Recall that $\hat{X}_t$ models the material released in period t so that it finishes processing and is available for shipping in period $t+L$. This definition, while capturing the nature of fixed lead times, does not allow a direct comparison between the two models. Hence, we replace the variable $\hat{X}_t$ with two variables: $\hat{R}_t$ to denote the material release in period t, and $\hat{X'}_t$, the actual production in period t. The two variables are related by the expression $\hat{R}_t = \hat{X'}_{t+L}$. Hence, the $\hat{R}_t$ units of work released in period t will remain in WIP for L time periods, which we explicitly include in the objective function. The modified FLT model is thus as follows:

$$\sum_{t=1}^{T}\left[\hat{P}_t\left(M - a_t\hat{P}_t - b_t L\right) - c_t\hat{R}_t - \phi_t\hat{X}'_t - h_t\hat{I}_t - \omega_t\sum_{j=t-L+1}^{t}\hat{R}_j\right] \tag{3.21}$$

s.t.

$$\hat{X}'_{t+L} = \hat{R}'_t \qquad \forall t$$

$$\hat{I}_t = \hat{I}_{t-1} + \hat{X}'_t - \hat{Y}_t \qquad \forall t \tag{3.22}$$

$$\hat{X}'_t \le K_1 \qquad \forall t \tag{3.23}$$

$$M - a_t\hat{P}_t - b_t L \ge 0 \qquad \forall t \tag{3.24}$$

$$\sum_{\tau=1}^{t}\hat{Y}_t \ge \sum_{\tau=1}^{t-L}(M - a_t\hat{P}_t - b_t L) \qquad \forall t \tag{3.25}$$

$$\sum_{\tau=1}^{t}\hat{Y}_t \le \sum_{\tau=1}^{t-L}(M - a_t\hat{P}_t - b_t L) + \nu \qquad \forall t \tag{3.26}$$

$$\hat{X}'_t, \hat{P}_t, \hat{I}_t, \hat{Y}_t, \hat{R}_t \ge 0 \qquad \forall t \tag{3.27}$$

In the following section we examine the structure of locally optimal solutions to both the FLT and CF models to explore the differences between them, induced by the different models of production capacity they use.

Model Analysis

In Appendix 3.1, we show that the revenue function of the FLT model is concave, resulting in a concave objective function. Further, the linear demand function results in constraints (3.15)–(3.19) being linear. Thus the FLT model aims to maximize a concave function over a convex constraint set, so a locally optimal solution is also

globally optimal. The CF model has a quasi-concave objective function if the sales variable is positive and the capacity constraint is tight (see Appendix 3.2). However, satisfying the capacity constraint at equality causes the constraint set to lose convexity and become concave. Hence the CF model does not have a unique global optimum. Nevertheless, all local optima should satisfy the Karush-Kuhn-Tucker (KKT) optimality conditions and since the global optimum must also be a local optimum, structural properties derived for a local optimum are valid for a global optimum.

We begin by examining the relationships between price, demand, lead time and capacity using the KKT conditions for a local optimum (Appendix 3.3). We then examine the relationship between the ending FGI and the delivery flexibility parameter ν and discuss properties of some Lagrange multipliers used in the formulations.

Sales, Price and Delivery Time at Optimality

We first develop expressions for price and sales quantity based on the KKT conditions for a local optimum. We are interested in local optima with non-trivial solutions, i.e., the firm operates in a reasonable manner that yields non-zero revenue, or in other words, both price and sales are non-zero. Using $P_t > 0$ in (3.46) and (3.61) we obtain the optimal prices for both models as follows:

Price(*FLT model*)

$$\hat{P}_t = \frac{M}{2a_t} - \frac{b_t}{2a_t}L - \frac{1}{2}\left(\hat{\mu}_t - \sum_{\tau=t+L}^{T}\hat{\sigma}_\tau + \sum_{\tau=t+L}^{T}\hat{\rho}_\tau\right) \tag{3.28}$$

Price(*CF model*)

$$P_t = \frac{M}{2a_t} - \frac{b_t}{2a_t}\left(\frac{\hat{W}_t}{X_t}\right) - \frac{1}{2}\left(\mu_t - \sum_{\tau=t+L_G}^{T}\sigma_\tau + \sum_{\tau=t+L_G}^{T}\rho_\tau\right) \tag{3.29}$$

Substituting these expressions into the demand functions for the respective models, we obtain the following expressions for the sales quantities:

Sales (*FLT model*)

$$\hat{D}_t = \frac{M}{2} - \frac{b_t}{2} - \frac{a_t}{2}\left(\sum_{\tau=t+L}^{T}\hat{\sigma}_t - \sum_{\tau=t+L}^{T}\hat{\rho}_t - \hat{\mu}_t\right) \tag{3.30}$$

Sales(*CF model*)

$$D_t = \frac{M}{2} - \frac{b_t}{2}\left(\frac{\hat{W}_t}{X_t}\right) - \frac{a_t}{2}\left(\sum_{\tau=t+L_G}^{T}\sigma_t - \sum_{\tau=t+L_G}^{T}\rho_t - \mu_t\right) \tag{3.31}$$

Equations (3.28)–(3.31) clearly show that under the CF model both price and sales decisions are dependent upon the observed lead time. Equation (3.29) is particularly interesting since price is expressed as a downward sloping function of lead time using the basic decision variables of the production system. Ray and Boyaci (2004) assume price to be a downward sloping function of lead time in order to investigate the effects of ignoring lead-time sensitivity of prices while making pricing decisions. However, our model does not require such an assumption, since the relationship between price and lead time emerges directly from the model. The last terms in all four expressions represent the interactions between the cumulative shipment constraints and can be interpreted in terms of the Lagrange multiplier of the finished goods inventory balance constraints of the respective models. We discuss this in the section "Properties of Lagrange Multipliers".

It is interesting to examine the behavior of the model as lead times approach the threshold delivery time L_G. We can write

$$\left(\frac{\hat{W}_t}{X_t}\right) = L_G - \left(L_G - \left(\frac{\hat{W}_t}{X_t}\right)\right) = L_G - \Delta L$$

which, in turn, allows us to rewrite (3.29) and (3.31) as:

$$P_t = \frac{M}{2a_t} - \frac{b_t}{2a_t}L_G + \frac{b_t}{2a_t}\Delta L - \frac{1}{2}\left(\mu_t - \sum_{\tau=t+L_G}^{T}\sigma_\tau + \sum_{\tau=t+L_G}^{T}\rho_\tau\right) \tag{3.32}$$

$$D_t = \frac{M}{2} - \frac{b_t}{2}L_G + \frac{b_t}{2}\Delta L - \frac{a_t}{2}\left(\sum_{\tau=t+L_G}^{T}\sigma_\tau - \sum_{\tau=t+L_G}^{T}\rho_\tau - \mu_t\right) \tag{3.33}$$

The ΔL term represents the difference between the maximum allowable delivery time and the average delivery time quotation (i.e., the average manufacturing lead time). When $\Delta L < 0$, i.e., the average delivery time quotation exceeds the maximum allowable delivery time, our model penalizes the firm by reducing demand per (3.33), thus reducing the WIP in the production system and hence the average lead time. This self-regulating behavior removes the need to include explicit penalty terms for exceeding the delivery time guarantee in the objective function of model CF.

This behavior can be visualized by examining relationships between different variables by means of a causal loop diagram (Sterman 2000) in Fig. 3.2. The variable at the tail of an arc is linked to the variable at the head of the arc by the sign on the head that indicates whether an increase in the variable at the tail causes a corresponding increase or a decrease in the variable at the head. Average delivery times eventually have a negative feedback on sales, since an increase in sales will cause an increase in quoted average delivery times, which in turn will reduce ΔL, making it negative. Negative values of ΔL cause a reduction in sales, keeping the average delivery time and sales variables in close relation with each other. Recall that these two variables are tightly coupled with the price variable through the demand function. Hence a reduction in both sales and average delivery time would require an increase in price.

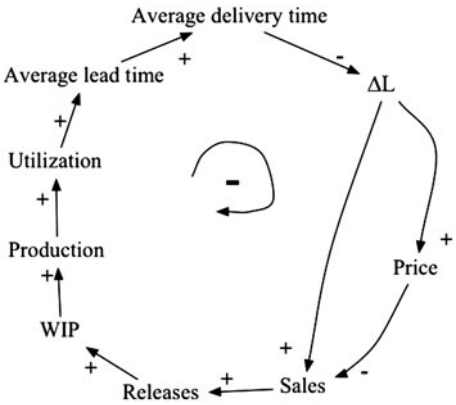


Fig. 3.2 Relationship between variables

Thus the firm can use both price and delivery time in an aggregate planning framework to manage sales. Webster (2002) proposes a similar feedback loop in his model for determining equilibrium values for price and lead time in face of a changing demand function. His model changes capacity in response to a change in sales, keeping lead time fixed, but does not consider the costs of changing capacity.

Prices and Utilization

If $\Delta L > 0$, the quoted average delivery time is less than the guaranteed delivery time. From (3.33), it would appear that this would cause sales to increase, increasing resource utilization and delivery times. Recall that a target guaranteed delivery time corresponds to a particular utilization level. However, it is not possible to determine a priori whether a targeted utilization level will allow satisfying the guaranteed delivery time, since we only have information about average delivery times and the maximum realized delivery time may exceed the guaranteed delivery time at high utilizations. Attaining higher utilization levels in the CF model requires additional WIP, and the marginal increase in utilization decreases with each unit increase in WIP. The CF model uses this information to determine a utilization level that may be lower than the target utilization level corresponding to the guaranteed delivery time parameter. This decision is implemented by limiting sales (to control utilization) by increasing prices instead ($\Delta L > 0$ in (3.32)). At high values of L_G, ΔL could be significant enough that a large difference may exist between the prices quoted by the two models.

Analytically, this behavior can be explained as follows. Let u_t denote the utilization level in period t due to production level X_t, i.e., $X_t = u_t K_1$. Using this relationship we obtain $L_t = \frac{\hat{W}_t}{K_1 u_t}$. Initially it appears counterintuitive that lead time decreases with increasing utilization. However, this expression must be viewed in the light of the relationship between WIP W_t and utilization u_t. As shown in Fig. 3.1, a unit increase in utilization will cause WIP, and thus lead times, to increase by a

larger amount. Conversely, if utilization threatens to increase, the CF model can lower WIP, thus controlling it and keeping average delivery time in check. The FLT model is blind to the effects of utilization on delivery time and thus has one less lever for managing sales and system operation. The impact of this can be observed by simulating the key decisions of the FLT model in a congestion-prone system, which we do in the section "Low Utilization: $u = 0.8$, $L = 1$".

The key decisions made by a joint pricing and production planning model are prices and material releases. For the FLT model, these correspond to the variables $\hat{P}_t$ and $\hat{X}_t$ respectively. If we define $\hat{W}_t^{'}$ to be the WIP level arising from the material release $\hat{X}_t$, then the production in a given period is found to be $f(\hat{W}_t^{'})$ from the CF. Thus, the average delivery time can be written as $\frac{\hat{W}_t^{'}}{f(\hat{W}_t^{'})}$. Now let $D_t^{'}$ be the sales decision arising from this average delivery time quotation and price quotation $\hat{P}_t$. Then we have

$$D_t^{'} = M - a_t \hat{P}_t - b_t \left(\frac{\hat{W}_t^{'}}{f(\hat{W}_t^{'})} \right) \tag{3.34}$$

Further, if $D_t{>}0$ and $\hat{W}_t^{'} > 0$, then $\mu_t = 0$ and $\hat{\mu}_t = 0$ from complementary slackness conditions (3.52) and (3.65). Using this in (3.28) and (3.29), we can express the difference in prices as

$$\begin{aligned} P_t - \hat{P}_t &= \frac{b_t}{2a_t} \left(L - \frac{\hat{W}_t}{X_t} \right) + \sum_{\tau=t+L_G}^{T} (\sigma_t - \hat{\sigma}_t) - \sum_{\tau=t+L_G}^{T} (\rho_t - \hat{\rho}_t) \\ &= \frac{b_t}{2a_t} \Delta L^{'} + \sum_{\tau=t+L_G}^{T} (\sigma_t - \hat{\sigma}_t) - \sum_{\tau=t+L_G}^{T} (\rho_t - \hat{\rho}_t) \end{aligned}$$

At high values of L (corresponding to high utilization), it is possible that $\Delta L^{'}{>}0$ and is large enough for $\hat{P}_t$ to be significantly less than P_t. In this scenario, we have $D_t^{'} > D_t$. Further, since $\Delta L^{'} > 0$, the material release decisions $\hat{X}_t$ could load the system with significantly higher WIP than the release decisions R_t made by the CF model, leading to larger queue sizes. Average quoted lead times will not be met and there will not be enough FGI to satisfy sales $D_t^{'}$. If we allow unsatisfied sales to be lost, revenues will drop since the prices quoted are lower than those in the CF model. We discuss this further through a numerical example in the section "Experiments with Early Delivery Flexibility: $\nu > 0$".

On the other hand, at low values of L (corresponding to low utilization), $\Delta L^{'}$ will be small and will have less influence over the differences in prices quoted by the two models. From Fig. 3.1, it can be seen that lower utilizations imply lower WIP levels and hence lower average delivery times. Further, the marginal increase in throughput with a unit increase in WIP is higher at low utilization than at high utilization. This allows the production system to fulfill demand in a timely manner more easily. Lower utilizations are achieved by having low sales or ample excess capacity. Neither of these alternatives is practical in a capital-intensive environment, motivating our interest in high-utilization environments.

Properties of Lagrange Multipliers

Proposition 1 In an optimal solution to the CF model the capacity constraint is always tight (i.e., $\theta_t > 0$) if $\hat{W}_t$, R_t, P_t and D_t are all strictly positive.

Proof See Appendix 3.4.

Capacity in the CF model is expressed in terms of the amount of WIP in the system that can be cleared in a given period. The above proposition implies that the release pattern will be coordinated with the sales pattern, so that there is just enough WIP to create the capacity required for fulfilling sales.

Proposition 2 In an optimal solution to the CF model, the marginal cost of holding finished goods inventory is always positive (i.e., $\pi_t > 0$) if $\theta_t > 0, X_t > 0$, and $\hat{W}_t > 0$.

Proof See Appendix 3.5.

Proposition 3 In an optimal solution to the FLT model, the marginal cost of holding finished goods inventory is always positive (i.e., $\gamma_t > 0$).

Proof See Appendix 3.6.

Since we quote an average delivery time, production that is realized earlier than due can be held as finished goods inventory to fulfill orders by the guaranteed delivery time. This can be clarified further when the marginal FGI costs are seen in relationship to the shipment Y_t. Considering the FLT model, if shipments $\hat{Y}_t > 0$, for some period t, then from condition (3.63), we have

$$\gamma_t - \sum_{\tau=t}^{T} \hat{\sigma}_\tau + \sum_{\tau=t}^{T} \hat{\rho}_\tau = 0$$

$$\Leftrightarrow - \sum_{\tau=t}^{T} \hat{\sigma}_\tau + \sum_{\tau=t}^{T} \hat{\rho}_\tau = -\gamma_t$$

If $Y_{t+L_G} > 0$, we have $- \sum_{\tau=t}^{T} \hat{\sigma}_\tau + \sum_{\tau=t}^{T} \hat{\rho}_\tau = -\gamma_{t+L_G}$

Since $\hat{P}_t > 0$, from condition (3.61), we have

$$-M + a_t \hat{P}_t + b_t L_G + a_t \left(\hat{P}_t + \hat{\mu}_t - \sum_{\tau=t+L_G}^{T} \hat{\sigma}_\tau + \sum_{\tau=t+L_G}^{T} \hat{\rho}_\tau \right) = 0$$

when $\hat{D}_t > 0, \hat{\mu}_t = 0$. Hence $-M + a_t \hat{P}_t + b_t L_G + a_t(\hat{P}_t - \gamma_{t+L_G}) = 0$. Rewriting, we obtain

$\gamma_{t+L_G} = 2\hat{P}_t - \frac{M}{a_t} + \frac{b_t}{a_t} L_G > 0$ by Proposition 3. Thus when there are positive sales in period t, it is beneficial to have FGI in period $t + L_G$ in order to meet the

quoted delivery date. The analogous expression for the CF model is

$$\pi_{t+L_G} = 2P_t - \frac{M}{a_t} + \frac{b_t}{a_t}\left(\frac{\hat{W}_t}{X_t}\right).$$

Since D_t, P_t, and the average quoted delivery time for the CF model are strictly positive (section "CF model"), we find that the marginal cost of the FGI constraint in period $t+L$ is strictly positive by a simple manipulation of the demand function. In addition, the marginal cost of the FGI constraint in the CF model varies with both the price and the average delivery time, whereas for the FLT model it can vary only with price.

We also investigate the optimal sales decision made by our CF model if the linear demand function is replaced by the power function used by So and Song (1998). The demand function itself is expressed as $D_t = M P_t^{-a_t} L_t^{-b_t}$, where all symbols have the same meaning as before. By repeating the steps in Appendix 3.3 and the section "CF model", we obtain the optimal sales decision as:

$$\ln D_t = \ln M - a_t \ln\left[\left(\frac{a_t}{1-a_t}\right)\left(\sum_{\tau=t+L_G}^{T} \rho_\tau - \sum_{\tau=t+L_G}^{T} \sigma_\tau\right)\right] - b_t \ln\left(\frac{\hat{W}_t}{X_t}\right) \quad (3.35)$$

We find that the negative feedback loop discussed in the section "Sales, Price, and Delivery Time at Optimality" for the linear demand function also holds for the power demand function, though on a logarithmic scale. We conjecture that the negative feedback relationship between sales and average delivery times would exist in case of any demand function form that is downward sloping in delivery time. We now present a numerical study to compare the behavior of the CF and FLT models.

Numerical Study

The length of the planning horizon is chosen to be 24 periods where each period corresponds to a month. The price and lead time sensitivities for each period are presented in Table 3.1. Price sensitivity is low in the first half of the horizon and increases in the latter half. This change in sensitivity represents a typical scenario in semiconductor products where, as other manufacturers bring competing devices to market, the price for the device will begin to decrease significantly (Akcali et al. 2000; Leachman and Ding 2007). Lead-time sensitivity, on the other hand, is low in the first and third quarters, and high in the second and fourth quarters of the horizon. The high-sensitivity periods represent seasonal effects where the market is unwilling to wait for a longer time interval between placing the order and taking delivery of the product.

Values of other input parameters are given in Table 3.2. The value of the curvature parameter K_2 is selected such that the slope of the CF at $\hat{W}_t = 0$ does not exceed the reciprocal of the raw process time.

Table 3.1 Price and lead-time sensitivities

Period range	Price sensitivity (a_t)	Lead-time sensitivity (b_t)
1–6	1	1
7–12	1	2
13–18	2	1
19–24	2	2

The relationship between utilization u and fixed lead time L for chosen values of the CF parameters K_1 and K_2 is obtained as seen in Appendix 3.7. We chose the value for guaranteed lead time as $L_G = L + 1$ periods. Thus sales will only be lost if the realized lead time exceeds the planned lead time L by more than one period. We consider four combinations of unit costs given in Table 3.3. Combination 1 is the base case. Combination 2 allows comparison of objective function values when unit material cost is less than the unit production cost. Combination 3 allows for a similar comparison when WIP holding cost is less than the FGI holding cost. To facilitate direct comparison of the objective function values, we use the modified FLT (MFLT) model that considered WIP costs in the objective function instead of the original FLT model used for the analytical results.

We assume there is no residual demand from earlier planning periods to be met in the current planning horizon. Both CF and modified FLT models are initialized with WIP equal to the targeted production in period 1, i.e., $W_0 = uK_1$. We also require that ending WIP in periods 23 and 24 for both CF and FLT model equals uK_1. WIP inventory in the FLT model at the end of a period is the sum of the releases in the previous L periods; we impose this boundary condition on the FLT model by controlling the material releases. By imposing these boundary conditions, we aim to avoid ramp-up and end-effects that would normally influence behavior at the beginning and end of the horizon.

Table 3.2 Input parameter values

Length of planning horizon	T	24 periods
Theoretical production capacity per period	K_1	500 units
Curvature parameter	K_2	100
Demand at zero price and zero lead time	M	1,000 units
Early delivery flexibility	ν	0 units
Fixed lead time L and corresponding target utilizations u	L	1 period ($u = 0.8$), 2 periods ($u = 0.9$), 4 periods ($u = 0.95$)
Initial WIP for CF and FLT models ($u = 0.8$)	W_o	400
WIP at ending of period 23 for CF and FLT models ($u = 0.8$)	W_{23}	400
WIP at ending of period 24 for CF and FLT models ($u = 0.8$)	W_{24}	400
Initial WIP for CF and FLT models ($u = 0.9$)	W_o	900
WIP at ending of period 23 for CF and FLT models ($u = 0.9$)	W_{23}	900
WIP at ending of period 24 for CF and FLT models ($u = 0.9$)	W_{24}	900
Initial WIP for CF and FLT models ($u = 0.95$)	W_o	1,900
WIP at ending of period 23 for CF and FLT models ($u = 0.95$)	W_{23}	1,900
WIP at ending of period 24 for CF and FLT models ($u = 0.95$)	W_{24}	1,900

Table 3.3 Unit cost combinations

Combination	Unit material cost	Unit production cost	Unit WIP holding cost	Unit FGI holding cost
1	1/unit	1/unit	1/unit/period	1/unit/period
2	0.5/unit	1/unit	1/unit/period	1/unit/period
3	1/unit	1/unit	0.5/unit/period	1/unit/period
4	0.5/unit	1/unit	0.125/unit/period	0.25/unit/period

Both models are solved using the CONOPT solver in the general algebraic modeling system (GAMS) optimization suite (www.gams.com). Since this solver does not guarantee a globally optimal solution for the nonconvex CF model, we used six different starting points for both models, and found that for both models all initial starting points led to the same values for the objective function and decision variables.

The primary question of interest is how important it is to consider the effects of congestion explicitly—do they lead to significant differences in profit, and, if so under what conditions? One way to approach this issue is to examine how much profit-planned solutions from the FLT model would yield if the production system is subject to the type of congestion represented in the CF model. In other words, how much profit is lost if we plan using a fixed lead time when our production system is, in reality, subject to congestion as represented by a CF?

In order to examine this question, we simulate the behavior of the production system period by period using expressions (3.6) and (3.7). The material releases obtained from the MFLT model are used to determine the WIP level in each period, and the CF is used to determine the production at this WIP level. The WIP level in a period can be calculated using the ending WIP in the previous period and the release in the current period using expressions derived in Appendix 3.8. We estimate the average delivery time resulting from the WIP level as $L_t = \hat{W}_t / X_t$. These provide estimates of the realized production, WIP and finished goods inventory available when the system operates as represented by the CF, allowing us to obtain actual shipments. Projected sales in each period under the price quoted by the FLT model are given by the linear demand function, allowing us to calculate the revenue that would be realized if the production system were able to produce exactly the quantities planned by the FLT model in each period. We assume sales are lost if not enough finished goods are available for order fulfillment. The realized shipments are multiplied by the quoted price to give the realized revenue in each period. We deduct the material, production and inventory costs incurred due to the release and sales decisions to obtain the realized profit for both models. We impose no boundary conditions on the system during this simulation.

Experiments without Early Delivery Flexibility: $\nu = 0$ In our base case we use the unit costs described by Combination 1 to study the behavior of each model with no early delivery flexibility, i.e., $\nu = 0$. We will discuss the results for each planned utilization level u, and hence each planned lead time L, separately. In all figures, the captions "FLT," and "CF" denote the quantity computed by the respective optimization models. "Realized FLT" denotes the quantities that are realized when the plans computed from the FLT model are implemented in a system that is subject to congestion as represented by the CF.

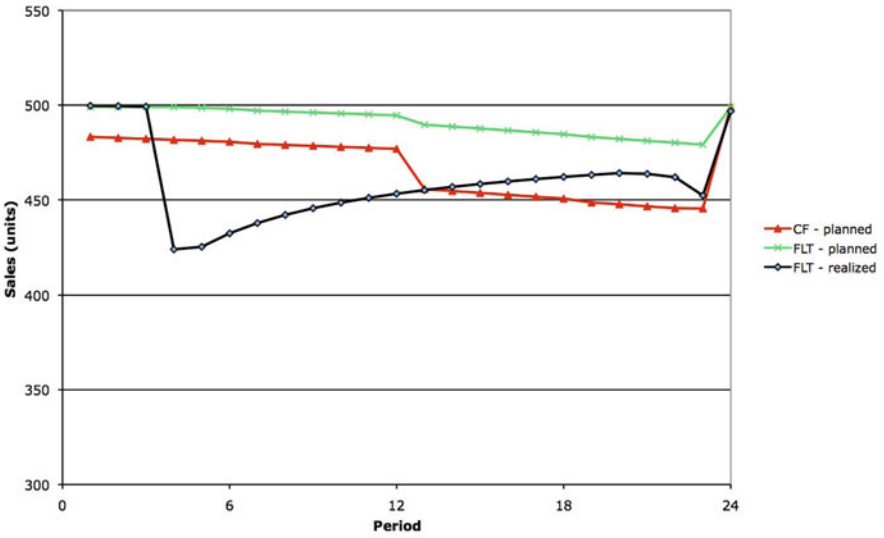


Fig. 3.3 Sales comparison at $u = 0.8$

Low utilization: $u = 0.8$, $L = 1$ The results of this experiment are summarized in Figs. 3.3, 3.4 and 3.5. As seen in Fig. 3.4, the CF model consistently sets prices somewhat higher than MFLT model, but not by a great deal. Both models reduce prices in the second half of the planning horizon when the market becomes sensitive to price. However, Fig. 3.3 shows that the FLT model realizes substantially lower sales than the CF model in the later periods. Examination of Figs. 3.5 and 3.6, which show the planned lead times and FGI levels, explains the situation. The CF model plans to operate at a higher utilization level with longer lead times from the start of the horizon. It must meet demands within the maximum lead time L_G, but accomplishes this by building finished goods inventory early in the horizon which it draws down over time, allowing the model to meet demand within the specified maximum lead time L_G that the market will bear. As a result of this approach and the slightly higher prices it sets, the planned sales of the CF model are lower than those of the FLT model.

However, Fig. 3.6 shows that the finished goods inventory realized when the material releases and prices from the FLT plan are implemented in the presence of congestion is very different from that planned. The FLT model assumes that any demand that does not exceed the capacity of the system can be met within the planned lead time $L = 1$, allowing it to set lower prices than CF. However, the low prices and low-quoted lead time lead to high demand, which the congested system cannot meet within the planned lead time L_G. This results in a stock out in periods 9 through 11 where there is no available product to ship and sales are lost. The net result, seen in Fig. 3.7, is an approximately 20 % difference in planned and realized revenue for the FLT model in periods 11 through 20.

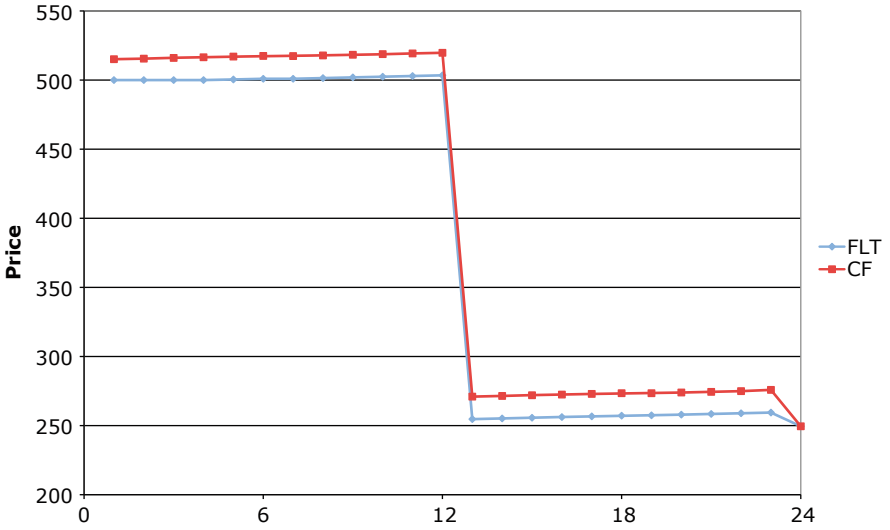


Fig. 3.4 Pricing comparison at $u = 0.8$

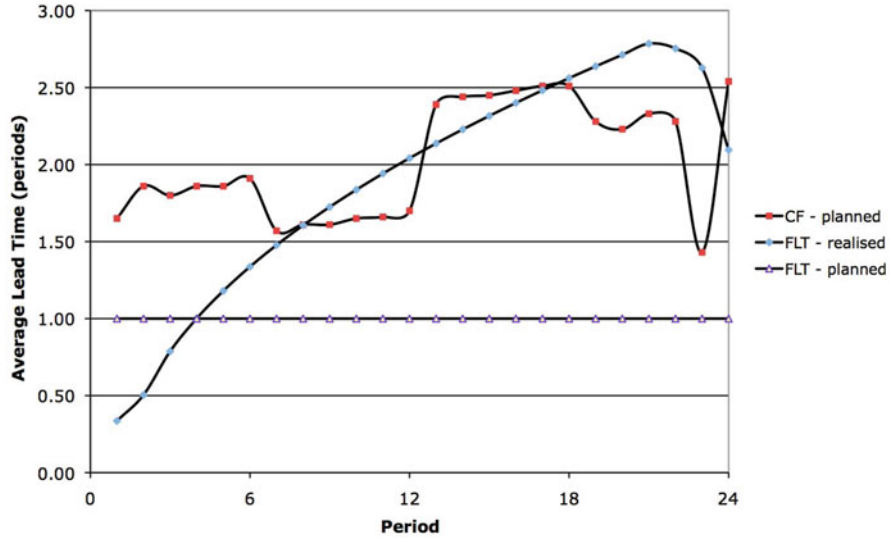


Fig. 3.5 Planned and realized average lead times at $u = 0.8$

Figure 3.7 shows that both models plan to achieve very similar revenue, but the CF is able to achieve its aim while the FLT model is not. The difference is almost entirely due to the FLT model's assumption that the fixed lead time L can be maintained regardless of utilization. The FLT model loads the system to its available theoretical capacity, which results in utilization levels incompatible with the maximum lead time

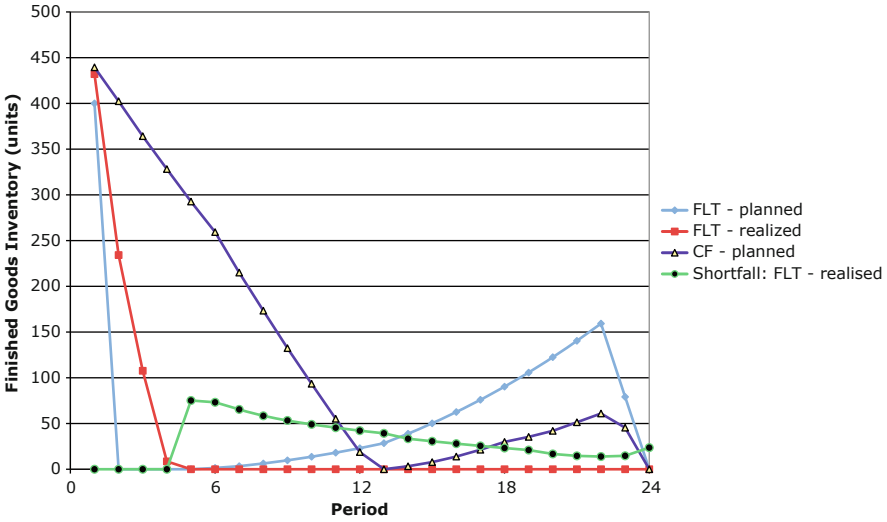


Fig. 3.6 Finished goods inventory levels for $u = 0.8$

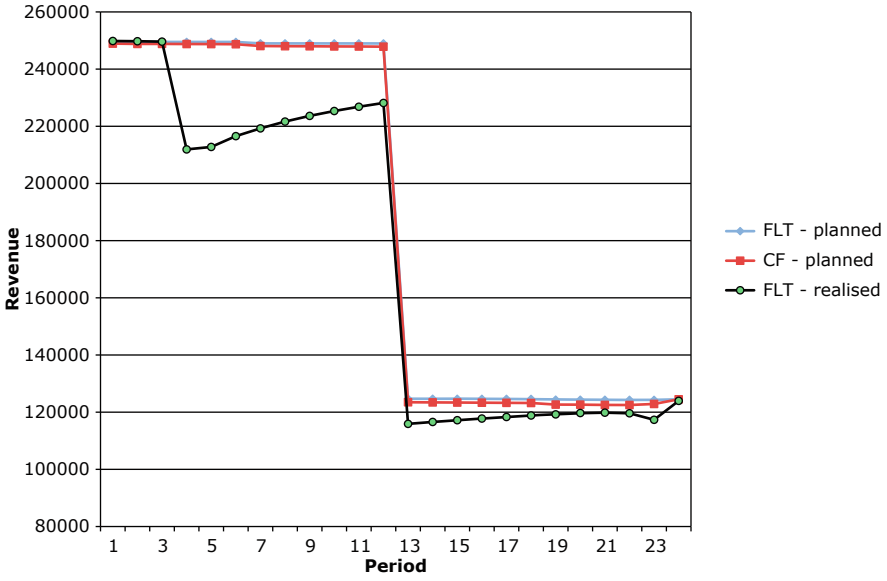


Fig. 3.7 Revenue comparison for u = 0.8

L_G. It is also interesting that this significant difference in behavior occurs despite low demand sensitivity to both prices and lead times.

The results of this experiment highlight what we believe is the principal reason for an FLT model to perform poorly in an environment subject to congestion. The basic

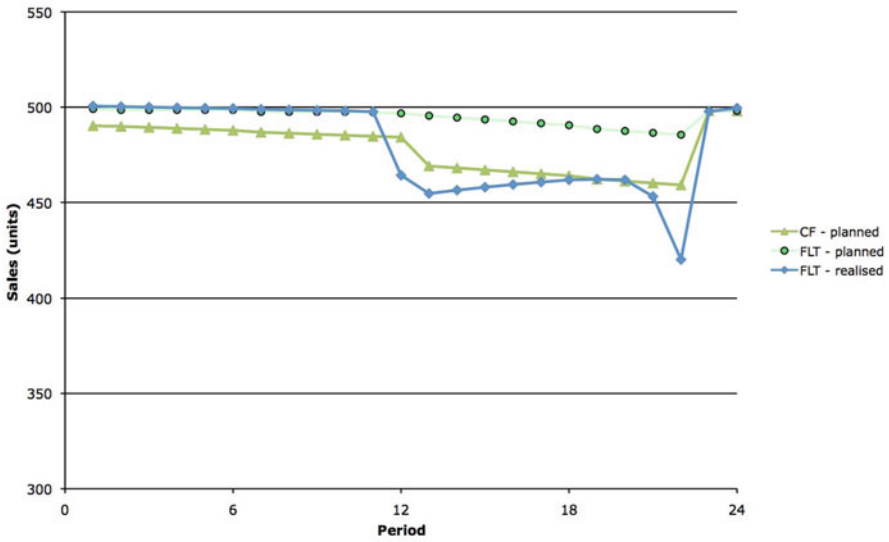


Fig. 3.8 Sales comparison for $u = 0.9$

issue is that the planning model fails to represent accurately the realized behavior of the production system, which is manifested in the realized lead time. Figure 3.5 clearly indicates that the planned lead time L is a gross underestimate of the realized lead time that becomes worse as the planning horizon advances.

Intermediate utilization: $u = 0.9$, $L = 2$, $L_G = 3$ The results of this experiment are given in Figs. 3.8–3.11. In this situation the difference between the two models is rather less than might be expected, although the behavior of the inventory and lead times differs somewhat between the models. This is because the maximum lead time L_G is consistent with a high level of utilization. The FLT model again loads the system to its capacity, resulting in lead times higher than L_G, but because L_G is already quite high the impact on predicted lead times is not as severe as at the lower utilization level. The CF model, on the other hand, varies lead times over the horizon, keeping them below L_G. Hence in this case both models plan very similar total revenues and both achieve them, although with quite different production plans. The reason both are able to achieve their plans to a large extent is the low sensitivity of demand to prices and lead times.

High utilization level: $u = 0.95$, $L = 4$, $L_G = 5$ In this case, again the difference between the two models is closer than before (Figs. 3.12–3.14). The primary reason for this is the high WIP level imposed at the beginning of the horizon for both models. Both models behave similarly, choosing not to make any releases into the system in the first few periods and consuming the initial WIP. This allows lead times to be low for both the CF and the realized FLT decisions in this initial part of the horizon. The FLT model again loads the system to capacity in the following periods, due to which the realized lead times gradually rise over the horizon. This is the only case

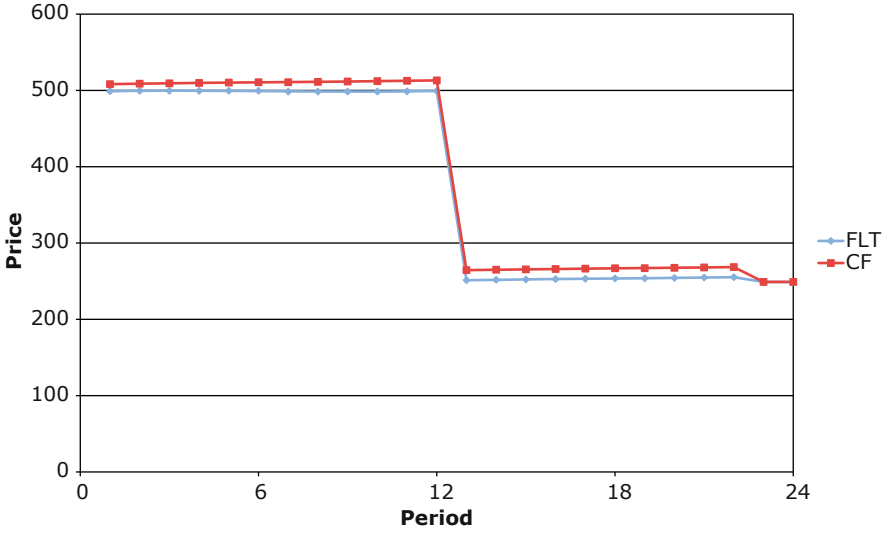


Fig. 3.9 Price comparison for $u = 0.9$

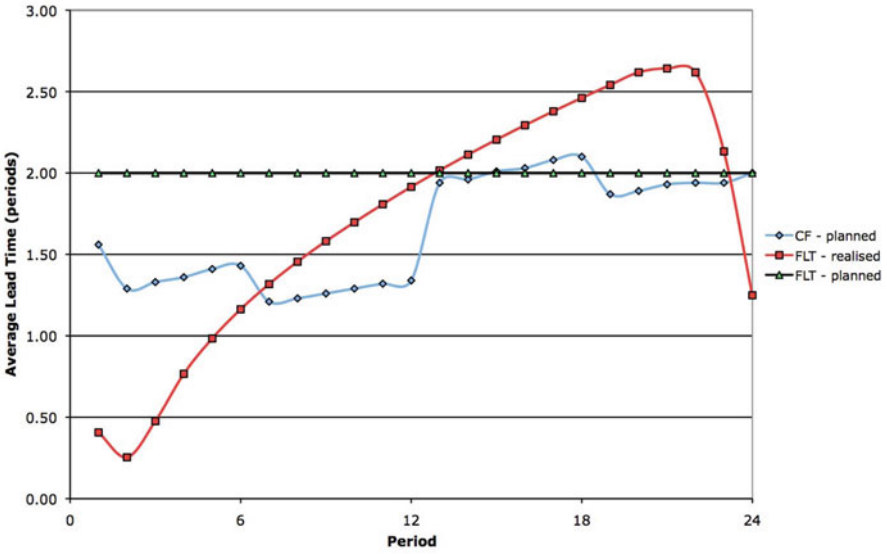


Fig. 3.10 Lead time comparison for $u = 0.9$

where we impose the ending WIP conditions on the simulated decisions of the FLT model, because otherwise the ending WIP does not rise to a value high enough to satisfy the ending conditions. This is again due to the fact that there are no releases early in the horizon, resulting in low WIP levels that do not rise fast enough during

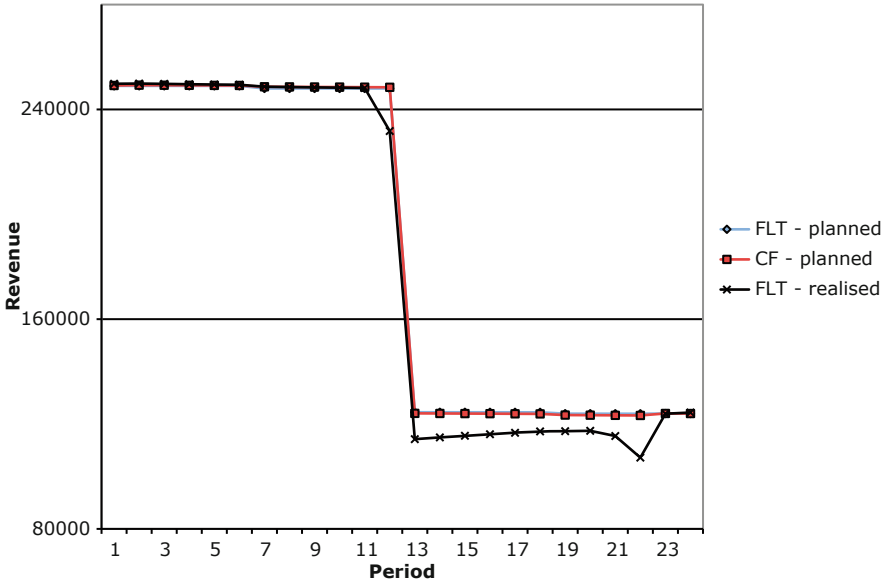


Fig. 3.11 Revenue comparison for $u = 0.9$

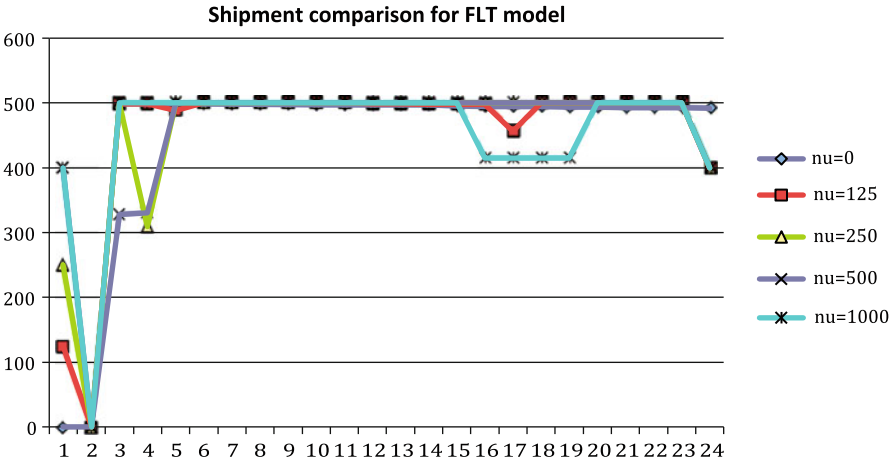


Fig. 3.12 Shipment decision comparisons for FLT model with changing v

the course of the horizon. The detailed results of this experiment are omitted for the sake of brevity.

Objective function values The discussion to this point has demonstrated that the production and pricing plans developed by the CF and FLT models result in quite different plans over the planning horizon. When the planned lead time substantially

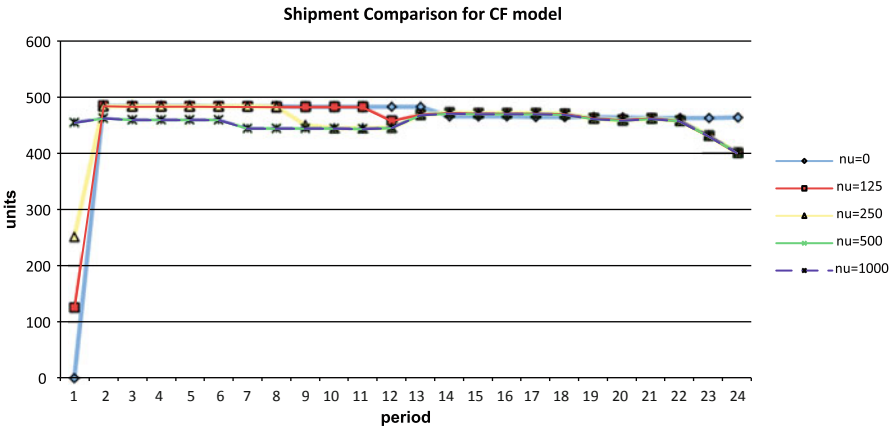


Fig. 3.13 Shipment decision comparisons for CF model with changing ν

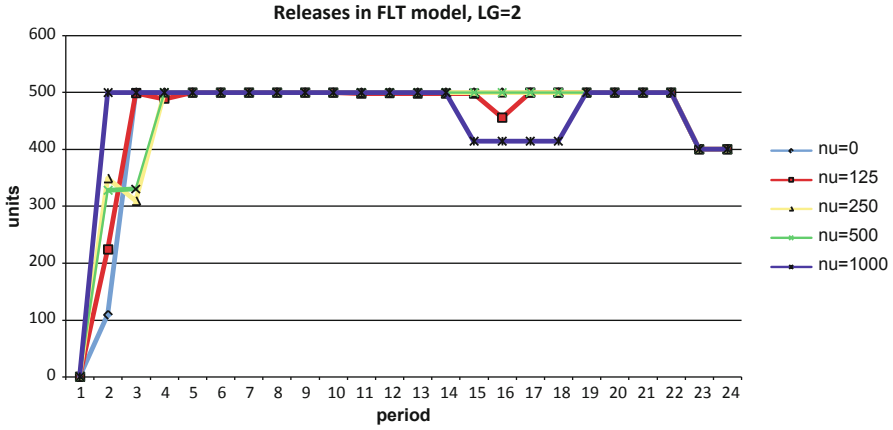


Fig. 3.14 Release decision comparisons for FLT model with changing ν

underestimates the manufacturing lead time that can actually be realized, severe discrepancies between the plans and actual deliveries to customers can result, as was the case for our experiment with $u = 0.8$. Table 3.4 presents a comparison of the realized objective function values planned by the FLT model, and those realized when the FLT plans are simulated in the presence of congestion. All quantities are expressed as a ratio to the objective function value obtained by the CF model for the same cost combination.

When $u = 0.8$, we find that even though the FLT model predicts an objective function value higher than the CF model, the realized objective (Fixed Lead Time-Simulated FLT-SIM) is significantly lower than both FLT and CF models. This is

Table 3.4 Comparison of planned and realized objective function values

(u, L, L_G)	Cost scenario (c_t, f_t, h_t, w_t)	FLT	FLT-SIM
(0.8, 1,2)	(1,1,1,1)	1.065	0.930
	(0.5, 1,1,1)	1.065	0.930
	(1,1,0.5,1)	1.064	0.930
	(0.5,1,0.125,0.25)	1.062	0.930
(0.9, 2,3)	(1,1,1,1)	0.999	0.978
	(0.5, 1,1,1)	0.999	0.978
	(1,1,0.5,1)	1.000	0.978
	(0.5,1,0.125,0.25)	0.999	0.977
(0.95, 4, 5)	(1,1,1,1)	0.987	0.993
	(0.5, 1,1,1)	0.988	0.993
	(1,1,0.5,1)	0.990	0.993
	(0.5,1,0.125,0.25)	0.991	0.994

due to the release decisions proposed by the FLT model that result in high WIP levels, high lead times and product shortages, all of which lead to lower revenue and profit margin. When the discrepancy between planned and realized lead times is less severe, when $u = 0.9$, the same effect is observed although at a much smaller level. When $u = 0.95$, the FLT-SIM is very slightly higher than the planned FLT objective, because the realized lead time is shorter than the planned lead time for most of the planning horizon. It is notable that the CF model gives the highest objective function value in all scenarios considered, most markedly when the discrepancy between planned and realized FLT lead time is most severe.

Experiments with Early Delivery Flexibility: $\nu > 0$ The combinations with early delivery flexibility ν provide more interesting insights. Early delivery flexibility allows both plans to shift production away from periods with high-delivery-time sensitivity to those with low-delivery-time sensitivity without the need to carry all the production as finished goods inventory. This should result in an increase in profit margins as flexibility increases, due to reduction in cost of carrying finished goods inventory. The two models use this flexibility differently. The shipment decisions made by the FLT model for different values of ν and cost Combination 4 when $L_G = 2$ are seen in Fig. 3.12 and those for the CF model in Fig. 3.13. The FLT model applies all of its flexibility in the early part of the horizon, choosing to make zero shipments in period 2. The model also chooses to increase the load in the system as ν increases by releasing more orders, which has a detrimental effect on the profit margin when its decisions are subjected to congestion.

Table 3.5 summarizes the planned and realized objective function values of the models, again using the objective function value of the CF model as a base. We observe that the profit margin for FLT-SIM decreases as ν increases from 125 to 500. The CF model also uses its flexibility early on for lower values of ν, but for $\nu = 500$ and 1,000, it spreads this flexibility over the horizon. It is interesting to note that the realized objective function value FLT-SIM first decreases and then increases with ν, suggesting that the choice of an optimal value for ν may improve the realized performance of the FLT model. However, it is again striking that the CF model produces higher objective function values consistently across all scenarios.

Table 3.5 Objective function comparison for experiments with early delivery

(u, L, L_G)	n	FLT	FLT-SIM
(0.8, 1,2)	0	1.062	0.930
	125	1.062	0.938
	250	1.062	0.935
	500	1.062	0.934
	1,000	1.062	0.962
(0.9, 2,3)	0	0.999	0.977
	125	0.999	0.984
	250	0.999	0.977
	500	0.998	0.993
	1,000	0.998	0.997
(0.95, 4, 5)	0	0.991	0.994
	125	0.991	0.996
	250	0.990	0.995
	500	0.990	0.995
	1,000	0.990	0.995

Conclusions and Future Directions

In this chapter we have used the concept of CFs developed in the production planning literature to develop an integrated model for jointly planning production and pricing over time for a manufacturing firm whose resources are subject to congestion. Our analytical results show that the interplay between lead times and prices in the demand function requires careful consideration of the implications of pricing decisions for lead times. Pricing decisions made under a naïve capacity model that assumes any level of demand up to the theoretical capacity of the system can be met within a fixed lead time independent of workload have the potential to lead to significant difficulties when low prices and optimistic lead time estimates lead to the system being unable to meet demand within a reasonable time, causing lost sales and possibly loss of customer goodwill. It is interesting that noticeable effects can be observed even when the demand is not very sensitive to prices or lead times.

The critical issue is the difference between the lead times assumed in the planning model and the realized lead times. A FLT model may perform satisfactorily in terms of achieving its planned revenue even at high utilization if the planned lead time is set consistently with the realized utilization levels and remains within the maximum lead time the market will bear. However, such a model will have difficulties when lead times are underestimated or when sensitivity to lead times changes abruptly, since it has no ability to modulate the lead times quoted based on system state and market sensitivities. It is also noteworthy that the CF model consistently sets higher prices than the FLT model, which upon reflection is intuitive; the price set by the CF model considers the costs incurred due to congestion such as WIP accumulation, whereas the FLT model does not. When solved at an aggregate level considering product families and planning horizons of 18–24 months, the models can be solved sufficiently rapidly to permit extensive what-if analysis to provide decision makers with intuition as to the likely results of their decisions.

The CF model is, as far as we are aware, the first model to integrate dynamic pricing and production planning over time in a manner that represents the effects of congestion due to queuing. Most queuing-based models provide steady-state results, while most prior models that plan prices and production over time have adopted conventional models of capacity that do not capture the effect of workload on lead times, and do not permit the joint manipulation of lead times and prices to maximize profit.

These results highlight the importance of a well-designed and functional manufacturing-marketing interface for firms operating in markets where price and lead-time sensitivity may change over time. The problem is aggravated by the fact that lead times are generally the responsibility of the supply chain organization, while pricing is determined by sales and marketing groups. A common solution to this issue we have observed in industry, and which has been advocated by a number of authors (Graves 1986; de Kok and Fransoo 2003) is to simplify the situation by requiring the supply chain organization to maintain a constant lead time which is agreeable to marketing. However, in capital intensive industries where resources must be run at high utilization for the firm to be profitable, small changes in utilization make maintaining a constant lead time a very challenging task. The CF model proposed here in fact addresses exactly this—modulating prices and releases to optimize profit within the constraints of the lead time imposed by the market's "reservation" lead time L_G. In addition, the ability to change both prices and lead times in response to changing market sensitivity may result in higher revenues and profits relative to using price as the only control variable.

Several directions for future work present themselves. A natural direction is the extension of the models developed in this chapter to environments with multiple product families that may serve quite different markets but share capacity. Many semiconductor wafer fabs operating as foundries produce circuits for quite different markets, such as controllers and communication devices, in the same fab using largely the same technology and equipment. Another natural extension is to embed these models in a multistage stochastic programming framework where scenarios would consider different price sensitivities for different products over time. This model presents a number of challenges due to the rapid growth of the scenario tree, but may still be practical for the aggregate models of the type suggested in this chapter, and considered by Allison et al. (1997).

Appendix 3.1: Concavity of Revenue Function for the FLT Model

Our revenue function has the form $R = PD = P(M - aP - bL) = MP - aP^2 - bLP$. Thus there is only one variable, P. Taking the second derivative of the revenue function w.r.t P, we obtain

$$\frac{d^2R}{dP^2} = -2a \leq 0$$

Since we assume $a \geq 1$,we have $-2a < 0$. Hence the revenue function is strictly concave.

Appendix 3.2: Nature of Demand and Revenue Function for the CF Model

The demand function has the form. $D_t = M - a_t P_t - b_t \left(\frac{\hat{W}_t}{X_t}\right)$. Under very reasonable conditions (see Proposition 3), we can show that the capacity constraint is tight, i.e., $X_t = f(\hat{W}_t) = \frac{K_1 \hat{W}_t}{K_2 + \hat{W}_t}$. Under these conditions, the demand function takes the form $D_t = M - a_t P_t - \frac{b_t}{K_1}(K_2 + \hat{W}_t)$. Thus, the demand function has a linear form. The CF constraint (3.8) is convex, hence the constraint set is also convex.

Then, revenue function has the form $g_t = P_t D_t = P_t(M - a_t P_t - \frac{b_t}{K_1}(K_2 + \hat{W}_t))$. We have:

$$\frac{\partial g_t}{\partial \hat{W}_t} = -\frac{b_t}{K_1} P_t; \frac{\partial^2 g_t}{\partial \hat{W}_t^2} = 0$$

$$\frac{\partial g_t}{\partial P_t} = M - 2a_t P_t - \frac{b_t}{K_1}(K_2 + \hat{W}_t); \frac{\partial^2 g_t}{\partial {P_t}^2} = -2a_t$$

$$\frac{\partial^2 g_t}{\partial P_t \partial \hat{W}_t} = -\frac{b_t}{K_1}$$

In order to have quasi-concavity, we require that

$$-\left(\frac{\partial g_t}{\partial \hat{W}_t}\right)^2 = -\left(-\frac{b_t}{K_1} P_t\right)^2 = -\frac{{b_t}^2}{K_1^2} P_t^2 \tag{3.36}$$

and

$$2\frac{\partial^2 g_t}{\partial \hat{W}_t \partial P_t} \cdot \frac{\partial g_t}{\partial \hat{W}_t} \cdot \frac{\partial g_t}{\partial P_t} - \frac{\partial^2 g_t}{\partial \hat{W}_t^2}\left(\frac{\partial g_t}{\partial \hat{W}_t}\right)^2 - \frac{\partial^2 g_t}{\partial {P^2}_t}\left(\frac{\partial g_t}{\partial \hat{W}_t}\right) > 0$$

After some algebra the expression above reduces to $2\frac{b_t^2}{K_1^2} P_t\left(M - a_t P_t - \frac{b_t}{K_1}(K_2 + \hat{W}_t)\right)$. If sales $D_t = M - a_t P_t - \frac{b_t}{K_1}(K_2 + \hat{W}_t)) 0$, it is clear that this expression is nonnegative. Thus from (3.36) and this expression we conclude that the revenue function is quasi-concave.

Appendix 3.3: KKT Conditions for CF and FLT Models

KKT conditions for CF model The Lagrangian for this planning model is as below:

$$L = -\sum_{t=1}^{T}\left[MP_t - a_t P^2 - b_t P_t\left(\frac{\hat{W}_t}{X_t}\right) - c_t R_t - \phi_t X_t - h_t I_t - \omega_t W_t\right]$$

$$+\sum_{t=1}^{T}\lambda_t(W_t - W_{t-1} + X_t - R_t) + \sum_{t=1}^{T}\pi_t(I_t - I_{t-1} - X_t + Y_t)$$
$$+\sum_{t=1}^{T}\theta_t(K_2X_t + X_t\hat{W}_t) + \sum_{t=1}^{T}\mu_t\left(-M + a_tP_t + b_t\left(\frac{\hat{W}_t}{X_t}\right)\right)$$
$$+\sum_{t=1}^{T}\pi_t\left(-\sum_{\tau=1}^{t}Y_\tau + \sum_{\tau=1}^{\tau=t-L_G}\left(-M + a_\tau P_\tau + b_\tau\left(\frac{\hat{W}_\tau}{X_\tau}\right)\right)\right)$$
$$+\sum_{t=1}^{T}\rho_t\left(\sum_{\tau=1}^{t}Y_\tau - \sum_{\tau=1}^{\tau=t-L_G}\left(-M + a_\tau P_\tau + b_\tau\left(\frac{\hat{W}_\tau}{X_\tau}\right)\right) - \upsilon\right)$$
$$+\sum_{t=1}^{T}\chi_t\left(\hat{W}_t - \frac{1}{2}(W_{t-1} + W_t)\right)$$

The first order optimality conditions and complementary slackness conditions follow:

First Order Optimality Conditions

$$\frac{\partial L}{\partial I_t} = h_t + \pi_t - \pi_{t-1} \geq 0$$
$$\frac{\partial L}{\partial I_t} = h_T + \pi_T \tag{3.37}$$

$$I_t\frac{\partial L}{\partial I_t} = I_T(h_T + \pi_t - \pi_{t-1}) = 0$$
$$I_t\frac{\partial L}{\partial I_t} = I_T(h_T + \pi_T) \tag{3.38}$$

$$\frac{\partial L}{\partial R_t} = c_T - \lambda_t \geq 0 \tag{3.39}$$

$$R_t\frac{\partial L}{\partial R_t} = R_T(c_T - \lambda_T) = 0 \tag{3.40}$$

$$\frac{\partial L}{\partial W_t} = \omega_t + \lambda_t - \lambda_{t+1} - \frac{1}{2}(\chi_t + \chi_{t+1}) \geq 0 \tag{3.41}$$

$$W_t\frac{\partial L}{\partial W_t} = 0 \tag{3.42}$$

$$\frac{\partial L}{\partial X_t} = \phi_t + \lambda_t - \pi_t + \theta_t(K_2 + \hat{W}_t)$$
$$- \frac{b_t\hat{W}_t}{X_t^2}\left(P_t + \mu_t - \sum_{\tau=t+L_G}^{T}\sigma_\tau + \sum_{\tau=t+L_G}^{T}\rho_\tau\right) \tag{3.43}$$

$$X_t \frac{\partial L}{\partial X_t} = 0 \tag{3.44}$$

$$\frac{\partial L}{\partial P_t} = -M + a_t P_t + b_t \left(\frac{\hat{W}_t}{X_t}\right) + a_t \left(P_t + \mu_t - \sum_{\tau=t+L_G}^{T} \sigma_\tau + \sum_{\tau=t+L_G}^{T} \rho_\tau\right) \geq 0 \tag{3.45}$$

$$P_t \frac{\partial L}{\partial P_t} = 0 \tag{3.46}$$

$$\frac{\partial L}{\partial Y_t} = \pi_t - \sum_{\tau=t}^{T} \sigma_\tau + \sum_{\tau=t}^{T} \rho_\tau \geq 0 \tag{3.47}$$

$$Y_t \frac{\partial L}{\partial Y_t} = 0 \tag{3.48}$$

$$\frac{\partial L}{\partial \hat{W}_t} = \theta_t (X_t - K_1) + \chi_t + \frac{b_t}{X_t} \left(P_t + \mu_t - \sum_{\tau=t+L_G}^{T} \sigma_\tau + \sum_{\tau=t+L_G}^{T} \rho_\tau\right) \geq 0 \tag{3.49}$$

$$\hat{W}_t \frac{\partial L}{\partial \hat{W}_t} = 0 \tag{3.50}$$

Complementary Slackness Conditions

$$\theta_t (K_2 X_t + X_t \hat{W}_t - K_t \hat{W}_t) \tag{3.51}$$

$$\mu_t \left(-M + a_t P_t + b_t \left(\frac{\hat{W}_t}{X_t}\right)\right) = 0 \tag{3.52}$$

$$\sigma_t \left(-\sum_{\tau=1}^{t} Y_\tau + \sum_{\tau=1}^{t-L_G} \left(M - a_\tau P_\tau - b_\tau \left(\frac{\hat{W}_\tau}{X_\tau}\right)\right)\right) = 0 \tag{3.53}$$

$$\rho_t \left(\sum_{\tau=1}^{t} Y_\tau - \sum_{\tau=1}^{t-LG} \left(M - a_\tau P_\tau - b_\tau \left(\frac{\hat{W}_\tau}{X_\tau}\right)\right) - v\right) = 0 \tag{3.54}$$

$$\chi_t \left(\hat{W}_t - \frac{1}{2}(W_{t-1} + W_t)\right) = 0 \tag{3.55}$$

Nonnegativity conditions λ_t, π_t: unrestricted, $\theta_t, \mu_t, \sigma_t, \rho_t, \chi_t \geq 0$

KKT conditions for FLT model The Lagrangian for this planning model is as below:

$$L = -\sum_{\tau=1}^{t} \left[M \hat{P}_t - a_t \hat{P}_t^2 - b_t \hat{P}_t L_G - c_t \hat{X}_t - h_t \hat{I}_t\right] + \sum_{\tau=1}^{t} \gamma_t \left(\hat{I}_t - \hat{I}_{t-1} - \hat{X}_{t-L_G} + \hat{Y}_t\right)$$

$$+\sum_{t=1}^{T}\delta_t(\hat{X}_t - K_1) + \sum_{t=1}^{T}\hat{\mu}_t(-M + a_t\hat{P}_t + b_t L_G)$$
$$+\sum_{t=1}^{T}\hat{\sigma}_t\left(-\sum_{\tau=1}^{t}\hat{Y}_t + \sum_{T=1}^{t-L_G}(M - a_\tau\hat{P}_\tau - b_\tau L_G)\right)$$
$$+\sum_{t=1}^{T}\hat{\rho}_t\left(\sum_{\tau=1}^{t}\hat{Y}_\tau - \sum_{T=1}^{t-LG}(M - a_\tau\hat{P}_\tau - b_\tau L_G) - v\right)$$

The first order optimality conditions and complementary slackness conditions follow.

First Order Optimality Conditions

$$\frac{\partial L}{\partial \hat{I}_t} = h_t + \gamma_t - \gamma_{t+1} \geq 0 \tag{3.56}$$

$$\hat{I}t\frac{\partial L}{\partial \hat{I}_t} = 0 \tag{3.57}$$

$$\frac{\partial L}{\partial \hat{X}_t} = c_t - \gamma_{t+L_G} + \delta_t \geq 0 \tag{3.58}$$

$$\hat{X}_t\frac{\partial L}{\partial \hat{X}_t} = 0 \tag{3.59}$$

$$\frac{\partial L}{\partial \hat{P}_t} = -M + a_t\hat{P}_t + b_t L + a_t\left(\hat{P}_t + \hat{\mu}_t - \sum_{\tau=t+L}^{T}\hat{\sigma}_\tau + \sum_{\tau=t+L}^{T}\hat{\sigma}_\tau\right) \geq 0 \tag{3.60}$$

$$\hat{P}_t\frac{\partial L}{\partial \hat{P}_t} = 0 \tag{3.61}$$

$$\frac{\partial L}{\partial Y_t} = \gamma_t - \sum_{l=t}^{T}\hat{\sigma}_l + \sum_{l=t}^{T}\hat{P}_l \geq 0 \tag{3.62}$$

$$\hat{Y}_t\frac{\partial L}{\partial \hat{Y}_t} = 0 \tag{3.63}$$

Complementary Slackness Conditions

$$\delta_t(\hat{X}_t - K) = 0 \tag{3.64}$$

$$\hat{\mu}_t(-M + a_t\hat{P}_t + b_t L_G) = 0 \tag{3.65}$$

$$\hat{P}(\sum_{l=1}^{t}\hat{Y}_l + \sum_{l=1}^{t-L_G}(M - a_l\hat{P}_l - b_l L_G)) = 0 \tag{3.66}$$

$$\hat{P}_t(\sum_{l=1}^{t}\hat{Y}_l - \sum_{l=1}^{t-L_G}(M - a_l\hat{P} - b_l L_G) - v) = 0 \tag{3.67}$$

Nonnegativity conditions γ_t: unrestricted, $\delta_t,\ \hat{\mu}_t,\ \hat{\sigma}_t,\ \hat{\rho}_t \geq 0$

Appendix 3.4: Proof of Proposition 1

From equation (3.50), we have $\hat{W}_t \frac{\partial L}{\partial \hat{W}_t} = 0$.

From $\hat{W}_t > 0$, we have

$$\frac{\partial L}{\partial \hat{W}_t} = \theta_t(X_t - K_1) + \chi_t + \frac{b_t}{X_t}\left(P_t + \mu_t - \sum_{\tau=t+L_G}^{T} \sigma_\tau + \sum_{\tau=t+L_G}^{T} \rho_\tau\right) = 0$$

$$\Rightarrow \theta_t = \frac{1}{(K_1 - X_t)}\left(\chi_t + \frac{b_t}{X_t}\left(P_t + \mu_t - \sum_{\tau=t+L_G}^{T} \sigma_\tau + \sum_{\tau=t+L_G}^{T} \rho_\tau\right)\right) \tag{3.68}$$

From equation (3.46), we have $P_t \frac{\partial L}{\partial P_t} = 0$.

From $P_t > 0$,

$$\frac{\partial L}{\partial P_t} = -M + a_t P_t + b_t\left(\frac{\hat{W}_t}{X_t}\right) + a_t\left(P_t + \mu_t - \sum_{\tau=t+L_G}^{T} \sigma_\tau + \sum_{\tau=t+L_G}^{T} \rho_\tau\right) = 0$$

$$\Rightarrow \left(P_t + \mu_t - \sum_{\tau=t+L_G}^{T} \sigma_\tau + \sum_{\tau=t+L_G}^{T} \rho_\tau\right) = \frac{1}{a_t}\left(M - a_t P_t - b_t\left(\frac{\hat{W}_t}{X_t}\right)\right) \geq 0$$

Since sales $D_t > 0$, we have

$$\frac{1}{a_t}\left(M - a_t P_t - b_t\left(\frac{\hat{W}_t}{X_t}\right)\right) > 0 \tag{3.69}$$

For the CF model, we have $K_1 > X_t$. This statement can be inferred from Fig. 3.1, where K_1 refers to the theoretical capacity indicated by the "fixed capacity" line and X_t is the output of the concave CF. Using this fact and Eq. (3.69) in (3.68), we have $\theta_t > 0$. From complementary slackness condition (3.51), if $\theta_t > 0$, $K_2 X_t + X_t \hat{W}_t - K_1 \hat{W}_t = 0$, implying that the capacity constraint is tight. *QED*.

Appendix 3.5: Proof of Proposition 2

From equation (3.44), we have $X_t \frac{\partial L}{\partial X_t} = 0$.

From $X_t > 0$, we have

$$\frac{\partial L}{\partial X_t} = \phi_t + \lambda_t - \pi_t + \theta_t(K_2 + \hat{W}_t) - \frac{b_t \hat{W}_t}{X_t^2}\left(P_t + \mu_t - \sum_{\tau=t+L_G}^{T} \sigma_\tau + \sum_{\tau=t+L_G}^{T} \rho_\tau\right) = 0$$

yielding

$$\pi_t = \phi_t + \lambda_t + \theta_t(K_2 + \hat{W}_t) - \frac{b_t \hat{W}_t}{X_t^2}\left(P_t + \mu_t - \sum_{\tau=t+L_G}^{T} \sigma_\tau + \sum_{\tau=t+L_G}^{T} \rho_\tau\right).$$

From equation (3.40), we have $R_t\,(c_t - \lambda_t) = 0$. From $R_t > 0$, we have $\lambda_t = c_t$. Thus

$$\pi_t = c_t + \theta_t(K_2 + \hat{W}_t) - \frac{b_t \hat{W}_t}{X_t^2}\left(P_t + \mu_t - \sum_{\tau=t+L_G}^{T} \sigma_\tau + \sum_{\tau=t+L_G}^{T} \rho_\tau\right).$$

From (3.68), we have

$$\theta_t = \frac{1}{(K_1 - X_t)}\left(\chi_t + \frac{b_t}{X_t}\left(P_t + \mu_t - \sum_{\tau=t+L_G}^{T} \sigma_\tau + \sum_{\tau=t+L_G}^{T} \rho_\tau\right)\right)$$

$$\Rightarrow \frac{b_t}{X_t}\left(P_t + \mu_t - \sum_{\tau=t+L_G}^{T} \sigma_\tau + \sum_{\tau=t+L_G}^{T} \rho_\tau\right) = \theta_t(K_1 - X_t) - \chi_t$$

Using this relation in the expression for π_t, we have

$$\pi_t = \phi_t + c_t + \theta_t(K_2 + \hat{W}_t) + \frac{\hat{W}_t}{X_t}(\chi_t - \theta_t(K_1 - X_t)) \tag{3.70}$$

Since $\theta_t > 0$, we have $K_2 X_t + X_t \hat{W}_t - K_1 \hat{W}_t = 0$ from complementary slackness condition (3.51). Rewriting, we obtain $K_2 + \hat{W}_t = \frac{K_1 \hat{W}_t}{X_t}$. Using this in (3.70), we have

$$\pi_t = \phi_t + c_t + \frac{\theta_t K_1 \hat{W}_t}{X_t} + \frac{\hat{W}_t \chi_t}{X_t} - \frac{\theta_t K_1 \hat{W}_t}{X_t} + \hat{W}_t \theta_t = \phi_t + c_t + \frac{\hat{W}_t \chi_t}{X_t} + \hat{W}_t \theta_t > 0$$

QED.

Appendix 3.6: Proof of Proposition 3

We have two cases. Shipments can be made from the production quantity in the current period or from ending inventory from the previous period.
Case 1: $\hat{X}_t > 0$
In this case, sales are fulfilled from production in that period.

From equation (3.59),

$$\hat{X}_{t-L_G} \frac{\partial L}{\partial \hat{X}_{t-L_G}} = \hat{X}_{t-L_G}(c_{t-L_G} - \gamma_t + \delta_{t-L_G}) = 0$$

Since $\hat{X}_t > 0$ the expression in brackets equals zero. Rearranging the terms in the bracket, we obtain $\gamma_t = c_{t-L_G} + \delta_{t-L_G} > 0$, which implies that $\gamma_t > 0$ for any period with positive production.

Case 2: $\hat{I}_t > 0$
In this case, shipments take place from ending inventory of previous period.

From equation (3.57) for period $t-1$, we have

$$\gamma_t = h_{t-1} + \gamma_{t-1} \tag{3.71}$$

Let the last positive production have taken place t-τ periods before and sales in all subsequent periods be met from inventory resulting from this production. In other words, $\hat{X}_{t-L_G-\tau} > 0$ and $\hat{X}_{t-L_G-\tau+1} = ... = \hat{X}_{t-L_G} = 0$. Then from Case (1) we have $\gamma_{t\text{-}\tau} > 0$.

In addition, we have $\hat{I}_{t-\tau}, \hat{I}_{t-\tau+1}, ..., \hat{I}_{t-1} > 0$. Writing equation (3.71) for periods t-$\tau + 1$ to t, we have

$$\begin{aligned}
\gamma_{t-\tau+1} &= h_{t-\tau} + \gamma_{t-\tau} \\
\gamma_{t-\tau+2} &= h_{t-\tau+1} + \gamma_{t-\tau+1} \\
&\vdots \\
\gamma_t &= h_{t-1} + \gamma_{t-1}
\end{aligned}$$

Adding the above expressions, we get

$$\gamma_t = \sum_{i=t-\tau}^{t-1} h_i + \gamma_{t-\tau}$$

Since both terms on the right hand side are positive, we have $\gamma_t > 0$. *QED*.

Appendix 3.7

By Little's Law we have $L = \frac{\hat{W}_t}{X_t}$, implying that $\hat{W}_t = LX_t$. Noting that the capacity constraints will be tight, we have $X_t = \frac{K_1 \hat{W}_t}{K_2 + \hat{W}_t}$. The utilization u_t in period t can thus be calculated as $u_t = \frac{X_t}{K_1} = \frac{LX_t}{LX_t + K_2}$. Solving for K2, we obtain $K_2 = L(1-u_t)K_1$. Choosing $K_1 = M/2$, for $M = 1{,}000$ we obtain $K_1 = 500$. For $L = 1$ and $u_t = 0.8$, we obtain $K_2 = 100$.

Appendix 3.8

From WIP Balance constraint of CF model, we have:

$$W_t = W_{t-1} - X_t + R_t$$

Writing the constraint for WIP level as an equality, we have $\hat{W}_t = \frac{W_{t-1}+W_t}{2}$, implying $W_t = 2\hat{W}_t - W_{t-1}$. Setting the two expressions for W_t equal to each other, we find that $X_t = R_t - 2\hat{W}_t + 2W_{t-1}$

Writing the CF constraint as an equality,

$$X_t = f(\hat{W}_t) = \frac{K_1 \hat{W}_t}{K_2 + \hat{W}_t}$$

Comparing both equations for X_t, we have

$$R_t - 2\hat{W}_t + 2W_{t-1} = \frac{K_1 \hat{W}_t}{K_2 + \hat{W}_t}$$

Solving the resulting quadratic in $\hat{W}_t$

$$\hat{W}_t = \frac{(R_t - 2K_2 + 2W_{t-1} - K_1) \pm \sqrt{(R_t - 2K_2 + 2W_{t-1} - K_1)^2 + 8K_2(R_t + W_{t-1})}}{4}$$

We use the positive root for calculating WIP level in a period when simulating FLT decisions under congestion.

References

Adida, E., & Perakis, G. (2006). A robust optimization approach to dynamic pricing and inventory control with no backorders. *Mathematical Programming Series B, 107,* 97–129.

Adida, E., & Perakis, G. (2010). Dynamic pricing and inventory control: robust vs. stochastic uncertainty models: A computational study. *Annals of Operations Research, 181,* 125–157.

Agnew, C. (1976). Dynamic modeling and control of some congestion prone systems. *Operations Research, 24*(3), 400–419.

Ahn, H., Gumus, M., & Kaminsky, P. (2007). Pricing and manufacturing decisions when demand is a function of prices in multiple periods. *Operations Research, 55*(6), 1039–1057.

Akcali, E., Nemoto, K., & Uzsoy, R. (2000). Quantifying the benefits of cycle-time reduction in semiconductor wafer fabrication. *IEEE Transactions on Electronics Packaging Manufacturing, 23,* 39–47.

Allison, R. A. H., Yu, J., Tsai, L. H., Liu, C., Drummond, M., Kayton, D., Sustae, T., & Witte, J. (1997). Macro model development as a bridge between factory level simulation and LP enterprise systems. *IEEE/CPMT International Electronics Manufacturing Technology Symposium*: 408–416.

Asmundsson, J. M., Rardin, R. L., Turkseven, C. H., & Uzsoy, R. (2009). Production planning models with resources subject to congestion. *Naval Research Logistics, 56,* 142–157.

Asmundsson, J. M., Rardin, R. L., & Uzsoy, R. (2006). Tractable nonlinear production planning models for semiconductor wafer fabrication facilities. *IEEE Transactions on Semiconductor Manufacturing, 19,* 95–111.

Boyaci, T., & Ray, S. (2003). Product differentiation and capacity cost interaction in time and price sensitive markets. *Manufacturing and Service Operations Management, 5*(1), 18–36.

Buzacott, J. A., & Shanthikumar, J. G. (1993). *Stochastic models of manufacturing systems*. Englewood Cliffs, NJ, Prentice-Hall.

Charnsirisakskul, K., Griffin, P., & Keskinocak, P. (2004). Order selection and scheduling with leadtime flexibility. *IIE Transactions, 36,* 697–707.

Charnsirisakskul, K., Griffin, P., & Keskinocak, P. (2006). Pricing and scheduling decisions with leadtime flexibility. *European Journal of Operational Research, 171,* 153–169.

Chatterjee, S., Slotnick, S. A., & Sobel, M. J. (2002). Delivery guarantees and the interdependence of marketing and operations. *Production and Operations Management, 11*(3), 393–410.

Chen, Z. L., & Hall, N. G. (2010). The coordination of pricing and scheduling decisions. *Manufacturing and Service Operations Management, 12*(1), 77–92.

de Kok, A. G., & Fransoo, J. C. (2003). *Planning supply chain operations: definition and comparison of planning concepts*. *OR Handbook on supply chain management*. A. G. de Kok & S. C. Graves (597–675). Amsterdam: Elsevier.

Dellaert, N. P. (1991). Due date setting and production control. *International Journal of Production Economics, 23,* 59–67.

Deng, S., & Yano, C. A. (2006). Joint production and pricing decisions with setup costs and capacity constraints. *Management Science, 52,* 741–756.

Donohue, K. L. (1994). The economics of capacity and marketing measures in a simple manufacturing environment. *Production and Operations Management, 3*(2), 78–99.

Duenyas, I. (1995). Single facility due date setting with multiple customer classes. *Management Science, 41*(4), 608–619.

Duenyas, I., & Hopp, W. J. (1995). Quoting customer lead times. *Management Science, 41,* 608–619.

Easton, F. F., & Moodie, D. R. (1999). Pricing and lead time decisions for make-to-order firms with contingent orders. *European Journal of Operational Research, 116,* 305–318.

Elhafsi, M. (2000). An operational decision model for lead-time and price quotation in congested manufacturing systems. *European Journal of Operational Research, 126,* 355–370.

Elhafsi, M., & Rolland, E. (1999). Negotiating price/delivery date in a stochastic manufacturimg environment. *IIE Transactions, 31,* 255–270.

Eliashberg, J., & Steinberg, R. (1991). Marketing-production joint decision-making. *Management science in marketing, handbooks in operations research and management science*. J. Eliashberg and J. D. Lilien, North Holland: 827–880.

Elmaghraby, W., & Keskinocak, P. (2003). Dynamic pricing in the presence of inventory considerations: Research overview, current practices and future directions. *Management Science, 49*(10), 1287–1309.

Graves, S. C. (1986). A tactical planning model for a job shop. *Operations Research, 34,* 552–533.

Hackman, S. T., & Leachman, R. C. (1989). A general framework for modeling production. *Management Science, 35,* 478–495.

Hopp, W. J., & Spearman, M. L. (2001). *Factory physics: Foundations of manufacturing management*. Boston, Irwin/McGraw-Hill.

Johnson, L. A., & Montgomery, D. C. (1974). *Operations research in production planning, scheduling and inventory control*. New York: John Wiley.

Kacar, N. B., & Uzsoy, R. (2010). Estimating clearing functions from simulation data. *Winter Simulation Conference*. B. Johansson, Jain, S., Montoya-Torres, J., Hugan, J., Yucesan, E. Baltimore, MD.

Karmarkar, U. S. (1989). Capacity loading and release planning with work-in-progress (WIP) and lead-times. *Journal of Manufacturing and Operations Management, 2*(105-123).

Kefeli, A., Uzsoy, R., Fathi, Y., & Kay, M. (2011). Using a mathematical programming model to examine the marginal price of capacitated resources. *International Journal of Production Economics, 131*(1), 383–391.

Keskinocak, P., & Tayur, S. (2004). Due-date management policies. In D. Simchi-Levi, S. D. Wu, & Z. M. Shen (Eds.), *Supply chain analysis in the e-business era: Handbook of quantitative supply chain analysis*. Kluwer Academic Publishers.

Leachman, R. C., & Ding, S. (2007). Integration of speed economics into decision-making for manufacturing management. *International Journal of Production Economics, 107,* 39–55.

Liu, L. M., Parlar, M., & Zhu, S. X. (2007). Pricing and lead time decisions in decentralized supply chains. *Management Science, 53*(5), 713–725.

Low, D. W. (1974). Optimal dynamic pricing policies for an M/M/s queue. *Operations Research, 22,* 545–561.

Medhi, J. (1991). *Stochastic models in queuing theory*. Academic Press.

Missbauer, H. (2009). Models of the transient behaviour of production units to optimize the aggregate material flow. *International Journal of Production Economics, 118*(2), 387–397.

Missbauer, H., & Uzsoy, R. (2010). *Optimization models for production planning. Planning production and inventories in the extended enterprise: A state of the art handbook*. K. G. Kempf, P. Keskinocak and R. Uzsoy (437–508). New York: Springer.

Orcun, S., Uzsoy, R., & Kempf, K. G. (2006). Using system dynamics simulations to compare capacity models for production planning. *Winter Simulation Conference*. Monterey, CA.

Pahl, J., Voss, S., & Woodruff, D. L. (2005). Production planning with load dependent lead times. *4OR: A Quarterly Journal of Operations Research, 3,* 257–302.

Pahl, J., Voss, S., & Woodruff, D. L. (2007). Production planning with load dependent lead times: An update of research. *Annals of Operations Research, 153,* 297–345.

Palaka, K., Erlebacher, S., & Kropp, D. H. (1998). Lead-time setting capacity utilization, and pricing decisions under lead-time dependent demand. *IIE Transactions, 30,* 151–163.

Pekgun, P., Griffin, P. M., & Keskinocak, P. (2008). Coordination of marketing and production for price and leadtime decisions. *IIE Transactions, 40*(1), 12–30.

Plambeck, E. L. (2004). Optimal leadtime differentiation via diffusion approximation. *Operations Research, 52*(2), 213–228.

Ray, S., & Jewkes, E. M. (2004). Customer lead time management when both demand and price are lead time sensitive. *European Journal of Operational Research, 153,* 769–781.

Selçuk, B., Fransoo, J. C., & de Kok, A. G. (2007). Work in process clearing in supply chain operations planning. *IIE Transactions, 40,* 206–220.

So, K. C., & Song, J.-S. (1998). Price, delivery time guarantees and capacity selection. *European Journal of Operational Research, 111,* 28–49.

Spearman, M. L. (1991). An analytic congestion model for closed production systems with ifr processing times. *Management Science, 37*(8), 1015–1029.

Spitter, J. M., A. G. de Kok and N. P. Dellaert (2005a). Timing production in LP models in a rolling schedule. *International Journal of Production Economics, 93–94,* 319–329.

Spitter, J. M., Hurkens, C. A. J., de Kok, A. G., Lenstra, J. K., & Negenman, E. G. (2005b). Linear programming models with planned lead times for supply chain operations planning. *European Journal of Operational Research, 163,* 706–720.

Srinivasan, A., Carey, M., & Morton, T. E. (1988). Resource pricing and aggregate scheduling in manufacturing systems. *Graduate School of Industrial Administration, Carnegie-Mellon University*. Pittsburgh, PA

Sterman, J. D. (2000). *Business dynamics: Systems thinking and modeling for a complex world*. New York: McGraw-Hill.

Swann, J. L. (2001). Dynamic pricing models to improve supply chain performance. *Department of Industrial Engineering and Management Sciences*. Evanston, IL 60601, Northwestern University.

Tardif, V., & Spearman, M. L. (1997). Diagnostic scheduling in finite-capacity production environments. *Computers and Industrial Engineering, 32,* 867–878.

Upasani, A., & Uzsoy, R. (2008). Incorporating manufacturing lead times in joint production-marketing models: A review and further directions. *Annals of Operations Research, 161,* 171–188.

Webster, S. (2002). Dynamic pricing and lead time policies for make to order systems. *Decision Sciences, 33*(4), 579–599.

Yano, C. A., & Gilbert, S. M. (2003). Coordinated pricing and production/procurement decisions: A review. *Managing business interfaces: Marketing, engineering and manufacturing perspectives*. A. Charkarvarty and J. Eliashberg, Kluwer Academic Publishers: 65–103.

Chapter 4
Refined EM Method for Solving Linearly Constrained Global Optimization Problems

Lu Yu and Shu-Cherng Fang

Introduction

In recent years global optimization has become a rapidly developing field. Many real life problems in areas such as physics, chemistry, and molecular biology involve nonlinear objective functions where multiple local optima may exist. These problems can be difficult to optimize by conventional mathematical tools, such as gradient methods.

To locate a global optimum among many local optima, various stochastic search methods have been proposed. Commonly used algorithms include simulated annealing (Ingber 1994), multilevel methods (Kan and Timmer 1987), evolutionary methods (Michalewicz 1996), partitioning methods (Wood 1991), and particle swarm optimizer (Kennedy and Eberhart 1995). These methods utilize a stochastic mechanism to search for better bounds on an objective function to be optimized. Some of these methods may combine the search process with local refinements like hill-climbing or gradient-based methods (Hart 1994).

Recently, Birbil and Fang proposed a new population-based stochastic search algorithm (Birbil and Fang 2002, 2004). The method is called electromagnetism-like method (EM), which utilizes an attraction-repulsion mechanism to move a population of points toward optimality. The computational results have shown that EM converges rapidly (in terms of the number of functions evaluations) to the global optimal solutions and produce better results than other known methods in solving problems without the using higher order information of the objective functions. In this paper we extend the EM method to solve optimization problems defined by general linear constraints without using the derivatives of the objective function.

L. Yu (✉) · Shu-C. Fang
Edward P. Fitts Department of Industrial and Systems Engineering,
North Carolina State University, Raleigh, North Carolina
e-mail: lyu@ncsu.edu

Shu-C. Fang
e-mail: fang@ncsu.edu

P. S. Pulat et al. (eds.), *Essays in Production, Project Planning and Scheduling,*
International Series in Operations Research & Management Science 200,
DOI 10.1007/978-1-4614-9056-2_4, © Springer Science+Business Media New York 2014

Let $f(\cdot)$ be a real-valued function, A be an $m \times n$ real matrix, and $\mathbf{b}$ be an $m-$vector. In this paper, we consider the following global optimization problem with linear constraints:

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & A\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \in R^n. \end{aligned} \tag{4.1}$$

We do not require any special information or structure of the objective function, as long as we know how to evaluate its value at each point. But the feasible domain $S = \{\mathbf{x} \in R^n \mid A\mathbf{x} \leq \mathbf{b}\}$ is assumed to be a nonempty bounded set. The refined EM seeks for the global solution of (4.1) while maintaining feasibility in each iteration.

The paper is organized as follows. In Sect. 4.2, the main structure of refined EM, including its major steps, is given. The computational results and comparisons between refined EM and existing global optimizers are given in Sect. 4.3. Conclusions are given in Sect. 4.4.

Refined Electromagnetism-Like Mechanism (Refined EM)

We assume that for problem (4.1), the following parameters are given: the dimension of the problem (n), the objective function ($f(\cdot)$), the matrix $A \in R^{m\times n}$ and the vector $\mathbf{b} \in R^m$. Since EM works on a set of sample points (population), there is an additional predetermined parameter, r, which denotes the number of points in the population.

Our goal is to design a refined EM seeking for the global solutions while maintaining feasibility in each iteration. In this way, the algorithm always provides a meaningful solution even when it stops prematurely. The refined EM contains four major steps, namely, "Initialization", "Local Search", "Calculation of Aggregated Force" and "Movement". The main structure of refined EM is given in Algorithm 1. The details of these procedures will be given in Sects. 4.2.1–4.2.5.

Algorithm 1 EM for linear-constrained Problems

```
Define parameters.
Initialize(r)
iteration = 1.
while termination criteriaare not satisfied do
    Local ()
    CalcF ()
    Move ()
    Check termination criteria
    iteration = iteration + 1.
end while
Output x^best and f(x^best).
```

Initialization

The *Initialization* procedure is used to sample a certain number (r) of points randomly from the feasible domain,

$$\mathcal{S} = \{\mathbf{x} \in R^n | \ A\mathbf{x} \leq \mathbf{b}\} \tag{4.2}$$

which is an n dimensional polyhedron.

Before the algorithm starts, some parameters are defined as follows:

Parameters

Δ_{tol}: the stopping tolerance.
Δ: the stopping parameter.
$\phi > 1,\ 0 < \theta < 1$: the increasing and decreasing factors, respectively.

Δ_{tol} and Δ will be used to check termination criteria. $\phi > 1$ and $0 < \theta < 1$ are used to increase and decrease the stopping parameter Δ.

After the parameters have been defined, initialization starts. There are four methods considered in refined EM to generate initial feasible solutions. The first method is to ignore the linear constraints at the beginning and randomly generate points. Then a newly generated point is accepted if it satisfies the linear constraints. This strategy is straightforward and easy to implement. However, such a strategy may not be efficient for generating a diverse feasible population.

The second method for generating initial feasible population works as described below. Since the constraint functions are all linear, we may produce a linear programming problem by using the feasible domain S and an artificial linear function as the objective function. Then we apply the simplex method or interior point method to solve the problem. During the solving procedure, by recording the location of the point in each iteration, we are able to obtain some feasible solutions to the problem. Finally, the convex combinations of these solutions can be used as the initial feasible points. Since the linear programming problem can be solved in polynomial time by the interior point method, this method may finish generating the initial feasible population in polynomial time.

The third way of providing an initial feasible population is explained below. First, find an interior point $\mathbf{x}^*$ that lies inside the feasible domain S by solving the following linear programming problem

$$\begin{aligned} \max \quad & t \\ \text{s.t.} \quad & A\mathbf{x} + t\mathbf{e} \leq \mathbf{b} \end{aligned} \tag{4.3}$$

with the optimal solution $(\mathbf{x}^*, t^*)$, where $\mathbf{e} = (1, 1, \ldots, 1)^T \in R^m$. If $t^* > 0$, $\mathbf{x}^*$ can be accepted. Then, from $\mathbf{x}^*$, a set of random vectors $\{\mathbf{v}^1, \mathbf{v}^2, \ldots, \mathbf{v}^q\}$ pointing to different directions are generated. Before extending these vectors to hit the boundary of S, if we choose the step length α_i carefully, we can use the points

$$\mathbf{x}^i = \mathbf{x}^* + \alpha_i \mathbf{v}^i,\ \alpha_i > 0,\ i = 1, 2, \ldots, q, \tag{4.4}$$

and their convex combinations as the initial feasible points. This method involves solving only one linear programming problem.

The fourth method of generating the initial feasible solutions is first find the maximum-volume ellipsoid which is inscribed in the feasible region S. The ellipsoid can be represented as $(\mathbf{x}^*, E^*)$, where $\mathbf{x}^* \in R^n$ is the center of the ellipsoid and $E \in \mathcal{S}_+^n$, where $\mathcal{S}_+^n$ is the cone of all symmetric positive definite matrices in R^n. Then, an initial population within the ellipsoid is randomly generated as below:

$$\mathbf{x}^i = \mathbf{x}^* + E^* \boldsymbol{\eta}^i,\ i = 1, 2, \ldots, r, \tag{4.5}$$

where $\boldsymbol{\eta}^i \in R^n$ and $\|\boldsymbol{\eta}^i\| \leq 1,\ i = 1, 2, \ldots, r$, are vectors generated along different directions. Since the ellipsoid is inside the feasible region, the points generated are feasible. Furthermore, the maximum-volume ellipsoid helps us to distribute the points as diverse as possible.

The computation of the maximum-volume ellipsoid inscribed inside the feasible region is carried out by the interior point method developed by (Zhang and Gao 2001). In their article, there exists a good state-of-the-art optimization software to calculate the maximum-volume ellipsoid.

In our algorithm, the method of finding the maximum volume ellipsoid is first applied since it may generate diverse initial solutions. If the interior point method is unable to find the ellipsoid in a certain number of iterations, we turn to use the third method which is described previously. When the calculated step lengths in the third method are too small, which means the generated points are too close to $\mathbf{x}^*$ in (4.4), the second method is applied. Finally, if the second method still does not provide enough initial feasible solutions, the first method has to be utilized, though it appears to be inefficient.

Local Search

After the initial population has been generated, the procedure *Local Search* is used to find better solutions in their neighborhoods. Many powerful local search methods can be utilized in this step. In this paper a direct search method is applied only at the current best point $\mathbf{x}^{\text{best}}$. In each iteration of the direct search method, we evaluate the objective function value of each selected point in the neighborhood of $\mathbf{x}^{\text{best}}$. The new points in the neighborhood are obtained by adding to $\mathbf{x}^{\text{best}}$ a set of feasible directions $\{\mathbf{d}^i,\ i = 1, 2, \ldots, k\}$:

$$\mathbf{x}_{\text{nbr}}^i = \mathbf{x}^{\text{best}} + a_i \mathbf{d}^i,\ i = 1, 2, \ldots, k, \tag{4.6}$$

where $a_i > 0$ is the step length. Then we keep the one with the lowest function value in $\{\mathbf{x}^{\text{best}}, \mathbf{x}_{\text{nbr}}^i,\ i = 1, 2, \ldots, k\}$ as the updated $\mathbf{x}^{\text{best}}$. This procedure is repeated until the maximum number of iterations has been reached.

The calculation of the directions $\{\mathbf{d}^i,\ i = 1, 2, \ldots, k\}$ is a key step. When $\mathbf{x}^{\text{best}}$ is not close to the boundary of the feasible domain S, a good choice of the directions

is $[\mathbf{d}^1, \mathbf{d}^2, \ldots, \mathbf{d}^k] = [I \; -I]$, where I is the identity matrix. Otherwise, the set of searching directions $\{\mathbf{d}^1, \ldots, \mathbf{d}^k\}$ should reflect the geometry of any portion of the boundary of the feasible region near $\mathbf{x}^{\text{best}}$.

To calculate $\{\mathbf{d}^1, \ldots, \mathbf{d}^k\}$ in the latter case, we adopt the idea in (Kolda et al. 2003). Let $\mathbf{a}^i$ be the i^{th} row of A in (4.1) and b_i be the i^{th} element of $\mathbf{b}$ in (4.1). For a given $\epsilon > 0$, define

$$\mathcal{I} = \{i \mid \mathbf{a}^i \mathbf{x}^{\text{best}} - b_i \geq -\epsilon, \; i = 1, 2, \ldots, m\}, \tag{4.7}$$

to be the active set whose elements are the indices of the active constraints. Let $\widetilde{A}, \widetilde{\mathbf{b}}$ be the matrix and vector that correspond to the active set:

$$\widetilde{A} = \begin{bmatrix} \mathbf{a}^i \end{bmatrix} \text{and } \widetilde{\mathbf{b}} = \begin{bmatrix} b_i \end{bmatrix}, \; i \in \mathcal{I}. \tag{4.8}$$

Define

$$\mathbf{v}^i \triangleq \mathbf{v}^i(\mathbf{x}^{\text{best}}, \epsilon) = \mathbf{a}^{i\,T}, \; i \in \mathcal{I}. \tag{4.9}$$

Geometrically $\mathbf{v}^i(\mathbf{x}^{\text{best}}, \epsilon)$ is the outward-pointing normal to the corresponding facet of $\mathcal{S}$. Define $K(\mathbf{x}^{\text{best}}, \epsilon)$ to be the cone generated by the vectors in $\{\mathbf{v}^i(\mathbf{x}^{\text{best}}, \epsilon), \; i \in \mathcal{I}\}$, and its polar cone $K^0(\mathbf{x}^{\text{best}}, \epsilon) = \{\mathbf{u} \mid \mathbf{u}^T \mathbf{v} \leq 0, \; \forall \; \mathbf{v} \in K(\mathbf{x}^{\text{best}}, \epsilon)\}$. Then the search directions can be formed by the vectors which generate the cone $K^0(\mathbf{x}^{\text{best}}, \epsilon)$.

If the vectors $\{\mathbf{v}^i\}$ that generate the cone $K(\mathbf{x}^{\text{best}}, \epsilon^*)$ are linearly independent for some $\epsilon^* > 0$, one can construct the generators of the cone $K^0(\mathbf{x}^{\text{best}}, \epsilon)$ in the following way: let V denote the matrix whose columns are $\{\mathbf{v}^i\}$. Suppose there are s vectors, V is an $n \times s$ matrix. Let N be the matrix whose columns are in the basis of the null space of V^T. Then one can show that for any $\epsilon, \; 0 < \epsilon < \epsilon^*$, a set of generators of $K^0(\mathbf{x}^{\text{best}}, \epsilon)$ can be found among the columns of $N, \; V(V^T V)^{-1}$ and $-V(V^T V)^{-1}$.

The next task is to determine a set of linearly independent vectors $\{\mathbf{v}^i\}$. If ϵ is set to be too large, there could be more rows in $\widetilde{A}$ and there is a higher probability that the rows are linearly dependent. If ϵ is too small, the directions obtained could be useless. Thus the direct search used in refined EM method tries to dynamically decrease ϵ to achieve our goal. The calculation of $\{\mathbf{d}^i, \; i = 1, 2, \ldots, k\}$ is summarized in Algorithm 2.

Calculate Force

The computation of the total force vector is inspired by the superposition principle of electromagnetism theory (Cowan 1968). In each iteration, a charge q_i of each point $\mathbf{x}^i$ is calculated according to $f(\mathbf{x}^i)$ in (4.10). The charge reflects the efficiency of the objective function value of the corresponding point in the population. The point with a higher charge has a lower function value and tends to attract other points to come

Algorithm 2 CalcJ($\mathbf{x}^{\text{best}}, \epsilon_{\text{tol}}$)

$\mathbf{x}^{\text{best}}$: the current best point.
$\epsilon_{\text{tol}} > 0$: the stopping tolerance.
Set $\epsilon > 0$ to be the stopping parameter.
Set $0 < \vartheta < 1$ to be the decreasing factor.
while $\epsilon > \epsilon_{\text{tol}}$ **do**
  Let $\widetilde{A}$ be the matrix defined in (4.8).
  if $\widetilde{A}$ is of full row rank and $rank(\widetilde{A}) > 0$ **then**
    Find a QR factorization of $\widetilde{A}^T$: $[Q, R] = \mathsf{qr}(\widetilde{A}^T)$.
    Set $B = Q(R^T)^{-1}$, $N = I - B\widetilde{A}$.
    Set $J = [B \ -B \ N \ -N]$.
  else if $\dim(\widetilde{A}) = 0$ **then**
    Set $J = [I \ -I]$.
  else
    Set $\epsilon = \vartheta \times \epsilon$ (decrease ϵ to make $\widetilde{A}$ full row rank).
  end if
end while
Output the matrix $J = [\mathbf{d}^1, \mathbf{d}^2, \ldots, \mathbf{d}^k]$ where $\{\mathbf{d}^k,\ i = 1, 2, \ldots, k\}$ are the directions used to generate new trial points.

closer to it, while the one with lower charge repulses other particles. The charge of particles are defined as

$$q_i = \exp\left(-n \times \frac{f(\mathbf{x}^i) - f(\mathbf{x}^{\text{best}})}{\sum_{k=1}^{r}[f(\mathbf{x}^k) - f(\mathbf{x}^{\text{best}})]}\right). \tag{4.10}$$

Then the total force vector $\mathbf{f}^i$ exerted on point $\mathbf{x}^i$ is calculated by adding together the individual component forces, $\mathbf{f}^{ij}$, between pairs of points $\mathbf{x}^i$ and $\mathbf{x}^j$, $j = 1, 2, \ldots, r,\ j \neq i$. The magnitude of this component force is inversely proportional to the Euclidean distance between the points and directly proportional to the product of their charges.

$$\mathbf{f}^i = \sum_{j \neq i}^{r} \mathbf{f}^{ij},\ i = 1, 2, \ldots, r, \tag{4.11}$$

where

$$\mathbf{f}^{ij} = \begin{cases} (\mathbf{x}^j - \mathbf{x}^i)\frac{q_i q_j}{\|\mathbf{x}^j - \mathbf{x}^i\|^2}, & \text{if } f(\mathbf{x}^j) < f(\mathbf{x}^i) \\ (\mathbf{x}^i - \mathbf{x}^j)\frac{q_i q_j}{\|\mathbf{x}^j - \mathbf{x}^i\|^2}, & \text{if } f(\mathbf{x}^j) \geq f(\mathbf{x}^i) \end{cases},\ i = 1, 2, \ldots, r. \tag{4.12}$$

Closely examining the algorithm, we see that the determination of a direction via the total force vector is similar to the statistical estimation of the gradient vector of f. But the Euclidean distance between two points also affects the magnitude of the force. Therefore, the points that become close enough may lead each other to a direction other than the statistically estimated gradient of f.

To prevent the algorithm from converging prematurely (for details, please refer (Birbil and Fang 2004)), a modification is performed by adding a perturbed point $\mathbf{x}^p$ which is the farthest point from the current best point $\mathbf{x}^{\text{best}}$ defined by

$$\mathbf{x}^p = \text{argmax}\{\|\mathbf{x}^{\text{best}} - \mathbf{x}^i\|,\ i = 1, 2, \ldots, r\}. \tag{4.13}$$

At $\mathbf{x}^p$, the component forces are perturbed by a random number $\lambda \sim U(0,1)$,

$$\mathbf{f}^{pj} = \begin{cases} (\mathbf{x}^j - \mathbf{x}^p)\frac{\lambda q_p q_j}{\|\mathbf{x}^j - \mathbf{x}^p\|^2}, & \text{if } f(\mathbf{x}^j) < f(\mathbf{x}^p), \\ (\mathbf{x}^p - \mathbf{x}^j)\frac{\lambda q_p q_j}{\|\mathbf{x}^j - \mathbf{x}^p\|^2}, & \text{if } f(\mathbf{x}^j) \geq f(\mathbf{x}^p). \end{cases} \tag{4.14}$$

Also, the directions of the component forces are perturbed. That is, if λ is less than a given parameter ν, then the direction of the component force is reversed. Consequently, there exists one point in the population for which the direction of movement may be reversed. The purpose of introducing the perturbed point is to allow the algorithm explore more areas in the feasible region so that a global convergent property can be proved. For details of the convergence proof, please refer to Birbil and Fang (2004).

Movement

In refined EM, the mechanism of movement of the points is similar to that of the original EM for bounded constraints. But instead of moving inside the feasible region formed only by the lower and upper bounds, the points have to shift inside a polyhedron formed by the bounds and linear constraints.

To simplify the notation, let $\mathbf{f}^i \leftarrow \frac{\mathbf{f}^i}{\|\mathbf{f}^i\|}$, $(i = 1, 2, \ldots, r \text{ and } i \neq \text{best})$ be the normalized force vector. A feasible point $\mathbf{x}^i$ is moved according to the following equation:

$$\mathbf{x}^i_{\text{new}} = \mathbf{x}^i + \lambda \mathbf{f}^i (RNG_i),\ i = 1, 2, \ldots, r \text{ and } i \neq \text{best}, \tag{4.15}$$

where $\lambda \sim U(0,1]$ and $\mathbf{x}^i_{\text{new}}$ is the updated point. Our goal is to calculate the appropriate $RNG_i > 0$ so that $\mathbf{x}^i_{\text{new}}$ is still feasible.

For a point $\mathbf{x}^i$ in the population, let $\mathcal{L}_i$ be the set of indices corresponding to the constraints which may lead to infeasibility:

$$\mathcal{L}_i = \{j \mid \mathbf{a}^j \mathbf{f}^i > 0,\ j = 1, 2, \ldots, m\}, \tag{4.16}$$

where $\mathbf{a}^j$ is the j^{th} row of matrix A. We can see that if $\mathbf{a}^j \mathbf{f}^i \leq 0$, no matter how large the step length is, the new point remains feasible. It simply goes further away from the boundary of the feasible region. Therefore, only the rows $\mathbf{a}^j$ whose indices are in $\mathcal{L}_i$ need to be considered. The maximum step length allowed along the force vector $\mathbf{f}^i$ is given by

$$RNG_i = \begin{cases} \min_{j \in \mathcal{L}_i} \left(\dfrac{b_j - \mathbf{a}^j \mathbf{x}^i}{\mathbf{a}^j \mathbf{f}^i} \right), & \text{if } \mathcal{L}_i \text{ is not empty}, \\ 1, & \text{if } \mathcal{L}_i \text{ is empty}. \end{cases} \tag{4.17}$$

Then, if a direction $\mathbf{f}^i$ is pointing outward to the boundary that may lead the point $\mathbf{x}^i$ to infeasibility, RNG_i will prevent $\mathbf{x}^i$ from going too far. Thus Eq. (4.15) guarantees the feasibility of the updated point.

If a point $\mathbf{x}^i$ is close to the boundary of $\mathcal{S}$ and the direction $\mathbf{f}^i$ exerted on it is pointing outward to that boundary, RNG_i could be very small. This boundary is called an active constraint. In this case, the point will not be updated to a new position. To resolve this problem, one more step is applied at $\mathbf{x}^i$.

The first thing is to find the active constraints. Let ϵ be a small positive number, $\mathbf{a}^j$ be the j^{th} row of A and b_j be the j^{th} element of $\mathbf{b}$. For a given ϵ, define

$$\mathcal{M}_i = \left\{ j \mid \frac{b_j - \mathbf{a}^j \mathbf{x}^i}{\mathbf{a}^j \mathbf{f}^i} < \epsilon,\ j = 1, 2, \ldots, m \right\},$$

and

$$A_i = \left[\ \mathbf{a}^j\ \right],\ j \in \mathcal{M}_i. \tag{4.18}$$

Then, we project the direction $\mathbf{f}^i$ onto the null space of A_i, i.e.,

$$\hat{\mathbf{f}}^i = (I - A_i^T (A_i A_i^T)^{-1} A_i) \mathbf{f}^i. \tag{4.19}$$

Since $A_i \hat{\mathbf{f}}^i = 0$, $\hat{\mathbf{f}}^i$ does not point outward to the active constraint and RNG_i is significantly larger than 0. Thus, $\hat{\mathbf{f}}^i$ can be used as the new force vector exerted on $\mathbf{x}^i$.

Finally, the current best point $\mathbf{x}^{\text{best}}$ is not moved since the current best record should be kept and carried to the subsequent iteration. This suggests that we may avoid the calculation of the total force on the current best point.

Termination

The original EM method stops when the number of iterations exceeds a maximum limit. We keep this as an important criterion, and there are other ways of defining the stopping criteria.

Notice that in the searching procedure there are two cases in which the method could fail to find a better point. The first one is that, before performing the local search procedure, all the updated points are not better than the best point obtained in the previous iteration. The second case is that in the local search step, no better point is found. When the two cases happen consecutively, it indicates that refined EM has not find an improved solution in the current iteration. In this case the stopping parameter Δ defined in Algorithm 1 is decreased. Otherwise, the stopping parameter Δ is increased. Hence a sufficiently small Δ indicates that the algorithm could not find a better point in relatively many iterations and it is the time to stop.

In this paper the algorithm stops when either the iterations or function evaluations exceed the corresponding limits, or when Δ is less than a certain threshold Δ_{tol}.

Computational Experiments

After developing refined EM, we collect a set of 73 problems found in the literature to test its performance. The problems are from the following resources: Vanderbei (2013), CUTEr collection Gould, Orban and Toint (2013), GLOBALLib, Runarsson and Yao (2000), Ji et al (2007) and Michalewicz (1994, 1996). Among the 73 problems, there are 14 problems whose objective functions are linear functions and 59 problems whose objective functions are nonlinear functions. The dimension of the problems ranges between 2 and 100.

We then apply refined EM and other derivative free global optimizers to solve the test problems and compare the results provided by refined EM and other optimizers. Besides refined EM, the optimizers used in our numerical experiments are PSwarm (PSO) (Vaz and Vicente 2009) and Genetic Algorithm (GA) in the MATLAB toolbox. These two methods are both population-based stochastic search methods.

All the parameters used in the optimizers are set to be the default values. The population size of each solver is set to be 40. Refined EM, GA, and PSO are all run 10 times. We terminate the iteration using a combination of relative and absolute measures of $f(\mathbf{x})$, i.e., when

$$|f(\mathbf{x}^*) - f_{\text{glob}}| \leq \tau_r |f_{\text{glob}}| + \tau_a, \tag{4.20}$$

where $f(\mathbf{x}^*)$ is the solution obtained by the algorithm and f_{glob} is the known global optimum. τ_r and τ_a are relative and absolute error tolerances, respectively. In our experiments, we set $\tau_r = 10^{-3}$ and $\tau_a = 10^{-4}$.

When an optimizer cannot achieve the known global optimum, it stops when it reaches the maximum number of function evaluations which is set to be 10,000 in our experiments. There is one problem (p.63: *s340*) whose global optimal solution is unknown and this case will be omitted in Fig. 4.1.

Performances on All Test Problems

Figure 4.1 shows the average number of function evaluations used by refined EM to solve the 72 problems (*s340* omitted). Figure 4.2 shows the number of problems solved under different (average) function evaluations.

Figures 4.1 and 4.2 suggest that a large portion of the problems (more than 50) can be solved by refined EM in under 2,000 function evaluations. Most problems (more than 60) are solved in under 5,000 function evaluations. The points whose number of function evaluations are more than 10,000 in Fig. 4.1 represent the problems that are not solved optimally in 10,000 function evaluations. (Since we allow the optimizers to finish the current iteration before they stop, the number of function evaluations for some cases could exceed 10,000). Note that problem *s340* is excluded in Fig. 4.1.

Table 4.1 lists some of the test problems solved by refined EM. The problems have various dimensions and numbers of constraints. It indicates that refined EM has

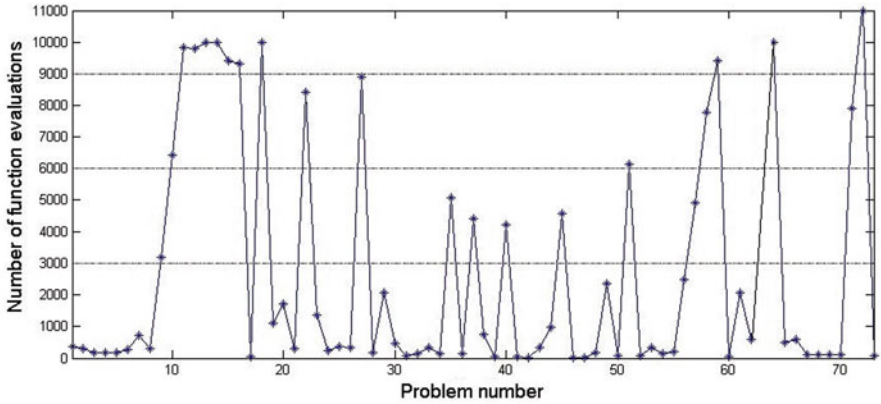


Fig. 4.1 Number of function evaluations used by refined EM for solving the test problems

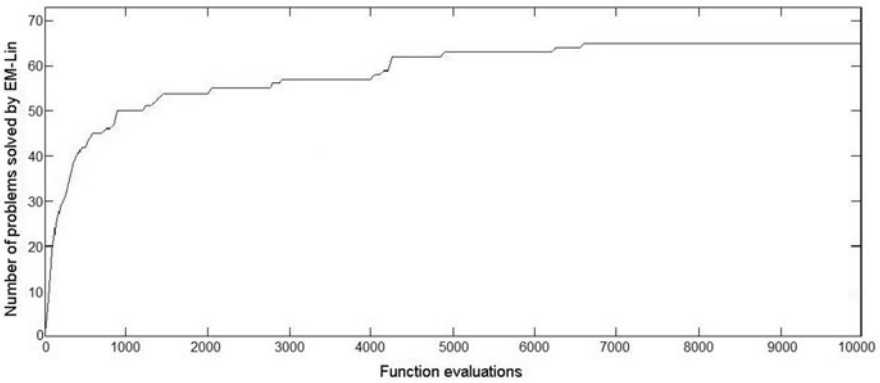


Fig. 4.2 Number of problems solved by refined EM under different function evaluations

the ability to find optimal solutions or near optimal solutions for problems of different sizes. We can also find out that, in general, the number of function evaluations grows when the dimension and/or number of constraints become larger. Furthermore, refined EM met some difficulties in solving problem *s224*, which is a linear programming (LP) problem. This fact is true for many LP problems and gives us an area where refined EM needs to be improved.

To investigate the relationship between dimensionalities of the problems and function evaluations used by refined EM, we divide the problems into groups of different dimensions. The median of the problem dimension is 4, and the interval of the dimension is divided into 4 parts: $[1, 2], [3, 4], [5, 10]$ and $[11, 100]$. The number of problems in each interval is $19, 21, 18$ and 15, respectively. Then the average dimensions, average function evaluations and the 95 % confidence intervals of the average function evaluations are calculated for the problems in these intervals.

Table 4.1 Problems of various dimensions and number of constraints solved by refined EM

Prob	Dim	Cons	Avg. Evals.	Stdev. Evals.
Ji3	2	1	348	31.7007
s224	2	2	6115	50.3919
hs076	4	3	137	23.1442
bunnag5	6	4	298	79.9111
avgasa	6	6	373	54.3612
s278	6	6	4909	2842.3621
bunnag12	20	10	9401	1610.9595
ex2_1_7	20	10	8489	3666.4410
goffin	51	50	8892	1308.2944
himmelbi	100	12	2088	689.3390
Prob	Avg $f(\mathbf{x})$	Best $f(\mathbf{x})$	Stdev. $f(\mathbf{x})$	Known Best
Ji3	-5.9948	-5.9955	0.0012	-6.0000
s224	5.7009	0.0000	6.3076	0.0000
hs076	-4.6792	-4.6816	0.0016	-4.6818
bunnag5	-11	-11	0.0000	-11
avgasa	-4.1685	-4.1687	0.0002	-4.1687
s278	7.8470	7.8434	0.0023	7.8385
bunnag12	-2782.3868	-4105.2779	1534.9321	-4105.2779
ex2_1_7	-3688.4141	-4147.5819	637.8759	-4150.4101
goffin	0.0029	0.0001	0.0017	0.0000
himmelbi	-1754.3000	-1755.0000	0.5516	-1755.0000

The result shown in Fig. 4.3 indicates a trend that the number of function evaluations increases as the dimension of the problem grows.

Performance Profile for Function Evaluations

In the next part we are interested in comparing the existing solvers with refined EM. We present the numerical results in the form of *performance profiles*, as described in Dolan and Moré (2002), for evaluating and comparing the performances of optimization softwares. The performance profile for a solver is the (cumulative) distribution function for a performance metric. This procedure was developed to benchmark optimization softwares, i.e., to compare different solvers on several (possibly many) test problems. One advantage of the performance profiles is that the tested solvers can be presented in one figure where each solver has a cumulative distribution function that represents its performance.

Benchmark results are generated by running a set of solvers $\mathbb{S}$ on a set $\mathbb{P}$ of problems and recording information of interests such as the number of function evaluations and the objective function values. We assume that we have n_s solvers and n_p problems.

Firstly, we are interested in using function evaluations as a performance measure; although, the ideas below can be used with other measures. For each problem p and solver s, we define

$$t_{p,s} = \text{the \# of function evaulations required to solve problem } p \text{ by solver } s \tag{4.21}$$

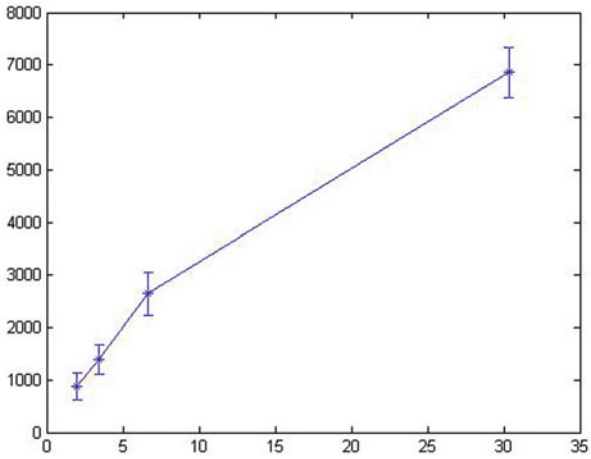


Fig. 4.3 Number of function evaluations vs. problem dimensions

and performance ratio

$$r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s} |\ s \in \mathbb{S}\}} (\geq 1). \tag{4.22}$$

We assume that a parameter $r_M \geq r_{p,s}$, for all p and s, is chosen, and $r_{p,s} = r_M$ if and only if solver s does not solve problem p.

Then we define

$$\rho_s(\tau) = \frac{1}{n_p} (\text{the \# of elements in } \{p \in \mathbb{P} |\ r_{p,s} \leq \tau\}). \tag{4.23}$$

$\rho_s(\tau)$ represents the probability for solver $s \in \mathbb{S}$ whose performance ratio $r_{p,s}$ is within a factor $\tau \geq 1$.

In our study, the term of *performance profile* is kept in use as is in Dolan and Moré (2002). According to Dolan and Moré (2002), a plot of the performance profile reveals the major performance characteristics. In particular, if the set of problems $\rho_s(\tau)$ is suitably large and representative for problems that are likely to occur in applications, then the solvers with large probability $\rho_s(\tau)$ are preferred.

Figures 4.4, 4.6 and 4.7 are the performance profiles of the three tested solvers in terms of the average, minimum and maximum number of function evaluations, respectively. In each figure, the x-axis is τ, and the y-axis is $\rho_s(\tau)$ defined above. There are several points to be addressed in these figures. First, the line of refined EM is higher than the other two algorithms in all three figures, which means it has the highest probability of being the optimal solver. For instance, in Fig. 4.4, refined EM solves about 60 % problems using least function evaluations ($\tau = 0$), while the percentages for PSO and GA are 23 % and 11 %, respectively.

In Fig. 4.4, the line of refined EM is higher than the line of PSO for most of the τ, and both of them are higher than the line of GA. Also, refined EM is especially competitive for the smaller factors ($\tau \in [1, 3.3]$), which suggests that refined EM

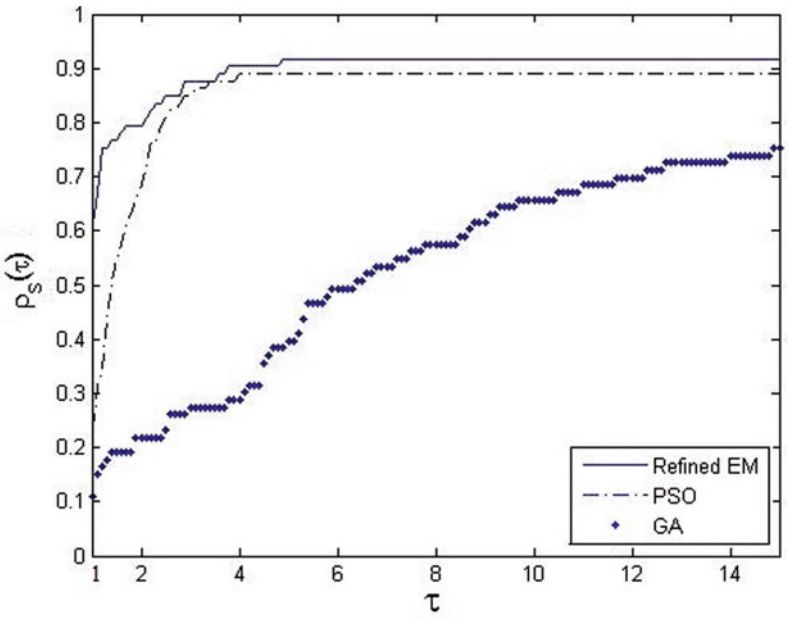


Fig. 4.4 Performance profile of the average number of function evaluations on [1, 15]

can solve more problems with relatively small number of function evaluations. It can be seen that refined EM solves approximately 75 % of the problems with a factor $\tau = 1.2$, which means that in any of these 75 % problems, say, in problem p, if the smallest number of function evaluations used in all the solvers is t_p^*, then the number of function evaluations used by refined EM is no more than $1.2t_P^*$. The corresponding factors for PSO and GA are 2.1 and 14.8, respectively. Similarly, if we are interested in the solver that can solve 80 % of the problems with the greatest efficiency, refined EM also stands out.

When τ is large enough, $\rho_s(\tau)$ represents the percentage of problems that a solver s could eventually solve at least once. We set $\tau = r_M = 1,000$, scale the plot and present Fig. 4.5, which is the performance profile for $\log_2(r_{p,s})$. Here, the probability $\rho_s(\tau)$ is defined below:

$$\rho_s(\tau) = \frac{1}{n_p}(\text{the } \# \text{ of elements in } \{p \in \mathbb{P} |\ \log_2(r_{p,s}) \leq \tau\}). \tag{4.24}$$

Figure 4.5 indicates that refined EM can solve 91.8 % of all problems at least once, while PSO and GA can solve 90.4 % and 82.2 % of all problems at least once. Also, we can see that in the interval $[2, 4]$, the line of GA has a relatively quick increment.

Figure 4.6 shows that, in terms of the minimum number of function evaluations used in 10 runs, 60 % of the solutions provided by refined EM has the term τ smaller than 1.5, the corresponding percentages for PSO and GA are 50 % and 35 %. In general, refined EM has the best $\rho_s(\tau)$ for $\tau \in [1, 2.2]$. PSO has the best $\rho_s(\tau)$ for $\tau \in [2.2, 4.1]$. Moreover, refined EM and PSO perform similarly in the interval $[4.1, 15]$.

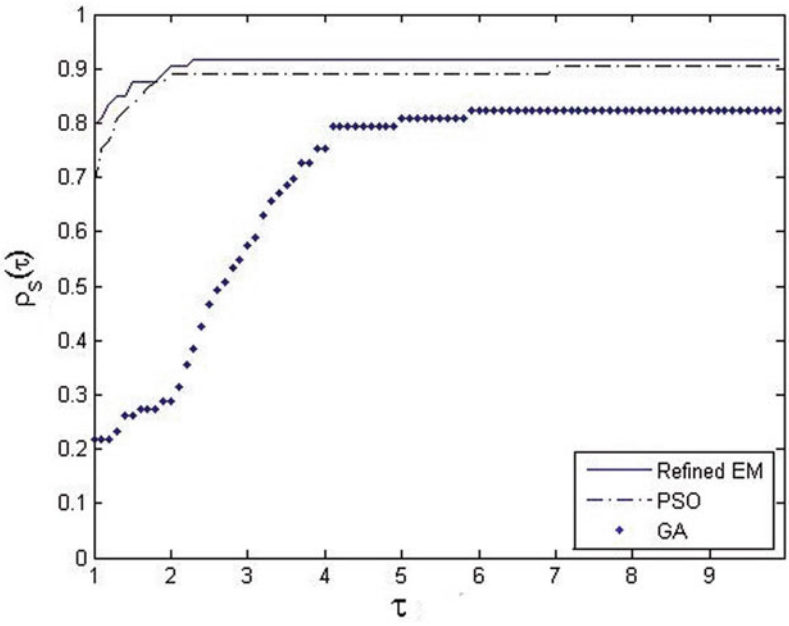


Fig. 4.5 Performance profile of the average number of function evaluations on $[1, 2^{10}]$ in $\log_2$ scale

From Fig. 4.7, we conclude that refined EM is the most efficient solver in terms of the maximum number of function evaluations used in 10 runs. Moreover, it can be seen that refined EM can solve 65.8 % of all problems in all 10 runs, which is because the maximum numbers of function evaluations used in these problems are smaller than $10,000$. Similarly, PSO solves 65.8 % of the problems and GA solves 64.4 % of them.

Figure 4.8 shows the band of 95 % confidence interval for each solver. We can see that for small τ, the band of refined EM is above the band of PSO and they do not overlap, which means that refined EM is significantly better than PSO in terms of the number of function evaluations when $\tau \in [1, 2]$. Moreover, the bands of refined EM and PSO are above the band of GA for $\tau \in [1, 15]$.

Performance Profile for Solution Quality

Now we are interested in the objective function values obtained by the tested solvers. Here, the performance ratio is defined as

$$r_{p,s} = \frac{f_{p,s} - f_p^*}{f_p^w - f_p^*} (\leq 1), \tag{4.25}$$

where $f_{p,s}$ is the (average, minimum or maximum) objective value for problem p obtained by solver s, f_p^* and f_p^w are the best and worst objective values for problem

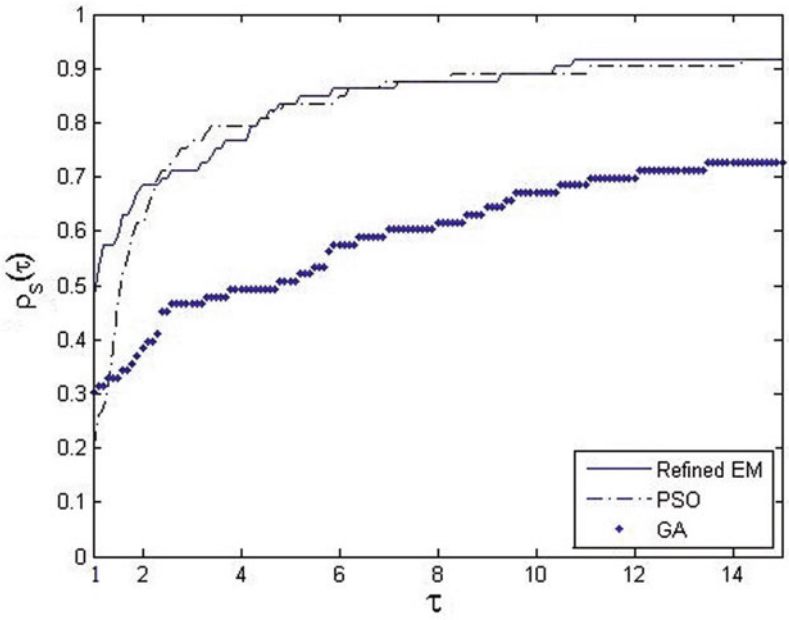


Fig. 4.6 Performance profile of the minimum number of function evaluations on [1, 15]

p among the solutions of all solvers, respectively. If $f_p^* = f_p^w$, which means that all the solvers have found the same solution, let $r_{p,s} = 0$ for all $s \in \mathbb{S}$. Define the probability

$$\rho_s(\tau) = \frac{1}{n_p}(\text{the \# of elements in } \{p \in \mathbb{P} |\ r_{p,s} \leq \tau\}). \tag{4.26}$$

with $0 \leq \tau \leq 1$ in this situation.

For a fixed $0 \leq \tau \leq 1$, a solver with a higher line suggests that it has the ability of providing "τ–good" solutions in more problems. A "τ–good" solution is the one whose performance ratio (4.25) is smaller than or equal to τ.

Figures 4.9, 4.10, and 4.11 give the performance profiles for the average, minimum, and maximum solutions obtained by refined EM, PSO and GA in their 10 runs.

Figure 4.9 shows that refined EM always provides "τ–good" average solutions in more problems than GA and PSO, since the line of refined EM is higher than PSO and GA for every τ.

Figure 4.10 indicates that, in terms of the minimum objective function values, PSO is better when $\tau \in [0.05, 0.49]$ and refined EM is better for $\tau \in [0.49, 1]$.

From Fig. 4.11, we can see that, in terms of the maximum objective function values, refined EM is almost always better than PSO, while GA becomes competitive for $\tau \geq 0.52$.

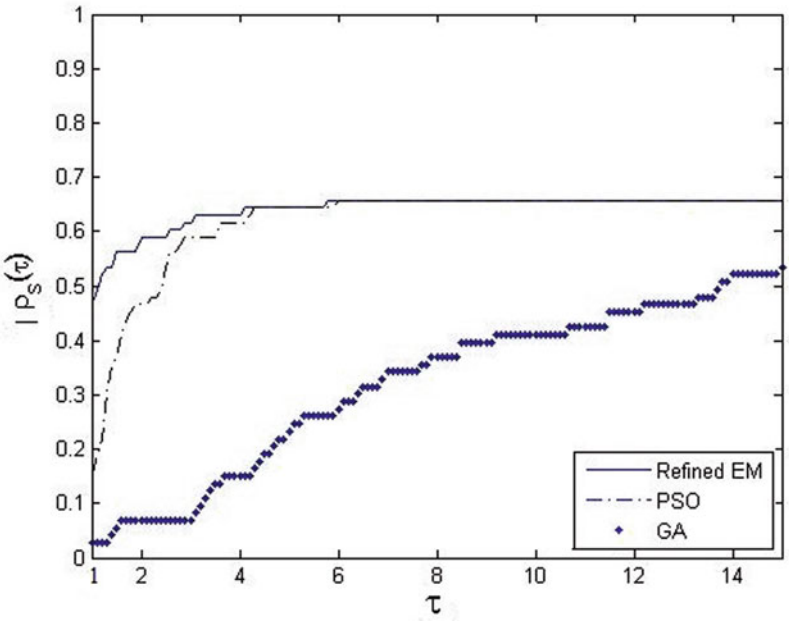


Fig. 4.7 Performance profile of the maximum number of function evaluations on [1, 15]

Figure 4.12 shows the band of 95 % confidence interval of each solver in terms of the objective function values. It can be seen that the bands of refined EM and PSO overlap heavily, which means that the performances of refined EM and PSO are similar in terms of the objective function values.

In summary, if one is interested in the average objective function values, refined EM is a good choice, which means that refined EM is more stable. While if one seeks best solutions, both refined EM and PSO are very competitive.

Solution Quality on Hard Problems

In this section, we mark the hard problems as those were not solved by a solver *s* in 4 runs or more out of the 10 runs. The problem with unknown optimum is excluded. There are in total 15 hard problems for refined EM, 19 for PSO and 18 for GA. We are interested in the 15 hard problems for refined EM. The computational results are shown in Table 4.2. In the table, every solution is the best solution obtained by the corresponding solver in the 10 runs. N/A means that the solver did not find a feasible solution.

In Table 4.2, we notice that

1) Among the 15 problems, refined EM solved optimally 10 problems at least once: *bunnag7*, *bunnag8*, *bunnag9*, *bunnag12*, *bunnag13*, *ex*2_1_7, *hs044*, *s279*, *s280* and *tfi2*.

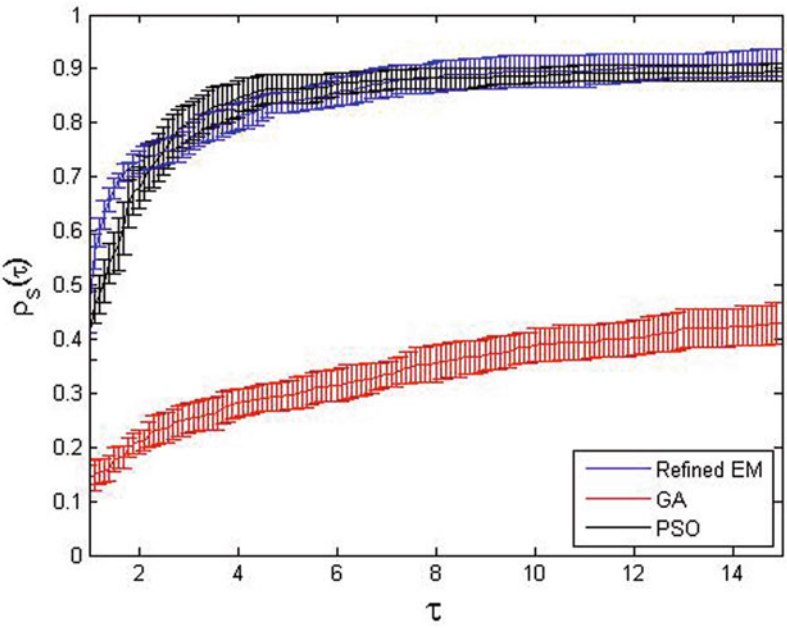


Fig. 4.8 Bands of 95 % confidence intervals for the performance profile of the number of function evaluations on [1, 15]

Table 4.2 Comparison of different methods using 10,000 function evaluations

Problem	*n*	EM-lin	PSO	GA	Known best
bunnag7	10	−39.0000	−39.0000	−38.9921	−39.0000
bunnag8	20	−394.4029	−394.3840	−160.1755	−394.7506
bunnag9	20	−883.9029	−828.8544	−679.7276	−884.7506
bunnag10	20	−8224.4559	−8315.1801	−3486.4060	−8695.0119
bunnag11	20	−682.5842	−753.9962	−630.1617	−754.7506
bunnag12	20	−4105.2779	−4105.2779	−524.7473	−4105.2779
bunnag13	20	49359.1934	51431.0000	154925.5218	49318.0000
ex2_1_10	20	101183.1904	121737.4909	153480.6965	49318.0180
ex 2_1_7	20	−4147.5819	−4146.4153	−378.5011	−4150.4101
hs044	4	−14.9923	−14.9987	−14.9864	−15.0000
s279	8	10.6106	10.6157	10.6168	10.6059
s280	10	13.3886	13.3869	13.3906	13.3754
s359	5	−5.4711E+06	−5.4958E+06	N/A	−5.5045E+06
s392	30	−1.0662E+06	−1.0418E+06	−331242.8649	−1.1012E+6
tfi2	3	0.6496	0.6493	N/A	0.6490

2) PSO also solved 10 problems optimally at least once: *bunnag7*, *bunnag8*, *bunnag11*, *bunnag12*, *ex*2_1_7, *hs044*, *s279*, *s280*, *s359*, and *tfi2*.
3) GA solved four problems optimally at least once: *bunnag7*, *hs044*, *s279*, and *s280*.

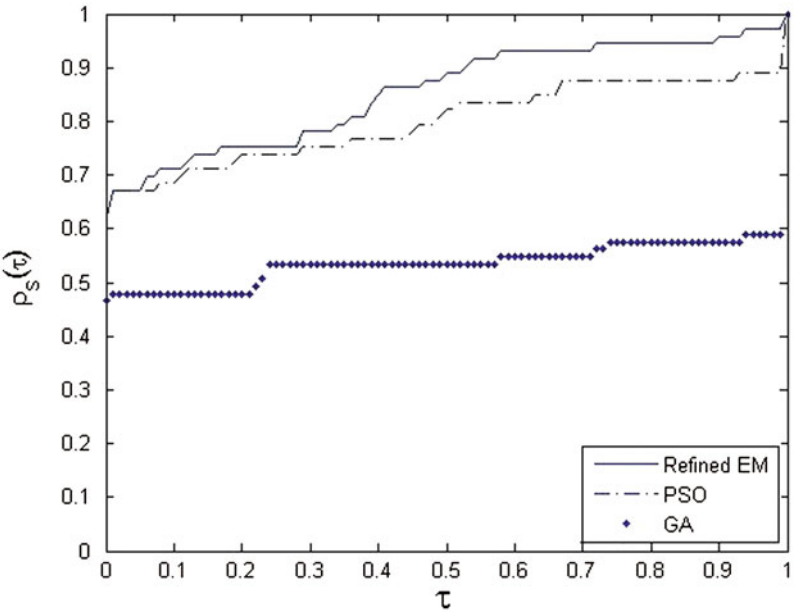


Fig. 4.9 Performance profile of the average objective function values

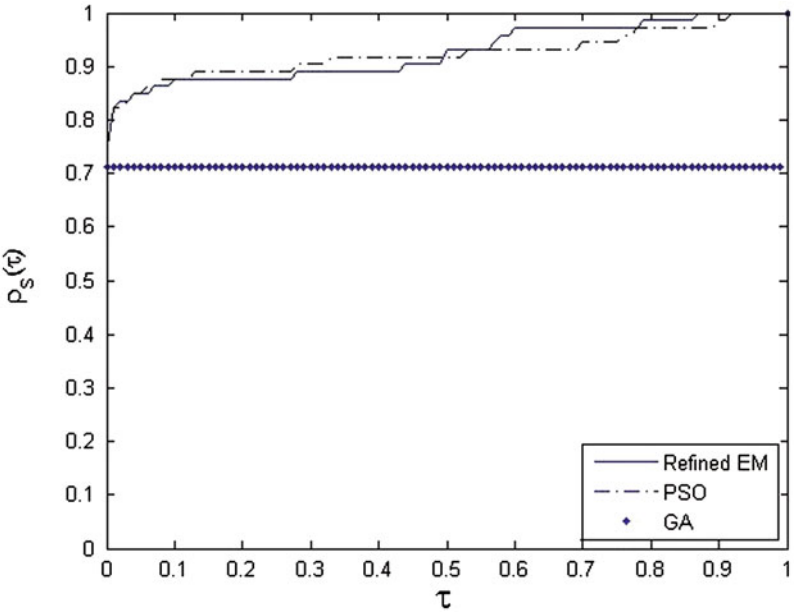


Fig. 4.10 Performance profile of the minimum objective function values

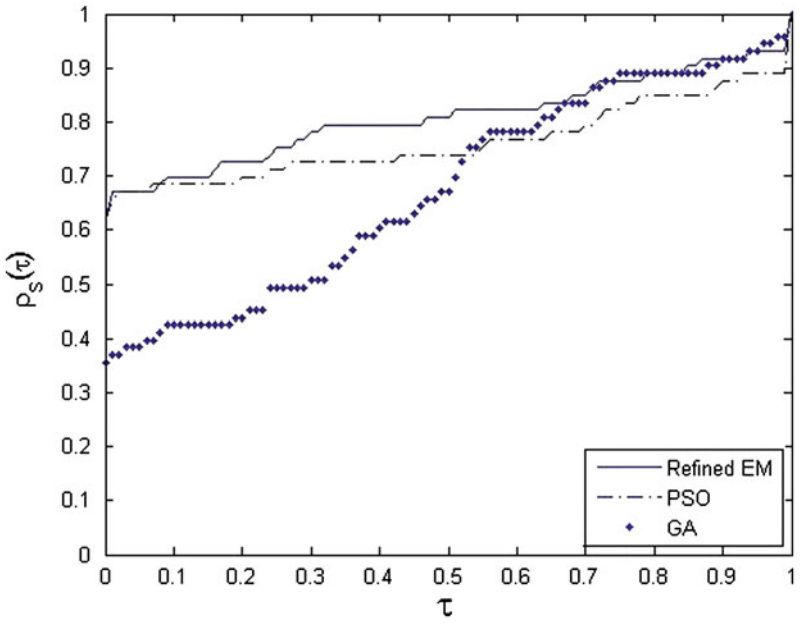


Fig. 4.11 Performance profile of the maximum objective function values

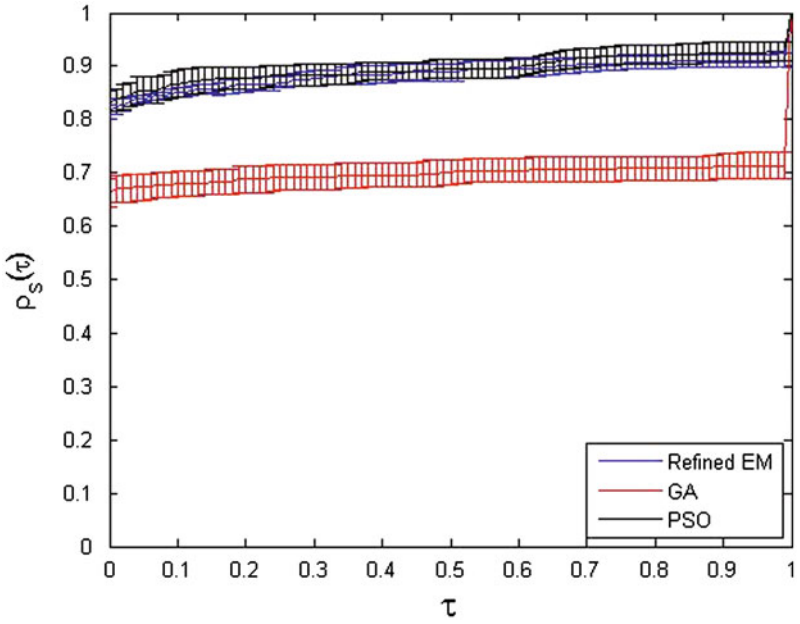


Fig. 4.12 Bands of 95 % confidence intervals for the performance profile of the objective function values

Table 4.3 Comparison of errors under the budget of 10,000 objective function evaluations

Problem	e_{EM}	e_{PSO}	e_{GA}
bunnag7	*0.0000 %*	*0.0000 %*	0.0203 %
bunnag8	*0.0881 %*	0.0929 %	59.4236 %
bunnag9	*0.0958 %*	6.3177 %	23.1730 %
bunnag10	5.4118 %	*4.3684 %*	59.9034 %
bunnag11	9.5616 %	*0.1000 %*	16.5073 %
bunnag12	*0.0001 %*	0.0002 %	87.2177 %
bunnag13	*0.0835 %*	4.2844 %	214.1359 %
ex2_1_10	*105.1648 %*	146.8418 %	211.2061 %
ex2_1_7	*0.0681 %*	0.0963 %	90.8804 %
hs044	0.0513 %	*0.0087 %*	0.0907 %
s279	*0.0443 %*	0.0924 %	0.1028 %
s280	0.0987 %	*0.0860 %*	0.1136 %
s359	1.8778 %	*0.1574 %*	N/A
s392	*3.1784 %*	5.3941 %	69.9198 %
tfi2	0.0924 %	*0.0462 %*	N/A

4) Refined EM outperforms PSO in five problems (*bunnag9*, *bunnag13*, *ex*2_1_7, *ex*2_1_10, and *s392*) while PSO outperforms refined EM in three problems (*bunnag10*, *bunnag11* and *s359*).
5) None of the solvers found a close-to-optimal solution for the linear programming problem *ex*2_1_10. This shows that their abilities to solve linear programming problems need to be improved.

To quantify the distances of the results given by refined EM, PSO, and GA from the known best solutions, in Table 4.3, the errors of refined EM, PSO, and GA are defined:

$$e_{EM} = \frac{|f^*_{EM} - f_{\text{glob}}|}{|f_{\text{glob}}|}, e_{PSO} = \frac{|f^*_{PSO} - f_{\text{glob}}|}{|f_{\text{glob}}|} \text{ and } e_{GA} = \frac{|f^*_{GA} - f_{\text{glob}}|}{|f_{\text{glob}}|}, \quad (4.27)$$

where f^*_{EM}, f^*_{PSO} and f^*_{GA} are the best values obtained by refined EM, PSO, and GA, respectively. f_{glob} is the known best solution and $f_{\text{glob}} \neq 0$.

In Table 4.3, N/A means GA cannot find a feasible solution for that problem. The *bold* numbers are the smallest errors for the corresponding problems.

Table 4.3 shows that

1) Refined EM solved 10 problems with error smaller than 0.1 %, they are considered as solved optimally by refined EM.
2) Other than these 10 problems, there are two problems whose errors are between 0.1 % and 5 %: *s359* and *s392*. There are two problems whose errors are between 5 % and 10 %: *bunnag10* and *bunnag11*.
3) There are nine problems in which refined EM achieves best solutions. And there are seven problems in which PSO achieves best solutions.

In summary, refined EM is able to solve problems with relative high level of difficulty to optimal or near optimal. Moreover, refined EM has achieved the best performance among all the three test solvers in terms of objective function values obtained.

Conclusions and Future Research

In this paper we have refined the original EM method which solves bounded constrained problems and made it capable to solve linearly constrained problems. We have applied this algorithm as well as other existing optimizers to different test problems in the literature and compared their performances. Our testing results support the claim that refined EM solves linearly constrained global optimization problems in an effective manner. Our computational results indicate that refined EM outperforms other two tested optimizers in terms of the number of function evaluations and/or the quality of best solutions obtained.

Future research will focus on refining EM to handle more complicated constraints, such as general convex constraints and nonlinear constraints. To handle general convex constraints, we need to redesign the initialization, local search, and movement procedures. Particularly in the movement procedure, a new method of calculating the range parameters needs to be developed. To handle general nonlinear constraints, calculating range parameters may not be useful since the feasible region is more complicated. Therefore a different method is needed.

Since the results in this paper have shown that the essential scheme utilized in the EM method is quite efficient as compared to other heuristics such as GA and PSO, good performance can be expected if the difficulty of handling different constraints is overcome.

References

Birbil, Ş.I. (2002). *Stochastic global optimization techniques. PhD Thesis*. Raleigh: North Carolina State University.

Birbil, Ş.I., & Fang, S.-C. (2002). An electromagnetism-like mechanism for global optimization. *Journal of Global Optimisation*, *25*(3), 263-282.

Birbil, Ş.I., Fang, S.-C., & Sheu, R.L. (2004). On the convergence of a population based global optimization algorithm. *Journal of Global Optimisation*, *30*(2), 301-318.

Bonnans, J.F., Gilbert, J.C., Lemaréchal, C., & Sagastizábal, C.A. (2006). *Numerical optimization: theoretical and practical aspects*. Berlin: Springer-Verlag.

Cowan, E.W. (1968). *Basic electromagnetism*. New York: Academic Press.

Dolan, E.D., & Moré, J.J. (2002). Benchmarking optimization software with performance profiles. *Math Program*, *91*, 201-213.

Eberhart R.C., & Kennedy, J. (1995). A new optimizer using particle swarm theory. Proceedings of the Sixth International Symposium on Micromachine and Human Science, Nagoya, Japan, 39-43

Floudas, C.A., & Pardalos, P.M. (1990). A collection of test problems for constrained global optimization algorithms. *Lecture notes in computer science 455*. Berlin: Springer.

GLOBAL Library, http://www.gamsworld.org/global/globallib.htm.

Goldberg, D.E. (1989). *Genetic algorithms in search, optimization & machine learning*. Boston: Addison-Wesley Longman Publishing Company, Inc.

Gould, N.I.M., Orban, D., & Toint Ph.L. (2013). CUTEr, a constrained and unconstrained test environment, revisited. http://cuter.rl.ac.uk.

Hart, W.E. (1994). *Adaptive global optimization with local search, PhD Thesis*. San Diego: University of California.

Ingber, L. (1994). Simulated annealing: practice versus theory. *Journal of Mathematical Computation Modeling*, *18*, 29-57.

Ji, Y., Zhang, K.-C., & Qu, S.-J. (2007). A deterministic global optimization algorithm. *Applied Mathematics and Computation*, *185*, 382-387.

Kan, A.H.G.R., & Timmer, G.T. (1987). Stochastic global optimization methods Part II: Multi level methods. *Math Program*, *39*, 57-78.

Kennedy, J., & Eberhart, R.C. (1995). Particle swarm optimization. Proceedings of IEEE International Conference on Neural Networks, IV, 1942-1948. Piscataway, NJ, IEEE Service Center.

Kolda, T.G., Lewis, R.M., & Torczon, V. (2003). Optimization by direct search: new perspectives on some classical and modern methods. *SIAM Review*, *45*, 385-482.

Michalewicz, Z. (1994). Evolutionary computation techniques for nonlinear programming problems. *International Transactions in Operational Research*, *1*:223-240.

Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs, 3rd edition*. Berlin: Springer.

Runarsson, T.P., & Yao, X. (2000). Stochastic ranking for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, *4*, 284-294.

Vanderbei, R.J. (2013). Benchmarks for nonlinear optimization. http://www.princeton.edu/rvdb bench.html.

Vaz, A.I.F., & Vicente, L.N. (2009). PSwarm: a hybrid solver for linearly constrained global derivative-free optimization. *Optimization Methods Software*, *24*(4-5), 669-685.

Wood, G.R. (1991). Multidimensional bisection and global optimization. *Computers & Mathematics with Applications*, *21*, 161-172

Zhang, Y., & Gao, L. (2001). On numerical solution of the maximum volume ellipsoid problem. *Society for Industrial and Applied Mathematics*, *14*, 53-76.

Chapter 5
The Price of Anarchy for a Network of Queues in Heavy Traffic

Shaler Stidham

Introduction

A recurring theme in the literature on optimal design and control of queueing systems is the distinction between individually and socially optimal solutions. Roughly speaking, *individual optimization* refers to a situation in which each individual user (customer) makes decisions (e.g., whether to join the system, which facility or route to choose) based on the cost (e.g., waiting time) incurred by the user. By contrast, in the case of *social optimization*, an agent acting on behalf of the collective of all customers makes decisions with the objective of minimizing the sum of the costs of all users. In the language of welfare economics, individually optimal solutions are Nash equilibria and socially optimal solutions are Pareto optima.

The literature on vehicle traffic flow contains some of the earliest references to individual and social optimization in the context of congestion phenomena. In a pioneering paper, Wardrop (1952) introduced the two optimality criteria in the setting of the *traffic assignment* problem, in which given origin/destination demands for travel in a road network are to be assigned to different routes, where the travel time on each link in the network is an increasing function of the flow on that link. Subsequent books and papers include Beckmann et al. (1956), Dafermos and Sparrow (1969), Dafermos (1980), and Dafermos and Nagourney (1984).

Naor (1969) brought the concepts of individual and social optimization to the attention of the queueing theory community in the context of an *M/M/1* queueing model in which arriving customers choose whether or not to join, based on real-time observation of the queue length. This paper initiated an extensive literature on this topic, with respect to both optimal design (in which queue lengths cannot be observed) and optimal control (in which queue lengths can be observed). Surveys and books on this topic include Sobel (1974), Stidham and Prabhu (1974), Crabill et al (1977), Serfozo (1981), Stidham (1978, 1984, 1985, 1988), Kitaev and Rykov (1995), Hassin and Haviv (2003), and Stidham (2009).

S. Stidham (✉)
Prof. Emeritus, Department of Statistics and Operations Research, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599, Chapel Hill, USA
e-mail: sandy@email.unc.edu

P. S. Pulat et al. (eds.), *Essays in Production, Project Planning and Scheduling,*
International Series in Operations Research & Management Science 200,
DOI 10.1007/978-1-4614-9056-2_5,

It is well known from the research on social and individual optimization that congestion tolls can be used to induce users concerned only with their own costs (e.g., travel or waiting times) to behave in a manner that minimizes the total cost incurred by all users. By setting the toll equal to the *external effect*, an individually optimal solution can be rendered socially optimal. In systems with congestion (e.g., queueing facilities and traffic networks) the external effect is the additional cost of congestion (e.g., waiting or travel time) imposed on other users by a user's decision to join a facility or traverse a link. (See Chap. 1 of Stidham (2009) for an introduction to these concepts in the setting of queueing systems.)

More recently, researchers in the telecommunications community have examined the problem of route assignment for traffic in a communication network (such as the Internet), using variants of the Wardrop model for road traffic networks. It is in this context of communication networks that attention has recently turned to establishing approximations and upper bounds for the ratio of the total congestion cost of an individually optimal solution to that of a socially optimal solution: the so-called *price of anarchy* (POA). (A more accurate term might be the price of free choice.) See, e.g., Roughgarden (2002, 2005, 2006), Roughgarden and Tardos (2002), Chau and Sim (2003), Schultz and Stier-Moses (2003), Perakis (2004), and Correa et al. (2004a, b, 2005). The primary goal of this research has been to find upper bounds that are independent of the topology of the network and only minimally dependent on the form of the congestion cost (e.g., travel time) on each link.

By establishing upper limits on the additional cost incurred under individual optimization relative to social optimization, research on the POA can provide some insight into the potential benefit of setting up a toll-collecting mechanism to achieve social optimization. Inasmuch as such mechanisms have their own administrative costs and inconveniences, it is important to have some idea of how much the total cost to society could be reduced by their imposition. Since the costs incurred by users are larger in heavy traffic, it is particularly useful to have bounds and approximations that hold in such circumstances.

In each of these settings—queueing theory, vehicular traffic-flow theory, and the theory of telecommunication networks—the object of interest is a *congestion network*: a network in which using a facility (e.g., traversing a link) has an associated cost per user which is an increasing function of the flow at that facility. In this paper we shall focus on facility congestion-cost functions of the form that arise when the facilities of a congestion network are modeled as queues with infinite waiting rooms. An important property of such queueing models is that the cost (e.g., waiting time) approaches infinity as the arrival rate (flow) approaches the service rate. As we shall see, this property has a crucial effect on the POA in the associated network. In particular, the POA does not grow without bound as the flow approaches the capacity, in contrast to the "conventional" estimates and upper bounds in the literature. In fact we are able to derive finite, closed-form expressions for the POA in heavy traffic for a variety of networks of queues.

The rest of the paper is organized as follows. In Sect. 5.2 we introduce our basic model of a congestion network. Following the literature on road-traffic and communication networks, our model is a deterministic network-flow model with nonlinear

cost functions on each link. The flows on the different routes are the decision variables. (In the language of queueing-optimization theory, ours is a design rather than a control model.) Sect. 5.3 provides an introduction and summary of previous results for the POA. (Sects. 5.2 and 5.3 follow closely the development in Stidham (2009)).

In Sect. 5.4 we focus on a network of parallel facilities. For the case in which each facility is an *M/M/1* queue we derive closed-form expressions for the individually and socially optimal flow allocations, their associated total costs, and the POA. We compare this result to the upper bound from the literature on the POA, and show that this upper bound can be quite weak, particularly as the traffic intensity increases. We pay particular attention to the heavy-traffic limit, in which our expression for the POA has a particularly simple form. We also show how to extend the heavy-traffic analysis to the case of parallel *GI/GI/1* facilities.

In Sect. 5.5 we continue the focus on heavy traffic and show how to extend our results for parallel facilities to more general networks.

(An earlier version of the present paper appeared in 2008 as a technical report (Stidham 2008)).

General Model of a Congestion Network

The system under study is a network consisting of a set $J = \{1, \ldots, n\}$ of service facilities and a set R of routes. Each route $r \in R$ consists of a subset of facilities, and we use the notation $j \in r$ to indicate that facility j is on route r.[1]

The network operates over a finite or infinite time interval, which we refer to as the *period*. At this stage, rather than specify a particular congestion model for each service facility, we prefer to describe the system in general terms, keeping structural and stochastic assumptions at a minimum. (We shall later consider specific examples.)

There is a single class of jobs (customers). The arrival rate—the average number of jobs entering the system per unit time during the period—is denoted by λ and is assumed to be a fixed parameter in our model. (Later we shall examine the behavior of the system as this parameter varies.) Each job that enters the system must be assigned to one of the routes, $r \in R$. Let λ_r denote the flow (average number of jobs per unit time) assigned to route r, $r \in R$. The flows, λ_r, $r \in R$, are decision variables, subject to the constraint that the total flow must equal Λ:

$$\sum_{r \in R} \lambda_r = \Lambda. \tag{5.1}$$

[1] This abstract characterization of a network is sufficiently general to include both classical models of networks of queues and road traffic networks, as well as more recent models of communication networks. In queueing-network models (e.g., a Jackson network), each queue (service facility) is modeled as a node, with a directed arc from node j to node k if service at queue j may be followed immediately by service at queue k. In communication-network models it is more common (and more natural) to consider each transmission link as a service facility, with a queue of jobs (messages or packets) at the node (router/server) at the head of the link, waiting to be transmitted. In road traffic networks, both nodes (intersections) and links (road segments between intersections) are service facilities in the sense that they are potential sources of congestion and waiting.

A flow λ_r on route r may be implemented by independently assigning each arriving job to route r with probability $p_r = \lambda_r/\Lambda$.

We assume that at each facility the average waiting cost per job is a function of the total flow (arrival rate) at that facility. The total flow at each facility $j \in J$ is denoted by ν_j and equals the sum of the flows on all the routes that use that facility. That is,

$$\nu_j = \sum_{r:j\in r} \lambda_r, \quad j \in J. \tag{5.2}$$

Let $G_j(\nu_j)$ denote the average waiting cost of a job at facility j, as a function of the flow, ν_j. We assume that $G_j(\nu_j)$ takes values in $[0, \infty]$ and is nondecreasing and differentiable in ν_j, $0 \le \nu_j < \infty$, with

$$G_j(\nu_j) \to \infty, \quad \text{as } \nu_j \to \infty, \quad j \in J. \tag{5.3}$$

The meaning of the word "average" depends on the specific model context. For example, it may mean a sample-path time average or (in the case of an infinite time period) the expectation of a steady-state random variable.

In general waiting cost is a measure of the disutility to a customer of the time spent waiting in the queue or in the queue plus in service. In some cases (see the following example) the waiting cost is proportional to the total time spent in queue and in service. This is a useful paradigm to keep in mind, but we prefer to keep the development as general as possible until specific assumptions are needed.

Example 1 As an example, suppose facility j is a single-server queue operating in steady state, with

$$G_j(\nu_j) = \mathrm{E}[h_j(\mathcal{W}_j(\nu_j))],$$

where $h_j(t)$ is the waiting cost incurred by a job that spends a length of time t at facility j and, for each $\nu_j \ge 0$, $\mathcal{W}_j(\nu_j)$ is the steady-state random waiting time in the system for the queueing system induced by an arrival rate equal to ν_j. For the special case of a *M/M/1* queue with an *(FCFS)* queue discipline and a linear waiting-cost function, $h_j(t) = h_j \cdot t$, $t \ge 0$, we have

$$G_j(\nu_j) = \begin{cases} \dfrac{h_j}{\mu_j - \nu_j}, & \text{if } \nu_j < \mu_j, \\ \infty, & \text{if } \nu_j \ge \mu_j, \end{cases} \tag{5.4}$$

where μ_j is the service rate (i.e., the reciprocal of the average service time).

Remark 1 The assumption that the waiting cost at each facility depends only on the flow at that facility puts restrictions on the applicability of our general model to a network of queues. In a classical *Jackson network* (Poisson arrival process and independent and exponentially distributed service times at the facilities) the facilities behave like independent *M/M/1* queues in steady state, with facility j having Poisson arrivals at rate ν_j, exponential service times with service rate μ_j, and average waiting

time $1/(\mu_j - \nu_j)$. But for a *generalized Jackson network*—that is, a network with a renewal arrival process and i.i.d. service times and *FCFS* queue discipline at each facility—the different facilities do not exhibit independent behavior in steady state. Hence, the expected steady-state waiting time at facility j is in general a function of the flows at other facilities as well as facility j. (There are some examples of networks of queues with general interarrival-time and service-time distributions and non-*FCFS* queue disciplines in which the facilities behave independently and the expected steady-state waiting times at each facility depend only on the average flow at that facility. These are sometimes called *Kelly* networks: see, e.g., Kelly (1979)). In Sect. 5.5 we shall consider a more general model in which the waiting cost at a facility may depend on other flows as well as the flow at that facility. This generalized model accommodates generalized Jackson networks.

Given the flows, ν_j, at the various facilities, the total waiting cost incurred by a job that follows a particular route is the sum of the resulting waiting costs at the facilities on that route:

$$\sum_{j:j\in r} G_j(\nu_j), \quad r \in R.$$

There may also be a toll δ_j which is charged to each customer who uses facility j, $j \in J$. In this case the total cost (or *full price*) for a job assigned to route r is given by

$$\sum_{j:j\in r} (\delta_j + G_j(\nu_j)).$$

The solution to the decision problem depends on who is making the decision and what criteria are being used. The decision may be made by the individual customers, each concerned only with his/her own waiting cost (*individual optimality*) or by an agent for the customers as a whole who might be interested in minimizing the aggregate waiting cost incurred by all customers per unit time (*social optimality*).

Socially Optimal Arrival Rates and Routes

First let us consider the problem from the point of view of social optimization. The objective is to choose a vector of route flows, $\lambda = (\lambda_r, r \in R)$, to minimize the average total cost per unit time,

$$C(\lambda) := \sum_{r\in R} \lambda_r \sum_{j:j\in r} G_j(\nu_j),$$

among all feasible flows. Let $\lambda^s = (\lambda_r^s, r \in R)$ denote a vector of socially optimal arrival rates.

Thus we have the following constrained minimization problem:

$$\begin{aligned} C(\lambda^s) = \quad & \min_{\{\lambda_r, r\in R; \nu_j, j\in J\}} C(\lambda) \\ \text{s.t.} \quad & \sum_{r\in R} \lambda_r = \Lambda, \\ & \sum_{r: j\in r} \lambda_r = \nu_j, j \in J, \\ & \lambda_r \geq 0, r \in R. \end{aligned}$$

The necessary and sufficient *Karush-Kuhn-Tucker (KKT)* conditions for this problem are:

$$\begin{aligned} \sum_{r\in R} \lambda_r &= \Lambda, \\ \sum_{j: j\in r} (G_j(\nu_j) + \nu_j G_j'(\nu_j)) &\geq \alpha, r \in R, \\ \lambda_r \left(\sum_{j: j\in r} (G_j(\nu_j) + \nu_j G_j'(\nu_j)) - \alpha \right) &= 0, r \in R, \\ \sum_{r: j\in r} \lambda_r &= \nu_j, j \in J, \\ \lambda_r &\geq 0, r \in R. \end{aligned}$$

Individually Optimal Arrival Rates and Routes

Individually optimal arrival rates are characterized by the Nash-equilibrium property that no individual user will have an incentive to deviate unilaterally from the equilibrium behavior implied by these rates.

Consider a given arrival rate Λ and a feasible allocation of flows, that is, $(\lambda_r, r \in R; \nu_j, j \in J)$ satisfying (5.1) and (5.2). Let π denote the minimum value of the full price on all routes $r \in R$. That is,

$$\pi = \min_{r\in R} \sum_{j: j\in r} (\delta_j + G_j(\nu_j))$$

Consider the behavior of a user entering the system with this arrival rate and flow allocation (a *marginal user*). At equilibrium, such a user will choose a route which offers the minimum full price. If, to the contrary, a route with a larger full price receives positive flow, then such a solution cannot be an equilibrium, since there is an incentive to divert some of this flow to a route that achieves the minimum price. Thus $\lambda_r^e > 0$ only if $\sum_{j: j\in r} (\delta_j + G_j(\nu_j)) = \pi$.

Therefore, an allocation of route flows, $\lambda = (\lambda_r, r \in R)$, is individually optimal (denoted $\lambda^e = (\lambda_r^e, r \in R)$) if and only if it satisfies the following system of equations and inequalities, for some $\pi > 0$:

$$\sum_{r \in R} \lambda_r = \Lambda, \tag{5.5}$$

$$\sum_{j:j \in r} (\delta_j + G_j(\nu_j)) \geq \pi, r \in R, \tag{5.6}$$

$$\lambda_r \left(\sum_{j:j \in r} (\delta_j + G_j(\nu_j)) - \pi \right) = 0, r \in R, \tag{5.7}$$

$$\sum_{r:j \in r} \lambda_r = \nu_j, j \in J, \tag{5.8}$$

$$\text{and } \lambda_r \geq 0, r \in R. \tag{5.9}$$

Together with conditions (5.6) and (5.9), the complementary-slackness conditions (5.7) ensure that $\pi = \min_{r \in R} \sum_{j:j \in r} (\delta_j + G_j(\nu_j))$ and that only the routes with the minimal price have positive flows. Note that it will be typical for an equilibrium solution to have more than one route sharing the minimal price and, therefore, having a positive flow.

It can be shown that the equilibrium conditions for an individually optimal allocation have a unique solution, by noting that the equilibrium conditions are the optimality conditions for the following minimization problem:

$$\begin{aligned} \min_{\{\lambda; \lambda_r, r \in R; \nu_j \in J\}} \quad & \sum_{j \in J} \int_0^{\nu_j} (\delta_j + G_j(\eta)) d\eta \\ \text{s.t.} \quad & \sum_{r \in R} \lambda_r = \Lambda, \\ & \sum_{r:j \in r} \lambda_r = \nu_j, \\ & \lambda_r \geq 0, r \in R. \end{aligned}$$

Since the objective function is jointly convex in $(\lambda, \nu_j, j \in J)$ and the constraints are linear, the *KKT* conditions are necessary and sufficient for a global minimum to this problem. These conditions have a unique solution and it is easily verified that they are identical to the equilibrium conditions, (5.5)–(5.9), for an individually optimal solution.

Note that the above minimization problem has the property that the marginal impact of flow at facility j on the objective function, namely, the integrand,

$$\delta_j + G_j(\eta),$$

equals the cost incurred by a marginal user. This observation explains intuitively why the optimality conditions coincide with the equilibrium conditions for an individually optimal solution.

We can interpret $G_j(\nu_j)$ as the *internal effect* of a marginal increase in the flow (arrival rate) ν_j at facility j. It is the portion of the marginal increase in aggregate waiting cost that is borne by a marginal user at facility j when the arrival rate is ν_j. Similarly, we can interpret the term $\nu_j G_j'(\nu_j)$ as the *external effect*: the rate of increase in waiting cost borne by all users as a result of a marginal increase in the arrival rate ν_j. By charging a toll at each facility j equal to the external effect—that is, $\delta_j = \nu_j G_j'(\nu_j)$—one can render the individually optimal allocation socially optimal.

The Price of Anarchy in a General Congestion Network

If charging tolls is not a practical option, then an individually optimal allocation will typically have a higher total cost than a socially optimal allocation. How bad (relative to the socially optimal allocation) can a toll-free individually optimal allocation be? More precisely, what is the worst-case behavior of the ratio of the total cost of an individually optimal allocation to the total cost of the socially optimal allocation? Using more colorful language (cf. Roughgarden and Tardos 2002): what is the "POA"? In this setting, "anarchy" means letting customers make their own route choices.

Previous research on the POA has focussed primarily on the derivation of upper bounds on the ratio of the total cost of an individually optimal allocation to the cost of a socially optimal allocation. These bounds apply over the full range of values of the parameter, λ, and in some cases are independent of the topology of the network. Relevant references are Dafermos (1980), Roughgarden (2002, 2005, 2006), Roughgarden and Tardos (2002), Chau and Sim (2003), Schultz and Stier-Moses (2003), Perakis (2004), and Correa et al. (2004, 2004, 2005). In this section we provide a brief overview of this research, inspired by the approach of Correa et al. (2005).

It will sometimes be convenient to work with an alternative formulation of the social optimization problem. Using the equality constraints (5.5) we can rewrite the objective function for social optimization as follows:

$$\sum_{r\in R}\lambda_r \sum_{j:j\in r} G_j(\nu_j) = \sum_{j\in J}\sum_{r:j\in r}\lambda_r G_j(\nu_j) = \sum_{j\in J}\nu_j G_j(\nu_j) \tag{5.10}$$

Now define the set, $\mathcal{F}$, of feasible vectors of facility flows, $\nu = (\nu_j, j \in J)$, as follows:

$$\mathcal{F} := \left\{ \nu | \exists \lambda_r \geq 0, \quad r \in R : \nu_j = \sum_{r:j\in r}\lambda_r, \quad j \in J; \sum_{r\in R}\lambda_r = \Lambda \right\}$$

Then the social optimization problem can be rewritten with decision variables, $\nu = (\nu_j, j \in J)$, as follows:

$$\min_{\{\nu\in f\}} C(\nu)$$

where $C(\nu) := \sum_{j\in J} \nu_j G_j(\nu_j)$, $\nu \in \mathcal{F}$.

Derivation of Upper Bounds: Review

We first establish a relation between the total cost, $C(\nu)$, of an arbitrary flow vector, $\nu \in \mathcal{F}$, and the total cost, $C(\nu^e)$, of the individually optimal flow vector, ν^e, which can be used to bound the ratio, $C(\nu^e)/C(\nu^s)$, where ν^s is the vector of socially optimal flows.

Lemma 1 An individually optimal vector of facility flow rates, ν^e, satisfies the relation,

$$C(\nu^e) = C(\nu) + \sum_{r\in R} \lambda_r \left(\pi - \sum_{j:j\in r} G_j(\nu_j) \right), \tag{5.11}$$

for all $\nu = (\nu_j, j \in J) \in \mathcal{F}$, where π is the *imputed cost* of an individually optimal flow (see (5)–(9)).

Proof. First note that it follows from (5.5)–(5.9) and (5.10) that

$$C(\nu^e) = \Lambda\pi. \tag{5.12}$$

Therefore, for all $\nu \in \mathcal{F}$,

$$\begin{aligned} C(\nu^e) = \Lambda\pi &= \sum_{r\in R} \lambda_r \pi \\ &= C(\nu) + \sum_{r\in R} \lambda_r \pi - \sum_{r\in R} \lambda_r \sum_{j:j\in r} G_j(\nu_j) \\ &= C(\nu) + \sum_{r\in R} \lambda_r \left(\pi - \sum_{j:j\in r} G_j(\nu_j) \right). \end{aligned}$$

Remark. As an immediate consequence of (5.6), (5.10), and Lemma 1, we have the inequality,

$$C(\nu^e) \le C(\nu) + \sum_{j\in J} \nu_j (G_j(\nu_j^e) - G_j(\nu_j)), \tag{5.13}$$

which has been used in the POA literature to bound the ratio, $C(\nu^e)/C(\nu^s)$. The inequality (5.13) can also be derived from the following *variational inequality*:

$$\sum_{j\in J} (\nu_j^e - \nu_j) G_j(\nu_j^e) \le 0, \quad \text{for all } \nu \in \mathcal{F} \tag{5.14}$$

(see, for example, Correa et al (2005)). But it is important to note that the relation (5.11) is stronger than the variational inequality (5.14).

Now we show how (5.11) (or (5.14)) can be used to find an upper bound on $C(\nu^e)/C(\nu^s)$.

Theorem 2 *Suppose there exists a constant* $\sigma < 1$ *such that*

$$\sum_{r \in R} \lambda_r \left(\pi - \sum_{j: j \in r} G_j(\nu_j) \right) \leq \sigma C(\nu^e), \tag{5.15}$$

for all $\nu \in \mathcal{F}$. *Then* $C(\nu^e) \leq (1-\sigma)^{-1} C(\nu^s)$.

Proof. For any $\nu \in \mathcal{F}$, using (5.11) and (5.15) we have

$$\begin{aligned} C(\nu^e) &\leq C(\nu) + \sum_{r \in R} \lambda_r \left(\pi - \sum_{j: j \in r} G_j(\nu_j) \right) \\ &\leq C(\nu) + \sigma C(\nu^e). \end{aligned}$$

Since this inequality holds for all $\nu \in \mathcal{F}$, it holds in particular for the socially optimal vector, ν^s. Thus

$$C(\nu^e) \leq C(\nu^s) + \sigma C(\nu^e),$$

from which the desired result follows.

Corollary 3 *Suppose there exists a constant* $\sigma < 1$ *such that*

$$\sum_{j \in J} \nu_j (G_j(\nu_j^e) - G_j(\nu_j)) \leq \sigma C(\nu^e), \tag{5.16}$$

for all $\nu \in \mathcal{F}$. *Then* $C(\nu^e) \leq (1-\sigma)^{-1} C(\nu^s)$.

The Price of Anarchy in a Network of Parallel Queues

In this section and the next we focus our attention on congestion networks in which the individual facilities are modeled as queues with infinite waiting rooms. We derive exact formulas and bounds which exploit the specific characteristics of the queues and/or the topology of the network and compare these to the upper bounds derived in the previous section for a general congestion network. We begin with a network of parallel queues and then (in the next section) extend our analysis to a general network of queues.

Parallel M/M/1 Queues

Consider a system consisting of n independent parallel facilities, with facility j behaving as an *M/M/1* queue in steady state with service rate μ_j, $j \in J = \{1, \ldots, n\}$.

There is a single class of customers arriving according to a Poisson process at fixed rate λ. The decision variables are the arrival rates, ν_j, $j \in J$, at the various facilities, where $\sum_{j\in J} \nu_j = \Lambda$. The waiting cost per customer at facility j is linear, with waiting-cost coefficient h_j, so that

$$G_j(\nu_j) = \frac{h_j}{\mu_j - \nu_j}, \quad j \in J.$$

(See Example 1 above.) Without loss of generality, assume that the facilities are numbered so that

$$\frac{h_1}{\mu_1} \le \frac{h_2}{\mu_2} \le \cdots \le \frac{h_n}{\mu_n}. \tag{5.17}$$

In this case explicit expressions are available for the individually optimal arrival rates, the socially optimal arrival rates, and the associated costs. (See Stidham (1971, 1985, 2009) for derivations).

Individually Optimal Arrival Rates and Costs

Let ν_j^e denote the individually optimal arrival rate (flow) at facility j, $j \in J$. Let $\nu^e = (\nu_j^e, j \in J)$. Define

$$s_k := \sum_{i=1}^{k} (\mu_i - h_i \mu_k / h_k), \quad k = 1, \ldots, n,$$

$$s_{n+1} := \sum_{i=1}^{n} \mu_i.$$

Note that it follows from the ordering (5.17) that

$$0 = s_1 \le s_2 \le \cdots \le s_n \le s_{n+1} = \sum_{i=1}^{n} \mu_i.$$

Then the individually optimal allocation is as follows: for $k = 1, \ldots, n$, if $s_k \le \Lambda \le s_{k+1}$, then

$$\nu_j^e = \begin{cases} \mu_j - \left(\dfrac{h_j}{\sum_{i=1}^{k} h_i}\right)\left(\sum_{i=1}^{k} \mu_i - \Lambda\right), & j = 1, \ldots, k, \\ 0, & j = k+1, \ldots, n, \end{cases}$$

and

$$C(\nu^e) = \frac{\left(\sum_{i=1}^{k} h_i\right)\left(\sum_{i=1}^{k} \mu_i\right)}{\left(\sum_{i=1}^{k} \mu_i\right) - \Lambda} - \sum_{i=1}^{k} h_i = \frac{\left(\sum_{i=1}^{k} h_i\right)\Lambda}{\left(\sum_{i=1}^{k} \mu_i\right) - \Lambda}.$$

Note that the cost of the individually optimal allocation equals the waiting cost per unit time at a single *M/M/1* facility with service rate $\mu = \sum_{i=1}^{k} \mu_i$ and waiting-cost rate $h = \sum_{i=1}^{k} h_i$.

Socially Optimal Arrival Rates and Costs

Let ν_j^s denote the socially optimal arrival rate (flow) at facility j, $j \in J$. Let $\nu^s = (\nu_j^s, j \in J)$. Define

$$r_k := \sum_{i=1}^{k} \left(\mu_i - \sqrt{h_i \mu_i \mu_k / h_k} \right), \quad k = 1, \ldots, n,$$

$$r_{n+1} := \sum_{i=1}^{n} \mu_i.$$

Note that it follows from the ordering (5.17) that

$$0 = r_1 \le r_2 \le \cdots \le r_n \le r_{n+1} = \sum_{i=1}^{n} \mu_i.$$

Then, for $k = 1, \ldots, n$, if $r_k \le \Lambda \le r_{k+1}$,

$$\nu_j^s = \begin{cases} \mu_j - \left(\dfrac{\sqrt{h_j \mu_j}}{\sum_{i=1}^{k} \sqrt{h_i \mu_i}} \right) \left(\sum_{i=1}^{k} \mu_i - \Lambda \right), & j = 1, \ldots, k, \\ 0, & j = k+1, \ldots, n, \end{cases}$$

and

$$C(\nu^s) = \frac{\left(\sum_{i=1}^{k} \sqrt{h_i \mu_i} \right)^2}{\left(\sum_{i=1}^{k} \mu_i \right) - \Lambda} - \sum_{i=1}^{k} h_i.$$

It follows that the ratio of the individually optimal total cost to the socially optimal total cost is given by

$$\frac{C(\nu^e)}{C(\nu^s)} = \frac{\left(\sum_{j \in J} h_j \right) \left(\sum_{j \in J} \mu_j \right) - (\mu - \Lambda) \sum_{j \in J} h_j}{\left(\sum_{j \in J} \sqrt{h_j \mu_j} \right)^2 - (\mu - \Lambda) \sum_{j \in J} h_j} \tag{5.18}$$

(where $\mu := \sum_{i=1}^{n} \mu_i$), provided Λ is large enough that all n facilities have positive arrival rates in both allocations. From this expression we see that the ratio, $C(\nu^e)/C(\nu^s)$ *decreases* in heavy traffic as $\Lambda \to \mu$, approaching the finite limit,

$$\frac{\left(\sum_{j \in J} h_j \right) \left(\sum_{j \in J} \mu_j \right)}{\left(\sum_{j \in J} \sqrt{h_j \mu_j} \right)^2}. \tag{5.19}$$

Note the interesting property that this expression is symmetric in $\{h_j, j \in J\}$ and $\{\mu_j, j \in J\}$.

For the special case of equal waiting-cost rates, $h_j = 1$, $j \in J$, (5.18) simplifies to

$$\frac{C(\nu^e)}{C(\nu^s)} = \frac{n\left(\sum_{j\in J}\mu_j\right) - n(\mu - \Lambda)}{\left(\sum_{j\in J}\sqrt{h_j\mu_j}\right)^2 - n(\mu - \Lambda)}. \tag{5.20}$$

Stidham (1985) analyzed the the ratio, $C(\nu^e)/C(\nu^s)$, for this case in heavy traffic. It follows from (5.20) (or from (5.19)) that

$$\lim_{\Lambda\uparrow\mu} C(\nu^e)/C(\nu^s) = n\mu/\left(\sum_{j\in J}\sqrt{\mu_j}\right)^2 \le n.$$

This expression attains its lower bound, $\lim_{\lambda\uparrow\mu} C(\nu^e)/C(\nu^s) = 1$, in the symmetric case in which the service rates are equal at all facilities. In this case (by symmetry) the socially optimal and the individually optimal allocations both assign equal arrival rates, $\lambda_j = \lambda/n$, to all facilities. The upper bound, n, is tight, as can be seen by considering the case

$$\mu_1 = \mu - n\epsilon,$$
$$\mu_j = \epsilon, \quad j = 2, \ldots, n,$$

and letting $\epsilon \to 0$.

In the non-heavy-traffic setting, the behavior of the ratio, $C(\nu^e)/C(\nu^s)$, can be quite complicated. We illustrate this complexity below by presenting a numerical example.

Numerical Example. To keep the exposition simple, we restrict attention to the case of equal waiting-cost rates, $h_j = 1$, $j \in J$.

Suppose the system consists of four *M/M/1* queues in parallel. The service rates are

$$\mu_1 = 20, \quad \mu_2 = 15, \quad \mu_3 = 10, \quad \mu_4 = 5.$$

For this system the breakpoints at which each facility starts receiving positive flow are

$$s_1 = 0, \quad s_2 = 5, \quad s_3 = 15, \quad s_4 = 30, \quad s_5 = 50,$$

for the individually optimal solution and

$$r_1 = 0, \quad r_2 = 2.6795, \quad r_3 = 8.6104, \quad r_4 = 19.2687, \quad r_5 = 50,$$

for the socially optimal solution. Fig. 5.1 compares the exact behavior of $C(\nu^e)/C(\nu^s)$ with that of the conventional upper bound on the POA (calculated in this case using Theorem 2). Note that, for this example, the maximum value of the ratio, $C(\nu^e)/C(\nu^s)$, equals 1.1 and occurs at $\lambda = 30$.

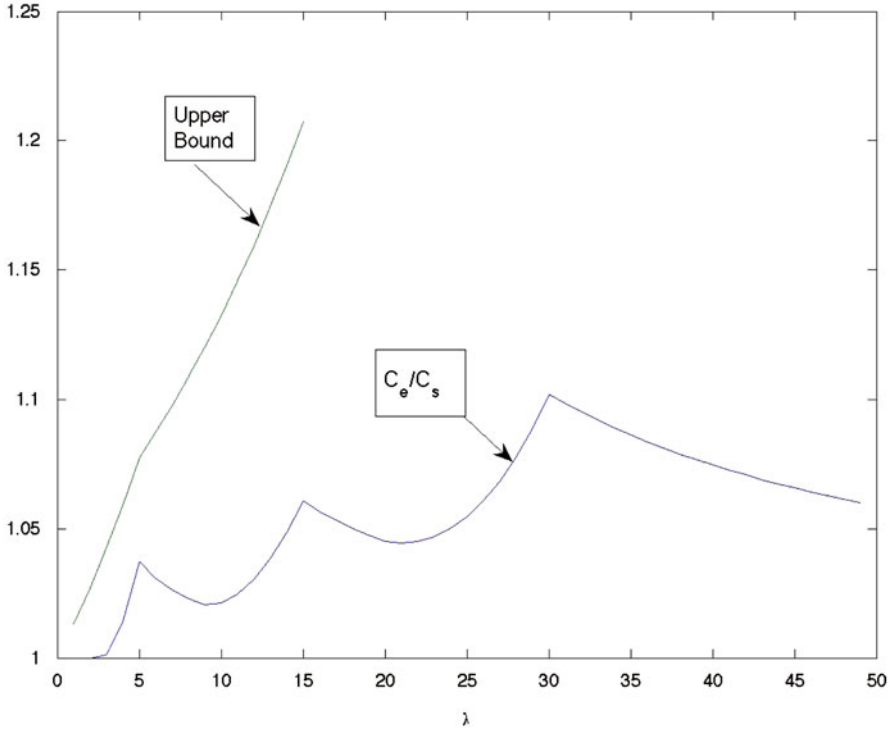


Fig. 5.1 Comparison of C_e/C_s with Upper Bound

By contrast the upper bound on the POA approaches ∞ as $\Lambda \to \mu$. To gain insight into why this is the case, let us look in more detail at the derivation of σ in Theorem 2 in the case of parallel facilities. It follows from Theorem 2 and (5.12) that any constant σ used in the upper bound on the POA must satisfy the inequality

$$\sigma \geq \frac{\sum_{j \in J} \nu_j^s(\pi - G_j(\nu_j^s))}{\Lambda \pi}.$$

Thus we see that any such upper bound must be at least as great as the upper bound derived by solving, for each facility j, a social optimization problem with linear utility in which the reward coefficient is π. Now π is the imputed reward that induces individually optimizing customers to join each facility j at a rate ν_j^e such that $\sum_{j \in J} \nu_j^e = \Lambda$. But the imputed reward (Lagrange multiplier) α required for social optimization is strictly larger than π. (See Chap. 7 of Stidham (2009)). Using π rather than $\alpha > \pi$ in the social optimization problem leads to facility arrival rates that are uniformly smaller than the individually optimal rates and therefore sum to a quantity strictly smaller than Λ. The result is an upper bound on the difference between $C(\nu^e)$ and $C(\nu^s)$ that is based on a systematic underestimate of $C(\nu^s)$.

This interpretation suggests an explanation for why the upper bound, $(1-\sigma)^{-1}$, on the ratio, $C(\nu^e)/C(\nu^s)$, increases to infinity as $\Lambda \to \mu$ in the case of parallel *M/M/1* facilities, whereas the ratio itself actually decreases.

Parallel GI/GI/1 Queues; Heavy Traffic

Now consider a system consisting of n independent parallel facilities, with each facility behaving as an *GI/GI/1* queue in steady state. Our model and assumptions are basically those of Shanthikumar and Xu (1997).

Customers arrive to the system according to a renewal process. The generic inter-arrival time is denoted by $A = X/\Lambda$, where X is a fixed nonnegative random variable with mean 1 and squared coefficient of variation (scv) C_a^2. Upon arrival each customer joins facility j with probability p_j, where $p_j \geq 0$, for $j \in J = \{1, \ldots, n\}$, independently of all other customers,and $\sum_{j\in J} p_j = 1$. The service times of the customers who join facility j form an i.i.d. sequence of random variables distributed as S_j with finite mean $1/\mu_j$ and scv $C_{S_j}^2$, $j \in J$. We assume that $\Lambda < \mu := \sum_{j\in J} \mu_j$.

Let $\{A_t, t = 1,2,\ldots\}$ be a sequence of i.i.d. random variables with the same distribution as A and let Z_j be an independent geometric random variable with mean $1/p_j$, $j \in J$. Define

$$A^{(j)} := \sum_{t=1}^{Z_j} A_t, \quad j \in J.$$

Then facility j behaves as a *GI/GI/1* queue with a renewal arrival process which has a generic inter-arrival time $A^{(j)}$ with mean $1/(\Lambda p_j)$ and scv $p_j(C_a^2 - 1) + 1$, $j \in J$.

The decision variables are the routing probabilities, p_j, $j \in J$, or, equivalently, the arrival rates, ν_j, where $\nu_j = \Lambda p_j$, $j \in J$, and $\sum_{j\in J} \nu_j = \Lambda$. The waiting cost per customer at facility j is linear, with waiting-cost coefficient h_j, so that

$$G_j(\nu_j) = h_j W_j(\nu_j), \quad j \in J,$$

where $W_j(\nu_j)$ is the steady-state expected waiting time (in queue plus in service) of a customer at facility j, $j \in J$. The total waiting cost per unit time is therefore given by

$$C(\nu) = \sum_{j\in J} \nu_j G_j(\nu_j) = \sum_{j\in J} \nu_j h_j W_j(\nu_j),$$

where $\nu = (\nu_j, j \in J)$. As usual, we denote the individually optimal and socially optimal flow allocations by ν^e and ν^s, respectively.

We use variants of techniques from Shanthikumar and Xu (1997) to derive upper and lower bounds on $W_j(\nu_j)$, $j \in J$, which are the basis for the derivation of the POA in heavy traffic, that is,

$$\lim_{\Lambda\to\mu} C(\nu^e)/C(\nu^s).$$

The main result of this section is that the POA for a system of parallel *GI/GI/1* facilities coincides with the POA for a system of parallel *M/M/1* facilities with a modified waiting-cost function.

Let the arrival rate, Λ, be given. Consider a particular facility $j \in J$ with a given routing probability, p_j, and corresponding flow rate, $\nu_j = \Lambda p_j$. Define

$$\hat{W}_j(\nu_j) := \frac{\mu_j(C_a^2 - 1) + \Lambda(C_{S_j}^2 + 1)}{2\Lambda(\mu_j - \nu_j)} = \frac{f_j}{\mu_j - \nu_j}, \tag{5.21}$$

where

$$f_j := \frac{\mu_j(C_a^2 - 1) + \Lambda(C_{S_j}^2 + 1)}{2\Lambda}.$$

We shall use $\hat{W}_j(\nu_j)$ as a heavy-traffic approximation of $W_j(\nu_j)$. Following Shanthikumar and Xu (1997) it can be shown that

$$\hat{W}_j(\nu_j) - \left(\frac{C_{S_j}^2}{2\mu_j} + \frac{C_a^2 - 1 + 2\alpha + 2\beta}{2\Lambda}\right) \le W_j(\nu_j) \le \hat{W}_j(\nu_j) + \left(\frac{1}{2\mu_j} + \frac{1}{\nu_j}\right), \tag{5.22}$$

for all $\nu_j \in (0, \mu_j)$, where α and β are positive, finite constants defined in Shanthikumar and Xu (1997). Note that the lower bound on $W_j(\nu_j) - \hat{W}_j(\nu_j)$ is independent of ν_j, whereas the upper bound approaches ∞ as $\nu_j \to 0$ and approaches 0 as $\nu_j \to \mu_j$. Since, for any feasible flow allocation, it must be the case that $\nu_j \to \mu_j$ as $\Lambda \to \mu = \sum_{j \in J} \mu_j$, this upper bound will suffice in heavy traffic. This observation leads to the following lemma.

Lemma 4 *Let $\epsilon > 0$ be given. Then there exists a $\delta > 0$ such that*

$$\hat{W}_j(\nu_j) - l_j \le W_j(\nu_j) \le \hat{W}_j(\nu_j) + u_j, \tag{5.23}$$

for all $\Lambda \in (\mu - \delta, \mu)$, where

$$l_j := \frac{C_{S_j}^2}{2\mu_j} + \frac{C_a^2 - 1 + 2\alpha + 2\beta}{2\Lambda},$$

$$u_j := \frac{1}{2\mu_j} + \frac{1}{\mu_j - \epsilon}$$

For all feasible flow allocations, ν, let

$$\hat{C}(\nu) := \sum_{j \in J} \tilde{h}_j \left(\frac{\nu_j}{\mu_j - \nu_j}\right), \tag{5.24}$$

where

$$\tilde{h}_j := h_j f_j. \tag{5.25}$$

Note that $\hat{C}(\nu)$ coincides with the total cost per unit time in a system consisting of n parallel *M/M/1* facilities, where facility j has service rate μ_j and linear waiting-cost function with $\tilde{h}_j$ as the waiting cost per customer per unit time, $j \in J$.

The following lemma restates a result from Shanthikumar and Xu (1997) (cf. (27)).

Lemma 5 *There exist finite constants, L and U, such that for all feasible flow allocations, ν,*

$$\hat{C}(\nu) - L \leq C(\nu) \leq \hat{C}(\nu) + U.$$

F rom (5.22) we obtain the following upper and lower bounds on $L_j(\nu_j) = \nu_j W_j(\nu_j)$ for all $\nu_j \in (0, \mu_j)$:

$$\nu_j \hat{W}_j(\nu_j) - \left(\frac{C_{S_j}^2}{2} + \frac{p_j(C_a^2 - 1 + 2\alpha + 2\beta)}{2} \right) \leq \nu_j W_j(\nu_j) \leq \nu_j \hat{W}_j(\nu_j) + \frac{3}{2}.$$

Lemma 5 follows directly from these inequalities.

For given Λ, let $\hat{\nu}^s = (\hat{\nu}_j^s, j \in J)$ denote a vector of flows that minimizes the approximate cost function, $\hat{C}(\nu)$, subject to $\sum_{j \in J} \nu_j = \Lambda$. As a consequence of Lemma 5 we have the following theorem (cf. Theorem 3 in Shanthikumar and Xu (1997)), which demonstrates that $\hat{\nu}^s$ is *strongly asymptotically (socially) optimal* in heavy traffic.

Theorem 6 *For all $\Lambda < \mu$,*

$$0 \leq C(\hat{\nu}) - C(\nu^s) \leq L + U, \tag{5.26}$$

and therefore

$$\frac{C(\hat{\nu}^s)}{C(\nu^s)} \to 1, \text{ as } \Lambda \to \mu. \tag{5.27}$$

Now we turn our attention to individually optimal flows. We show that the same approximation can be used to construct asymptotically individually optimal flows.

For given Λ, let $\hat{\nu}^e = (\hat{\nu}_j^e, j \in J)$ denote a vector of flows that is individually optimal for the approximate cost function, $\hat{C}(\nu)$, subject to $\sum_{j \in J} \nu_j = \Lambda$.

Theorem 7

$$\frac{C(\hat{\nu}^e)}{C(\nu^e)} \to 1, \quad \text{as } \Lambda \to \mu. \tag{5.28}$$

Proof. Assuming that Λ is large enough that all facilities have positive flows, and using (5.21), we can write the equilibrium conditions satisfied by $\hat{\nu}^e$ as

$$\frac{h_j f_j}{\mu_j - \nu_j} = \pi, \quad j \in J, \tag{5.29}$$

$$\sum_{j \in J} \nu_j = \Lambda, \tag{5.30}$$

for some $\pi > 0$. Similarly, the equilibrium conditions satisfied by ν^e, the individually optimal vector of flows for the original problem, take the form

$$h_j W_j(\nu_j) = \pi, \quad j \in J, \tag{5.31}$$

$$\sum_{j \in J} \nu_j = \Lambda. \tag{5.32}$$

From (5.23) we obtain the following inequalities,

$$\frac{h_j f_j}{\mu_j - \nu_j} - h_j l_j \leq h_j W_j(\nu_j) \leq \frac{h_j f_j}{\mu_j - \nu_j} + h_j u_j, \quad j \in J, \tag{5.33}$$

for all $\Lambda \in (\mu - \delta, \mu)$.

First we consider the Eqs. (5.29) and (5.31) for a fixed, arbitrary value of the parameter π. Let $\nu_j(\pi)$ denote the solution to Eq. (31) and let $\hat{\nu}_j(\pi)$ denote the solution to Eq. (5.29), $j \in J$. Solving Eq. (5.29) for ν_j in terms of π yields

$$\hat{\nu}_j(\pi) = \mu_j - \frac{h_j f_j}{\pi}, \quad j \in J. \tag{5.34}$$

Now consider the following equations,

$$\frac{h_j f_j}{\mu_j - \nu_j} - h_j l_j = \pi, \quad j \in J,$$

$$\frac{h_j f_j}{\mu_j - \nu_j} + h_j u_j = \pi, \quad j \in J,$$

noting that the solution to the former is $\hat{\nu}_j(\pi + h_j l_j)$ and the solution to the latter is $\hat{\nu}_j(\pi - h_j u_j)$. It follows from (33) that

$$\hat{\nu}_j(\pi - h_j u_j) \leq \nu_j(\pi) \leq \hat{\nu}_j(\pi + h_j l_j), \quad j \in J. \tag{5.35}$$

Let $\gamma := \max_{j \in J} h_j u_j$, $\delta := \max_{j \in J} h_j l_j$. Then (5.35) implies that

$$\hat{\nu}_j(\pi - \gamma) \leq \nu_j(\pi) \leq \hat{\nu}_j(\pi + \delta), \quad j \in J. \tag{5.36}$$

It follows from (5.36) that

$$C(\hat{\nu}(\pi - \gamma)) \leq C(\nu(\pi)) \leq C(\hat{\nu}(\pi + \delta)). \tag{5.37}$$

Dividing all three terms by $C(\hat{\nu}(\pi))$ yields

$$\frac{C(\hat{\nu}(\pi-\gamma))}{C(\hat{\nu}(\pi))} \le \frac{C(\nu(\pi))}{C(\hat{\nu}(\pi))} \le \frac{C(\hat{\nu}(\pi+\delta))}{C(\hat{\nu}(\pi))}. \tag{5.38}$$

Our intermediate goal is to show that

$$\frac{C(\nu(\pi))}{C(\hat{\nu}(\pi))} \to 1, \quad \text{as } \pi \to \infty.$$

It suffices to show that both the upper and lower bounds in (5.38) approach one as $\pi \to \infty$. We shall do this by approximating $C(\,\cdot\,)$ by $\hat{C}(\,\cdot\,)$.

From (5.24) and (5.34) we obtain the following simple formula for $\hat{C}(\hat{\nu}(\pi))$:

$$\hat{C}(\hat{\nu}(\pi)) = \pi\mu - \sum_{j\in J} h_j f_j. \tag{5.39}$$

Lemma 5 implies that

$$\begin{aligned} \hat{C}(\hat{\nu}(\pi)) - L \le C(\hat{\nu}(\pi)) &\le \hat{C}(\hat{\nu}(\pi)) + U, \\ \hat{C}(\hat{\nu}(\pi-\gamma)) - L \le C(\hat{\nu}(\pi-\gamma)) &\le \hat{C}(\hat{\nu}(\pi-\gamma)) + U, \end{aligned}$$

from which we obtain the following inequalities:

$$\frac{\hat{C}(\hat{\nu}(\pi-\gamma)) - L}{\hat{C}(\hat{\nu}(\pi)) + U} \le \frac{C(\hat{\nu}(\pi-\gamma))}{C(\hat{\nu}(\pi))} \le \frac{\hat{C}(\hat{\nu}(\pi-\gamma)) + U}{\hat{C}(\hat{\nu}(\pi)) - L}.$$

Substituting for $\hat{C}$ from (5.39) yields

$$\frac{(\pi-\gamma)\mu - (\sum_{j\in J} h_j f_j) - L}{\pi\mu - (\sum_{j\in J} h_j f_j) + U} \le \frac{C(\hat{\nu}(\pi-\gamma))}{C(\hat{\nu}(\pi))} \le \frac{(\pi-\gamma)\mu - (\sum_{j\in J} h_j f_j) + U}{\pi\mu - (\sum_{j\in J} h_j f_j) - L}.$$

It is easily seen that both the lower and the upper bound approach one as $\pi \to \infty$. Therefore,

$$\frac{C(\hat{\nu}(\pi-\gamma))}{C(\hat{\nu}(\pi))} \to 1, \quad \text{as } \pi \to \infty.$$

A similar argument shows that

$$\frac{C(\hat{\nu}(\pi+\delta)}{C(\hat{\nu}(\pi))} \to 1, \quad \text{as } \pi \to \infty.$$

It follows from (5.38) that

$$\frac{C(\nu(\pi))}{C(\hat{\nu}(\pi))} \to 1, \quad \text{as } \pi \to \infty, \tag{5.40}$$

which is the desired intermediate result.

Now we return to our ultimate goal of proving (5.23):

$$\frac{C(\hat{\nu}^e)}{C(\nu^e)} \to 1, \quad \text{as } \Lambda \to \mu.$$

For $0 \le \Lambda < \mu$, let $\pi(\Lambda)$ be the solution to

$$\sum_{j \in J} \nu_j(\pi) = \Lambda,$$

and let $\hat{\pi}(\Lambda)$ be the solution to

$$\sum_{j \in J} \hat{\nu}_j(\pi) = \Lambda.$$

Then

$$\frac{C(\hat{\nu}^e)}{C(\nu^e)} = \frac{C(\hat{\nu}(\hat{\pi}(\Lambda)))}{C(\nu(\pi(\Lambda)))} = \frac{C(\hat{\nu}(\hat{\pi}(\Lambda)))}{C(\hat{\nu}(\pi(\Lambda)))} \cdot \frac{C(\hat{\nu}(\pi(\Lambda)))}{C(\nu(\pi(\Lambda)))}.$$

Since $\pi(\Lambda) \to \infty$ as $\Lambda \to \mu$,

$$\frac{C(\hat{\nu}(\pi(\Lambda)))}{C(\nu(\pi(\Lambda)))} \to 1, \text{ as } \Lambda \to \mu,$$

by (5.40). It remains to show that

$$\frac{C(\hat{\nu}(\hat{\pi}(\Lambda)))}{C(\hat{\nu}(\pi(\Lambda)))} \to 1, \text{ as } \Lambda \to \mu. \tag{5.41}$$

From (5.34) and (5.36) we obtain the following inequalities, for all $\pi > 0$:

$$\mu_j - \frac{h_j f_j}{\pi - \gamma} \le \nu_j(\pi) \le \mu_j - \frac{h_j f_j}{\pi + \delta}, \quad j \in J.$$

Summing over $j \in J$ yields

$$\mu - \frac{\sum_{j \in J} h_j f_j}{\pi - \gamma} \le \sum_{j \in J} \nu_j(\pi) \le \mu - \frac{\sum_{j \in J} h_j f_j}{\pi + \delta}. \tag{5.42}$$

For given λ, let $\hat{\pi}_u(\lambda)$ denote the solution to

$$\mu - \frac{\sum_{j \in J} h_j f_j}{\pi - \gamma} = \Lambda,$$

and let $\hat{\pi}_l(\lambda)$ denote the solution to

$$\mu - \frac{\sum_{j \in J} h_j f_j}{\pi + \delta} = \Lambda.$$

Then

$$\hat{\pi}_l(\Lambda) = \frac{\sum_{j\in J} h_j f_j}{\mu - \Lambda} - \delta = \hat{\pi}(\Lambda) - \delta,$$

and

$$\hat{\pi}_u(\Lambda) = \frac{\sum_{j\in J} h_j f_j}{\mu - \Lambda} + \gamma = \hat{\pi}(\Lambda) + \gamma.$$

From these observations and (5.42) it follows that

$$\hat{\pi}(\Lambda) - \delta \le \pi(\Lambda) \le \hat{\pi}(\Lambda) + \gamma,$$

so that

$$\frac{\hat{\pi}(\Lambda)}{\pi(\Lambda)} \to 1, \text{ as } \Lambda \to \mu. \tag{5.43}$$

Recall that our goal is to show that

$$\frac{C(\hat{\nu}(\hat{\pi}(\Lambda)))}{C(\hat{\nu}(\pi(\Lambda)))} \to 1, \text{ as } \Lambda \to 1. \tag{5.44}$$

Now

$$\frac{C(\hat{\nu}(\hat{\pi}(\Lambda)))}{C(\hat{\nu}(\pi(\Lambda)))} = \frac{C(\hat{\nu}(\hat{\pi}(\Lambda)))}{\hat{C}(\hat{\nu}(\hat{\pi}(\Lambda)))} \cdot \frac{\hat{C}(\hat{\nu}(\hat{\pi}(\Lambda)))}{\hat{C}(\hat{\nu}(\pi(\Lambda)))} \cdot \frac{\hat{C}(\hat{\nu}(\pi(\Lambda)))}{C(\hat{\nu}(\pi(\Lambda)))}.$$

The first and third factors approach one as $\Lambda \to \mu$ by Lemma 5. It remains to show that

$$\frac{\hat{C}(\hat{\nu}(\hat{\pi}(\Lambda)))}{\hat{C}(\hat{\nu}(\pi(\Lambda))} \to 1, \text{ as } \Lambda \to \mu. \tag{5.45}$$

Recall (cf. (5.39)) that

$$\hat{C}(\hat{\nu}(\pi)) = \pi\mu - \sum_{j\in J} h_j f_j.$$

Therefore,

$$\frac{\hat{C}(\hat{\nu}(\hat{\pi}(\Lambda)))}{\hat{C}(\hat{\nu}(\pi(\Lambda)))} = \frac{\hat{\pi}(\Lambda)\mu - \sum_{j\in J} h_j f_j}{\pi(\Lambda)\mu - \sum_{j\in J} h_j f_j}.$$

Since $\pi(\Lambda) \to \infty$ and $\hat{\pi}(\Lambda) \to \infty$ as $\Lambda \to \mu$, it follows from (5.43) that (5.45) holds.

This completes the proof of the theorem.

Theorem 8 *The POA for the system of parallel GI/GI/1 facilities is given by*

$$\lim_{\Lambda\to\mu} C(\nu^e)/C(\nu^s) = \lim_{\Lambda\to\mu} \hat{C}(\nu^e)/\hat{C}(\nu^s) = \frac{\left(\sum_{j\in J} \tilde{h}_j\right)\left(\sum_{j\in J} \mu_j\right)}{\left(\sum_{j\in J} \sqrt{\tilde{h}_j \mu_j}\right)^2},$$

where $\tilde{h}_j$ is given by (25), $j \in J$.

Proof. First observe that

$$\frac{C(\nu^e)}{C(\nu^s)} = \frac{C(\nu^e)}{C(\hat{\nu}^e)} \cdot \frac{C(\hat{\nu}^e)}{\hat{C}(\hat{\nu}^e)} \cdot \frac{\hat{C}(\hat{\nu}^e)}{\hat{C}(\hat{\nu}^s)} \cdot \frac{\hat{C}(\hat{\nu}^s)}{C(\hat{\nu}^s)} \cdot \frac{C(\hat{\nu}^s)}{C(\nu^s)}.$$

The first and fifth ratios approach one as $\Lambda \to \mu$ by Theorems 7 and 6, respectively. The second and fourth ratios approach one as $\Lambda \to \mu$ by Lemma 5. Therefore,

$$\lim_{\Lambda\to\mu} \frac{C(\nu^e)}{C(\nu^s)} = \lim_{\Lambda\to\mu} \frac{\hat{C}(\hat{\nu}^e)}{\hat{C}(\hat{\nu}^s)} = \frac{\left(\sum_{j\in J} h_j f_j\right)\left(\sum_{j\in J} \mu_j\right)}{\left(\sum_{j\in J} \sqrt{h_j f_j \mu_j}\right)^2},$$

where the last equality follows from substituting the explicit expressions for $\hat{\nu}^e$ and $\hat{\nu}^s$ (cf. Sect. 5.4.1 into (5.24) and taking the limit as $\Lambda \to \mu$.

The implication of Theorem 8 is that the POA for a system of parallel *GI/GI/1* facilities coincides with the POA for a system of parallel *M/M/1* facilities with h_j replaced by $\tilde{h}_j = h_j f_j$, $j \in J$.

General Network of Queues; Heavy Traffic

In this section we return to a general congestion network and show how the heavy-traffic results for parallel queues can be extended to certain networks satisfying weak regularity conditions which hold for most queueing models.

We shall work with a more general model of a congestion network than the one introduced in Sect. 5.5. In the generalized model the waiting cost at each facility j may depend on the entire allocation of flows to routes, $\lambda := (\lambda_r, r \in R)$, rather than just on the flow at facility j, that is, $\nu_j = \sum_{R:\, j\in r} \lambda_r$.

(Recall that the steady-state expected waiting time at a facility in a network of queues depends only on the flow at that facility when interarrival times and service times at all facilities are exponentially distributed (a Jackson network), but not for a generalized Jackson network. See Remark 1 in Sect. 5).

Let $G_j(\lambda)$ denote the average waiting cost of a job at facility j, $j \in J$, as a function of the allocation vector, λ. Let $H_j(\lambda)$ denote the average waiting cost incurred per unit time at facility j, $j \in J$, as a function of the allocation vector, λ. In our standard model it follows from the formula, $H = \lambda G$ (cf., e.g., Chap. 6 of El-Taha and Stidham (1998)), that $H_j(\lambda) = \nu_j G_j(\lambda)$. In general we shall make the following assumption about the network and the waiting cost functions.

Assumption 1 For each facility j, the feasible set for ν_j is $A_j = [0, \mu_j]$, where μ_j is a positive constant. For a given value of the total arrival rate $\Lambda = \sum_{r \in R} \lambda_r$, the feasible set for $\lambda = (\lambda_r, r \in R)$ is

$$\mathcal{L} := \left\{ \lambda | \lambda_r \geq 0, \quad r \in R; \nu_j = \sum_{r: j \in r} \lambda_r, \quad 0 \leq \nu_j \leq \mu_j, \quad j \in J; \sum_{r \in R} \lambda_r = \lambda \right\}$$

The function $G_j(\lambda)$ takes values in $[0, \infty]$, with $G_j(\lambda) = \infty$ for all λ such that $\nu_j = \mu_j$ and $G_j(\lambda) \to \infty$ for any sequence of feasible values of λ such that $\nu_j \to \mu_j$.

(Note that we extend the domain of ν_j to the closed interval $[0, \mu_j]$ by setting $G_j(\lambda) = \infty$ for λ such that $\nu_j = \mu_j$).

The waiting-cost functions for most classical queueing models satisfy this assumption, with μ_j as the service rate of the facility. (See Example 1 in Sect. 5.2.) Because our interest here is in flows in a network, however, we shall refer to μ_j as the *capacity* of facility j.

For a feasible flow vector $\lambda \in \mathcal{L}$, the total cost per unit time is given by:

$$C(\lambda) := \sum_{j \in J} \nu_j G_j(\lambda).$$

For a given value of Λ, the social optimization problem may now be written as follows:

$$\min_{\{\lambda \in \mathcal{L}\}} C(\lambda).$$

Let $\lambda^s = (\lambda_r^s, r \in R)$ denote a socially optimal allocation of flows, that is, an allocation that achieves the above minimum. Let $\lambda^e = (\lambda_r^e, r \in R)$ denote an individually optimal allocation.

The Nash-equilibrium property of an individually optimal allocation can be expressed as follows (cf. Section 5.2). A flow vector $\lambda \in \mathcal{L}$ is individually optimal if and only if there exists a constant π such that:

$$\sum_{j: j \in r} G_j(\lambda) \geq \pi, \quad r \in R, \tag{5.46}$$

$$\lambda_r \left(\sum_{j: j \in r} G_j(\lambda) - \pi \right) = 0 = 0, \quad r \in R. \tag{5.47}$$

Our interest is in the heavy-traffic behavior of the ratio, $C(\lambda^e)/C(\lambda^s)$.

In the remainder of this section we shall restrict our attention to a classical Ford-Fulkerson network (*FFN*). A Ford-Fulkerson network consists of a set of nodes $\mathcal{N}$ and a set of (directed) links $\mathcal{A}$. (The links correspond to the facilities in our general network model.) Each link $a \in \mathcal{A}$ (equivalently, each facility $j \in J$) corresponds to a pair (i, k) of nodes, $i \in \mathcal{N}$, $k \in \mathcal{N}$, such that the flow in link a proceeds from node i to node k. One node $s \in \mathcal{N}$ is designated as the source and another node $t \in \mathcal{N}$ is

designated as the sink. In this network a route r consists of an ordered set of links connecting the source to the sink and the set R consists of all such routes.

We shall continue to use the notation and definitions for a general network (namely, a set of facilities J and a set of routes R), but it is important to note that our results only hold for a classical Ford-Fulkerson network. In particular, it is essential that *all* routes between the source and the sink be included in the set R.

We shall need the following definitions from the theory of (deterministic) network flows (cf. Bertsekas (1998)). A *cut* is a subset of facilities $j \in J$ such that each route $r \in R$ from s to t contains at least one facility j in the subset. Thus the removal of the subset from the set J of all facilities makes it impossible to find a feasible flow for any positive Λ. A *minimal cut* is a cut (denoted $\mathcal{C}$) whose total capacity, $\mu := \sum_{j\in\mathcal{C}} \mu_j$, is no larger than that of any other cut.

The Max-Flow-Min-Cut Theorem (cf. Bertsekas (1998)) states that the maximal feasible total flow through the network equals the total capacity of a minimal cut. Thus, the set of feasible values for the parameter Λ is $[0, \mu]$.

We can therefore write the heavy-traffic POA as

$$\lim_{\Lambda\to\mu} \frac{C(\lambda^e)}{C(\lambda^s)}. \tag{5.48}$$

To simplify the analysis we shall make the following technical assumption.

Assumption 2 The minimal cut (denoted $\mathcal{C}$) is unique. A feasible allocation, $\lambda \in \mathcal{L}$, can saturate facility j (that is, $\nu_j = \mu_j$) if and only if $j \in \mathcal{C}$.

For the remainder of this section we shall confine our attention to a general congestion network with flows and waiting cost functions satisfying Assumption 1 in which the underlying network is a Ford-Fulkerson network satisfying Assumption 2. We shall refer to such a network as a *FFGCN*.

We shall show (Theorem 10 below) that the behavior of the heavy-traffic limit (5.48) of the POA for a *FFGCN* is determined completely by the waiting-cost functions, $H_j(\lambda)$, at the facilities $j \in \mathcal{C}$, that is, the facilities in the (unique) minimal cut. This result makes it possible to reduce the problem for a *FFGCN* to an equivalent problem for a network of parallel facilities.

First we need some more notation. Consider an alternative network with the same topology as the original network, but with a revised set of facility cost functions, $\tilde{H}_j(\cdot)$ such that, for any feasible allocation, $\lambda \in \mathcal{L}$,

$$\tilde{H}_j(\lambda) = H_j(\lambda), \quad j \in \mathcal{C}$$
$$\tilde{H}_j(\lambda) = 0, \quad j \notin \mathcal{C}$$

Define

$$\tilde{C}(\lambda) := \sum_{J\in J} \tilde{H}_j(\lambda) = \sum_{J\in\mathcal{C}} H_j(\lambda).$$

That is, $\tilde{C}(\lambda)$ is the total waiting cost per unit time incurred at the facilities in the minimal cut.

Lemma 9 For any feasible allocation, $\lambda \in \mathcal{L}$,

$$\lim_{\Lambda \to \mu} \frac{\tilde{C}(\lambda)}{C(\lambda)} = 1.$$

Proof. By Assumption 2, there exists a constant, $M < \infty$, such that

$$\sum_{j \notin \mathcal{C}} H_j(\lambda) \leq M.$$

for all $\lambda \in \mathcal{L}$. It follows that

$$\begin{aligned} 1 \geq \frac{\tilde{C}(\lambda)}{C(\lambda)} &= \frac{\sum_{j \in \mathcal{C}} H_j(\lambda)}{\sum_{j \in \mathcal{C}} H_j(\lambda) + \sum_{j \notin \mathcal{C}} H_j(\lambda)} \\ &\geq \frac{\sum_{j \in \mathcal{C}} H_j(\lambda)}{\sum_{j \in \mathcal{C}} H_j(\lambda) + M} \to 1, \text{ as } \Lambda \to \mu. \end{aligned}$$

Now let $\tilde{\lambda}^e$ and $\tilde{\lambda}^s$ denote, respectively, individually optimal and socially optimal allocations for the network with the revised facility-cost functions, $\tilde{H}_j(\,\cdot\,)$, $j \in J$.

Theorem 10 *Suppose*

$$\lim_{\Lambda \to \mu} \frac{\tilde{C}(\tilde{\lambda}^e)}{\tilde{C}(\tilde{\lambda}^s)} = \kappa < \infty. \tag{5.49}$$

Then

$$\lim_{\Lambda \to \mu} \frac{C(\lambda^e)}{C(\lambda^s)} = \kappa. \tag{5.50}$$

Proof. It suffices to show that

$$\left| \frac{C(\lambda^e)}{C(\lambda^s)} - \frac{\tilde{C}(\tilde{\lambda}^e)}{\tilde{C}(\tilde{\lambda}^s)} \right| \to 0, \quad \text{as } \Lambda \to \mu. \tag{5.51}$$

Note that

$$\left| \frac{C(\lambda^e)}{C(\lambda^s)} - \frac{\tilde{C}(\tilde{\lambda}^e)}{\tilde{C}(\tilde{\lambda}^s)} \right| \leq \left| \frac{C(\lambda^e)}{C(\lambda^s)} - \frac{\tilde{C}(\lambda^e)}{\tilde{C}(\lambda^s)} \right| + \left| \frac{\tilde{C}(\lambda^e)}{\tilde{C}(\lambda^s)} - \frac{\tilde{C}(\tilde{\lambda}^e)}{\tilde{C}(\tilde{\lambda}^s)} \right| \tag{5.52}$$

We first show that the second term on the right-hand side of the inequality approaches zero as $\Lambda \to \mu$. Consider socially optimal flows for the original network and for the parallel network constructed from the facilities in the minimal cut. We claim that

$$\lim_{\Lambda \to \mu} \frac{\tilde{C}(\tilde{\lambda}^s)}{\tilde{C}(\lambda^s)} = 1. \tag{5.53}$$

To show this, let $\epsilon > 0$ be arbitrary. It suffices to show that

$$\frac{\tilde{C}(\tilde{\lambda}^s)}{\tilde{C}(\lambda^s)} > 1 - \epsilon, \quad \text{for all } \Lambda < \mu \text{sufficiently close to } \mu.$$

Suppose, to the contrary, there exists an $\epsilon > 0$ such that

$$\frac{\tilde{C}(\tilde{\lambda}^s)}{\tilde{C}(\lambda^s)} \leq 1 - \epsilon, \quad \text{for all } \Lambda < \mu.$$

Now $C(\tilde{\lambda}^s) = \tilde{C}(\tilde{\lambda}^s) + \sum_{j \notin \mathcal{C}} H_j(\tilde{\lambda}^s)$. Since $\sum_{j \notin \mathcal{C}} H_j(\tilde{\lambda}^s) \leq M < \infty$ and $C(\lambda^s) \to \infty$ as $\Lambda \to \mu$, we can choose a $\Lambda < \mu$ sufficiently close to μ such that

$$\frac{\sum_{j \notin \mathcal{C}} H_j(\tilde{\Lambda}^s)}{C(\Lambda^s)} < \epsilon.$$

It then follows that

$$\frac{C(\tilde{\lambda}^s)}{C(\lambda^s)} < 1 - \epsilon + \epsilon = 1,$$

which implies that $C(\tilde{\lambda}^s) < C(\lambda^s)$, contradicting the assumed social optimality of λ^s. Thus, (5.53) holds.

By an argument similar to that used to prove Theorem 7 in Sect. 5.4 one can show that

$$\lim_{\Lambda \to \mu} \frac{\tilde{C}(\tilde{\lambda}^e)}{\tilde{C}(\lambda^e)} = 1. \tag{5.54}$$

From (5.53) and (5.54) and the assumption that $\tilde{C}(\tilde{\lambda}^e)/\tilde{C}(\tilde{\lambda}^s) \to \kappa < \infty$, it follows that

$$\begin{aligned}\left| \frac{\tilde{C}(\lambda^e)}{\tilde{C}(\lambda^s)} - \frac{\tilde{C}(\tilde{\lambda}^e)}{\tilde{C}(\tilde{\lambda}^s)} \right| &= \left| \frac{\tilde{C}(\lambda^e)}{\tilde{C}(\tilde{\lambda}^e)} \cdot \frac{\tilde{C}(\tilde{\lambda}^s)}{\tilde{C}(\lambda^s)} \cdot \frac{\tilde{C}(\tilde{\lambda}^e)}{\tilde{C}(\tilde{\lambda}^s)} - \frac{\tilde{C}(\tilde{\lambda}^e)}{\tilde{C}(\tilde{\lambda}^s)} \right| \\ &= \frac{\tilde{C}(\tilde{\lambda}^e)}{\tilde{C}(\tilde{\lambda}^s)} \cdot \left| \frac{\tilde{C}(\lambda^e)}{\tilde{C}(\tilde{\lambda}^e)} \cdot \frac{\tilde{C}(\tilde{\lambda}^s)}{\tilde{C}(\lambda^s)} - 1 \right| \to 0, \text{ as } \Lambda \to \mu,\end{aligned}$$

and hence

$$\frac{\tilde{C}(\lambda^e)}{\tilde{C}(\lambda^s)} \to \kappa < \infty, \quad \text{as } \Lambda \to \mu. \tag{5.55}$$

Now we show that the first term on the right-hand side of the inequality (52) also approaches zero as $\Lambda \to \mu$. We have

$$\left| \frac{C(\lambda^e)}{C(\lambda^s)} - \frac{\tilde{C}(\lambda^e)}{\tilde{C}(\lambda^s)} \right| = \left| \frac{C(\lambda^e)}{\tilde{C}(\lambda^e)} \cdot \frac{\tilde{C}(\lambda^s)}{C(\lambda^s)} \cdot \frac{\tilde{C}(\lambda^e)}{\tilde{C}(\lambda^s)} - \frac{\tilde{C}(\lambda^e)}{\tilde{C}(\lambda^s)} \right|$$

$$= \frac{\tilde{C}(\lambda^e)}{\tilde{C}(\lambda^s)} \cdot \left| \frac{C(\lambda^e)}{\tilde{C}(\lambda^e)} \cdot \frac{\tilde{C}(\lambda^s)}{C(\lambda^s)} - 1 \right|,$$

which approaches zero as $\Lambda \to \mu$ by Lemma 9 and the fact that $\tilde{C}(\lambda^e)/\tilde{C}(\lambda^s)$ approaches a finite limit as $\Lambda \to \mu$ (cf. Eq. (5.55)). This completes the proof of the theorem.

Application to a Network of Queues

We now consider the special case of a generalized Jackson network satisfying the assumptions of this section. The detailed conditions of our model for the arrival process and the service times at the facilities are essentially the same as for the model for parallel *GI/GI/1* queues in Section 5.4 For convenience we restate the conditions here.

Customers arrive to the source node s according to a renewal process. The generic interarrival time is denoted by $A = X/\Lambda$, where X is a fixed nonnegative random variable with mean 1 and squared coefficient of variation (scv) C_a^2. Upon arrival each customer is assigned to route $r \in R$ with probability p_r, where $p_r \geq 0$, for $r \in R$, independently of all other customers, and $\sum_{r \in R} p_r = 1$. The service times of the customers who use facility $j \in J$ form an i.i.d. sequence of random variables distributed as S_j with finite mean $1/\mu_j$ and scv $C_{S_j}^2$, $j \in J$.

Let $\{A_t, t = 1,2,\dots\}$ be a sequence of i.i.d. random variables with the same distribution as A and let Z_r be an independent geometric random variable with mean $1/p_r$, $r \in R$. Define

$$A^{(r)} := \sum_{t=1}^{Z_r} A_t, \quad r \in R.$$

Then the interarrival times of customers assigned to route $r \in R$ are i.i.d. random variables distributed as the generic random variable $A^{(r)}$.

The waiting cost incurred by a customer who spends a length of time t at facility j is $h_j(t)$, $t \geq 0$, where $h_j(\cdot)$ is non-decreasing with $h_j(0) = 0$, $j \in J$. Thus

$$G_j(\lambda) = \mathrm{E}[h_j(\mathcal{W}_j(\lambda))], \quad j \in J,$$

where $\mathcal{W}_j(\lambda)$ is the steady-state random waiting time (in queue plus in service) of a customer at facility j, $j \in J$. The total waiting cost per unit time is therefore given by

$$C(\lambda) = \sum_{j \in J} \nu_j G_j(\lambda) = \sum_{j \in J} \nu_j \mathrm{E}[h_j(\mathcal{W}_j(\lambda))],$$

where $\lambda = (\lambda_r, r \in R)$.

The decision variables are the routing probabilities, p_r, $r \in R$, or, equivalently, the arrival rates, λ_r, where $\lambda_r = \Lambda p_r, r \in R$, and $\sum_{r \in R} \lambda_r = \Lambda$. As usual, we denote individually optimal and socially optimal flow allocations by λ^e and λ^s, respectively.

We shall refer to such a network satisfying Assumptions 1 and 2 as a *Ford-Fulkerson Generalized Jackson Network* (*FFGJN*).

Now we construct an alternative network, in which the generic service times at the facilities $j \in J$ are defined as follows:

$$\tilde{S}_j = S_j, \quad j \in \mathcal{C}$$
$$\tilde{S}_j = 0, \quad j \notin \mathcal{C}$$

In this alternative network, the waiting times at the facilities $j \notin \mathcal{C}$ are identically zero, so that these facilities become in effect transparent to the customers as they move through the network. It follows that the facilities $j \in \mathcal{C}$ behave exactly like a network of parallel independent *GI/GI/1* queues as studied in Sect. 5.4.2. Specifically, the arrivals to each facility $j \in \mathcal{C}$ – namely, those customers who are assigned to the routes r that include facility j – have i.i.d. inter-arrival times with mean $1/\nu_j$ and scv given by

$$(\sum_{r: j \in r} p_r)(C_a^2 - 1) + 1, \quad j \in \mathcal{C},$$

where

$$\nu_j = \Lambda \sum_{r: j \in r} p_r, \quad j \in \mathcal{C}.$$

Since the facilities $j \in \mathcal{C}$ in the alternative network act independently of each other, just as in the network of parallel queues considered in Sect. 5.4.2, the waiting times and costs at facility j, $j \in \mathcal{C}$, are completely determined by ν_j. Therefore we can write $\tilde{G}_j(\nu_j)$ rather than $\tilde{G}_j(\lambda)$ for the waiting cost function at facility j, $j \in \mathcal{C}$, and $\tilde{C}(\nu)$ rather than $\tilde{C}(\lambda)$ for the total waiting cost in the alternative network. Let $\tilde{\nu}^s$ and $\tilde{\nu}^e$ denote the socially optimal and individually optimal facility flow allocations, respectively, in the alternative network.

We can now state the following corollary of Theorem 10.

Corollary 11 *Consider a* FFGJN. *Suppose*

$$\lim_{\Lambda \to \mu} \frac{\tilde{C}(\tilde{\nu}^e)}{\tilde{C}(\tilde{\nu}^s)} = \kappa < \infty. \tag{5.56}$$

Then

$$\lim_{\Lambda \to \mu} \frac{C(\lambda^e)}{C(\lambda^s)} = \kappa. \tag{5.57}$$

For the special case of linear waiting cost functions, we have the following corollary of Theorems 8 and 10.

Corollary 12 *Consider a* FFGJN *with linear waiting cost functions,*

$$G_j(\lambda) = \mathrm{E}[h_j \cdot \mathcal{W}(\lambda)], \quad j \in J.$$

The POA for this system is given by

$$\lim_{\Lambda \to \mu} C(\lambda^e)/C(\lambda^s) = \lim_{\Lambda \to \mu} \tilde{C}(\tilde{\nu}^e)/\tilde{C}(\tilde{\nu}^s) = \frac{\left(\sum_{j \in \mathcal{C}} \tilde{h}_j\right)\left(\sum_{j \in \mathcal{C}} \mu_j\right)}{\left(\sum_{j \in \mathcal{C}} \sqrt{\tilde{h}_j \mu_j}\right)^2},$$

where $\tilde{h}_j$ *is given by* (5.25), $j \in \mathcal{C}$.

The implication of Corollary 12 is that the POA for a *FFGJN* with linear waiting cost functions coincides with the POA for a system consisting only of the facilities in the minimal cut $\mathcal{C}$, operating in parallel, where each facility $j \in \mathcal{C}$ operates as an *M/M/1* queue with h_j replaced by $\tilde{h}_j = h_j f_j$.

Conclusions

The POA for a general congestion network is the ratio of the total cost of an individually optimal (competitive equilibrium) allocation of flows to the total cost of a socially optimal allocation. In this paper we have considered the POA for a congestion network in which the waiting costs at the facilities have the property that the average waiting cost of a customer approaches infinity as the flow at that facility approaches the capacity of the facility. The expected steady-state waiting time in a single-server queue typically has this property, with the mean service rate playing the role of the capacity. For special cases of such a network, we have shown that the heavy-traffic limit of the POA is finite. We were able to calculate this limit in closed form for the special case of a network of parallel *GI/GI/1* queues. For certain cases of generalized Jackson networks, we have shown that the heavy-traffic limit of the POA coincides with the limit for a network consisting only of the facilities in a minimal cut, operating in parallel. Our results contrast with those in the previous literature on the POA for general congestion networks, in which upper bounds on the POA are derived which typically grow without bound in heavy traffic.

There are a number of possibilities for future research in this area. An extension of our results in Sect. 5 from Ford-Fulkerson networks to more general single-class networks would allow for applications in which not all routes are permitted. In such networks, the allocation of flows in heavy traffic is more complicated and it no longer suffices to consider only the flows in a minimal cut. Multiclass networks (e.g., networks with different types of customers with different waiting costs and/or many origin-destination pairs, each with its own demand), also will require a more sophisticated analysis.

References

Beckmann, M., McGuire, C., & Winsten, C. (1956). *Studies in the economics of transportation*. New Haven: Yale University Press.

Bertsekas, D. (1998).*Network optimization: Continuous and discrete models*. Nashua:Athena Scientific.

Chau, C., & Sim, K. (2003). The price of anarchy for nonatomic congestion games with symmetric cost maps and elastic demands. *Operational Research Letters*, *31*, 327–334.

Correa, J., Schulz, A., & Stier-Moses, N. (2004a). Computational complexity, fairness, and the price of anarchy of the maximum latency problem. *Integer Programming and Combinatorial Optimization* (Vol. 3064, p. 59–73). Springer Berlin/Heidelberg, Lecture Notes in Computer Science.

Correa, J., Schulz, A. & Stier-Moses, N. (2004b). Selfish routing in capacitated networks. *Mathematics of Operations Research*, *29,* 961–976.

Correa, J., Schulz A., & Stier-Moses, N. (2005). On the inefficiency of equilibria in congestion games. M. Junger & V. Kaibel (eds.), *IPCO 2005, LNCS 3509*, (p. 167–181). Berlin: Springer-Verlag.

Crabill, T., Gross D., & Magazine, M. (1977). A classified bibliography of research on optimal design and control of queues. Operations Research, *25*, 219–232.

Dafermos S. (1980). Traffic equilibrium and variational inequalities. *Trans- portation Science*, *14*, 42–54.

Dafermos, S., & Nagourney, A. (1984). On some traffic equilibrium theory paradoxes. Transportation Research, *18B*, 101–110.

Dafermos, S., & Sparrow, F. (1969). The traffic assignment problem for a general network. *Journal of Research of the National Bureau of Standards*, *73B*, 91–118.

El-Taha, M. & Stidham, S. (1998). *Sample-path analysis of queueing systems*. Boston: Kluwer Academic Publishing.

Hassin, R. & Haviv, M. (2003). *To queue or not to queue equilibrium behavior in queueing systems*. Boston: Kluwer Academic Publishers.

Kelly, F. (1979). *Reversibility and stochastic networks*. Chichester: John Wiley.

Kitaev, M., & Rykov, V. (1995). *Controlled queueing systems*. Boca Raton: CRC Press.

Naor, P. (1969). On the regulation of queue size by levying tolls. Econometrica, *37*, 15–24.

Perakis, G. (2004). The price of anarchy under nonlinear and asymmetric costs. *Integer Programming and Combinatorial Optimization* (Vol. 3064, p. 46–58). Springer Berlin/Heidelberg, Lecture Notes in Computer Science.

Roughgarden T. (2002). The price of anarchy is independent of the network topology. Proceedings of ACM Symposium on Theory of Computing (Vol. 34, p. 428–437).

Roughgarden, T. (2005). *Selfish Routing and the Price of Anarchy*. Cambridge: MIT Press.

Roughgarden, T. (2006). On the severity of Braesss paradox: Designing networks for selfish users is hard. *Journal of Computer and System Science*, *72*, 922–953.

Roughgarden, T., & Tardos E. (2002). How bad is selfish routing? *Journal of the Association of Computer Machinery*, *49*, 236–259.

Schulz, A., & Stier-Moses, N. (2003). On the performance of user equilibria in traffic networks. Proceeding of ACM-SIAM Symposium on Discrete Algorithms (Vol. 14, p. 86–87). Baltimore.

Serfozo, R. (1981). Optimal control of random walks, birth and death processes, and queues. *Advances in Applied Probability*, *13*, 61–83.

Shanthikumar, G., & Xu, S. (1997). Asymptotically optimal routing and service rate allocation in a multi server queueing system. *Operations Research*, *45*, 464–469.

Shanthikumar, G, & Xu, S. (2000). Strongly asymptotically optimal design and control of production and service systems. *IIE Transactions*, *32*, 881–890.

Sobel, M. (1974). Optimal operation of queues. A. B. Clarke (ed.), *Mathematical methods in queueing theory* (Vol. 98, p. 145–162, Berlin: Springer-Verlag, Lecture Notes in Economics and Mathematical Systems.

Stidham, S. (1971). *Stochastic design models for location and allocation of public service facilities: Part I*. Technical Report, Department of Environmental Systems Engineering, College of Engineering, Cornell University.

Stidham, S. (1978). Socially and individually optimal control of arrivals to a GI/M/1 queue. *Management Science*, *24*, 1598–1610.

Stidham, S. (1984). Optimal control of admission, routing, and service in queues and networks of queues: A tutorial review. Proceedings ARO Workshop: Analytic and Computational Issues in Logistics R and D (p. 330–377). George Washington University.

Stidham, S. (1985). Optimal control of admission to a queueing system. *The IEEE Transactions on Automatic Control*, *30*, 705–713.

Stidham, S. (1988). Scheduling, routing, and ow control in stochastic networks. In W. Fleming, P. L. Lions (eds.), *Stochastic Differential Systems, Stochastic Control Theory and Applications* (Vol. IMA-10, p. 529–561). New York: Springer-Verlag.

Stidham, S. (2008). The price of anarchy for a single-class network of queues. Technical Report, Department of Statistics and Operations Research. University of North Carolina at Chapel Hill.

Stidham, S. (2009) *Optimal design of queueing systems*. Boca Raton, FL: CRC Press, Taylor and Francis Group (A Chapman & Hall Book).

Stidham, S., & Prabhu, N. (1974). Optimal control of queueing systems. In A. B. Clarke (ed.), *Mathematical methods in queueing theory* (Vol. 98, p. 263–294). Berlin:Springer-Verlag. Lecture Notes in Economics and Mathematical Systems.

Wardrop, J. (1952). Some theoretical aspects of road traffic research. *Proceeding of the Institution of Civil Engineers, Part II*, *1*, 325–378.

Chapter 6
A Comparative Study of Procedures for the Multinomial Selection Problem

Eric Tollefson, David Goldsman, Anton J. Kleywegt and Craig A. Tovey

Introduction

How many games are needed in a playoff series to identify the best team with specified confidence? How many potential voters should be surveyed to identify the most popular candidate in a particular political campaign? How many households does one need to include in a survey to identify the most watched television show during a certain time slot? How many wine connoisseurs have to participate in a tasting competition to identify the wine most likely to be preferred (by a connoisseur)? How many times does one have to send packages to a destination with different couriers to identify the courier that is fastest on average (Bartholdi 2010)? These are all questions that can be formulated as multinomial selection problems (MSPs), where one attempts to identify the alternative (or outcome or category) of a multinomial distribution that has the largest probability of occurrence, and in which one may be subject to a budget constraint that limits the number of trials to be conducted.

We want to design an experiment to choose the best among k alternatives. An experiment consists of a chosen number of trials, in each of which all the alternatives compete. In each trial, alternative i has probability $p_i > 0$ of winning, where $\sum_{i=1}^{k} p_i = 1$. Denote the ordered p_i's by $p_{[1]} \leq p_{[2]} \leq \cdots < p_{[k]}$. The alternative associated with $p_{[k]}$ is the *most probable* or *best*, and is denoted i^* (it is assumed throughout the paper that the best alternative is unique). The purpose of the experiment is to identify correctly with high probability the best alternative i^*. Prior to

E. Tollefson (✉) · D. Goldsman · A. J. Kleywegt · C. A. Tovey
H. Milton Stewart School of Industrial and Systems Engineering,
Georgia Institute of Technology, Atlanta, GA, 30332-0205, USA
e-mail: eric.tollefson@us.army.mil

D. Goldsman
e-mail: sman@gatech.edu

A. J. Kleywegt
e-mail: anton@isye.gatech.edu

C. A. Tovey
e-mail: ctovey@isye.gatech.edu

P. S. Pulat et al. (eds.), *Essays in Production, Project Planning and Scheduling,*
International Series in Operations Research & Management Science 200,
DOI 10.1007/978-1-4614-9056-2_6,

an experiment, all that is known is the number of alternatives, how each trial will be conducted, and how the winning alternative will be chosen. No information is known concerning the probabilities $\boldsymbol{p} = (p_1, p_2, \ldots, p_k)$ or the likelihood that any alternative is better than any other.

Let $x_{ij} = 1[0]$ if alternative i is [is not] the winner of trial j, and let $\eta_{im} \equiv \sum_{j=1}^{m} x_{ij}$ denote the total number among the first m trials won by alternative i. We denote the cumulative win vector by $\boldsymbol{\eta}_m \equiv (\eta_{1m}, \eta_{2m}, \ldots, \eta_{km})$, and we denote the ordered η_{im}'s by $\eta_{[1]m} \leq \cdots \leq \eta_{[k]m}$. Let N be a random variable that denotes the total number of trials conducted during an experiment. A procedure $\mathcal{M}$ is a method to choose the number of trials conducted and to select one alternative at the conclusion of the trials. When needed, we include a subscript on $\mathcal{M}$ and N to refer to a particular MSP procedure, e.g., N_{J} is the number of trials conducted by procedure $\mathcal{M}_{\mathrm{J}}$. We consider only procedures for which, after N trials, an alternative with the most wins is selected; that is, an alternative i with $\eta_{iN} = \eta_{[k]N}$ is chosen. For cases when multiple alternatives are tied for the most wins, we choose each with equal probability. The chosen alternative is denoted $\hat{i}_N$.

For a given procedure, the probability of correct selection, denoted by $\mathsf{P}_{\boldsymbol{p}}(\mathrm{CS})$, or simply $\mathsf{P}(\mathrm{CS})$, is the probability that alternative i^* is chosen. Clearly, $\mathsf{P}(\mathrm{CS})$ depends on $\boldsymbol{p}$. For any MSP procedure, a reasonable objective is to minimize the expected number of trials while requiring that

$$\mathsf{P}_{\boldsymbol{p}}(\mathrm{CS}) \;\geq\; P^* \text{ for all } \boldsymbol{p} \text{ such that } p_{[k]}/p_{[k-1]} \;\geq\; \theta^*, \tag{6.1}$$

where the desired probability of correct selection $P^* > 1/k$ and the so-called relative-ratio indifference-zone parameter $\theta^* > 1$ are constants that are both specified by the user. The quantity θ^* can be regarded as the "smallest ratio $p_{[k]}/p_{[k-1]}$ worth detecting."

To guarantee (6.1), we require additional information. Let $\mathcal{P} \equiv \{\boldsymbol{p} \in [0,1]^k : \sum_{i=1}^{k} p_i = 1\}$ be the set of all possible probability configurations $\boldsymbol{p}$. The *preference zone* (PZ) is denoted $\mathcal{P}_{\mathrm{PZ}} \equiv \{\boldsymbol{p} \in \mathcal{P} : p_{[k]}/p_{[k-1]} \geq \theta^*\}$. Its complement, $\mathcal{P}_{\mathrm{IZ}}$, is the *indifference zone* (IZ). Given a procedure, the *least favorable configuration* (LFC) is the probability configuration $\boldsymbol{p} \in \mathcal{P}_{\mathrm{PZ}}$ that minimizes $\mathsf{P}_{\boldsymbol{p}}(\mathrm{CS})$. While guaranteeing (6.1), the goal is to minimize the expected number of trials when $\boldsymbol{p}$ is the LFC. However, for some MSP procedures, the LFC has not yet been identified, so we instead attempt to minimize the expected number of trials when $\boldsymbol{p}$ is the *slippage configuration* (SC):

$$\mathrm{SC} \;\equiv\; \left(\frac{1}{\theta^* + k - 1}, \cdots, \frac{1}{\theta^* + k - 1}, \frac{\theta^*}{\theta^* + k - 1}\right),$$

which is the LFC for many procedures. In that case, our objective is to minimize the expected number of trials when $\boldsymbol{p}$ is the SC, while requiring that

$$\mathsf{P}_{\boldsymbol{p}}(\mathrm{CS}) \;\geq\; P^* \text{ when } \boldsymbol{p} \text{ is the SC}, \tag{6.2}$$

a weaker condition than condition (6.1).

There are different versions of the MSP described above. In the *single-stage* or *static* MSP, the experimenter has to choose in advance the number N of trials. Researchers typically assume that the experimenter wants the smallest number N such that if $\hat{i}_N$ is chosen as described above, then conditions (6.1) or (6.2) hold. Since the number of trials is committed in advance, the experimenter does not consider or allow the possibility of conducting one trial at a time for the purpose of deciding whether to stop or conduct another trial based on the outcomes of the previous trials.

In the *sequential* or *dynamic* MSP, the experimenter may conduct one trial at a time and dynamically determine the (random) number N of trials. In this setting, researchers typically assume that the experimenter wants to choose a procedure $\mathcal{M}$ that minimizes the expectation of N_{M} under configuration $\boldsymbol{p}$, denoted $\mathsf{E}_{\boldsymbol{p}}[N_{\mathrm{M}}]$; special attention is placed on cases for which $\boldsymbol{p}$ is the LFC (which depends on the procedure $\mathcal{M}$) or at least the SC, and such that conditions (6.1) or (6.2) hold. Sequential procedures can be one of three types:

- *Unbounded* sequential procedures for which there is no a priori bound on the number of trials taken during an experiment.
- *Bounded* sequential procedures for which the chosen procedure parameters provide an upper bound on the number of trials taken during an experiment.
- *Constrained* sequential procedures (a special case of bounded procedures) for which the decision maker specifies a maximum number of trials that can be taken, called the budget b.

An example of a setting modeled as a static MSP is an agricultural experiment, in which each trial consists of dividing a plot of land into k parts, one part for each of the k crops that are planted. The experimenter has to decide how many such plots of land to prepare before the growing season, and does not want to conduct one trial in each growing season before deciding whether to conduct another trial. Many sports competitions are dynamic "best-out-of-m" type tournaments. For example, the Major League Baseball World Series is a best-out-of-7 tournament (experiment), in which the first team to win 4 games (trials) is the winner of the tournament. Many tennis matches are best-out-of-3 matches (experiments), in which the first side to win 2 sets (trials) wins the match. As pointed out later, a best-out-of-m procedure is a specific type of procedure for dynamic MSPs.

In the next section, we review the static and dynamic procedures that will be compared in the paper. After this, we describe the methodology and metrics we will use to evaluate the performance of the procedures. We then compare the procedures, and finally we give conclusions. New, more-accurate and extensive parameter tables are given in the appendices.

Review of Procedures

The classic single-stage procedure is due to Bechhofer, Elmaghraby, and Morse (BEM) (1959), and proceeds as follows.

Procedure $\mathcal{M}_{\text{BEM}}$

- For the given k, θ^*, and P^*, choose the number n_{BEM} of trials. Tables that give the minimum value of n_{BEM} subject to (6.2) have been prepared, e.g., BEM (1959) or Bechhofer, Santner, and Goldsman (BSG) (1995).
- Conduct n_{BEM} multinomial trials in a single stage.
- Select $\hat{i}_{n_{\text{BEM}}}$ as the best alternative, using randomization to break ties.

Remark 1 For Procedure $\mathcal{M}_{\text{BEM}}$, Kesten and Morse (1959) prove that the SC is the LFC. Thus, the parameter n_{BEM} in BEM (1959), chosen based upon the procedure performance in the SC, satisfies condition (6.1) for the given k, θ^*, and P^*.

In principle, any static procedure can be used to choose the number of trials for a dynamic MSP. However, it is clear that such a procedure may sometimes conduct more trials than needed, because it does not exploit the information provided as trial outcomes are observed. For example, if $k = 2$ and $n_{\text{BEM}} = 100$ trials are chosen, and we obtain $\eta_{1,100} = 99$ and $\eta_{2,100} = 1$, then for many values of θ^* and P^*, we could have stopped before trial 100 and still have reached the conclusion that alternative 1 is the most probable for the given P^*-requirement on $\mathsf{P}(\text{CS})$. The bounded sequential procedure of Bechhofer and Kulkarni (BK) (1984) capitalizes on such favorable sample paths, that is, sample paths that allow the procedure to stop before conducting all n_{BEM} trials required by the single-stage Procedure $\mathcal{M}_{\text{BEM}}$.

Procedure $\mathcal{M}_{\text{BK}}$

- For the given k, θ^*, and P^*, choose the parameter n_{BK} (usually $n_{\text{BK}} = n_{\text{BEM}}$). Sources are BSG (1995) or Appendix A of this paper.
- At the mth stage of experimentation, $m \geq 1$, conduct a multinomial trial.
- Calculate the ordered cumulative wins $\eta_{[i]m}$, $i = 1, 2, \ldots, k$. Stop the experiment at the first stage when

$$\eta_{[k]m} - \eta_{[k-1]m} \geq n_{\text{BK}} - m. \tag{6.3}$$

- Select $\hat{i}_m$ as the best alternative, using randomization to break ties.

In other words, Procedure $\mathcal{M}_{\text{BK}}$ employs a curtailment strategy that stops sampling at the first stage m for which the alternative currently in first place can do no worse than tie if the remaining $n_{\text{BK}} - m$ trials were to be conducted. Let N_{BK} be a random variable denoting the value of m at the termination of the experiment. It can be shown that the curtailed Procedure $\mathcal{M}_{\text{BK}}$ yields the same $\mathsf{P}(\text{CS})$ as the single-stage Procedure $\mathcal{M}_{\text{BEM}}$, yet with a smaller expected number of trials, i.e., for all $\boldsymbol{p}$,

$$\mathsf{P}_{\boldsymbol{p}}(\text{CS using Procedure } \mathcal{M}_{\text{BK}}) = \mathsf{P}_{\boldsymbol{p}}(\text{CS using Procedure } \mathcal{M}_{\text{BEM}})$$

and

$$\mathsf{E}_{\boldsymbol{p}}[N_{\text{BK}}] \leq n_{\text{BEM}}.$$

Remark 2 Since $\mathsf{P}_{\boldsymbol{p}}(\mathrm{CS})$ for both procedures is identical when $n_{\mathrm{BK}} = n_{\mathrm{BEM}}$, the SC for Procedure $\mathcal{M}_{\mathrm{BK}}$ must be the LFC as Kesten and Morse (1959) proved for Procedure $\mathcal{M}_{\mathrm{BEM}}$. Thus, the parameter n_{BK}, chosen based upon the procedure performance in the SC, satisfies condition (6.1) for the given k, θ^*, and P^*.

Another sequential procedure is due to Ramey and Alam (RA) (1979), who combine the stopping rule of Alam's (1971) unbounded procedure, which stops when one alternative has a sufficient lead over the remaining alternatives, with the inverse sampling stopping rule of Cacoullos and Sobel (1966), which stops when the alternative with the largest number of wins hits a certain stopping bound.

Procedure $\mathcal{M}_{\mathrm{RA}}$

- For the given k, θ^*, and P^*, choose the parameter pair (r, t). Sources are Bechhofer and Goldsman (1985a) or Appendix C of this paper.
- At the mth stage of experimentation, $m \geq 1$, conduct a multinomial trial.
- Calculate the ordered cumulative wins $\eta_{[i]m}$, $i = 1, 2, \ldots, k$. Stop the experiment at the first stage when

$$\eta_{[k]m} = t \quad \text{or} \quad \eta_{[k]m} - \eta_{[k-1]m} = r.$$

- Select $\hat{i}_m$ as the best alternative; ties are not possible.

Remark 3 In their paper, RA prove that the SC is the LFC for their procedure when $k = 2$, and use empirical evidence to conjecture that it is so for $k > 2$. The (r, t)-values in BG (1985a) have been chosen to minimize the expected number of trials conducted by Procedure $\mathcal{M}_{\mathrm{RA}}$ when $\boldsymbol{p}$ is the SC, satisfying condition (6.2), but not necessarily condition (6.1).

Consider the special case of Procedure $\mathcal{M}_{\mathrm{RA}}$ with $r = t$. In that case, the first alternative to win t trials is chosen as the best alternative. In the case with $k = 2$, that corresponds to a best out of $2t - 1$ tournament.

In a slight modification to Procedure $\mathcal{M}_{\mathrm{RA}}$, Chen (1992) creates Procedure $\mathcal{M}_{\mathrm{RA}'}$ by adding truncation with curtailment at trial $n_{\mathrm{RA}'}$.

Procedure $\mathcal{M}_{\mathrm{RA}'}$

- For the given k, θ^*, and P^*, choose the parameter triplet $(n_{\mathrm{RA}'}, r, t)$. Sources are Chen (1992) or Appendix D of this paper.
- At the mth stage of experimentation, $m \geq 1$, conduct a multinomial trial.
- Calculate the ordered cumulative wins $\eta_{[i]m}$, $i = 1, 2, \ldots, k$. Stop the experiment at the first stage when

$$\eta_{[k]m} = t \quad \text{or} \quad \eta_{[k]m} - \eta_{[k-1]m} = r \quad \text{or} \quad \eta_{[k]m} - \eta_{[k-1]m} \geq n_{\mathrm{RA}'} - m.$$

- Select $\hat{i}_m$ as the best alternative, using randomization to break ties.

Remark 4 Chen conjectures that the LFC of Procedure $\mathcal{M}_{\mathrm{RA}'}$ is the SC. The $(n_{\mathrm{RA}'}, r, t)$-values in Chen (1992) have been chosen to minimize the expected number of trials conducted by Procedure $\mathcal{M}_{\mathrm{RA}'}$ when $\boldsymbol{p}$ is the SC, satisfying condition (6.2), but not necessarily condition (6.1).

Bechhofer and Goldsman (BG) (1985b, 1986) introduce Procedure $\mathcal{M}_{\mathrm{BG}}$, which truncates an unbounded sequential procedure due to Bechhofer, Kiefer, and Sobel (BKS) (1968) in order to save trials by reducing the inherent overprotection of P(CS) in the BKS procedure.

Procedure $\mathcal{M}_{\mathrm{BG}}$

- For the given k, θ^*, and P^*, choose the truncation parameter n_{BG}. Sources are BG (1986) or BSG (1995).
- At the mth stage of experimentation, $m \geq 1$, conduct a multinomial trial.
- Calculate the ordered cumulative wins $\eta_{[i]m}$, $i = 1, 2, \ldots, k$, and the quantity

$$z_m = \sum_{i=1}^{k-1} \left(\frac{1}{\theta^*}\right)^{\eta_{[k]m} - \eta_{[i]m}}.$$

Stop the experiment at the first stage when either

$$z_m \leq (1 - P^*)/P^* \quad \text{or} \quad \eta_{[k]m} - \eta_{[k-1]m} \geq n_{\mathrm{BG}} - m. \tag{6.4}$$

- Select $\hat{i}_m$ as the best alternative, using randomization to break ties.

Remark 5 For the unbounded procedure upon which Procedure $\mathcal{M}_{\mathrm{BG}}$ is based, BKS prove that the LFC is the SC; see also Levin (1984). BG (1986) acknowledge that both the BKS procedure and Procedure $\mathcal{M}_{\mathrm{BK}}$ share the same LFC, but they do not prove that combining the stopping rules of these two procedures by adding a truncation point to the BKS procedure actually preserves the LFC in the new procedure. The tabulated n_{BG}-values in BG (1986) and BSG (1995) minimize the expected number of trials taken by Procedure $\mathcal{M}_{\mathrm{BG}}$ when $\boldsymbol{p}$ is the SC, satisfying condition (6.2), but not necessarily condition (6.1).

Chen (1988) proposes a bounded sequential procedure that combines inverse sampling with a finite truncation point.

Procedure $\mathcal{M}_{\mathrm{C}}$

- For the given k, θ^*, and P^*, choose the parameter pair (n_{C}, t). A source is Chen (1988).
- At the mth stage of experimentation, $m \geq 1$, conduct a multinomial trial.
- Calculate the ordered cumulative wins $\eta_{[i]m}$, $i = 1, 2, \ldots, k$. Stop the experiment at the first stage when

$$\eta_{[k]m} = t \quad \text{or} \quad m = n_{\mathrm{C}}. \tag{6.5}$$

- Select $\hat{i}_m$ as the best alternative, using randomization to break ties.

Remark 6 Chen proves that the SC is the LFC. Thus, his tabulated (n_{C}, t)-pairs, based upon procedure performance in the SC, satisfy condition (6.1) for a given k, θ^*, and P^*.

Remark 7 Chen states that the strong curtailment stopping rule (see (6.3)) of Procedure $\mathcal{M}_{\mathrm{BK}}$ could be used to reduce the expected number of trials for his procedure without affecting P(CS), but he does not implement the change. We do so for this comparative study by incorporating curtailment, renaming it Procedure $\mathcal{M}_{\mathrm{C}'}$, and tabulating (in Appendix B) the results for common choices of k, θ^*, and P^*.

The classical sequential procedures reviewed above do not necessarily minimize the expected number of trials in the LFC or SC, but are heuristics for the dynamic MSP. Tollefson *et al.* (2013) develop an approach to find optimal sequential procedures that minimize the expected number of trials for a specified configuration such as the SC, for a constrained MSP with a specified trial budget b.

The sequential procedures reviewed above employ stopping rules that depend on simple relationships between the components of the cumulative win vector $\boldsymbol{\eta}$ and the specified procedure parameters. In general, one may consider all cumulative win vectors $\boldsymbol{\eta}_m$, and specify a decision whether to stop the experiment or continue with another trial for each $\boldsymbol{\eta}_m$. Note that each of the classical sequential procedures can be specified in such a general way.

If there is a trial budget b, then all cumulative win vectors $\boldsymbol{\eta}_m$ for $m \leq b$ are considered. Specifically, let $\mathcal{N} \equiv \left\{ \boldsymbol{\eta} \,:\, \sum_{i=1}^{k} \eta_i \leq b \right\}$ denote the set of all possible cumulative win vectors for a given budget b, and let $\mathcal{N}_b \equiv \left\{ \boldsymbol{\eta} \,:\, \sum_{i=1}^{k} \eta_i = b \right\}$ denote the set of possible cumulative win vectors after b trials. To find optimal sequential procedures, randomized stopping is allowed. Thus, for any $\boldsymbol{\eta} \in \mathcal{N}$, let $\phi_{\boldsymbol{\eta}} \in [0, 1]$ denote the conditional probability that the procedure stops when reaching cumulative win vector $\boldsymbol{\eta}$, given arrival at $\boldsymbol{\eta}$. Due to the budget constraint, $\phi_{\boldsymbol{\eta}} = 1$ for all $\boldsymbol{\eta} \in \mathcal{N}_b$. The classical sequential procedures are *nonrandomized* procedures, and for such procedures $\phi_{\boldsymbol{\eta}} \in \{0, 1\}$ for all $\boldsymbol{\eta} \in \mathcal{N}$. In other words, for nonrandomized procedures, the decision to stop at any point $\boldsymbol{\eta}$ is deterministic—the experiment either stops if it reaches $\boldsymbol{\eta}$ ($\phi_{\boldsymbol{\eta}} = 1$) or it does not ($\phi_{\boldsymbol{\eta}} = 0$). In contrast, a *randomized* procedure allows $\phi_{\boldsymbol{\eta}} \in [0, 1]$ for all $\boldsymbol{\eta} \in \mathcal{N} \setminus \mathcal{N}_b$. For a randomized procedure, the decision to stop at a particular point $\boldsymbol{\eta}$ may be deterministic (if $\phi_{\boldsymbol{\eta}} \in \{0, 1\}$) or may be random (if $\phi_{\boldsymbol{\eta}} \in (0, 1)$). Note that, if one allows $b = \infty$, then any nonrandomized procedure can be specified by a function $\phi : \mathcal{N} \mapsto \{0, 1\}$, and any randomized procedure can be specified by a function $\phi : \mathcal{N} \mapsto [0, 1]$. Recall that the experimenter does not know in advance which alternative is more or less likely to win than another alternative; and, therefore, the indexing of alternatives is arbitrary, and hence $\phi_{\boldsymbol{\eta}}$ is required to be invariant with respect to permutations of $\boldsymbol{\eta}$. In other words, for any $\boldsymbol{\eta}, \boldsymbol{\eta}' \in \mathcal{N}$ such that $\boldsymbol{\eta}$ and $\boldsymbol{\eta}'$ are permutations of each other, $\phi_{\boldsymbol{\eta}} = \phi_{\boldsymbol{\eta}'}$.

Next we specify generic nonrandomized and randomized Procedures $\mathcal{M}_{\mathrm{NR}}$ and $\mathcal{M}_{\mathrm{R}}$, respectively.

Procedures $\mathcal{M}_{\mathrm{NR}}$ and $\mathcal{M}_{\mathrm{R}}$

- For the given k, θ^*, P^*, and b, specify the function $\phi : \mathcal{N} \mapsto [0, 1]$ for the randomized case, or the function $\phi : \mathcal{N} \mapsto \{0, 1\}$ for the nonrandomized case.
- At the mth stage of experimentation, $m \geq 1$, conduct a multinomial trial.
- Calculate $\boldsymbol{\eta}$, the cumulative win vector. In the randomized case, generate a uniform(0, 1) random number υ_m; if $\upsilon_m < \phi_{\boldsymbol{\eta}}$, then stop and select alternative

$\hat{i}_m$, using randomization to break ties. In the nonrandomized case, if $\phi_{\boldsymbol{\eta}} = 1$, then stop and select alternative $\hat{i}_m$, using randomization to break ties. Otherwise, continue with the next trial.

Tollefson *et al.* (2013) present a linear program (LP) to choose the function $\phi : \mathcal{N} \mapsto [0, 1]$, such that the resulting Procedure $\mathcal{M}_{\mathrm{R}}$ is an optimal solution to the constrained MSP, in the sense that it minimizes the expected number of trials subject to a constraint on the probability of correct selection and a budget constraint, for specified parameters k, θ^*, P^*, b, and $\boldsymbol{p}$. Also, they present a mixed integer linear program (MIP) to choose the function $\phi : \mathcal{N} \mapsto \{0, 1\}$, such that the resulting Procedure $\mathcal{M}_{\mathrm{NR}}$ is an optimal nonrandomized solution to the constrained MSP for specified parameters k, θ^*, P^*, b, and $\boldsymbol{p}$. That paper's unique contribution lies in characterizing the problem as a network in which flows represent probabilities, and the nodes $\boldsymbol{\eta}$ in the network represent vectors through which the multinomial sample paths may go.

Remark 8 Tollefson *et al.* (2013) do not prove that the SC is the LFC for Procedures $\mathcal{M}_{\mathrm{NR}}$ and $\mathcal{M}_{\mathrm{R}}$. Empirical evidence drawn from Monte Carlo (MC) sampling suggests that it is so.

Remark 9 For Procedure $\mathcal{M}_{\mathrm{NR}}$, the required MIP formulation can be obtained from the LP formulation for Procedure $\mathcal{M}_{\mathrm{R}}$ by adding constraints and binary variables. It follows that

$$\mathsf{E}_{\boldsymbol{p}}[N_{\mathbf{R}}] \leq \mathsf{E}_{\boldsymbol{p}}[N_{\mathrm{NR}}].$$

Remark 10 Although Procedures $\mathcal{M}_{\mathrm{NR}}$ and $\mathcal{M}_{\mathrm{R}}$ in Tollefson *et al.* (2013) are optimal (over all nonrandomized or randomized procedures, respectively), they have a number of practical drawbacks compared with the classical procedures reviewed before.

1. Each of the classical procedures can be specified with a small number of procedure parameters (that can be published in tables for many values of the input parameters) and a small number of inequalities that are easy to compute. In contrast, the optimal procedures are specified by $\boldsymbol{\phi}_{\boldsymbol{\eta}}$ for each $\boldsymbol{\eta} \in \mathcal{N}$, and do not facilitate representation in concise tables.
2. Computation of an optimal procedure requires the solution of an LP if a randomized procedure is acceptable, or an MIP if a nonrandomized procedure is desired. This requires software to solve the LP or MIP, and software that specifies the formulation of the problem.
3. The optimal procedures require specification of a probability configuration $\boldsymbol{p}$. For some of the classical procedures, it has been shown that the LFC is the SC; but that has not yet been established for the optimal procedures.
4. The LP or MIP formulations for the optimal procedures take a budget constraint as input. All the classical procedures reviewed above are either single-stage or bounded sequential procedures, but the bounds result from the procedure parameters, and not from the input parameters. It is conjectured that for any input

parameters k, θ^*, and P^*, there exists a budget $b(k, \theta^*, P^*)$ such that the set of optimal solutions is the same for all $b \geq b(k, \theta^*, P^*)$ (and thus the LP and MIP formulations can be used to find optimal solutions even if the given problem has no budget constraint); but this conjecture has not yet been established. Also, the sizes (number of decision variables and number of constraints) of the LP and MIP formulations grow as b grows, and thus it is not desirable to choose an unnecessarily large value of b.

Due to these drawbacks, several of the classical procedures have not lost their practical appeal. One of the purposes of this paper is to investigate how much in optimality is sacrificed by using a classical procedure.

For more details concerning the LP and MIP formulations for Procedures $\mathcal{M}_{\mathrm{R}}$ and $\mathcal{M}_{\mathrm{NR}}$, see Tollefson *et al.* (2013).

Methodology

The purpose of the paper is to compare the performances of the MSP procedures reviewed above.

We will use the optimal procedures as a benchmark for comparing the relative performances of the classical procedures. First, we describe how we will compare the different types of MSP procedures, and then, we describe the metrics we will use for the comparisons.

Procedures

For all the procedures reviewed previously, the SC is either proven or conjectured to be the LFC. Therefore, we will conduct most of our comparisons of procedure performance when the probability configuration $\boldsymbol{p}$ is the SC. Furthermore, although we can evaluate the performance of both Procedures $\mathcal{M}_{\mathrm{R}}$ and $\mathcal{M}_{\mathrm{NR}}$ when $\boldsymbol{p}$ is the SC, we will not include Procedure $\mathcal{M}_{\mathrm{NR}}$ in our comparisons for three reasons:

1. The expected number of trials for Procedure $\mathcal{M}_{\mathrm{NR}}$ turns out to be very close to that of Procedure $\mathcal{M}_{\mathrm{R}}$ in most cases, especially for large b.
2. The maximum size of the MIPs that we are able to solve is much smaller than the maximum size of the LPs that we are able to solve. Considering only Procedure $\mathcal{M}_{\mathrm{R}}$ allows us to make comparisons across a larger set of problems than we could if we considered Procedure $\mathcal{M}_{\mathrm{NR}}$.
3. Most importantly, Procedure $\mathcal{M}_{\mathrm{R}}$ is optimal.

All the classical procedures that we consider are either single-stage or bounded sequential procedures. Bounded sequential procedures do not require the specification of a budget b as do the constrained Procedures $\mathcal{M}_{\mathrm{NR}}$ and $\mathcal{M}_{\mathrm{R}}$; rather, their procedure parameters are chosen in order to satisfy the P^*-requirement while minimizing $\mathsf{E}_{\mathrm{SC}}[N]$. In order to level the playing field and make like comparisons, we will choose

a b for each problem and then search only over the subset of possible procedure parameters that ensures that the maximum number of trials is at most b. For some problems, this may result in a particular procedure not being able to achieve P^* at all. Our choice of b for each problem is important and will be discussed later. It is of course possible for us to conduct procedure comparisons without the invocation of a budget constraint, but we do not do so here.

The following are the seven procedures that we will examine. We explain how the budget b will affect the search for the optimal procedure parameters for a given problem. In order to evaluate procedure performance, we developed algorithms to calculate the exact performance characteristics of each procedure. In some cases, that exercise allowed us to update the existing parameter tables found in the literature. When applicable, we refer the reader to those updated tables as well.

1. **Procedure $\mathcal{M}_{\mathrm{BEM}}$:** The single-stage procedure for which the truncation parameter $n_{\mathrm{BEM}} \leq b$.
2. **Procedure $\mathcal{M}_{\mathrm{BK}}$:** The bounded sequential procedure for which the truncation parameter $n_{\mathrm{BK}} \leq b$. We include updated tables for this procedure in Appendix A.
3. **Procedure $\mathcal{M}_{\mathrm{C}'}$:** A modified version of Chen's (1988) inverse sampling Procedure $\mathcal{M}_{\mathrm{C}}$ that includes the strong curtailment stopping rule (see (6.3)). Parameter tables for this new procedure can be found in Appendix B. Given b, choices for this procedure include all parameter combinations with $t \leq n_{\mathrm{C}'} \leq b$.
4. **Procedure $\mathcal{M}_{\mathrm{RA}}$:** The bounded sequential procedure that includes all parameter combinations with $r \leq t \leq (b-1)/k + 1$ (which ensures that the procedure will stop at or before the budget b). We include updated tables for this procedure in Appendix C.
5. **Procedure $\mathcal{M}_{\mathrm{RA}'}$:** The bounded sequential procedure that includes all parameter combinations with $n_{\mathrm{RA}'} \leq b$ and $r \leq t \leq n_{\mathrm{RA}'}/2$ (by strong curtailment). We include updated tables for this procedure in Appendix D.
6. **Procedure $\mathcal{M}_{\mathrm{BG}}$:** The bounded sequential procedure with truncation parameter $n_{\mathrm{BG}} \leq b$.
7. **Procedure $\mathcal{M}_{\mathrm{R}}$:** The optimal randomized constrained sequential procedure under budget b.

Metrics

In this section, we briefly describe some of the performance measures that we will use to compare procedures.

Expected Number of Trials

Given that a procedure meets the appropriate probability requirement, the most-common performance measure in the literature is $\mathsf{E}_{\mathrm{SC}}[N]$. Naturally, $\mathsf{E}_{\mathrm{SC}}[N]$ is quite important to the decision maker when considering a procedure to use, since his

primary goal is normally the minimization of $\mathsf{E}_{\mathrm{SC}}[N]$. In some cases, the decision maker may be concerned with minimizing the maximum possible number of trials taken; however, we assume that in setting a budget, the decision maker is more interested in the former than the latter. A decision maker might also be interested in the expected number of trials in the equal-probability configuration (EPC), $\boldsymbol{p} = (\frac{1}{k}, \frac{1}{k}, \ldots, \frac{1}{k})$. This worst-case expectation, $\mathsf{E}_{\mathrm{EPC}}[N]$, gives the decision maker insight into the possible procedure run length for the most challenging configuration with respect to determining the best alternative.

Procedure Inefficiency Metric

We may also be interested in the deviation of procedure performance with respect to the achievable lower bound on the expected number of trials in the SC, thereby using Procedure $\mathcal{M}_{\mathrm{R}}$ as a benchmark against which we compare other procedures. In order to facilitate an analysis across different problems with widely varying budgets, we use the following *procedure inefficiency* metric, W_{J}, for a given general procedure $\mathcal{M}_{\mathrm{J}}$:

$$W_{\mathrm{J}} \equiv \frac{\mathsf{E}_{\mathrm{SC}}[N_{\mathrm{J}}] - \mathsf{E}_{\mathrm{SC}}[N_{\mathrm{R}}]}{\mathsf{E}_{\mathrm{SC}}[N_{\mathrm{R}}]} = \frac{\mathsf{E}_{\mathrm{SC}}[N_{\mathrm{J}}]}{\mathsf{E}_{\mathrm{SC}}[N_{\mathrm{R}}]} - 1,$$

where $\mathsf{E}_{\mathrm{SC}}[N_{\mathrm{J}}]$ denotes the expected number of trials using procedure $\mathcal{M}_{\mathrm{J}}$ in the SC, and $\mathsf{E}_{\mathrm{SC}}[N_{\mathrm{R}}]$ is the expected number of trials using optimal Procedure $\mathcal{M}_{\mathrm{R}}$ in the SC. We can think of procedure inefficiency as the fractional increase in the expected number of trials due to using procedure $\mathcal{M}_{\mathrm{J}}$ instead of the optimal procedure.

Note that W_{J} depends on problem input k, θ^*, P^*, and b, although the notation does not show the dependence.

Often, we may want to evaluate procedure performance over a range of problem input. For that purpose, we extend the procedure inefficiency metric heuristically by calculating the *mean procedure inefficiency*, $\overline{W}_{\mathrm{J}}$, for a range of P^*-values, $P^\star_{\mathrm{J},0}, P^\star_{\mathrm{J},1}, \ldots, P^\star_{\mathrm{J},m_{\mathrm{J}}}$, where m_{J} is the total number of P^*-values at which we calculate $\mathsf{E}_{\mathrm{SC}}[N_{\mathrm{J}}]$, and where we assume for ease of exposition that $P^\star_{\mathrm{J},0} \le P^\star_{\mathrm{J},1} \le \cdots \le P^\star_{\mathrm{J},m_{\mathrm{J}}}$. (We could also carry out an analogous evaluation based on a range of θ^*-values, but we will not do so here.) Since we only calculate the performance at each P^* increment, the value $\mathsf{E}^i_{\mathrm{SC}}[N_{\mathrm{J}}]$ calculated at $P^\star_{\mathrm{J},i}$ will serve as the approximate expected number of trials for the entire interval $(P^\star_{\mathrm{J},i-1}, P^\star_{\mathrm{J},i}]$. Let I be the overall probability interval of P^*-values we are considering. The mean procedure inefficiency, $\overline{W}^I_{\mathrm{J}}$, for procedure $\mathcal{M}_{\mathrm{J}}$ over interval I is defined as

$$\overline{W}^I_{\mathrm{J}} \equiv \frac{\displaystyle\sum_{i=1}^{m_{\mathrm{J}}} \mathsf{E}^i_{\mathrm{SC}}[N_{\mathrm{J}}] \left(P^\star_{\mathrm{J},i} - P^\star_{\mathrm{J},i-1}\right)}{\displaystyle\sum_{i=1}^{m_{\mathrm{R}}} \mathsf{E}^i_{\mathrm{SC}}[N_{\mathrm{R}}] \left(P^\star_{\mathrm{R},i} - P^\star_{\mathrm{R},i-1}\right)} - 1.$$

Note that our definition does not require constant increment size, nor do we need to use the same increment sizes for both procedures. It does, however, require the same

overall P^*-interval I:

$$P^{\star}_{\mathrm{J},m_{\mathrm{J}}} \;=\; P^{\star}_{\mathrm{R},m_{\mathrm{R}}} \text{ and } P^{\star}_{\mathrm{J},0} \;=\; P^{\star}_{\mathrm{R},0}.$$

Keep in mind that the metric is specific to a particular combination of k, θ^*, and b, as well as a particular set of $P^{\star}_{\mathrm{J},i}$'s.

We must be careful here when comparing procedures, since $\overline{W}^{I}_{\mathrm{J}}$ compares each procedure with the optimum procedure, based upon the P^*-domain of each procedure, i.e., the range from $1/k$ to the maximum achievable $\mathsf{P}_{\mathrm{SC}}(\mathrm{CS})$ for each procedure. Procedures $\mathcal{M}_{\mathrm{BEM}}$, $\mathcal{M}_{\mathrm{BK}}$, $\mathcal{M}_{\mathrm{C}'}$, $\mathcal{M}_{\mathrm{RA}'}$, and $\mathcal{M}_{\mathrm{R}}$ have the same P^*-domain. On the other hand, procedures $\mathcal{M}_{\mathrm{RA}}$ and $\mathcal{M}_{\mathrm{BG}}$ may have different domains than each other and the remaining procedures. Considering only the mean procedure inefficiency metric fails to recognize that the domains of the procedures may be different. If we want to compare two procedures, say procedures $\mathcal{M}_{\mathrm{J}}$ and $\mathcal{M}_{\mathrm{L}}$, over a common domain, we create a new metric, which we call the *mean relative procedure performance*, defined as follows:

$$\overline{V}^{I}_{\mathrm{J,L}} \;\equiv\; \frac{\displaystyle\sum_{i=1}^{m_{\mathrm{J}}} \mathsf{E}^{i}_{\mathrm{SC}}[N_{\mathrm{J}}]\;\left(P^{\star}_{\mathrm{J},i} - P^{\star}_{\mathrm{J},i-1}\right)}{\displaystyle\sum_{i=1}^{m_{\mathrm{L}}} \mathsf{E}^{i}_{\mathrm{SC}}[N_{\mathrm{L}}]\;\left(P^{\star}_{\mathrm{L},i} - P^{\star}_{\mathrm{L},i-1}\right)} - 1,$$

where I is the intersection of the domains of procedures $\mathcal{M}_{\mathrm{J}}$ and $\mathcal{M}_{\mathrm{L}}$. A positive value indicates that procedure $\mathcal{M}_{\mathrm{L}}$ performs better than procedure $\mathcal{M}_{\mathrm{J}}$ over the interval of interest; a negative value indicates the opposite.

Distributional Metrics

We can enumerate all of the possible stopping vectors for any MSP procedure, and we can develop algorithms to determine the probability of arriving and stopping at each possible stopping vector. All MSP procedures under a finite budget have a finite number of stopping points; therefore, we have *complete* information about the discrete probability distribution of the number of trials required by the procedure. With this information, we can calculate metrics such as the mean, median, mode, variance, and quantiles of the random variable N.

Performance Comparison

We compared the seven procedures described previously using our proposed performance metrics. In Appendix E, we include comparison tables for the 36 possible combinations of $k \in \{2, 3, 4\}$, $\theta^* \in \{1.6, 2.0, 2.4, 3.0\}$, and $P^* \in \{0.75, 0.90, 0.95\}$,

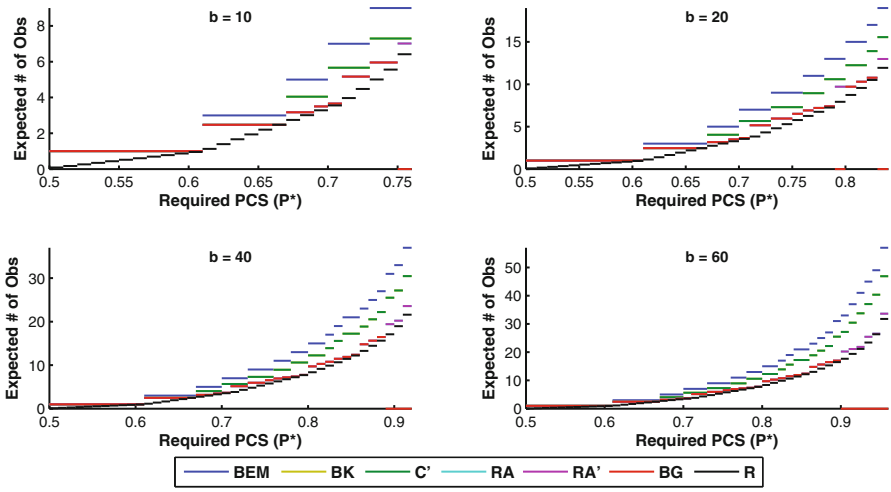


Fig. 6.1 Procedure Comparison Plots for $k = 2$, $\theta^* = 1.6$, $b \in \{10, 20, 40, 60\}$

with a single budget b for each. For the examples in that appendix, we set the budget b equal to the optimal truncation parameter n_{BG} for Procedure $\mathcal{M}_{\mathrm{BG}}$—a procedure which typically performs at least as well as the other classical procedures. Thus setting $b = n_{\mathrm{BG}}$ typically creates conditions favorable to the best of the existing procedures, thereby minimizing its inefficiency compared to the optimal procedure.

The tables give results for both $\mathsf{E}_{\mathrm{SC}}[N]$ and $\mathsf{E}_{\mathrm{EPC}}[N]$. For those 36 cases, Procedure $\mathcal{M}_{\mathrm{BG}}$ usually performs better, in terms of $\mathsf{E}_{\mathrm{SC}}[N]$, than Procedure $\mathcal{M}_{\mathrm{RA}'}$; however, Tables 6.10 and 6.18 in Appendix E show that this is not always the case. While certain trends may be evident across the tables, it is hard to draw any completely general conclusions, particularly since our choice of the budget b will affect the results and the relative performance of the procedures.

Relative Procedure Performance

We try to account for the effect of the budget b by examining procedure performance across the 12 combinations of $k \in \{2, 3, 4\}$ and $\theta^* \in \{1.6, 2.0, 2.4, 3.0\}$. For each combination, we choose four values for the budget b and then consider all possible P^* values between $1/k$ and 0.99, in increments of 0.01. We plot our results to visualize relative procedure performance.

Figure 6.1 shows a series of four charts, one each for $b \in \{10, 20, 40, 60\}$, with $k = 2$ and $\theta^* = 1.6$. The expected performance, $\mathsf{E}_{\mathrm{SC}}[N]$, of the seven procedures is plotted as a function of P^*. Procedure $\mathcal{M}_{\mathrm{R}}$, the lower bound, is shown in black.

Some Interesting Results for $k = 2$

The plotted results for $k = 2$ illustrate some intriguing findings, some of which have heretofore not been explained in the literature. First, Procedures $\mathcal{M}_{\mathrm{BK}}$ and $\mathcal{M}_{\mathrm{C}'}$ perform identically when $k = 2$. Specifically, for Procedure $\mathcal{M}_{\mathrm{C}'}$, stops due to inverse sampling (i.e., the parameter t) are identical to stops due to strong curtailment (i.e., the parameter $n_{\mathrm{C}'}$) when $n_{\mathrm{C}'} = 2t$. To see this, note that if $k = 2$ and $n = 2t$, then a stop due to curtailment implies

$$\eta_{[2]m} - \eta_{[1]m} \quad \geq \quad n - m \;=\; 2t - (\eta_{[2]m} + \eta_{[1]m}),$$

which is true if and only if $\eta_{[2]m} \geq t$. Since the procedure must stop when $\eta_{[2]m} = t$, the curtailment and inverse sampling stopping conditions are satisfied simultaneously. Choosing $n > 2t$ means that the procedure will always stop due to the inverse sampling condition. Choosing $t > n/2$ means that the procedure will always stop due to the curtailment condition. Therefore, when $k = 2$, we can represent any two-parameter Procedure $\mathcal{M}_{\mathrm{C}'}$ equivalently as the single-parameter Procedure $\mathcal{M}_{\mathrm{BK}}$ with $n_{\mathrm{BK}} = \min\{n_{\mathrm{C}'}, 2t\}$.

There is another unique characteristic of curtailment when $k = 2$ that is not obvious from the figure. Namely, for $k = 2$, the set of cumulative win vectors that are stops due to curtailment when $n = n_0$ with n_0 even, is identical to the set of cumulative win vectors that are stops due to curtailment when $n = n_0 - 1$. To see that this is true, we show first that when $k = 2$ and $n = n_0$ with n_0 even, a cumulative win vector at trial m, $\boldsymbol{\eta}_m$, is a stop due to curtailment if and only if $\eta_{[2]m} = n_0/2$. If $\boldsymbol{\eta}_m$ is a curtailment stop, then

$$\eta_{[2]m} - \eta_{[1]m} \quad \geq \quad n_0 - m \;=\; n_0 - (\eta_{[2]m} + \eta_{[1]m})$$

if and only if $\eta_{[2]m} \geq n_0/2$, where $n_0/2$ is an integer since n_0 is even. Let $\eta_{[2]m} = n_0/2 + \nu$, where $\nu \in \mathbb{N} \cup 0$, and therefore $\boldsymbol{\eta}_m = (n_0/2 + \nu, m - n_0/2 - \nu)$. Then

$$\eta_{[2]m} - \eta_{[1]m} = n_0/2 + \nu - (m - n_0/2 - \nu) = n_0 - m + 2\nu,$$

which always satisfies the curtailment condition. But not all values of ν may result in a feasible stopping point because it is possible that the procedure will have stopped at a previous trial. Therefore, we consider trial $m - 1$ under two possible cases: either the alternative associated with $\eta_{[2]m}$ won trial m or the other alternative won. In the former, the difference between the number of wins of the two alternatives after trial $m - 1$ is

$$n_0/2 + \nu - 1 - (m - n_0/2 - \nu) \;=\; n_0 - m + 2\nu - 1.$$

Since there are $n_0 - m + 1$ trials remaining, the curtailment condition is met at trial $m - 1$ if $\nu > 0$. In the latter case, the difference is

$$n_0/2 + \nu - (m - n_0/2 - \nu - 1) \;=\; n_0 - m + 2\nu + 1,$$

which is a stop for all $\nu \geq 0$. Only $\nu = 0$ allows the procedure to reach $\boldsymbol{\eta}_m$; and $\nu > 0$ results in an $\boldsymbol{\eta}_m$ that is infeasible. Therefore, if $\boldsymbol{\eta}_m$ is a stop due to curtailment, then $\eta_{[2]m} = n_0/2$.

To show the reverse, if $\eta_{[2]m} = n_0/2$, then

$$\eta_{[2]m} - \eta_{[1]m} = n_0/2 - (m - n_0/2) = n_0 - m,$$

which satisfies the curtailment stopping condition. We have now shown that when $k = 2$ and $n = n_0$, n_0 even, a cumulative win vector $\boldsymbol{\eta}_m$ is a stop due to curtailment if and only if $\eta_{[2]m} = n_0/2$.

Now, we show that when $k = 2$ and $n = n_0 - 1$ with n_0 even, a cumulative win vector $\boldsymbol{\eta}_m$ is a stop due to curtailment if and only if $\eta_{[2]m} = n_0/2$. If $\boldsymbol{\eta}_m$ is a curtailment stop, then

$$\eta_{[2]m} - \eta_{[1]m} \quad \geq \quad n - m \;=\; n_0 - 1 - (\eta_{[2]m} + \eta_{[1]m})$$

if and only if $\eta_{[2]m} \geq (n_0 - 1)/2$. But $(n_0 - 1)/2$ is not an integer since n_0 is even, so $\eta_{[2]m} = n_0/2$ is the smallest integer that meets the stopping condition. Let $\eta_{[2]m} = n_0/2 + \nu$, where $\nu \in \mathbb{N} \cup 0$, and therefore $\boldsymbol{\eta}_m = (n_0/2 + \nu, m - n_0/2 - \nu)$. By the same reasoning as the previous proof, if $\nu > 0$, then $\boldsymbol{\eta}_m$ is not a feasible stopping point. Only $\nu = 0$ results in a feasible stopping point at trial m. Therefore, if $\boldsymbol{\eta}_m$ is a stop due to curtailment, then $\eta_{[2]m} = n_0/2$.

To show the reverse, if $\eta_{[2]m} = n_0/2$, then

$$\eta_{[2]m} - \eta_{[1]m} = n_0/2 - (m - n_0/2) = n_0 - m,$$

which satisfies the curtailment stopping condition since there are $n_0 - m - 1$ trials remaining. We have now shown that when $k = 2$ and $n = n_0 - 1$ with n_0 even, a cumulative win vector at trial m, $\boldsymbol{\eta}_m$, is a stop due to curtailment if and only if $\eta_{[2]m} = n_0/2$. As a result, when $k = 2$, the set of cumulative win vectors that are stops due to curtailment when $n = n_0$ with n_0 even, is identical to the set of cumulative win vectors that are stops due to curtailment when $n = n_0 - 1$.

Returning to our discussion of the relationships between Procedures $\mathcal{M}_{\mathrm{BK}}$ and $\mathcal{M}_{\mathrm{C}'}$, we can now state that when $k = 2$, we can represent any two-parameter Procedure $\mathcal{M}_{\mathrm{C}'}$ equivalently as the single-parameter Procedure $\mathcal{M}_{\mathrm{BK}}$ with $n_{\mathrm{BK}} = \min\{n_{\mathrm{C}'}, 2t - 1\}$.

Similarly, Procedures $\mathcal{M}_{\mathrm{RA}}$ and $\mathcal{M}_{\mathrm{RA}'}$ also perform identically when $k = 2$. In this case, stops due to t are identical to stops due to $n_{\mathrm{RA}'}$ when $n_{\mathrm{RA}'} = 2t - 1$. Thus, when $k = 2$, Procedure $\mathcal{M}_{\mathrm{RA}'}$ with a particular $(n_{\mathrm{RA}'}, r', t')$-triplet is identical to Procedure $\mathcal{M}_{\mathrm{RA}}$ with a corresponding (r, t)-pair in which $r = r'$ and $t = \min\left\{t', \lfloor (n_{\mathrm{RA}'} + 1)/2 \rfloor\right\}$, where $\lfloor x \rfloor$ is the floor function (i.e., rounds x down to the nearest integer).

Procedure $\mathcal{M}_{\mathrm{BG}}$ has significant overlap with Procedures $\mathcal{M}_{\mathrm{RA}}$ and $\mathcal{M}_{\mathrm{RA}'}$ when $k = 2$. For Procedure $\mathcal{M}_{\mathrm{BG}}$, the parameter z_m in the stopping criteria (6.4) is based on the differences between the alternative with the most wins and the other alternatives. When $k = 2$, there is only one difference to consider, in which case exactly the same stopping behavior can be achieved with the r parameter in Procedures $\mathcal{M}_{\mathrm{RA}}$ and $\mathcal{M}_{\mathrm{RA}'}$. To see that this is true, recall the Procedure $\mathcal{M}_{\mathrm{BG}}$ stopping condition:

$$z_m \;=\; \left(\frac{1}{\theta^*}\right)^{\eta_{[2]m} - \eta_{[1]m}} \quad \leq \quad \frac{1 - P^*}{P^*}.$$

Solving for $\eta_{[2]m} - \eta_{[1]m}$, and using the fact that $\eta_{[2]m} - \eta_{[1]m}$ must be an integer, we get

$$\eta_{[2]m} - \eta_{[1]m} \;\geq\; r' \;\equiv\; \left\lceil \frac{\ln(P^*) - \ln(1-P^*)}{\ln(\theta^*)} \right\rceil,$$

where $\lceil x \rceil$ is the ceiling function (i.e., rounds x up to the nearest integer). As with the previous discussions, the parameter n_{BG} acts similarly to the t parameter when $k = 2$. The main aspect that makes Procedure $\mathcal{M}_{\mathrm{BG}}$ differ from Procedures $\mathcal{M}_{\mathrm{RA}}$ and $\mathcal{M}_{\mathrm{RA}'}$ in some cases is that parameter r can be chosen in the latter procedures independently of the other parameters, whereas z_m for Procedure $\mathcal{M}_{\mathrm{BG}}$ is determined by the problem parameters (i.e., cannot be chosen independently).

An Anomaly of Procedure $\mathcal{M}_{\mathrm{BG}}$

The lack of flexibility in the stopping criteria (6.4) leads to some interesting behavior with respect to Procedure $\mathcal{M}_{\mathrm{BG}}$. For some smaller values of P^*, there exists no $n_{\mathrm{BG}} \leq b$ such that $\mathsf{P}(\mathrm{CS})$ of Procedure $\mathcal{M}_{\mathrm{BG}}$ is greater than or equal to P^*, but (surprisingly) for some larger values of P^*, there exists $n_{\mathrm{BG}} \leq b$ such that $\mathsf{P}(\mathrm{CS})$ of Procedure $\mathcal{M}_{\mathrm{BG}}$ is greater than or equal to P^*. Such nonmonotonic behavior can be seen in the chart for $b = 20$ in Fig. 6.1. In that case, given $P^* = 0.80$, there exists no $n_{\mathrm{BG}} \leq 20$ such that $\mathsf{P}(\mathrm{CS})$ of Procedure $\mathcal{M}_{\mathrm{BG}}$ is greater than or equal to 0.80; but given $P^* = 0.81$, there exists $n_{\mathrm{BG}} \leq 20$ such that $\mathsf{P}(\mathrm{CS})$ of Procedure $\mathcal{M}_{\mathrm{BG}}$ is greater than 0.81. For this example, given $P^* = 0.80$, n_{BG} must be increased to 25 before $\mathsf{P}(\mathrm{CS})$ of Procedure $\mathcal{M}_{\mathrm{BG}}$ is greater than 0.80. This nonmonotonicity is a characteristic peculiar to Procedure $\mathcal{M}_{\mathrm{BG}}$ that is not shared by any of the other procedures discussed here. In some cases, such as for $k = 2$, $\theta^* = 3$, $P^* = 0.90$, there exists no n_{BG} at all such that $\mathsf{P}(\mathrm{CS})$ of Procedure $\mathcal{M}_{\mathrm{BG}}$ is greater than or equal to P^*. These anomalies are due to the fact that the stopping condition $z_m \leq (1-P^*)/P^*$ was originally developed by BKS (1968) for an unbounded procedure. When BG (1985b, 1986) added the additional stopping parameter, n_{BG}, thereby bounding the procedure to save trials in expectation, the truncated procedure lost the ability to achieve P^*-values for which the unboundedness of the number of trials was required.

Let $\Delta_{21} \equiv \eta_{[2]m} - \eta_{[1]m}$. Then for our example with $k = 2$ and $\theta^* = 1.6$,

$$z_m = \left(\frac{1}{\theta^*}\right)^{\Delta_{21}} = \begin{cases} 0.625 & \text{if } \Delta_{21} = 1 \\ 0.391 & \text{if } \Delta_{21} = 2 \\ 0.244 & \text{if } \Delta_{21} = 3 \\ 0.153 & \text{if } \Delta_{21} = 4. \end{cases}$$

Thus, for $P^* = 0.79$, we have $(1 - P^*)/P^* = 0.266$, and we stop at difference $\Delta_{21} = 2$ when $z_m = 0.391$; for $P^* = 0.8$, $(1 - P^*)/P^* = 0.25$, and we stop at $\Delta_{21} = 3$ when $z_m = 0.244$; and for $P^* = 0.81$, $(1 - P^*)/P^* = 0.235$, and we stop at $\Delta_{21} = 4$ when $z_m = 0.153$. As it turns out, stops from z_m values that are less than, but very close to, $(1 - P^*)/P^*$ require more trials to achieve P^*, as is the case for

$P^* = 0.8$ above. For the extreme example when $k = 2$, $\theta^* = 3$, and $P^* = 0.9$, the stopping condition quantity $(1 - P^*)/P^* = 1/9$, which is exactly equal to z_m when $\Delta_{21} = 2$. In that case, the procedure requires an infinite trial budget to achieve P^*.

How can we explain this phenomenon? We first refer to the discussion in Chap. 7 of Tollefson (2012) which shows that, for the SC with $k = 2$, the posterior (aka "conditional") P(CS), given a stop at cumulative win vector $\boldsymbol{\eta}$, is the remarkably simple expression

$$\frac{(\theta^*)^{\eta[2]}}{(\theta^*)^{\eta[1]} + (\theta^*)^{\eta[2]}}.$$

If we require this expression to be at least P^*, then a little algebra reveals that

$$z_m = \left(\frac{1}{\theta^*}\right)^{\eta[2]-\eta[1]} \quad \leq \quad \frac{1 - P^*}{P^*}.$$

Thus, the original BKS stopping rule requires stopping at the first point for which the posterior P(CS) $> P^*$ when the configuration is the LFC (which BKS prove is the SC). Now, back to our problem. The prior P(CS) is the expectation of the posterior P(CS), which is the quantity that we need to be at least P^*. For the above example with $k = 2$, $\theta^* = 3$, and $P^* = 0.9$, we always stop when $z_m = 1/9$, i.e., when $\eta_{[2]} - \eta_{[1]} = 2$. This means that we always stop exactly when the posterior P(CS) $= P^*$. Clearly then, we can *never* incorporate a truncation point n_{BG} for Procedure $\mathcal{M}_{\mathrm{BG}}$ because stops due to that criterion will result in a posterior P(CS) $< P^*$, while the other stopping points have exactly the required P(CS). Thus, any curtailment would result in a prior P(CS) $< P^*$ (although the difference would admittedly be small if n_{BG} were large). This is also the case when the stopping values of z_m are very close to (and obviously less than) their required value of $(1 - P^*)/P^*$. The closer that the posterior P(CS) of our z_m-condition stopping points gets to our required P^* (which occurs as z_m approaches $(1 - P^*)/P^*$ from below), the larger the curtailment point n_{BG} must be since we have very little excess posterior P(CS) that can be offset by the curtailment points. As n_{BG} increases, the contribution to the prior P(CS) due to curtailment stopping points decreases. As we increase P^*, we must increase n_{BG} until we need to move to the next discrete value of z_m (i.e., the next larger difference), at which point the required n_{BG} may not necessarily be larger than the previous value—as is the case when we move from $P^* = 0.79$ to 0.80 to 0.81, which is the particular case we are considering here.

Results for $k > 2$

We now examine similar plots when the number of alternatives is larger than $k = 2$. Figure 6.2 shows a series of charts for $b \in \{5, 10, 25, 40\}$ with $k = 4$ and $\theta^* = 2.4$. In this figure, we see that the relationships between the procedures are more complex than they were for $k = 2$. None of the procedures perform identically, as some did for $k = 2$, but some do perform similarly when b is low.

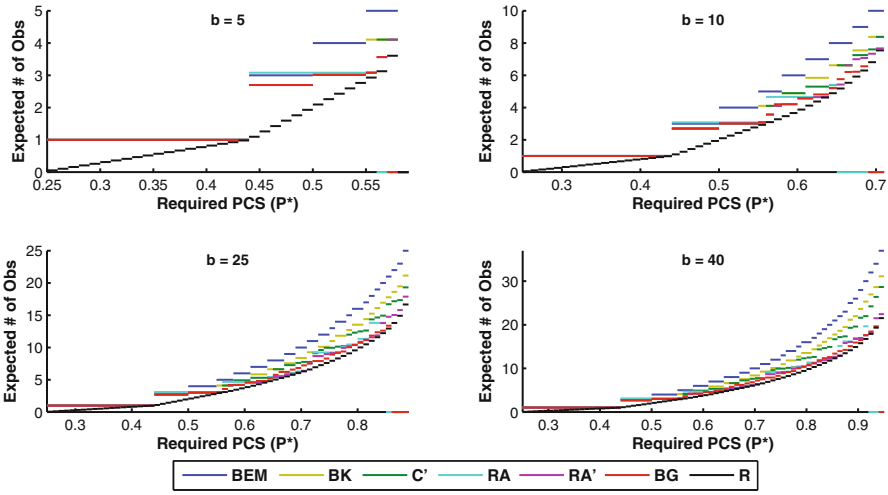


Fig. 6.2 Procedure Comparison Plots for $k = 4$, $\theta^* = 2.4$, $b \in \{5, 10, 25, 40\}$

We note some relationships between the procedures (regardless of k) that are reflected in Fig. 6.2.

- Procedure $\mathcal{M}_{\mathrm{BK}}$ with parameter n_{BK} is a special case of Procedure $\mathcal{M}_{\mathrm{C}'}$ with parameter pair $(n_{\mathrm{C}'}, t)$, where $n_{\mathrm{C}'} = n_{\mathrm{BK}}$ and $t \geq \lceil n_{\mathrm{C}'}/2 \rceil$.
- Procedure $\mathcal{M}_{\mathrm{C}'}$ with parameter pair $(n_{\mathrm{C}'}, t)$ is a special case of Procedure $\mathcal{M}_{\mathrm{RA}'}$ with parameter triplet $(n_{\mathrm{RA}'}, r', t')$, where $n_{\mathrm{RA}'} = n_{\mathrm{C}'}$, $r' \geq \lceil n_{\mathrm{C}'}/2 \rceil$, and $t' = t$.
- Procedure $\mathcal{M}_{\mathrm{RA}}$ with parameter pair (r, t) is a special case of Procedure $\mathcal{M}_{\mathrm{RA}'}$ with parameter triplet $(n_{\mathrm{RA}'}, r', t')$, where $n_{\mathrm{RA}'} \geq kt + 1$, $r' = r$, and $t' = t$.
- Procedure $\mathcal{M}_{\mathrm{BK}}$ always performs better than Procedure $\mathcal{M}_{\mathrm{BEM}}$.

These relationships guarantee a relative ordering between Procedures $\mathcal{M}_{\mathrm{RA}}$ and $\mathcal{M}_{\mathrm{RA}'}$, and among Procedures $\mathcal{M}_{\mathrm{BEM}}$, $\mathcal{M}_{\mathrm{BK}}$, $\mathcal{M}_{\mathrm{C}'}$, and $\mathcal{M}_{\mathrm{RA}'}$, which are reflected in Fig. 6.2, as well as in Fig. 6.1. Thus, when considering the best performing procedure (not including the optimal procedures), we need only compare Procedures $\mathcal{M}_{\mathrm{RA}'}$ and $\mathcal{M}_{\mathrm{BG}}$. Figure 6.2 shows that there are regions in which Procedure $\mathcal{M}_{\mathrm{RA}'}$ (and even Procedure $\mathcal{M}_{\mathrm{RA}}$) perform better than $\mathcal{M}_{\mathrm{BG}}$ and regions (seemingly more numerous) in which the opposite is true. We will address that issue in more detail later in this section.

Another insight from Fig. 6.2 is the seemingly counterintuitive fact that Procedure $\mathcal{M}_{\mathrm{BK}}$ and even Procedure $\mathcal{M}_{\mathrm{BEM}}$ perform better than Procedure $\mathcal{M}_{\mathrm{RA}}$ for some P^*-values. The reason for this phenomenon, which only occurs when b is low, is that the budget poses a more significant constraint on Procedure $\mathcal{M}_{\mathrm{RA}}$ than it does for Procedures $\mathcal{M}_{\mathrm{BEM}}$ and $\mathcal{M}_{\mathrm{BK}}$. For a Procedure $\mathcal{M}_{\mathrm{RA}}$ parameter pair (r, t) to be possible, we must have $b \geq k(t-1) + 1$. For example, if $k = 4$ and $b = 5$, then we require that $t \leq 2$, resulting in the possible parameter pairs in Table 6.1. Note that when $r = 1$, the procedure stops after one trial, regardless of t; therefore, there

Table 6.1 Parameters for Procedures $\mathcal{M}_{\mathrm{RA}}$ and $\mathcal{M}_{\mathrm{BK}}$ for $k = 4$, $\theta^* = 2.4$, and $b = 5$

Procedure $\mathcal{M}_{\mathrm{RA}}$			Procedure $\mathcal{M}_{\mathrm{BK}}$		
Parameters	$\mathsf{P}_{\mathrm{SC}}(\mathrm{CS})$	$\mathsf{E}_{\mathrm{SC}}[N]$	Parameters	$\mathsf{P}_{\mathrm{SC}}(\mathrm{CS})$	$\mathsf{E}_{\mathrm{SC}}[N]$
$(r = 1, t = 1)$	0.4444	1.000	$n_{\mathrm{BK}} = 1$	0.4444	1.000
$(r = 2, t = 2)$	0.5690	3.080	$n_{\mathrm{BK}} = 2$	0.4444	1.000
			$n_{\mathrm{BK}} = 3$	0.5085	2.700
			$n_{\mathrm{BK}} = 4$	0.5559	3.012
			$n_{\mathrm{BK}} = 5$	0.5849	4.104

is no need to include results for $(r = 1, t = 2)$. Consider $P^* = 0.5$. Procedure $\mathcal{M}_{\mathrm{BK}}$ with $n_{\mathrm{BK}} = 3$ can achieve P^* with $\mathsf{E}_{\mathrm{SC}}[N] = 2.7$, but Procedure $\mathcal{M}_{\mathrm{RA}}$ with $(r = 2, t = 2)$, the only parameter pair that achieves P^*, requires $\mathsf{E}_{\mathrm{SC}}[N] = 3.08$. Even Procedure $\mathcal{M}_{\mathrm{BEM}}$ with $n_{\mathrm{BEM}} = 3$ achieves P^* with a lower $\mathsf{E}_{\mathrm{SC}}[N] = 3$. These results agree with Fig. 6.2, although the results for Procedure $\mathcal{M}_{\mathrm{BK}}$ are masked by that of Procedure $\mathcal{M}_{\mathrm{BG}}$ at $P^* = 0.5$.

The anomalies that we noticed for Procedure $\mathcal{M}_{\mathrm{BG}}$ when $k = 2$ occasionally manifest for $k > 2$, but very rarely. In fact, such phenomena do not appear at all in Fig. 6.2 for $k = 4$. While these anomalies are more common for $k = 2$, we speculate that a larger k allows for a greater number of possible z_m-values and thus fewer anomalies from large gaps between the discrete z_m-values.

Mean Procedure Inefficiency

To supplement the visual insights provided by our charts, we also calculated the metrics W_{J} and $\overline{W}_{\mathrm{J}}^{I}$. The tables in Appendix E for our 36 procedure comparisons include values for W_{J} as a percentage in the column labeled "$100W_{\mathrm{J}}$." We also calculated $\overline{W}_{\mathrm{J}}^{I}$ for the 12 combinations of $k \in \{2, 3, 4\}$ and $\theta^* \in \{1.6, 2.0, 2.4, 3.0\}$ at four values of b. For those examples, we use a constant increment size of 0.01 for P^* (except for the interval between $1/3$ and 0.34 when $k = 3$). In addition, we use the same increment sizes for each procedure $\mathcal{M}_{\mathrm{J}}$ as we do for Procedure $\mathcal{M}_{\mathrm{R}}$. The following are the four relevant intervals:

- $\overline{W}_{\mathrm{J}}^{\star}$ is calculated from the entire interval from $1/k$ to the maximum achievable P^* by procedure $\mathcal{M}_{\mathrm{J}}$. For example, the maximum achievable P^* for Procedure $\mathcal{M}_{\mathrm{RA}}$ with $k = 2$, $\theta^* = 2$, and $b = 20$ is 0.9313. The interval considered in this comparison is then (for both Procedures $\mathcal{M}_{\mathrm{RA}}$ and $\mathcal{M}_{\mathrm{R}}$) from 0.50 to 0.93, even though Procedure $\mathcal{M}_{\mathrm{R}}$ can achieve a higher P^* at $b = 20$. Thus, we should qualify the mean procedure inefficiency metric by calling it the *mean procedure inefficiency over its achievable P^*-region* when that region is shorter than that of the optimal procedure. However, we omit the qualifier for the sake of brevity.
- $\overline{W}_{\mathrm{J}}^{75}$ is calculated from the interval from $1/k$ to the maximum achievable P^* or 0.75, whichever is less.

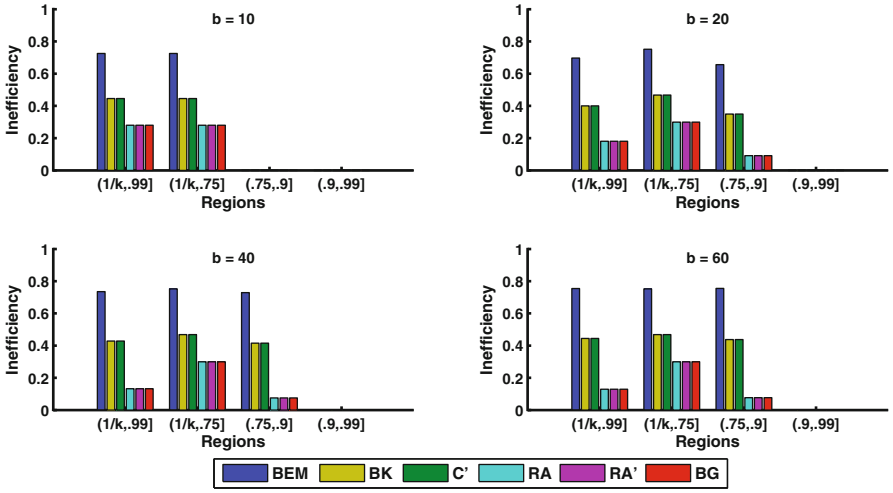


Fig. 6.3 Mean Procedure Inefficiency for $k = 2$, $\theta^* = 1.6$, $b \in \{10, 20, 40, 60\}$

- $\overline{W}_{\mathrm{J}}^{90}$ is calculated from the interval from 0.75 to the maximum achievable P^* or 0.9, whichever is less. If procedure $\mathcal{M}_{\mathrm{J}}$ cannot achieve a P^* above 0.75, this metric is not defined.
- $\overline{W}_{\mathrm{J}}^{95}$ is calculated from the interval from 0.9 to the maximum achievable P^* or 0.95, whichever is less. If procedure $\mathcal{M}_{\mathrm{J}}$ cannot achieve a P^* above 0.9, this metric is not defined.

Figure 6.3 shows the mean procedure inefficiencies for each of the four P^*-regions, with $b \in \{10, 20, 40, 60\}$, $k = 2$, and $\theta^* = 1.6$ (i.e., corresponding to the charts in Fig. 6.1). Here we see numerically what we noted in the plots of raw performance: Procedures $\mathcal{M}_{\mathrm{BK}}$ and $\mathcal{M}_{\mathrm{C}'}$ have the same performances, as do Procedures $\mathcal{M}_{\mathrm{RA}}$ and $\mathcal{M}_{\mathrm{RA}'}$. We also see that Procedure $\mathcal{M}_{\mathrm{BG}}$ performs similarly to Procedures $\mathcal{M}_{\mathrm{RA}}$ and $\mathcal{M}_{\mathrm{RA}'}$. The absence of a set of bars for any region means that none of the procedures can achieve P^* in that interval.

Figure 6.4 shows the mean procedure inefficiencies for each of the four regions, with $b \in \{5, 10, 25, 40\}$, $k = 4$, and $\theta^* = 2.4$. Due to the larger k and lower numbers of trials, there are again several regions within which none of the procedures can achieve a particular P^*, even though θ^* is larger than for Fig. 6.3. The relative ordering of procedure performance discussed in connection with Fig. 6.2 is evident here, as is the poorer performance of Procedure $\mathcal{M}_{\mathrm{RA}}$ when b is low.

The performances of Procedures $\mathcal{M}_{\mathrm{RA}'}$ and $\mathcal{M}_{\mathrm{BG}}$ dominate those of all other procedures except the optimal procedures. Therefore, we narrow our attention to just the two procedures by examining the mean relative procedure performance metric $\overline{V}^{I}_{\mathrm{RA}',\mathrm{BG}}$.

Figure 6.5 shows the results for this comparison over the same sets of k, θ^*, and b we have analyzed thus far. The figure consists of 12 charts, one for each possible combination of $k \in \{2, 3, 4\}$ and $\theta^* \in \{1.6, 2.0, 2.4, 3.0\}$. Each row of charts represents a particular number of alternatives k, and each column of charts represents

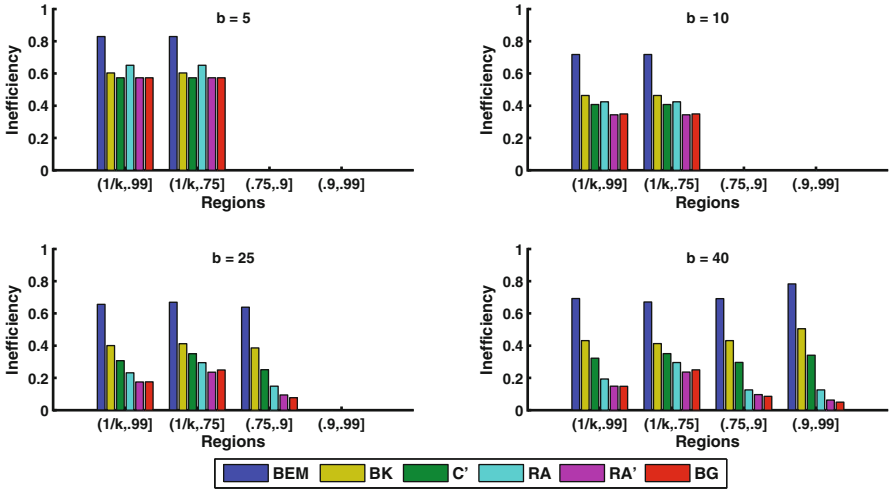


Fig. 6.4 Mean Procedure Inefficiency for $k = 4$, $\theta^* = 2.4$, $b \in \{5, 10, 25, 40\}$

a particular θ^*. Within each chart, the horizontal axis includes four groups of four bars. Each group of four bars represents a particular budget b, which is labeled on the axis. The four bars within each group represent the same four regions of interest (i.e., intervals I) as those described for $\overline{W}^I_{\mathrm{J}}$. The vertical axis on each chart is the value of the metric $\overline{V}^I_{\mathrm{RA'},\mathrm{BG}}$. Thus, within each group, the four bars from left to right represent values of $\overline{V}^{\star}_{\mathrm{RA'},\mathrm{BG}}$, $\overline{V}^{75}_{\mathrm{RA'},\mathrm{BG}}$, $\overline{V}^{90}_{\mathrm{RA'},\mathrm{BG}}$, and $\overline{V}^{95}_{\mathrm{RA'},\mathrm{BG}}$, respectively. Bars above the center line indicate regions within which Procedure $\mathcal{M}_{\mathrm{BG}}$ performs better than Procedure $\mathcal{M}_{\mathrm{RA'}}$. Bars below indicate regions within which the opposite is true. Points at which there are no bars indicate either identical or nearly identical performance, or a region within which the procedures cannot compete.

The greater frequency of bars above versus below shows that for the regions and problems we examined, Procedure $\mathcal{M}_{\mathrm{BG}}$ performs better than $\mathcal{M}_{\mathrm{RA'}}$ more often than the reverse. However, we point out again that this comparison is over the intersection of their domains. In some cases, Procedure $\mathcal{M}_{\mathrm{RA'}}$ can attain a higher maximum P^* for a problem than can Procedure $\mathcal{M}_{\mathrm{BG}}$, which may provide a decisive advantage for particular situations. Of course, we should not lose sight of the fact that Procedure $\mathcal{M}_{\mathrm{R}}$ (and Procedure $\mathcal{M}_{\mathrm{NR}}$) always perform as well as or better than all other existing procedures, and should be used if possible when minimization of the expected number of trials is the most important performance measure.

Distributional Comparisons

As discussed previously, we have complete distributional information for any procedure given the problem parameters (k and θ^*) and procedure parameters (e.g., n, r, t, etc.). We can calculate the population variance of N in the SC, $\mathsf{Var}_{\mathrm{SC}}[N]$, and thus

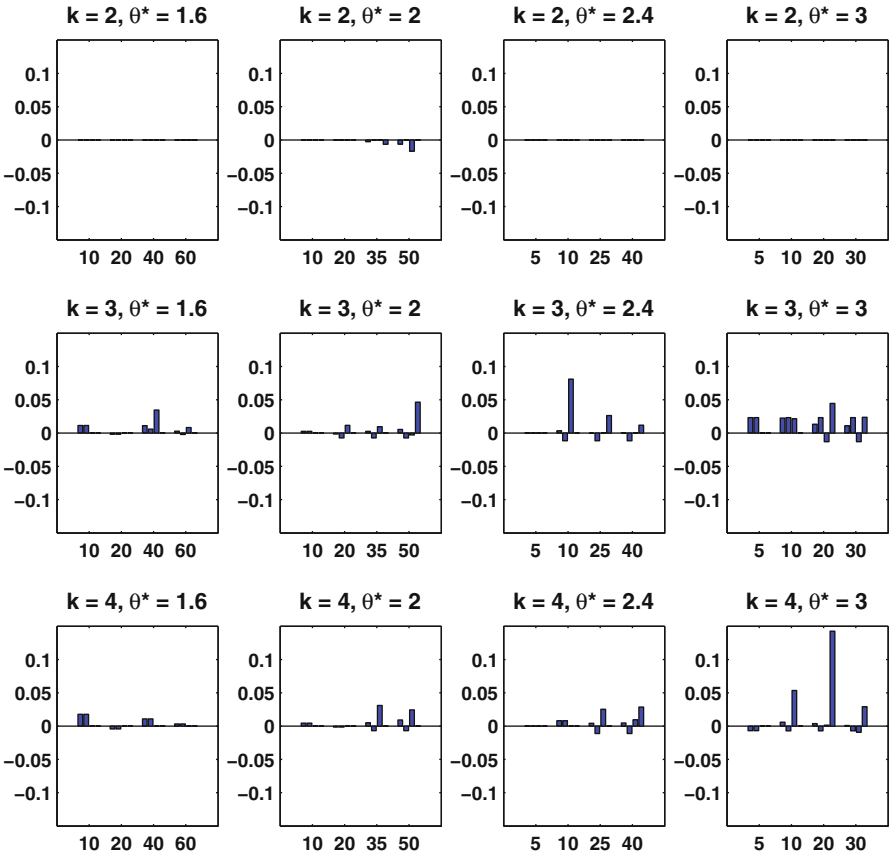


Fig. 6.5 Mean Relative Procedure Performance: $\mathcal{M}_{\mathrm{RA}'}$ Versus $\mathcal{M}_{\mathrm{BG}}$ (Bars Above Center Line Indicate Regions within which Procedure $\mathcal{M}_{\mathrm{BG}}$ Performs Better)

its standard deviation, $\mathsf{SD}_{\mathrm{SC}}[N]$, which we include in the tables in Appendix E. We were also interested in comparing the variance across the 36 cases in the appendix. Since $\mathsf{Var}_{\mathrm{SC}}[N]$ and $\mathsf{SD}_{\mathrm{SC}}[N]$ increase for all procedures except Procedure $\mathcal{M}_{\mathrm{BEM}}$ as $\mathsf{E}_{\mathrm{SC}}[N]$ increases (and are therefore dependent upon our choice of budget b), we chose to make our comparisons based on the coefficient of variation, $\mathsf{CV}_{\mathrm{SC}}[N]$, which measures variability relative to the mean, and is given by

$$\mathsf{CV}_{\mathrm{SC}}[N] \;=\; \frac{\mathsf{SD}_{\mathrm{SC}}[N]}{\mathsf{E}_{\mathrm{SC}}[N]}.$$

Table 6.2 shows the mean $\mathsf{CV}_{\mathrm{SC}}[N]$ across the possible combinations of $k \in \{2,3,4\}$, $\theta^* \in \{1.6, 2.0, 2.4, 3.0\}$, and $P^* \in \{0.75, 0.90, 0.95\}$, with $b = n_{\mathrm{BG}}$, as well as minimum and maximum values of $\mathsf{CV}_{\mathrm{SC}}[N]$ for each procedure. We did not consider the trivial case when $b = 1$ or when there is no entry for a procedure; therefore, the number of cases considered is less than 36. The relative ordering of the procedures, in

Table 6.2 Coefficient of Variation Results

Procedure	Cases	Mean $\mathsf{CV}_{\mathrm{SC}}[N]$	Min $\mathsf{CV}_{\mathrm{SC}}[N]$	Max $\mathsf{CV}_{\mathrm{SC}}[N]$
$\mathcal{M}_{\mathrm{R}}$	34	0.47	0.34	0.59
$\mathcal{M}_{\mathrm{NR}}$	29	0.45	0.20	0.60
$\mathcal{M}_{\mathrm{BG}}$	34	0.48	0.20	0.61
$\mathcal{M}_{\mathrm{RA}'}$	34	0.43	0.15	0.61
$\mathcal{M}_{\mathrm{RA}}$	25	0.41	0.20	0.61
$\mathcal{M}_{\mathrm{C}'}$	34	0.18	0.11	0.28
$\mathcal{M}_{\mathrm{BK}}$	35	0.13	0.05	0.20
$\mathcal{M}_{\mathrm{BEM}}$	35	0	0	0

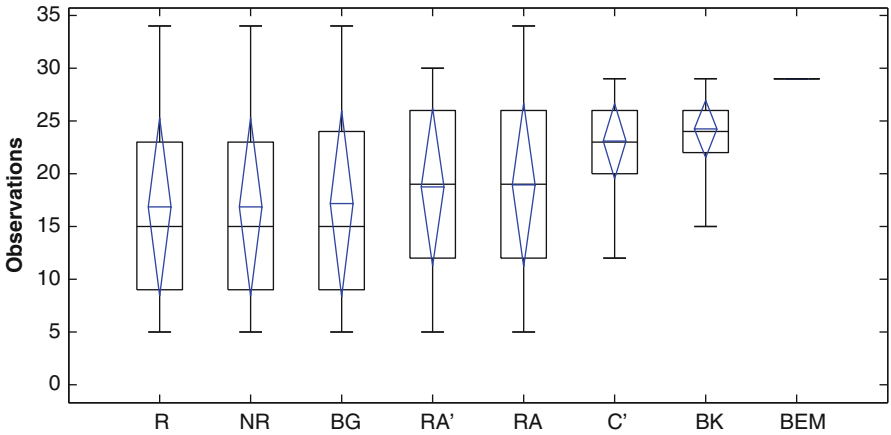


Fig. 6.6 Procedure Distribution Boxplots for $k = 3$, $\theta^* = 2$, $P^* = 0.9$, and $b = 34$

terms of their mean $\mathsf{CV}_{\mathrm{SC}}[N]$, generally holds for each of the cases. The following lists the procedures in increasing order of variability for the cases we examined, including Procedure $\mathcal{M}_{\mathrm{NR}}$. This order was not necessarily intact for all cases, but summarizes the observed trend.

1. Procedure $\mathcal{M}_{\mathrm{BEM}}$
2. Procedure $\mathcal{M}_{\mathrm{BK}}$
3. Procedure $\mathcal{M}_{\mathrm{C}'}$
4. Procedures $\mathcal{M}_{\mathrm{RA}}$ and $\mathcal{M}_{\mathrm{RA}'}$
5. Procedures $\mathcal{M}_{\mathrm{NR}}$ and $\mathcal{M}_{\mathrm{R}}$
6. Procedure $\mathcal{M}_{\mathrm{BG}}$

Note that the better performing procedures tend towards higher variability.

We may also be interested in more information about relative procedure performance. For example, a decision maker might care about the minimum, maximum, or median of the trial distribution, N, as well. One tool we can use is a modified boxplot (or box-and-whisker plot). Figure 6.6 displays boxplots of the distribution of N for each procedure when $k = 3$, $\theta^* = 2$, $P^* = 0.9$, and $b = 34$, corresponding to Table 6.12 in Appendix E, but including Procedure $\mathcal{M}_{\mathrm{NR}}$ as well. The bottom,

middle, and top of the boxes represent the 25th, 50th (median), and 75th percentiles of the procedure distributions, respectively. The ends of the whiskers represent the minimum and maximum of the distributions. We have also added information about the mean and standard deviation in the blue triangular regions. The horizontal line in the center of the triangular region represents the mean ($\mathsf{E}_{\mathrm{SC}}[N]$); the triangles extend one standard deviation from the mean.

The figure confirms our relative ordering for procedure variability. We also see that the distributions of Procedures $\mathcal{M}_{\mathrm{R}}$, $\mathcal{M}_{\mathrm{NR}}$, and $\mathcal{M}_{\mathrm{BG}}$ are noticeably skewed towards lower numbers of trials, since their medians are below the centers of the rectangles. Plots such as these can provide decision makers with the additional information necessary to compare other aspects of procedure performance in the SC or any other probability configuration. We could go a step further and plot the probability mass functions for each procedure; however, we feel that such detail is not necessary here.

Conclusions

We developed a procedure comparison methodology, which includes a method to compare different types of MSP procedures as well as a number of metrics that allow the user to examine different aspects of procedure performance. We used those metrics and selected charts to demonstrate some important relationships between the procedures in terms of performance, particularly when $k = 2$, as well as some interesting anomalies in the performance of Procedure $\mathcal{M}_{\mathrm{BG}}$. We also focused on a more thorough comparison of Procedures $\mathcal{M}_{\mathrm{BG}}$ and $\mathcal{M}_{\mathrm{RA}'}$, showing that Procedure $\mathcal{M}_{\mathrm{BG}}$ usually performs better in terms of $\mathsf{E}_{\mathrm{SC}}[N]$, but that Procedure $\mathcal{M}_{\mathrm{RA}'}$ can sometimes attain a higher maximum P^*. Then, we reported additional information provided by the distribution of N for each MSP procedure. In particular, we examined and compared MSP variabilities. The procedures followed a ranking with fairly good consistency, with Procedure $\mathcal{M}_{\mathrm{BEM}}$ having the least variability and Procedure $\mathcal{M}_{\mathrm{BG}}$ having the greatest variability. In general, procedures with better average performance had greater variability.

We provide updated tables for several procedures in the appendices. Our tables include the expected number of trials and the probability of correct selection for a larger number of problem parameter combinations than had been available previously.

Although our intent was not to show that the new class of procedures are better than classical MSP procedures, it should not be lost on the reader that, if the capability to derive the required functions for the new procedures is available, Procedures $\mathcal{M}_{\mathrm{R}}$ and $\mathcal{M}_{\mathrm{NR}}$ should be used.

A. Updated Tables for Procedure $\mathcal{M}_{\mathrm{BK}}$

Table 6.3 identifies the n_{BK}-values that minimize $\mathsf{E}_{\mathrm{SC}}[N]$ while still achieving P^*. We searched all n_{BK}-values up to $n_{\mathrm{BK}} = 400$. Table entries with ">400" in the column for n_{BK} indicate P^* requirements that cannot be achieved within our search space for the given k and θ^*.

Table 6.3 Updated Performance Characteristics for Procedure $\mathcal{M}_{\mathrm{BK}}$

P^*	θ^*	$k=2$				$k=3$				$k=4$			
		n_{BK}	$\mathsf{P}_{\mathrm{SC}}(\mathrm{CS})$	$\mathsf{E}_{\mathrm{SC}}[N]$	$\mathsf{E}_{\mathrm{EPC}}[N]$	n_{BK}	$\mathsf{P}_{\mathrm{SC}}(\mathrm{CS})$	$\mathsf{E}_{\mathrm{SC}}[N]$	$\mathsf{E}_{\mathrm{EPC}}[N]$	n_{BK}	$\mathsf{P}_{\mathrm{SC}}(\mathrm{CS})$	$\mathsf{E}_{\mathrm{SC}}[N]$	$\mathsf{E}_{\mathrm{EPC}}[N]$
0.95	3.0	9	0.9511	6.540	7.539	17	0.9554	12.958	15.079	23	0.9527	18.523	21.215
	2.8	11	0.9565	8.021	9.293	19	0.9535	14.687	16.946	26	0.9508	21.216	24.093
	2.6	13	0.9577	9.565	11.067	22	0.9527	17.273	19.784	31	0.9525	25.657	28.926
	2.4	15	0.9552	11.188	12.858	26	0.9511	20.781	23.596	37	0.9512	31.115	34.731
	2.2	19	0.9573	14.391	16.476	32	0.9502	26.091	29.321	46	0.9505	39.362	43.469
	2.0	23	0.9520	17.806	20.132	42	0.9509	35.023	38.921	61	0.9513	53.207	58.092
	1.8	33	0.9544	26.227	29.382	59	0.9508	50.489	55.325	86	0.9504	76.641	82.551
	1.6	49	0.9501	40.331	44.386	93	0.9502	82.011	88.365	138	0.9506	125.956	133.642
	1.4	97	0.9513	83.599	90.121	185	0.9503	168.952	178.424	277	0.9504	259.708	270.845
	1.2	327	0.9504	299.917	313.561	>400				>400			
0.90	3.0	7	0.9294	5.163	5.813	11	0.9014	8.460	9.482	16	0.9024	12.969	14.493
	2.8	7	0.9167	5.222	5.813	13	0.9073	10.103	11.312	19	0.9076	15.602	17.380
	2.6	7	0.9009	5.286	5.813	15	0.9054	11.833	13.222	22	0.9052	18.312	20.245
	2.4	9	0.9082	6.823	7.539	18	0.9056	14.436	16.035	26	0.9017	21.980	24.093
	2.2	11	0.9068	8.435	9.293	22	0.9034	17.985	19.784	33	0.9040	28.372	30.857
	2.0	15	0.9118	11.681	12.858	29	0.9044	24.242	26.455	43	0.9022	37.669	40.557
	1.8	19	0.9013	15.146	16.476	40	0.9018	34.299	36.984	61	0.9020	54.557	58.092
	1.6	31	0.9054	25.505	27.522	63	0.9007	55.643	59.203	97	0.9005	88.796	93.341
	1.4	59	0.9023	50.720	53.845	125	0.9004	114.286	119.609	195	0.9003	183.217	189.829
	1.2	199	0.9009	182.011	188.730	>400				>400			
0.75	3.0	1	0.7500	1.000	1.000	5	0.7690	3.950	4.111	8	0.7701	6.430	6.911
	2.8	3	0.8287	2.388	2.500	6	0.7803	4.493	4.926	9	0.7642	7.394	7.883
	2.6	3	0.8114	2.401	2.500	6	0.7536	4.549	4.926	10	0.7526	8.288	8.820
	2.4	3	0.7914	2.415	2.500	7	0.7502	5.559	5.786	12	0.7518	10.102	10.688
	2.2	3	0.7681	2.430	2.500	9	0.7545	7.286	7.675	15	0.7511	12.871	13.566
	2.0	5	0.7901	3.963	4.125	12	0.7577	9.902	10.431	20	0.7533	17.481	18.319
	1.8	5	0.7536	4.005	4.125	17	0.7580	14.412	15.079	29	0.7572	25.909	26.990
	1.6	9	0.7647	7.295	7.539	26	0.7517	22.732	23.596	46	0.7544	42.072	43.469
	1.4	17	0.7588	14.253	14.662	52	0.7529	47.206	48.550	91	0.7503	85.460	87.455
	1.2	55	0.7513	49.135	50.056	180	0.7510	170.794	173.516	323	0.7502	312.486	316.358

BK (1984) focus on proving various theorems and lemmas associated with curtailment, not on providing tables for the user. Their tables only include results for $n_{\mathrm{BK}} \leq 20$ and are tabulated by k, n_{BK}, and θ^*. We supplement those tables by providing results for common choices of P^*, including a greater range of θ^*-values, and searching over a much larger search space for n_{BK}. We also provide the expected number of trials in the EPC (i.e., $\mathsf{E}_{\mathrm{EPC}}[N]$).

B. Updated Tables for Procedure $\mathcal{M}_{\mathrm{C}'}$

Table 6.4 identifies the $(n_{\mathrm{C}'}, t)$-pairs that minimize $\mathsf{E}_{\mathrm{SC}}[N]$ while still achieving P^*. We do not include a table for $k = 2$, since, as discussed previously, Procedures $\mathcal{M}_{\mathrm{C}'}$ and $\mathcal{M}_{\mathrm{BK}}$ are identical in that case; and so we can consult the Procedure $\mathcal{M}_{\mathrm{BK}}$ table in Appendix A.

We searched all possible $(n_{\mathrm{C}'}, t)$-pairs up to $n_{\mathrm{C}'} = 125$. Rows with no entries in the table are θ^*-values for which P^* cannot be achieved within the search space. These tables improve upon those in Chen (1988), in which his values for $\mathsf{E}_{\mathrm{SC}}[N]$ did not incorporate curtailment. Also, his tables only provided performance characteristics for $n_{\mathrm{C}} \leq 30$. In addition, we provide the expected number of trials in the EPC (i.e., $\mathsf{E}_{\mathrm{EPC}}[N]$).

Table 6.4 Updated Performance Characteristics for Procedure $\mathcal{M}_{C'}$

P^*	θ^*	$k=3$					$k=4$				
		$n_{C'}$	t	$P_{SC}(CS)$	$E_{SC}[N]$	$E_{EPC}[N]$	$n_{C'}$	t	$P_{SC}(CS)$	$E_{SC}[N]$	$E_{EPC}[N]$
0.95	3.0	17	7	0.9509	11.470	14.356	25	8	0.9508	15.773	20.953
	2.8	19	8	0.9508	13.451	16.486	28	9	0.9500	18.384	24.040
	2.6	23	9	0.9513	15.707	19.523	31	11	0.9505	23.151	28.453
	2.4	27	11	0.9527	19.840	23.981	43	12	0.9501	26.711	34.594
	2.2	33	13	0.9505	24.478	29.390	48	15	0.9501	35.040	43.523
	2.0	43	17	0.9516	33.488	39.228	64	19	0.9503	47.068	57.894
	1.8	60	23	0.9508	47.966	55.370	91	26	0.9504	68.832	82.933
	1.6	95	35	0.9503	78.097	88.608					
	1.4										
	1.2										
0.90	3.0	12	5	0.9066	8.026	9.669	17	6	0.9026	11.553	14.350
	2.8	13	6	0.9062	9.727	11.196	19	7	0.9032	13.891	16.779
	2.6	15	7	0.9049	11.583	13.162	22	8	0.9015	16.498	19.668
	2.4	18	8	0.9045	13.916	15.876	27	9	0.9004	19.607	23.688
	2.2	23	9	0.9025	16.691	19.523	34	11	0.9018	25.269	30.211
	2.0	29	12	0.9028	23.088	26.073	44	14	0.9006	34.069	39.933
	1.8	40	16	0.9001	32.667	36.449	63	19	0.9011	49.624	57.514
	1.6	64	24	0.9001	52.906	58.699	100	29	0.9003	82.010	92.877
	1.4	125	48	0.9001	112.579	119.348					
	1.2										
0.75	3.0	5	3	0.7690	3.950	4.111	8	3	0.7508	5.154	5.826
	2.8	6	3	0.7803	4.493	4.926	9	4	0.7627	7.051	7.733
	2.6	6	3	0.7536	4.549	4.926	11	4	0.7585	7.611	8.620
	2.4	7	4	0.7502	5.559	5.786	13	5	0.7627	9.952	11.065
	2.2	10	4	0.7581	6.744	7.348	16	6	0.7601	12.600	13.886
	2.0	12	5	0.7505	8.927	9.669	21	7	0.7527	15.900	17.641
	1.8	17	7	0.7533	13.374	14.356	29	10	0.7539	24.179	26.166
	1.6	26	11	0.7507	22.193	23.325	49	14	0.7511	37.863	41.335
	1.4	52	20	0.7509	45.454	47.565	94	27	0.7510	81.002	86.274
	1.2										

C. Updated Tables for Procedure $\mathcal{M}_{\mathrm{RA}}$

Table 6.5 identifies the (r, t)-pairs that minimize $\mathsf{E}_{\mathrm{SC}}[N]$ while still achieving P^*. We searched all possible (r, t)-pairs up to $t = 150$ for $k = 2$ and 3, and up to $t = 75$ for $k = 4$. Rows with no entries in the table are θ^*-values for which P^* cannot be achieved within the search space.

These tables improve upon those in BG (1985a) by including a greater range of θ^*-values (theirs included $\theta^* = 2.0, 2.4, 3.0$ with some entries for $\theta^* = 1.6$), as well as a few corrections to their original paper. In the table, the symbol † represents an entry in our table that is different from that in BG. For that particular instance, BG allow $\mathsf{P}_{\mathrm{SC}}(\mathrm{CS})$ to be slightly below P^*; in our table, we do not. The symbol ‡ represents a value that is different from that in BG due to either our improved algorithm or our ability to calculate an exact result when BG estimated the result using MC sampling.

D. Updated Tables for Procedure $\mathcal{M}_{\mathrm{RA}'}$

Table 6.6 identifies the $(n_{\mathrm{RA}'}, r, t)$-triplets that minimize $\mathsf{E}_{\mathrm{SC}}[N]$ while still achieving P^*. We do not include a table for $k = 2$, since, as discussed previously, Procedures $\mathcal{M}_{\mathrm{RA}'}$ and $\mathcal{M}_{\mathrm{RA}}$ are identical in that case; and so we can consult the Procedure $\mathcal{M}_{\mathrm{RA}}$ table in Appendix C when $k = 2$.

We searched all possible $(n_{\mathrm{RA}'}, r, t)$-triplets up to $n_{\mathrm{RA}'} = 125$ for $k = 3$ and 4. Rows with no entries in the table are θ^*-values for which P^* cannot be achieved within the search space. These tables improve upon those in Chen (1992) by including a greater range of θ^*-values (his included $\theta^* = 2.0, 2.4, 3.0$), as well as corrections to some numerical errors found in his original paper. We use a † to identify entries that are corrections to values found in Table 1 of Chen (1992).

E. Procedure Comparison Tables

This appendix includes tables for all possible combinations of $k \in \{2, 3, 4\}$, $P^* \in \{0.75, 0.90, 0.95\}$, and $\theta^* \in \{1.6, 2.0, 2.4, 3.0\}$. We require that all procedures operate under a firm budget constraint, b, on the maximum number of trials, which sometimes results in a procedure not being able to achieve P^*.

All of the table entries have been verified via MC sampling. For each entry, we conducted 100,000 independent replications of the procedure. For any MC result outside of two standard errors of the tabulated data, we first determined if the tabulated data could be verified via a published source. If so, we did not pursue those any further. If not, we took 100,000 more MC samples. In all ten of those cases, the MC results were within two standard errors of our tabulated data. Thus, we have reasonable confidence that our results are accurate.

Table 6.5 Updated Performance Characteristics for Procedure $\mathcal{M}_{\mathrm{RA}}$

P^*	θ^*	$k=2$					$k=3$					$k=4$				
		r	t	P(CS)	$\mathsf{E}_{\mathrm{SC}}[N]$	$\mathsf{E}_{\mathrm{EPC}}[N]$	r	t	P(CS)	$\mathsf{E}_{\mathrm{SC}}[N]$	$\mathsf{E}_{\mathrm{EPC}}[N]$	r	t	P(CS)	$\mathsf{E}_{\mathrm{SC}}[N]$	$\mathsf{E}_{\mathrm{EPC}}[N]$
0.95	3.0	3	6	0.9522	5.251	6.943	4	7	0.9505	9.768	13.731	4	9	0.9577	13.747	22.193
	2.8	3	8	0.9515	5.650	7.843	4	9	0.9583	11.211	17.101	4	10	0.9527	15.006	24.349
	2.6	4	7	0.9536	7.541	9.658	4	10	0.9513	12.169	18.528	4	13	0.9533	17.009	29.714
	2.4	4	9	0.9548	8.465	11.380	4	14	0.9509	13.837	22.819	5	13	0.9531	22.297	34.652
	2.2	4	12	0.9505	9.426	13.127	5	14	0.9518	18.626	27.937	5	17	0.9518	26.322	43.276
	2.0	5	14	0.9537	13.091	17.898	5	22	0.9505	22.401	37.027	6	21	0.9534	36.255	57.562
	1.8	6	18	0.9508	18.034	24.301	6	28	0.9500	31.951	51.176	6	40	0.9502	46.412	86.080
	1.6	7	30	0.9502	26.559	37.094	7	80	0.9500	48.638	91.045	8	48	0.9503	78.184	128.640
	1.4	9	76	0.9501	48.312	72.356	10	99	0.9501	98.741	166.036					
	1.2															
0.90	3.0	3†	4†	0.9261†	4.560†	5.344†	3	5	0.9004	6.762	8.740	3	7	0.9091	9.869	14.709
	2.8	3	4	0.9132	4.633	5.344	3	6	0.9020	7.580	10.131	3	8	0.9008	10.750	16.168
	2.6	3	5	0.9168	5.234	6.258	3	8	0.9039	8.610	12.168	3	14	0.9002	12.220	20.826
	2.4	3	6	0.9113	5.718	6.943	4	8	0.9104	11.785	15.506	4	10	0.9107	16.980	24.349
	2.2	3	8	0.9037	6.326	7.843	4	10	0.9060	13.720	18.528	4	13	0.9055	19.976	29.714
	2.0	4	8	0.9033	8.899	10.587‡	4	15	0.9015	16.514	23.604	5	15	0.9060	27.928	39.218
	1.8	4	14	0.9021	11.040	13.907	5	18	0.9015	24.185	33.164	5	25	0.9017	36.201	55.197
	1.6	5	21	0.9006	17.001	21.482	6	30	0.9021	37.825	53.074	6‡	45‡	0.9005‡	57.123‡	90.224‡
	1.4	7	40	0.9000	32.924	41.835	8	64	0.9005	73.329	105.144	9	66	0.9001	121.152	173.668
	1.2	13	134	0.9003	112.282	143.502										
0.75	3.0	1	1	0.7500	1.000	1.000	2	3	0.7961	3.684	4.247	2	4	0.7718	5.148	6.386
	2.8	2	2	0.8287	2.388	2.500	2	3	0.7738	3.745	4.247	2	5	0.7639	5.668	7.187
	2.6	2	2	0.8114	2.401	2.500	2	4	0.7763	4.328	5.068	2	9	0.7506	6.226	8.243
	2.4	2	2	0.7914	2.415	2.500	2	5	0.7601	4.703	5.543	3	5	0.7731	9.221	10.900
	2.2	2	2	0.7681	2.430	2.500	3	4	0.7550	6.388	7.049	3	6	0.7629	10.895	12.963
	2.0	2	3	0.7737	3.086	3.250	4	5	0.7556	8.809	9.629	3	8	0.7525	13.371	16.168
	1.8	2	4	0.7524	3.437	3.625	3	9	0.7583	11.024	12.895	4	10	0.7578	20.687	24.349
	1.6	3	5	0.7559	5.956	6.258	4	12	0.7572	18.240	20.929	4	20	0.7514	30.051	37.290
	1.4	4	10	0.7555	11.354	12.056	5	26	0.7515	34.180	39.854	6	32	0.7535	64.368	76.981
	1.2	7	34	0.7514	36.748	39.283	9	90	0.7505	119.612	139.373					

Table 6.6 Updated Performance Characteristics for Procedure $\mathcal{M}_{RA'}$

P^*	θ^*	$k = 3$						$k = 4$					
		$n_{RA'}$	r	t	$P_{SC}(CS)$	$E_{SC}[N]$	$E_{EPC}[N]$	$n_{RA'}$	r	t	$P_{SC}(CS)$	$E_{SC}[N]$	$E_{EPC}[N]$
0.95	3.0	19	4	7	0.9505	9.768	13.731	25	4	9	0.9519	13.602	20.458
	2.8	20	4	9	0.9507	10.973	15.594	31	4	10	0.9510	14.976	23.712
	2.6	26	4	10	0.9503	12.157	18.351	39	4	13	0.9508	16.931	28.164
	2.4	36	4	14	0.9500	13.812	22.377	41†	5	13	0.9506	22.229	33.560
	2.2	37	5	14	0.9506	18.598	27.598	56	5	17	0.9505	26.279	42.318
	2.0	60	5	22	0.9501	22.389	36.742	68	6	21	0.9505†	36.094†	55.301†
	1.8	82	6	28	0.9500	31.951	51.176	101	7	28	0.9500	51.051	79.023
	1.6	111	8	39	0.9501	53.256	80.785						
	1.4												
	1.2												
0.90	3.0	13	3	5	0.9004	6.762	8.740	19	3	7	0.9016	9.739	13.762
	2.8	15	3	6	0.9004	7.574	10.089	26	3	8	0.9002	10.745	16.076
	2.6	19	3	8	0.9004	8.537	11.737	46	3	14	0.9000	12.217	20.626
	2.4	19	4	8	0.9038	11.637	14.857	29	4	10	0.9025	16.750	22.904
	2.2	24	4	11	0.9007	13.615	17.782	40	4	13	0.9012	19.832	28.528
	2.0	40	4	15	0.9006†	16.489†	23.396	47	5	15	0.9000	27.674	37.600
	1.8	49	5	18	0.9004	24.151	32.941	85	5	25	0.9003	36.125	54.247
	1.6	81	6	30	0.9003	37.691	52.270	111	7	32	0.9004	62.545	86.294
	1.4	125	14	49	0.9000	100.609	116.221						
	1.2												
0.75	3.0	5	2	3	0.7574	3.242	3.481	9	2	4	0.7513	4.895	5.740
	2.8	6	2	3	0.7623	3.699	4.148	12	2	5	0.7521	5.491	6.659
	2.6	7	2	4	0.7533	3.937	4.379	26	2	10	0.7500	6.217	8.154
	2.4	10	2	5	0.7507	4.534	5.187	13	3	5	0.7555	8.964	10.271
	2.2	10	3	4	0.7550	6.388	7.049	17	3	6	0.7529	10.733	12.534
	2.0	13†	3†	6†	0.7572†	8.395†	9.360†	26	3	8	0.7506	13.345	16.076
	1.8	21	3	9	0.7503	10.766	12.375	32	4	10	0.7528	20.551	23.974
	1.6	30	4	12	0.7505	17.927	20.327	70	4	20	0.7504	30.004	37.097
	1.4	72	5	26	0.7503	34.099	39.659	112	6	32	0.7503	64.011	75.963
	1.2												

Table 6.7 Comparative Results for $k = 2$ and $\theta^* = 1.6$

P^*	Proc	Parameters	$\mathsf{P}_{\mathrm{SC}}(\mathrm{CS})$	$\mathsf{E}_{\mathrm{SC}}[N]$	$\mathsf{SD}_{\mathrm{SC}}[N]$	$\mathsf{E}_{\mathrm{EPC}}[N]$	$100W_{\mathrm{J}}$
0.95	$\mathcal{M}_{\mathrm{R}}$		0.9500	26.355	14.113	36.396	
	$\mathcal{M}_{\mathrm{BG}}$	$n_{\mathrm{BG}} = 59$	0.9502	26.559	14.825	37.094	0.78
	$\mathcal{M}_{\mathrm{RA}'}$	$n_{\mathrm{RA}'} = 59, r = 7, t = 30$	0.9502	26.559	14.825	37.094	0.78
	$\mathcal{M}_{\mathrm{RA}}$	$r = 7, t = 30$	0.9502	26.559	14.825	37.094	0.78
	$\mathcal{M}_{\mathrm{C}'}$	$n_{\mathrm{C}'} = 49, t = 25$	0.9501	40.331	4.497	44.386	53.03
	$\mathcal{M}_{\mathrm{BK}}$	$n_{\mathrm{BK}} = 49$	0.9501	40.331	4.497	44.386	53.03
	$\mathcal{M}_{\mathrm{BEM}}$	$n_{\mathrm{BEM}} = 49$	0.9501	49.000	0.000	49.000	85.92
0.90	$\mathcal{M}_{\mathrm{R}}$		0.9000	16.865	10.019	21.185	
	$\mathcal{M}_{\mathrm{BG}}$	$n_{\mathrm{BG}} = 41$	0.9006	17.001	10.318	21.482	0.81
	$\mathcal{M}_{\mathrm{RA}'}$	$n_{\mathrm{RA}'} = 41, r = 5, t = 21$	0.9006	17.001	10.318	21.482	0.81
	$\mathcal{M}_{\mathrm{RA}}$	$r = 5, t = 21$	0.9006	17.001	10.318	21.482	0.81
	$\mathcal{M}_{\mathrm{C}'}$	$n_{\mathrm{C}'} = 31, t = 16$	0.9054	25.505	3.259	27.522	51.23
	$\mathcal{M}_{\mathrm{BK}}$	$n_{\mathrm{BK}} = 31$	0.9054	25.505	3.259	27.522	51.23
	$\mathcal{M}_{\mathrm{BEM}}$	$n_{\mathrm{BEM}} = 31$	0.9054	31.000	0.000	31.000	83.81
0.75	$\mathcal{M}_{\mathrm{R}}$		0.7500	5.558	2.056	5.814	
	$\mathcal{M}_{\mathrm{BG}}$	$n_{\mathrm{BG}} = 9$	0.7559	5.956	2.289	6.258	7.16
	$\mathcal{M}_{\mathrm{RA}'}$	$n_{\mathrm{RA}'} = 9, r = 3, t = 5$	0.7559	5.956	2.289	6.258	7.16
	$\mathcal{M}_{\mathrm{RA}}$	$r = 3, t = 5$	0.7559	5.956	2.289	6.258	7.16
	$\mathcal{M}_{\mathrm{C}'}$	$n_{\mathrm{C}'} = 9, t = 5$	0.7647	7.295	1.272	7.539	31.25
	$\mathcal{M}_{\mathrm{BK}}$	$n_{\mathrm{BK}} = 9$	0.7647	7.295	1.272	7.539	31.25
	$\mathcal{M}_{\mathrm{BEM}}$	$n_{\mathrm{BEM}} = 9$	0.7647	9.000	0.000	9.000	61.93

For all procedures except Procedure $\mathcal{M}_{\mathrm{R}}$, we report the parameters of the procedure that minimize $\mathsf{E}_{\mathrm{SC}}[N]$, while achieving the required P^* and remaining under the trial budget, b. In addition to $\mathsf{E}_{\mathrm{SC}}[N]$ and $\mathsf{P}_{\mathrm{SC}}(\mathrm{CS})$, we also report $\mathsf{E}_{\mathrm{EPC}}[N]$, $\mathsf{SD}_{\mathrm{SC}}(N)$, and W_{J} (as a percentage and labeled "100 W_{J}").

Blank rows for a particular procedure in a table indicate one of two situations. First, the procedure may not be able to achieve the given P^* under the budget constraint. These are marked by an "N/A" in the Parameters column. Second, the computational time or requirements for calculating $\mathsf{E}_{\mathrm{SC}}[N]$ and $\mathsf{P}_{\mathrm{SC}}(\mathrm{CS})$ for a particular procedure may be beyond our current capabilities. These are marked by "??" in the Parameters column in Tables 6.7–6.18.

Table 6.8 Comparative Results for $k = 2$ and $\theta^* = 2.0$

P^*	Proc	Parameters	P_{SC}(CS)	$\mathsf{E}_{\mathrm{SC}}[N]$	$\mathsf{SD}_{\mathrm{SC}}[N]$	$\mathsf{E}_{\mathrm{EPC}}[N]$	$100W_{\mathrm{J}}$
0.95	$\mathcal{M}_{\mathrm{R}}$		0.9500	12.411	5.992	16.625	
	$\mathcal{M}_{\mathrm{BG}}$	$n_{\mathrm{BG}} = 27$	0.9537	13.091	6.671	17.898	5.48
	$\mathcal{M}_{\mathrm{RA}'}$	$n_{\mathrm{RA}'} = 27, r = 5, t = 14$	0.9537	13.091	6.671	17.898	5.48
	$\mathcal{M}_{\mathrm{RA}}$	$r = 5, t = 14$	0.9537	13.091	6.671	17.898	5.48
	$\mathcal{M}_{\mathrm{C}'}$	$n_{\mathrm{C}'} = 23, t = 12$	0.9520	17.806	2.629	20.132	43.47
	$\mathcal{M}_{\mathrm{BK}}$	$n_{\mathrm{BK}} = 23$	0.9520	17.806	2.629	20.132	43.47
	$\mathcal{M}_{\mathrm{BEM}}$	$n_{\mathrm{BEM}} = 23$	0.9520	23.000	0.000	23.000	85.32
0.90	$\mathcal{M}_{\mathrm{R}}$		0.9000	8.511	3.519	10.048	
	$\mathcal{M}_{\mathrm{BG}}$	$n_{\mathrm{BG}} = 15$	0.9033	8.899	3.796	10.587	4.56
	$\mathcal{M}_{\mathrm{RA}'}$	$n_{\mathrm{RA}'} = 15, r = 4, t = 8$	0.9033	8.899	3.796	10.587	4.56
	$\mathcal{M}_{\mathrm{RA}}$	$r = 4, t = 8$	0.9033	8.899	3.796	10.587	4.56
	$\mathcal{M}_{\mathrm{C}'}$	$n_{\mathrm{C}'} = 15, t = 8$	0.9118	11.681	1.926	12.858	37.25
	$\mathcal{M}_{\mathrm{BK}}$	$n_{\mathrm{BK}} = 15$	0.9118	11.681	1.926	12.858	37.25
	$\mathcal{M}_{\mathrm{BEM}}$	$n_{\mathrm{BEM}} = 15$	0.9118	15.000	0.000	15.000	76.24
0.75	$\mathcal{M}_{\mathrm{R}}$		0.7500	2.625	1.409	2.752	
	$\mathcal{M}_{\mathrm{BG}}$	$n_{\mathrm{BG}} = 5$	0.7737	3.086	1.259	3.250	17.58
	$\mathcal{M}_{\mathrm{RA}'}$	$n_{\mathrm{RA}'} = 5, r = 2, t = 3$	0.7737	3.086	1.259	3.250	17.58
	$\mathcal{M}_{\mathrm{RA}}$	$r = 2, t = 3$	0.7737	3.086	1.259	3.250	17.58
	$\mathcal{M}_{\mathrm{C}'}$	$n_{\mathrm{C}'} = 5, t = 3$	0.7901	3.963	0.793	4.125	50.97
	$\mathcal{M}_{\mathrm{BK}}$	$n_{\mathrm{BK}} = 5$	0.7901	3.963	0.793	4.125	50.97
	$\mathcal{M}_{\mathrm{BEM}}$	$n_{\mathrm{BEM}} = 5$	0.7901	5.000	0.000	5.000	90.48

Table 6.9 Comparative Results for $k = 2$ and $\theta^* = 2.4$

P^*	Proc	Parameters	P_{SC}(CS)	$\mathsf{E}_{\mathrm{SC}}[N]$	$\mathsf{SD}_{\mathrm{SC}}[N]$	$\mathsf{E}_{\mathrm{EPC}}[N]$	$100W_{\mathrm{J}}$
0.95	$\mathcal{M}_{\mathrm{R}}$		0.9500	7.962	3.581	10.430	
	$\mathcal{M}_{\mathrm{BG}}$	$n_{\mathrm{BG}} = 17$	0.9548	8.465	4.085	11.380	6.32
	$\mathcal{M}_{\mathrm{RA}'}$	$n_{\mathrm{RA}'} = 17, r = 4, t = 9$	0.9548	8.465	4.085	11.380	6.32
	$\mathcal{M}_{\mathrm{RA}}$	$r = 4, t = 9$	0.9548	8.465	4.085	11.380	6.32
	$\mathcal{M}_{\mathrm{C}'}$	$n_{\mathrm{C}'} = 15, t = 8$	0.9552	11.188	1.884	12.858	40.51
	$\mathcal{M}_{\mathrm{BK}}$	$n_{\mathrm{BK}} = 15$	0.9552	11.188	1.884	12.858	40.51
	$\mathcal{M}_{\mathrm{BEM}}$	$n_{\mathrm{BEM}} = 15$	0.9552	15.000	0.000	15.000	88.39
0.90	$\mathcal{M}_{\mathrm{R}}$		0.9000	5.226	2.238	6.144	
	$\mathcal{M}_{\mathrm{BG}}$	$n_{\mathrm{BG}} = 11$	0.9113	5.718	2.736	6.943	9.40
	$\mathcal{M}_{\mathrm{RA}'}$	$n_{\mathrm{RA}'} = 11, r = 3, t = 6$	0.9113	5.718	2.736	6.943	9.40
	$\mathcal{M}_{\mathrm{RA}}$	$r = 3, t = 6$	0.9113	5.718	2.736	6.943	9.40
	$\mathcal{M}_{\mathrm{C}'}$	$n_{\mathrm{C}'} = 9, t = 5$	0.9082	6.823	1.283	7.539	30.55
	$\mathcal{M}_{\mathrm{BK}}$	$n_{\mathrm{BK}} = 9$	0.9082	6.823	1.283	7.539	30.55
	$\mathcal{M}_{\mathrm{BEM}}$	$n_{\mathrm{BEM}} = 9$	0.9082	9.000	0.000	9.000	72.21
0.75	$\mathcal{M}_{\mathrm{R}}$		0.7500	1.730	0.791	1.774	
	$\mathcal{M}_{\mathrm{BG}}$	$n_{\mathrm{BG}} = 3$	0.7914	2.415	0.493	2.500	39.58
	$\mathcal{M}_{\mathrm{RA}'}$	$n_{\mathrm{RA}'} = 3, r = 2, t = 2$	0.7914	2.415	0.493	2.500	39.58
	$\mathcal{M}_{\mathrm{RA}}$	$r = 2, t = 2$	0.7914	2.415	0.493	2.500	39.58
	$\mathcal{M}_{\mathrm{C}'}$	$n_{\mathrm{C}'} = 3, t = 2$	0.7914	2.415	0.493	2.500	39.58
	$\mathcal{M}_{\mathrm{BK}}$	$n_{\mathrm{BK}} = 3$	0.7914	2.415	0.493	2.500	39.58
	$\mathcal{M}_{\mathrm{BEM}}$	$n_{\mathrm{BEM}} = 3$	0.7914	3.000	0.000	3.000	73.37

Table 6.10 Comparative Results for $k = 2$ and $\theta^* = 3.0$

P^*	Proc	Parameters	$\mathsf{P}_{\mathrm{SC}}(\mathrm{CS})$	$\mathsf{E}_{\mathrm{SC}}[N]$	$\mathsf{SD}_{\mathrm{SC}}[N]$	$\mathsf{E}_{\mathrm{EPC}}[N]$	$100W_{\mathrm{J}}$
0.95	$\mathcal{M}_{\mathrm{R}}$		0.9500	5.140	2.363	6.682	
	$\mathcal{M}_{\mathrm{BG}}$	$n_{\mathrm{BG}} = 11$	0.9522	5.251	2.530	6.943	2.17
	$\mathcal{M}_{\mathrm{RA}'}$	$n_{\mathrm{RA}'} = 11, r = 3, t = 6$	0.9522	5.251	2.530	6.943	2.17
	$\mathcal{M}_{\mathrm{RA}}$	$r = 3, t = 6$	0.9522	5.251	2.530	6.943	2.17
	$\mathcal{M}_{\mathrm{C}'}$	$n_{\mathrm{C}'} = 9, t = 5$	0.9511	6.540	1.238	7.539	27.24
	$\mathcal{M}_{\mathrm{BK}}$	$n_{\mathrm{BK}} = 9$	0.9511	6.540	1.238	7.539	27.24
	$\mathcal{M}_{\mathrm{BEM}}$	$n_{\mathrm{BEM}} = 9$	0.9511	9.000	0.000	9.000	75.10
0.90	$\mathcal{M}_{\mathrm{R}}$		0.9000	3.405	1.559	3.973	
	$\mathcal{M}_{\mathrm{BG}}$	N/A					
	$\mathcal{M}_{\mathrm{RA}'}$	$n_{\mathrm{RA}'} = 7, r = 3, t = 4$	0.9261	4.560	1.493	5.344	33.94
	$\mathcal{M}_{\mathrm{RA}}$	$r = 3, t = 4$	0.9261	4.560	1.493	5.344	33.94
	$\mathcal{M}_{\mathrm{C}'}$	$n_{\mathrm{C}'} = 7, t = 4$	0.9294	5.163	1.020	5.813	51.65
	$\mathcal{M}_{\mathrm{BK}}$	$n_{\mathrm{BK}} = 7$	0.9294	5.163	1.020	5.813	51.65
	$\mathcal{M}_{\mathrm{BEM}}$	$n_{\mathrm{BEM}} = 7$	0.9294	7.000	0.000	7.000	105.61
0.75	$\mathcal{M}_{\mathrm{R}}$		0.7500	1.000	0.000	1.000	
	$\mathcal{M}_{\mathrm{BG}}$	$n_{\mathrm{BG}} = 1$	0.7500	1.000	0.000	1.000	0.00
	$\mathcal{M}_{\mathrm{RA}'}$	$n_{\mathrm{RA}'} = 1, r = 1, t = 1$	0.7500	1.000	0.000	1.000	0.00
	$\mathcal{M}_{\mathrm{RA}}$	$r = 1, t = 1$	0.7500	1.000	0.000	1.000	0.00
	$\mathcal{M}_{\mathrm{C}'}$	$n_{\mathrm{C}'} = 1, t = 1$	0.7500	1.000	0.000	1.000	0.00
	$\mathcal{M}_{\mathrm{BK}}$	$n_{\mathrm{BK}} = 1$	0.7500	1.000	0.000	1.000	0.00
	$\mathcal{M}_{\mathrm{BEM}}$	$n_{\mathrm{BEM}} = 1$	0.7500	1.000	0.000	1.000	0.00

Table 6.11 Comparative Results for $k = 3$ and $\theta^* = 1.6$

P^*	Proc	Parameters	$\mathsf{P}_{\mathrm{SC}}(\mathrm{CS})$	$\mathsf{E}_{\mathrm{SC}}[N]$	$\mathsf{SD}_{\mathrm{SC}}[N]$	$\mathsf{E}_{\mathrm{EPC}}[N]$	$100W_{\mathrm{J}}$
0.95	$\mathcal{M}_{\mathrm{R}}$		0.9500	50.049	27.313	79.216	
	$\mathcal{M}_{\mathrm{BG}}$	$n_{\mathrm{BG}} = 125$	0.9502	50.321	28.698	81.434	0.54
	$\mathcal{M}_{\mathrm{RA}'}$	$n_{\mathrm{RA}'} = 111, r = 8, t = 39$	0.9501	53.256	27.071	80.785	6.41
	$\mathcal{M}_{\mathrm{RA}}$	$r = 8, t = 39$	0.9504	53.270	27.102	81.005	6.44
	$\mathcal{M}_{\mathrm{C}'}$	$n_{\mathrm{C}'} = 95, t = 35$	0.9503	78.097	8.770	88.608	56.04
	$\mathcal{M}_{\mathrm{BK}}$	$n_{\mathrm{BK}} = 93$	0.9502	82.011	5.697	88.365	63.86
	$\mathcal{M}_{\mathrm{BEM}}$	$n_{\mathrm{BEM}} = 93$	0.9502	93.000	0.000	93.000	85.82
0.90	$\mathcal{M}_{\mathrm{R}}$		0.9000	36.949	19.345	50.761	
	$\mathcal{M}_{\mathrm{BG}}$	$n_{\mathrm{BG}} = 83$	0.9003	37.261	20.583	52.614	0.84
	$\mathcal{M}_{\mathrm{RA}'}$	$n_{\mathrm{RA}'} = 81, r = 6, t = 30$	0.9003	37.691	20.981	52.270	2.01
	$\mathcal{M}_{\mathrm{RA}}$	$r = 7, t = 25$	0.9010	41.243	17.791	53.841	11.62
	$\mathcal{M}_{\mathrm{C}'}$	$n_{\mathrm{C}'} = 64, t = 24$	0.9001	52.906	6.576	58.699	43.19
	$\mathcal{M}_{\mathrm{BK}}$	$n_{\mathrm{BK}} = 63$	0.9007	55.643	4.358	59.203	50.60
	$\mathcal{M}_{\mathrm{BEM}}$	$n_{\mathrm{BEM}} = 63$	0.9007	63.000	0.000	63.000	70.51
0.75	$\mathcal{M}_{\mathrm{R}}$		0.7500	17.242	8.101	19.548	
	$\mathcal{M}_{\mathrm{BG}}$	$n_{\mathrm{BG}} = 32$	0.7517	17.597	8.823	20.254	2.06
	$\mathcal{M}_{\mathrm{RA}'}$	$n_{\mathrm{RA}'} = 30, r = 4, t = 12$	0.7505	17.927	8.409	20.327	3.97
	$\mathcal{M}_{\mathrm{RA}}$	$r = 5, t = 11$	0.7628	20.515	6.778	22.903	18.98
	$\mathcal{M}_{\mathrm{C}'}$	$n_{\mathrm{C}'} = 26, t = 11$	0.7507	22.193	2.844	23.325	28.72
	$\mathcal{M}_{\mathrm{BK}}$	$n_{\mathrm{BK}} = 26$	0.7517	22.732	2.275	23.596	31.84
	$\mathcal{M}_{\mathrm{BEM}}$	$n_{\mathrm{BEM}} = 26$	0.7517	26.000	0.000	26.000	50.80

Table 6.12 Comparative Results for $k = 3$ and $\theta^* = 2.0$

P^*	Proc	Parameters	P_{SC}(CS)	$\mathsf{E}_{\mathrm{SC}}[N]$	$\mathsf{SD}_{\mathrm{SC}}[N]$	$\mathsf{E}_{\mathrm{EPC}}[N]$	$100W_{\mathrm{J}}$
0.95	$\mathcal{M}_{\mathrm{R}}$		0.9500	22.750	11.975	35.038	
	$\mathcal{M}_{\mathrm{BG}}$	$n_{\mathrm{BG}} = 52$	0.9508	23.032	12.475	35.972	1.24
	$\mathcal{M}_{\mathrm{RA}'}$	$n_{\mathrm{RA}'} = 48, r = 6, t = 17$	0.9502	25.113	11.011	36.327	10.39
	$\mathcal{M}_{\mathrm{RA}}$	$r = 6, t = 17$	0.9504	25.114	11.013	36.343	10.39
	$\mathcal{M}_{\mathrm{C}'}$	$n_{\mathrm{C}'} = 43, t = 17$	0.9516	33.488	4.933	39.228	47.20
	$\mathcal{M}_{\mathrm{BK}}$	$n_{\mathrm{BK}} = 42$	0.9509	35.023	3.511	38.921	53.95
	$\mathcal{M}_{\mathrm{BEM}}$	$n_{\mathrm{BEM}} = 42$	0.9509	42.000	0.000	42.000	84.62
0.90	$\mathcal{M}_{\mathrm{R}}$		0.9000	16.857	8.429	22.676	
	$\mathcal{M}_{\mathrm{BG}}$	$n_{\mathrm{BG}} = 34$	0.9016	17.165	8.813	23.296	1.83
	$\mathcal{M}_{\mathrm{RA}'}$	$n_{\mathrm{RA}'} = 30, r = 5, t = 12$	0.9001	18.749	7.470	23.902	11.22
	$\mathcal{M}_{\mathrm{RA}}$	$r = 5, t = 12$	0.9057	18.940	7.746	24.698	12.35
	$\mathcal{M}_{\mathrm{C}'}$	$n_{\mathrm{C}'} = 29, t = 12$	0.9028	23.088	3.563	26.073	36.96
	$\mathcal{M}_{\mathrm{BK}}$	$n_{\mathrm{BK}} = 29$	0.9044	24.242	2.716	26.455	43.80
	$\mathcal{M}_{\mathrm{BEM}}$	$n_{\mathrm{BEM}} = 29$	0.9044	29.000	0.000	29.000	72.03
0.75	$\mathcal{M}_{\mathrm{R}}$		0.7500	7.831	3.200	8.765	
	$\mathcal{M}_{\mathrm{BG}}$	$n_{\mathrm{BG}} = 13$	0.7512	7.966	3.315	8.934	1.72
	$\mathcal{M}_{\mathrm{RA}'}$	$n_{\mathrm{RA}'} = 13, r = 3, t = 6$	0.7572	8.395	3.399	9.360	7.21
	$\mathcal{M}_{\mathrm{RA}}$	$r = 4, t = 5$	0.7556	8.809	2.208	9.629	12.49
	$\mathcal{M}_{\mathrm{C}'}$	$n_{\mathrm{C}'} = 12, t = 5$	0.7505	8.927	1.912	9.669	13.99
	$\mathcal{M}_{\mathrm{BK}}$	$n_{\mathrm{BK}} = 12$	0.7577	9.902	1.453	10.431	26.45
	$\mathcal{M}_{\mathrm{BEM}}$	$n_{\mathrm{BEM}} = 12$	0.7577	12.000	0.000	12.000	53.24

Table 6.13 Comparative Results for $k = 3$ and $\theta^* = 2.4$

P^*	Proc	Parameters	P_{SC}(CS)	$\mathsf{E}_{\mathrm{SC}}[N]$	$\mathsf{SD}_{\mathrm{SC}}[N]$	$\mathsf{E}_{\mathrm{EPC}}[N]$	$100W_{\mathrm{J}}$
0.95	$\mathcal{M}_{\mathrm{R}}$		0.9500	14.177	7.046	21.303	
	$\mathcal{M}_{\mathrm{BG}}$	$n_{\mathrm{BG}} = 31$	0.9516	14.479	7.397	22.086	2.13
	$\mathcal{M}_{\mathrm{RA}'}$	$n_{\mathrm{RA}'} = 28, r = 5, t = 11$	0.9509	15.921	6.503	22.426	12.30
	$\mathcal{M}_{\mathrm{RA}}$	$r = 5, t = 11$	0.9530	15.969	6.590	22.903	12.64
	$\mathcal{M}_{\mathrm{C}'}$	$n_{\mathrm{C}'} = 27, t = 11$	0.9527	19.840	3.496	23.981	39.94
	$\mathcal{M}_{\mathrm{BK}}$	$n_{\mathrm{BK}} = 26$	0.9511	20.781	2.565	23.596	46.58
	$\mathcal{M}_{\mathrm{BEM}}$	$n_{\mathrm{BEM}} = 26$	0.9511	26.000	0.000	26.000	83.40
0.90	$\mathcal{M}_{\mathrm{R}}$		0.9000	10.235	5.160	13.784	
	$\mathcal{M}_{\mathrm{BG}}$	$n_{\mathrm{BG}} = 22$	0.9021	10.429	5.397	14.247	1.90
	$\mathcal{M}_{\mathrm{RA}'}$	$n_{\mathrm{RA}'} = 19, r = 4, t = 8$	0.9038	11.637	4.638	14.857	13.71
	$\mathcal{M}_{\mathrm{RA}}$	$r = 4, t = 8$	0.9104	11.785	4.862	15.506	15.15
	$\mathcal{M}_{\mathrm{C}'}$	$n_{\mathrm{C}'} = 18, t = 8$	0.9045	13.916	2.405	15.876	35.97
	$\mathcal{M}_{\mathrm{BK}}$	$n_{\mathrm{BK}} = 18$	0.9056	14.436	1.994	16.035	41.05
	$\mathcal{M}_{\mathrm{BEM}}$	$n_{\mathrm{BEM}} = 18$	0.9056	18.000	0.000	18.000	75.87
0.75	$\mathcal{M}_{\mathrm{R}}$		0.7500	4.910	1.864	5.390	
	$\mathcal{M}_{\mathrm{BG}}$	$n_{\mathrm{BG}} = 8$	0.7602	5.403	1.770	5.938	10.02
	$\mathcal{M}_{\mathrm{RA}'}$	$n_{\mathrm{RA}'} = 7, r = 4, t = 4$	0.7502	5.559	0.857	5.786	13.22
	$\mathcal{M}_{\mathrm{RA}}$	N/A					
	$\mathcal{M}_{\mathrm{C}'}$	$n_{\mathrm{C}'} = 7, t = 4$	0.7502	5.559	0.857	5.786	13.22
	$\mathcal{M}_{\mathrm{BK}}$	$n_{\mathrm{BK}} = 7$	0.7502	5.559	0.857	5.786	13.22
	$\mathcal{M}_{\mathrm{BEM}}$	$n_{\mathrm{BEM}} = 7$	0.7502	7.000	0.000	7.000	42.56

Table 6.14 Comparative Results for $k = 3$ and $\theta^* = 3.0$

P^*	Proc	Parameters	$\mathsf{P}_{\mathrm{SC}}(\mathrm{CS})$	$\mathsf{E}_{\mathrm{SC}}[N]$	$\mathsf{SD}_{\mathrm{SC}}[N]$	$\mathsf{E}_{\mathrm{EPC}}[N]$	$100W_{\mathrm{J}}$
0.95	$\mathcal{M}_{\mathrm{R}}$		0.9500	8.844	4.399	13.383	
	$\mathcal{M}_{\mathrm{BG}}$	$n_{\mathrm{BG}} = 20$	0.9505	8.901	4.474	13.573	0.64
	$\mathcal{M}_{\mathrm{RA}'}$	$n_{\mathrm{RA}'} = 19, r = 4, t = 7$	0.9505	9.768	3.830	13.731	10.44
	$\mathcal{M}_{\mathrm{RA}}$	$r = 4, t = 7$	0.9505	9.768	3.830	13.731	10.44
	$\mathcal{M}_{\mathrm{C}'}$	$n_{\mathrm{C}'} = 17, t = 7$	0.9509	11.470	2.414	14.356	29.69
	$\mathcal{M}_{\mathrm{BK}}$	$n_{\mathrm{BK}} = 17$	0.9554	12.958	1.895	15.079	46.51
	$\mathcal{M}_{\mathrm{BEM}}$	$n_{\mathrm{BEM}} = 17$	0.9554	17.000	0.000	17.000	92.21
0.90	$\mathcal{M}_{\mathrm{R}}$		0.9000	6.762	2.864	8.682	
	$\mathcal{M}_{\mathrm{BG}}$	$n_{\mathrm{BG}} = 12$	0.9029	6.969	3.023	8.933	3.06
	$\mathcal{M}_{\mathrm{RA}'}$	$n_{\mathrm{RA}'} = 12, r = 3, t = 6$	0.9029	6.969	3.023	8.933	3.06
	$\mathcal{M}_{\mathrm{RA}}$	N/A					
	$\mathcal{M}_{\mathrm{C}'}$	$n_{\mathrm{C}'} = 12, t = 5$	0.9066	8.026	1.864	9.669	18.70
	$\mathcal{M}_{\mathrm{BK}}$	$n_{\mathrm{BK}} = 11$	0.9014	8.460	1.353	9.482	25.11
	$\mathcal{M}_{\mathrm{BEM}}$	$n_{\mathrm{BEM}} = 11$	0.9014	11.000	0.000	11.000	62.67
0.75	$\mathcal{M}_{\mathrm{R}}$		0.7500	3.068	1.053	3.290	
	$\mathcal{M}_{\mathrm{BG}}$	$n_{\mathrm{BG}} = 5$	0.7574	3.242	1.143	3.481	5.66
	$\mathcal{M}_{\mathrm{RA}'}$	$n_{\mathrm{RA}'} = 5, r = 2, t = 3$	0.7574	3.242	1.143	3.481	5.66
	$\mathcal{M}_{\mathrm{RA}}$	N/A					
	$\mathcal{M}_{\mathrm{C}'}$	$n_{\mathrm{C}'} = 5, t = 3$	0.7690	3.950	0.642	4.111	28.76
	$\mathcal{M}_{\mathrm{BK}}$	$n_{\mathrm{BK}} = 5$	0.7690	3.950	0.642	4.111	28.76
	$\mathcal{M}_{\mathrm{BEM}}$	$n_{\mathrm{BEM}} = 5$	0.7690	5.000	0.000	5.000	62.97

Table 6.15 Comparative Results for $k = 4$ and $\theta^* = 1.6$

P^*	Proc	Parameters	$\mathsf{P}_{\mathrm{SC}}(\mathrm{CS})$	$\mathsf{E}_{\mathrm{SC}}[N]$	$\mathsf{SD}_{\mathrm{SC}}[N]$	$\mathsf{E}_{\mathrm{EPC}}[N]$	$100W_{\mathrm{J}}$
0.95	$\mathcal{M}_{\mathrm{R}}$	??					
	$\mathcal{M}_{\mathrm{BG}}$	$n_{\mathrm{BG}} = 181$	0.9500	76.265	41.254	128.931	
	$\mathcal{M}_{\mathrm{RA}'}$	??					
	$\mathcal{M}_{\mathrm{RA}}$	??					
	$\mathcal{M}_{\mathrm{C}'}$	??					
	$\mathcal{M}_{\mathrm{BK}}$	$n_{\mathrm{BK}} = 138$	0.9506	125.956	6.368	133.642	
	$\mathcal{M}_{\mathrm{BEM}}$	$n_{\mathrm{BEM}} = 138$	0.9506	138.000	0.000	138.000	
0.90	$\mathcal{M}_{\mathrm{R}}$		0.9000	58.189	29.900	83.609	
	$\mathcal{M}_{\mathrm{BG}}$	$n_{\mathrm{BG}} = 126$	0.9004	58.715	31.160	86.889	0.90
	$\mathcal{M}_{\mathrm{RA}'}$	$n_{\mathrm{RA}'} = 111, r = 7, t = 32$	0.9004	62.545	29.113	86.294	7.49
	$\mathcal{M}_{\mathrm{RA}}$	$r = 7, t = 32$	0.9029	62.755	29.470	87.896	7.85
	$\mathcal{M}_{\mathrm{C}'}$	$n_{\mathrm{C}'} = 100, t = 29$	0.9003	82.010	10.439	92.877	40.94
	$\mathcal{M}_{\mathrm{BK}}$	$n_{\mathrm{BK}} = 97$	0.9005	88.796	4.993	93.341	52.60
	$\mathcal{M}_{\mathrm{BEM}}$	$n_{\mathrm{BEM}} = 97$	0.9005	97.000	0.000	97.000	66.70
0.75	$\mathcal{M}_{\mathrm{R}}$		0.7500	30.549	14.559	36.195	
	$\mathcal{M}_{\mathrm{BG}}$	$n_{\mathrm{BG}} = 57$	0.7512	31.109	15.462	37.649	1.83
	$\mathcal{M}_{\mathrm{RA}'}$	$n_{\mathrm{RA}'} = 50, r = 5, t = 15$	0.7504	33.385	12.826	38.648	9.29
	$\mathcal{M}_{\mathrm{RA}}$	$r = 5, t = 15$	0.7551	33.601	13.115	39.218	9.99
	$\mathcal{M}_{\mathrm{C}'}$	$n_{\mathrm{C}'} = 49, t = 14$	0.7511	37.863	6.092	41.335	23.94
	$\mathcal{M}_{\mathrm{BK}}$	$n_{\mathrm{BK}} = 46$	0.7544	42.072	2.862	43.469	37.72
	$\mathcal{M}_{\mathrm{BEM}}$	$n_{\mathrm{BEM}} = 46$	0.7544	46.000	0.000	46.000	50.58

Table 6.16 Comparative Results for $k = 4$ and $\theta^* = 2.0$

P^*	Proc	Parameters	P_{SC}(CS)	$\mathsf{E}_{\mathrm{SC}}[N]$	$\mathsf{SD}_{\mathrm{SC}}[N]$	$\mathsf{E}_{\mathrm{EPC}}[N]$	$100W_{\mathrm{J}}$
0.95	$\mathcal{M}_{\mathrm{R}}$		0.9500	33.673	16.774	53.912	
	$\mathcal{M}_{\mathrm{BG}}$	$n_{\mathrm{BG}} = 74$	0.9500	33.824	17.611	55.326	0.45
	$\mathcal{M}_{\mathrm{RA}'}$	$n_{\mathrm{RA}'} = 68, r = 6, t = 21$	0.9505	36.094	16.872	55.301	7.19
	$\mathcal{M}_{\mathrm{RA}}$	$r = 8, t = 19$	0.9515	41.970	13.131	57.765	24.64
	$\mathcal{M}_{\mathrm{C}'}$	$n_{\mathrm{C}'} = 64, t = 19$	0.9503	47.068	7.664	57.894	39.78
	$\mathcal{M}_{\mathrm{BK}}$	$n_{\mathrm{BK}} = 61$	0.9513	53.207	4.034	58.092	58.01
	$\mathcal{M}_{\mathrm{BEM}}$	$n_{\mathrm{BEM}} = 61$	0.9513	61.000	0.000	61.000	81.16
0.90	$\mathcal{M}_{\mathrm{R}}$		0.9000	25.566	12.699	36.378	
	$\mathcal{M}_{\mathrm{BG}}$	$n_{\mathrm{BG}} = 53$	0.9000	25.706	13.518	37.305	0.55
	$\mathcal{M}_{\mathrm{RA}'}$	$n_{\mathrm{RA}'} = 47, r = 5, t = 15$	0.9000	27.674	12.181	37.600	8.24
	$\mathcal{M}_{\mathrm{RA}}$	$r = 6, t = 14$	0.9049	30.293	10.316	39.759	18.49
	$\mathcal{M}_{\mathrm{C}'}$	$n_{\mathrm{C}'} = 44, t = 14$	0.9006	34.069	5.802	39.933	33.26
	$\mathcal{M}_{\mathrm{BK}}$	$n_{\mathrm{BK}} = 43$	0.9022	37.669	3.172	40.557	47.34
	$\mathcal{M}_{\mathrm{BEM}}$	$n_{\mathrm{BEM}} = 43$	0.9022	43.000	0.000	43.000	68.19
0.75	$\mathcal{M}_{\mathrm{R}}$		0.7500	13.342	6.098	15.712	
	$\mathcal{M}_{\mathrm{BG}}$	$n_{\mathrm{BG}} = 24$	0.7541	13.781	6.448	16.449	3.29
	$\mathcal{M}_{\mathrm{RA}'}$	$n_{\mathrm{RA}'} = 22, r = 4, t = 7$	0.7534	14.973	4.636	17.137	12.22
	$\mathcal{M}_{\mathrm{RA}}$	N/A					
	$\mathcal{M}_{\mathrm{C}'}$	$n_{\mathrm{C}'} = 21, t = 7$	0.7527	15.900	3.254	17.641	19.17
	$\mathcal{M}_{\mathrm{BK}}$	$n_{\mathrm{BK}} = 20$	0.7533	17.481	1.799	18.319	31.02
	$\mathcal{M}_{\mathrm{BEM}}$	$n_{\mathrm{BEM}} = 20$	0.7533	20.000	0.000	20.000	49.90

Table 6.17 Comparative Results for $k = 4$ and $\theta^* = 2.4$

P^*	Proc	Parameters	P_{SC}(CS)	$\mathsf{E}_{\mathrm{SC}}[N]$	$\mathsf{SD}_{\mathrm{SC}}[N]$	$\mathsf{E}_{\mathrm{EPC}}[N]$	$100W_{\mathrm{J}}$
0.95	$\mathcal{M}_{\mathrm{R}}$		0.9500	20.471	10.064	32.570	
	$\mathcal{M}_{\mathrm{BG}}$	$n_{\mathrm{BG}} = 44$	0.9506	20.679	10.484	33.382	1.01
	$\mathcal{M}_{\mathrm{RA}'}$	$n_{\mathrm{RA}'} = 41, r = 5, t = 13$	0.9506	22.229	9.744	33.560	8.58
	$\mathcal{M}_{\mathrm{RA}}$	N/A					
	$\mathcal{M}_{\mathrm{C}'}$	$n_{\mathrm{C}'} = 43, t = 12$	0.9501	26.711	5.313	34.594	30.48
	$\mathcal{M}_{\mathrm{BK}}$	$n_{\mathrm{BK}} = 37$	0.9512	31.115	2.998	34.731	51.99
	$\mathcal{M}_{\mathrm{BEM}}$	$n_{\mathrm{BEM}} = 37$	0.9512	37.000	0.000	37.000	80.74
0.90	$\mathcal{M}_{\mathrm{R}}$		0.9000	15.604	7.378	21.869	
	$\mathcal{M}_{\mathrm{BG}}$	$n_{\mathrm{BG}} = 31$	0.9022	15.927	7.794	22.767	2.07
	$\mathcal{M}_{\mathrm{RA}'}$	$n_{\mathrm{RA}'} = 29, r = 4, t = 10$	0.9025	16.750	7.491	22.904	7.35
	$\mathcal{M}_{\mathrm{RA}}$	N/A					
	$\mathcal{M}_{\mathrm{C}'}$	$n_{\mathrm{C}'} = 27, t = 9$	0.9004	19.607	4.019	23.688	25.65
	$\mathcal{M}_{\mathrm{BK}}$	$n_{\mathrm{BK}} = 26$	0.9017	21.980	2.356	24.093	40.86
	$\mathcal{M}_{\mathrm{BEM}}$	$n_{\mathrm{BEM}} = 26$	0.9017	26.000	0.000	26.000	66.62
0.75	$\mathcal{M}_{\mathrm{R}}$		0.7500	7.922	3.731	9.377	
	$\mathcal{M}_{\mathrm{BG}}$	$n_{\mathrm{BG}} = 15$	0.7569	8.286	4.020	9.911	4.59
	$\mathcal{M}_{\mathrm{RA}'}$	$n_{\mathrm{RA}'} = 13, r = 3, t = 5$	0.7555	8.964	3.152	10.271	13.15
	$\mathcal{M}_{\mathrm{RA}}$	N/A					
	$\mathcal{M}_{\mathrm{C}'}$	$n_{\mathrm{C}'} = 13, t = 5$	0.7627	9.952	2.141	11.065	25.61
	$\mathcal{M}_{\mathrm{BK}}$	$n_{\mathrm{BK}} = 12$	0.7518	10.102	1.339	10.688	27.52
	$\mathcal{M}_{\mathrm{BEM}}$	$n_{\mathrm{BEM}} = 12$	0.7518	12.000	0.000	12.000	51.47

Table 6.18 Comparative Results for $k = 4$ and $\theta^* = 3.0$

P^*	Proc	Parameters	$\mathsf{P}_{\mathrm{SC}}(\mathrm{CS})$	$\mathsf{E}_{\mathrm{SC}}[N]$	$\mathsf{SD}_{\mathrm{SC}}[N]$	$\mathsf{E}_{\mathrm{EPC}}[N]$	$100W_{\mathrm{J}}$
0.95	$\mathcal{M}_{\mathrm{R}}$		0.9500	12.741	5.858	19.727	
	$\mathcal{M}_{\mathrm{BG}}$	$n_{\mathrm{BG}} = 26$	0.9513	12.968	6.114	20.341	1.78
	$\mathcal{M}_{\mathrm{RA}'}$	$n_{\mathrm{RA}'} = 25, r = 4, t = 9$	0.9519	13.602	6.070	20.458	6.76
	$\mathcal{M}_{\mathrm{RA}}$	N/A					
	$\mathcal{M}_{\mathrm{C}'}$	$n_{\mathrm{C}'} = 25, t = 8$	0.9508	15.773	3.590	20.953	23.80
	$\mathcal{M}_{\mathrm{BK}}$	$n_{\mathrm{BK}} = 23$	0.9527	18.523	2.222	21.215	45.38
	$\mathcal{M}_{\mathrm{BEM}}$	$n_{\mathrm{BEM}} = 23$	0.9527	23.000	0.000	23.000	80.53
0.90	$\mathcal{M}_{\mathrm{R}}$		0.9000	9.534	4.690	13.361	
	$\mathcal{M}_{\mathrm{BG}}$	$n_{\mathrm{BG}} = 19$	0.9036	9.844	5.048	13.852	3.25
	$\mathcal{M}_{\mathrm{RA}'}$	$n_{\mathrm{RA}'} = 19, r = 3, t = 7$	0.9016	9.739	4.912	13.762	2.15
	$\mathcal{M}_{\mathrm{RA}}$	N/A					
	$\mathcal{M}_{\mathrm{C}'}$	$n_{\mathrm{C}'} = 17, t = 6$	0.9026	11.553	2.748	14.350	21.17
	$\mathcal{M}_{\mathrm{BK}}$	$n_{\mathrm{BK}} = 16$	0.9024	12.969	1.741	14.493	36.02
	$\mathcal{M}_{\mathrm{BEM}}$	$n_{\mathrm{BEM}} = 16$	0.9024	16.000	0.000	16.000	67.82
0.75	$\mathcal{M}_{\mathrm{R}}$		0.7500	4.848	2.370	5.613	
	$\mathcal{M}_{\mathrm{BG}}$	$n_{\mathrm{BG}} = 9$	0.7517	4.907	2.526	5.747	1.22
	$\mathcal{M}_{\mathrm{RA}'}$	$n_{\mathrm{RA}'} = 9, r = 2, t = 4$	0.7513	4.895	2.513	5.740	0.98
	$\mathcal{M}_{\mathrm{RA}}$	$r = 3, t = 3$	0.7541	5.167	1.468	5.864	6.60
	$\mathcal{M}_{\mathrm{C}'}$	$n_{\mathrm{C}'} = 8, t = 3$	0.7508	5.154	1.437	5.826	6.32
	$\mathcal{M}_{\mathrm{BK}}$	$n_{\mathrm{BK}} = 8$	0.7701	6.430	1.099	6.911	32.64
	$\mathcal{M}_{\mathrm{BEM}}$	$n_{\mathrm{BEM}} = 8$	0.7701	8.000	0.000	8.000	65.03

References

Alam, K., (1971). On selecting the most probable category. *Technometrics 13*, 843–850.

Bartholdi, J. J. (2010). *The Great Package Race,* The Supply Chain & Logistics Institute. Atlanta: Georgia Institute of Technology. www2.isye.gatech.edu/people/faculty/John_Bartholdi/wh/package-race/package-race.html. Accessed 20 June 2012.

Bechhofer, R. E., Elmaghraby, S., & Morse, N. (1959). A single-sample multiple decision procedure for selecting the multinomial event which has the highest probability. *Annals of Mathematical Statistics 30*, 102–119.

Bechhofer, R. E., & Goldsman, D. (1985a). On the Ramey-Alam sequential procedure for selecting the multinomial event which has the largest probability. *Communications in Statistics—Simulation and Computation B14*, 263–282.

Bechhofer, R. E., & Goldsman, D. (1985b). Truncation of the Bechhofer-Kiefer-Sobel sequential procedure for selecting the multinomial event which has the largest probability. *Communications in Statistics—Simulation and Computation B14*, 283–315.

Bechhofer, R. E., & Goldsman, D. (1986). Truncation of the Bechhofer-Kiefer-Sobel sequential procedure for selecting the multinomial event which has the largest probability (II): Extended tables and an improved procedure. *Communications in Statistics—Simulation and Computation B15*, 829–851.

Bechhofer, R. E., Kiefer, J., & Sobel, M. (1968). *Sequential Identification and Ranking Procedures (with Special Reference to Koopman-Darmois Populations)*. University of Chicago Press: Chicago.

Bechhofer, R. E., & Kulkarni, R. V. (1984). Closed sequential procedures for selecting the multinomial events which have the largest probabilities. *Communications in Statistics—Theory and Methods A13*, 2997–3031.

Bechhofer, R. E., Santner, T. J., & Goldsman, D. (1995). *Design and Analysis of Experiments for Statistical Selection, Screening and Multiple Comparisons*. John Wiley and Sons: New York.

Cacoullos, T., & Sobel, M. (1966). An inverse sampling procedure for selecting the most probable event in a multinomial distribution. In P. Krishnaiah (Ed.), *Multivariate Analysis* (pp. 423–455) New York: Academic Press.

Chen, P. (1988). Closed inverse sampling procedure for selecting the largest multinomial cell probability. *Communications in Statistics—Simulation and Computation B17*, 969–994.

Chen, P. (1992). Truncated selection procedures for the most probable event and the least probable event. *Annals of the Institute of Statistical Mathematics 44*, 613–622.

Kesten, H., & Morse, N. (1959). A property of the multinomial distribution. *Annals of Mathematical Statistics 30*, 120–127.

Levin, B. (1984). On a sequential selection procedure of Bechhofer, Kiefer, and Sobel. *Statistics Probability Letters 2*, 91–94.

Ramey, J. T. Jr. & Alam, K. (1979). A sequential procedure for selecting the most probable multinomial event. *Biometrika 66*, 171–173.

Tollefson, E. (2012). *Optimal Randomized and Non-Randomized Procedures for Multinomial Selection Problems*, Ph.D. dissertation, H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA.

Tollefson, E., Goldsman, D., Kleywegt, A., & Tovey, C. (2013). Optimal selection of the most probable multinomial alternative. In review.

Chapter 7
Vulnerability Discussion in Multimodal Freight Systems

Saniye Gizem Aydin and Pakize Simin Pulat

Transportation infrastructure has been the subject of research mostly for passenger transportation. The impact of extreme events has led to the study of evacuation models and the determination of the most vital links for passenger safety. This chapter focuses on the vulnerability of the transportation infrastructure to the extreme events within a multimodal freight transportation context. Reliability, vulnerability, risk, and resilience terminologies are defined; their relationship with each other within the freight transportation context is discussed. The concepts are illustrated using Hurricane Katrina's impact on the freight flow transportation within a state and for the USA. The intent of the discussion is to promote further research on the vulnerability of multimodal freight transportation systems to extreme events.

Multimodal transportation, a critical component of the global economy, offers solutions to the ever-increasing congestion on the roadway network, helping address pollution and noise problems of large cities. On the other hand, inclusion of two or more modes into the analysis increases the problem complexity significantly, necessitating the creation of transfer points and the study of efficient and safe operations at these points. While the global economy relies heavily on the efficient movement of goods through the interdependent multimodal systems, vulnerability of multimodal transportation systems presents the challenge to understand, resist, prepare, and recover from unexpected events faster, cheaper, and better. Although, the discussion on the multimodal freight transportation and the importance of vulnerability analysis for the multimodal system are discussed in this chapter, we limit the definition and demonstration of the vulnerability measures to a single-mode freight transportation network. Further research is needed to expand the concepts to the multimodal network system.

S. G. Aydin (✉)
School of Industrial and Systems Engineering, The University of Oklahoma,
Room 448C, 202 W. Boyd Street, Norman, OK 73019, USA
e-mail: gizemaydin@ou.edu

P. S. Pulat
School of Industrial and Systems Engineering, The University of Oklahoma,
Room 107, 202 W. Boyd Street, Norman, OK 73019, USA
e-mail: pulat@ou.edu

P. S. Pulat et al. (eds.), *Essays in Production, Project Planning and Scheduling,*
International Series in Operations Research & Management Science 200,
DOI 10.1007/978-1-4614-9056-2_7, © Springer Science+Business Media New York 2014

Multimodal Freight Transportation System and Importance of Its Vulnerability to Extreme Events

The global economy relies heavily on the efficient movement of freight. In 2007, the US transportation system moved freight nearly 3.5 trillion ton-miles (5.6 trillion ton-kilometer (tkm), USDOT BTS 2009). Freight transportation is as important in other countries as it is in the USA. In 2008, China transported 6.85 billion ton-miles (11 billion tkm) of freight; in 2009, it was 2.3 trillion ton-miles (3.6 trillion tkm) for the European Union countries (EU-27) and 4.6 billion ton-miles (7.4 billion tkm) for Russia (EuroStat 2011). As global networks grow, their dependency on third-party logistics provider (3PL) services grows and hence, the importance of the freight transportation becomes more apparent across the globe.

The transportation sector's contribution to the economy through employment is substantial. In EU-27, the freight transport industry employed more than 6.9 million people in 2008. The transportation industry directly employed 10 million in 2009, accounting for 4.5 % of total employment and representing 4.6 % of the Gross Domestic Product (GDP, EuroStat 2011). In the USA, the transportation sector employed 20 million with transportation-related goods and services accounting for more than 10 % of the GDP in 2002; only housing, healthcare, and food industries contributed a larger share to the GDP (USDOT BTS 2004).

Freight utilizes several modes of transportation: trucks, railcars, planes, and ships. Trucks are used extensively on shorter hauls for valuable goods and time-sensitive freight, while rail is used mainly for long haul of heavy freight and waterways for long-haul transport of containers between ports. Inland waterway traffic is also very important, especially on the major inland rivers such as the Ohio and the Mississippi. Airways are used mostly for small, valuable, and urgently needed goods. The trucks have the largest modal share in freight transportation. In the USA the trucks carry 70 % of the freight annually. However, congestion and negative environmental impacts are challenging the freight system and have been the subject of study by the transportation planners in the last decade. As passenger and freight transportation increase, opportunities to link the different modes of transportation are created.

The multimodal transportation services combine advantages of the single-mode transportation and offer potential cost savings in addition to service advantages, such as speed, capacity, routing, and scheduling. Therefore, the multimodal transportation offers flexibility to the changing face of the global markets by meeting the competitive distribution requirements. It is also considered to be more environmentally friendly and can relieve the congestion on other modes. The share of the multimodal transport is small compared to the single mode. In order to benefit from the multimodal transportation opportunities, industries share risks within their supply chains. Increased information sharing, outsourcing of services to 3PL companies for the control, planning, and management of the transportation operations seamlessly from the origin to the destination, and public/private investments to improve the efficiency of the transportation facilities are making multimodal transportation systems a more viable alternative. Combining the multiple modes is not only a flexible option but also an environmentally friendly option. On the other hand, the system is as strong as its weakest link.

In this chapter, we focus on how to assess the vulnerability of a freight transportation network to extreme events. One can apply the methodology to determine what the impact of an extreme event under consideration would be on the transportation of freight from origin to its destination or to identify segments of the transportation network with the largest impact on freight transportation if they become nonfunctional due to an extreme event. We first review the concept of vulnerability and related terminologies, then we study the vulnerability of a single-mode freight transportation system and conclude with a discussion of an approach to studying the vulnerability of a multimodal transportation system.

Vulnerability becomes visible when an extreme event occurs. There was an upward trend in the occurrence of disasters during the period 2000–2008 (Vos et al. 2010) and no sign of a decline so far. History recorded many extreme events since the early ages; let it be natural disasters, by which dinosaurs went extinct, cities were buried under ground, economic development and daily lives were severely disrupted; or man-made disasters by which the society was targeted and cut off of critical sources on purpose. Any interruption in the transportation of people, goods, and services can have a devastating impact on the economy. In extreme cases, the spillover effects may cost even more than the primary damages. All extreme events show us how critical and vulnerable transportation systems are, and how dependent our lives and our economy are on an interconnected network of systems. Below are just a few examples of extreme events and their direct and indirect impacts in an interconnected network of systems.

In August 2003, a malfunction of a single electricity generation plant in Cleveland, Ohio, caused an estimated economic damage of US $ 6.4 billion (Anderson and Geckil 2003). This event triggered electrical systems' failure and resulted in a blackout covering eight US states and two Canadian provinces, leaving about 50 million people in complete darkness (North American Electric Reliability Corporation, NERC 2004). In New York City, the subway system failed trapping several thousand commuters. Telecommunication and water systems were also disrupted (Renesys Corporation 2004; NERC 2004). Investigation of the case revealed a complex matrix of environmental and engineering conditions on the day of the event. The conditions combined with several violations of operating and planning standards caused the widespread crisis (NERC 2004). More recent events, such as Hurricane Irene (August 2011) resulted in high winds and massive flooding, leaving many people homeless as well as taking many lives. The Fukushima earthquake (April 2011) resulted in a failure of a nuclear plant. The leaking gas increased the radiation levels so high that the region was evacuated and the radiation clouds traveled around the world. A tsunami triggered by the earthquake created enormous destruction on the coastal areas, pushing debris islands across the ocean.

More often than not, one event triggers another, and cascading effects are observed where the resulting damage increases exponentially. The 2003 North American blackout, Hurricane Irene, and the Fukushima earthquake validate that a systems approach is more appropriate to understand the reasons and spillover effects of extreme events.

Vulnerability of transportation systems has received attention from researchers only recently. On the other hand, vulnerability has been studied extensively in the

social sciences. With the increasing natural and man-made disasters, most existing vulnerability research focused on critical infrastructure protection. As methods used in transport reliability research are found to be inadequate to study interdependent system failures, new approaches and methods are necessary to assess the vulnerability of transportation systems (Berdica 2002; D'Este and Taylor 2003; Nicholson 2003).

In the next section, we define a multimodal freight system model and discuss the unique properties of single-mode and multimodal transport systems. In the third section, we define vulnerability and discuss how it can be analyzed in a multimodal transportation context. Related terms like reliability and risk are defined and their connection to vulnerability is demonstrated with an example. The section provides insights for the following question: If each of the modes used within a multimodal transportation system is subject to failure, then how does one study the overall risk and vulnerability of the integrated system as a function of an extreme event? We conclude the chapter with future research directions that will assist the transportation planners and operators to study the vulnerability of multimodal freight transportation systems to a set of extreme events.

Multimodal Freight Transport Systems

Multimodal freight systems can be represented by graphs composed of a series of single-mode transport systems connected through transfer points. Layered network models are used to represent various infrastructural systems in the literature, (see Johannson and Hassel 2010; Zhang and Peeta 2011; and Van Nes 2002), particularly on multimodal transportation network design. We will use a two-layered model for each single mode and then link the modes via transfer nodes and edges. Each single mode will be composed of a network layer and a service layer, modeled separately and then linked via transfer points.

Network Model

Consider a network model representation of a multimodal freight transport system by a graph G = (N, E) consisting of a set of nodes N and a set of edges (links) E. $|N| = n$ denotes the number of nodes while number of links is $|E| = m$. Let $G_i = (N_i, E_i)$ represent represent the graph of the subsystem for transport mode i. Let $G_{ij} = (N_{ij}^{'}, E_{ij})$ represent the graph connecting nodes common to N_i and N_j represented by the set $N_{ij}^{'}$ via set of links E_{ij}. Hence,

$$
\begin{aligned}
G &= \{G_i, G_{ij}\}, && \text{for all transport modes i and } j \neq i,\\
N &= \{N_i\}, && \text{for all } i,\\
E &= \{E_i, E_{ij}\}, && \text{for all transport modes i and } j \neq i.
\end{aligned}
$$

Table 7.1 Multimodal freight network model

	Roadways	Railways	Waterways	Airways
E_i	Highways Interstates Arterial roads	Railroad tracks	Routes	Routes
E_{ij}	Artificial edge connecting common locations across layers, i.e., G_i and G_j			
N_i	Location where the freight is originated Final destination for the freight Intermediate location in G_i where the freight changes mode			
N_{ij}	Transportation nodes common across layers, i.e., G_i and G_j			

For the sake of simplicity, we will use the notation $N_{ij}^{'}$ when referring to the transport nodes common to modes i and j and N_i when referring to the nodes only in the subnetwork for mode i. Table 7.1 describes examples of each node and link type. An example multimodal transportation model is presented in Fig. 7.1. Let $G_1 = Airways$, $G_2 = Roadways$, $G_3 = Railways$, and $G_4 = Waterways$.

Network and Service Layers

Network layers show the physical representation of corresponding modes (Fig. 7.1, left). According to the graph:

$$
\begin{aligned}
N &= \{n_1, n_2, ..., n_{22}, n_{23}\} \\
N_{12}^{'} &= \{(n_1, n_2, n_3, n_4) \in N_1 \text{and } (n_5, n_6, n_{11}, n_{12}) \in N_2\} \\
E_{12} &= (n_1, n_5), (n_2, n_6), (n_3, n_{11}), (n_4, n_{12}) \\
N_{23}^{'} &= \{(n_7, n_8, n_{10}) \in N_2 \text{and } (n_{17}, n_{15}, n_{19}) \in N_3\} \\
E_{23} &= (n_7, n_{17}), (n_8, n_{15}), (n_{10}, n_{19}) \\
N_{24}^{'} &= \{(n_7, n_9, n_{10}, n_{12}) \in N_2 \text{and } (n_{20}, n_{21}, n_{22}, n_{23}) \in N_4\} \\
E_{24} &= (n_7, n_{20}), (n_9, n_{21}), (n_{10}, n_{22}), (n_{12}, n_{23}) \\
N_{34}^{'} &= \{(n_{17}, n_{19}) \in N_3 \text{and } (n_{20}, n_{22}) \in N_4\} \\
E_{34} &= (n_{17}, n_{20}), (n_{19}, n_{22})
\end{aligned}
$$

Service layers include the operational characteristics of the corresponding mode (Fig. 7.1, right). A road service model, for instance, may include trucks as well as information on transfer locations and specific operational requirements for transfers and user preferences.

Waterways are composed of routes (represented as links) and connected via ports to mainland represented as nodes, where goods are transferred by intermodal connections. Port environment can be explained in Fig. 7.2 (WEF 2011). It is composed of

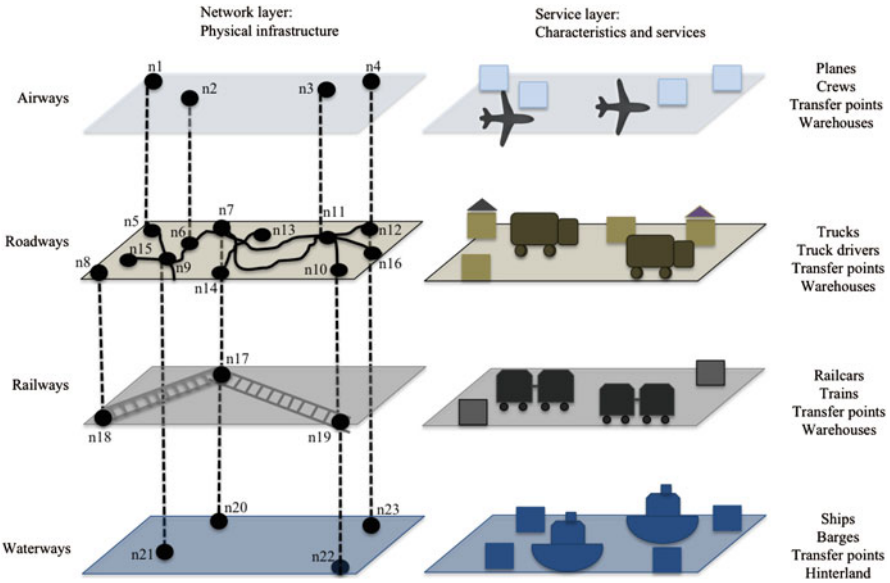


Fig. 7.1 Network and service layers of the multimodal transportation system

five elements: the vessels carrying the goods, navigable waterways that vessels use, the terminal operations such as loading and unloading, the intermodal connection point, and the intermodal connection to other modes by public infrastructure. These elements are modeled in the service layer. Railways and airways can be detailed in a similar fashion to roadways and waterways.

Multimodal transportation systems are represented via edges referred to as "transfer edges" and the incident nodes referred to as the "transfer nodes." (An exception may be a consolidation on a single-mode network, where goods may be transferred to another vehicle on the same network.) In reality, these transfer nodes correspond to the same geographic location. In a passenger flow example, they can represent a train/bus station or an airport. In the case of freight, they can be a port, an airport, or a warehouse. The transfer edges may have zero distance, or a value that represents the value of transfer, for instance in terms of time, service hours, or a cost value. Service network representation describes the set of activities included in the transfer process, such as, loading/unloading, transfer between vehicles, packaging, and sorting. A freight ship may transfer goods to barges or unload at a port to be stored until it is loaded on a railcar or a truck for its next destination. At an airport, goods may be delivered to the warehouse in containers, to be sorted, packaged, and loaded to trucks.

For practical applications, a single-mode freight system is easier to manage than a multientity, multimode freight transport chain. The next section discusses the definition of vulnerability within a freight transportation context. Definitions and formulations of related terminologies are given and illustrated by an example.

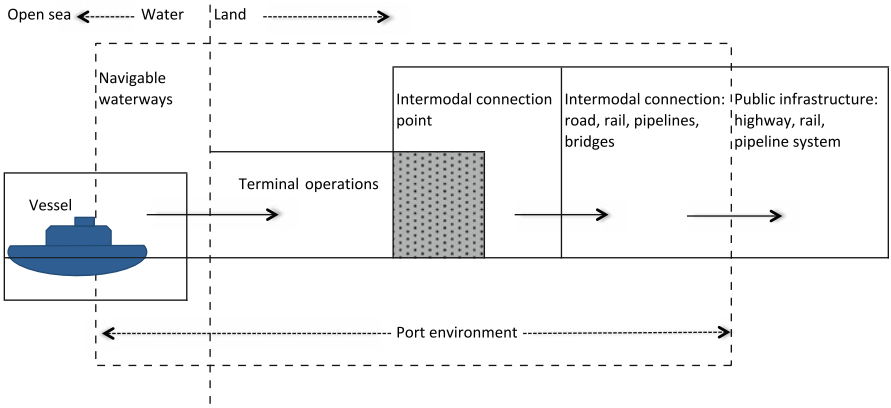


Fig. 7.2 Port environment. (WEF 2011)

Defining Vulnerability in Multimodal Freight Systems

Living in an imperfect world, we design and try to live with imperfect systems, because perfect is just too expensive, or not practical. As a result, failure, malfunction, and in general, vulnerability are inevitable. In relation to both risk and reliability, where does vulnerability stand? In reliability analysis, the objective is to minimize the occurrence/recurrence of failures by understanding and considering the design within budget (i.e., cost) limits (Modarres et al. 2010). In risk analysis, the objective is to minimize the occurrence (the attack ever happening), recurrence (i.e., by increased protection, security measures), as well as the consequences (minimize damage and improve resilience). Although incorporated in risk and reliability, the definition of vulnerability has not reached a consensus yet. Various opinions suggest vulnerability as a consequence, part of risk, and unreliability, and involve partial or complete loss of accessibility or serviceability, which may also change based on user perception.

Vulnerability has been defined in the literature from different perspectives. For example, by definition vulnerability means susceptibility to injury or attack (MW 2008), reduced accessibility due to disruptions (Chen et al. 2007), loss of utility, classified as connective vulnerability (D'Este and Taylor 2003), variations on the accessibility indices, referred to as access vulnerability (D'Este and Taylor 2003), susceptibility to incidents that can result in considerable reductions in (road) network serviceability (Berdica 2002), properties of a transportation system that may weaken or limit its ability to endure, handle, and survive threats and disruptive events (that originate both within and outside the system boundaries; Asbjornslett and Rausand 1999), probability and consequence of degradation on performance of the system (Nicholson and Du 1994; Murray 2011), and "success" of the threat, a manifestation of the inherent states of targeted system(s), each of which is dynamic and changes in response to the inputs and other building blocks (Haimes 2006).

Risk

Risk is the result of a threat causing adverse effects to a vulnerable system—where threat is intent and capability (motivation to harm, and, ability and capacity to attack a target and cause harm; Haimes 2006). The impact of a threat may vary depending on the situation. In general, one will identify possible scenarios and associate a probability and consequence (such as cost) associated with each scenario. Hence, risk is defined as a triplet of *scenario, frequency (probability), and consequence* associated with each scenario that may adversely diminish the system's ability to perform its mission (Kaplan and Garric 1981). As part of risk analysis, vulnerability of the system is identified based on the scenario.

Definition: Risk is a function of probability *p,* scenario *sc,* and consequence *c.*

$$risk = f(p, sc, c) \tag{7.1}$$

The calculation of a risk for a given scenario requires knowledge of the probability of the scenario occurring, the level of impact of the scenario on the performance of the system, and the recovery capability of the system.

Reliability

Reliability is defined as "the ability of an *item* to perform a *required function*, under given *environmental and operational conditions* and for *a stated period of time*" (ISO 8402). Here, the term *item* refers to any entity, which may be a component, system, or a subsystem. A *required function* refers to any function that is required to be performed by the entity and can be a single function or a combination of multiple functions. Therefore, defining the functions of the entity is crucial for reliability assessment. The environmental and operational conditions, as well as time dimensions set the expected/usual conditions and life cycle concepts within the definition. In relation to reliability (or unreliability), vulnerability is identified based on its diminished performance (in terms of capacity, time, or cost, for example).

Definition: Reliability is the probability of system "*s*" at an acceptable level f_n.

$$r_s = p_s(f_n) \tag{7.2}$$

$$ur_s = 1 - p_s(f_n) = q_s(f_n) \tag{7.3}$$

Hence, unreliability is the probability of the system not functioning at an acceptable level.

Reliability of transportation systems is defined in various ways. One of the commonly used and simplest measure of transport network reliability is the *terminal* or *connectivity reliability*, the probability that there is a connection between a pair of nodes in the network when one or more links are broken (Wakabayashi and Iida 1992; Bell and Iida 1997). Other measures include the *travel time reliability*, the

probability of a trip that will be completed within a specified time interval (Yang et al. 2000; Clark and Watling 2005), and the *capacity reliability*, the probability of accommodating a desired level of traffic for a given network (Yang et al. 2000; Chen et al. 2002). Early contributions to the problem of finding the most vital link or node include Garrison (1960), who studied using graph theoretical concepts, and Ratliff et al. (1975) and Ball et al. (1989), who developed various algorithms to determine most vital components of a network.

Vulnerability

Vulnerability as defined in risk analysis is part of the consequences of the identified risks. In reliability analysis, reliability is calculated by the design specifications, and *un*reliability includes every consequence related to a scenario leading to not satisfying one or more design specifications. The system is then assumed to function at an unacceptable level.

For a multimodal transportation system that is defined with a system of systems (network of networks), each of which is composed of a (physical) network and a service layer, connected via specific transfer nodes and edges, vulnerability is a multidimensional-state value of the system expressed as the performance degradation as a consequence of an extreme event that is caused by the dynamic inherent states of the system. The most important facets of multimodal freight transportation can be mapped as in Fig. 7.3: (1) condition and decay, (2) capacity and use, (3) interdependency, (4) spatial factors, (5) threats, (6) policy and political environment, (7) safeguards, (8) temporal factors (Grubesic et al. 2011), (9) economy, and (10) network design. In addition, most important vulnerability characteristics specific to the service layer can be listed as (11) communication, (12) demand, and (13) transfer (goods, vehicle, personnel, storage; Fig. 7.3). We explain each dimension in terms of its individual characteristics as well as its influences on other dimensions and its contribution to vulnerability.

The multimodal freight transportation system is a complex system, with multiple interdependent components. Each component's vulnerability contributes to the system vulnerability; the system is only as strong as its weakest link and as vulnerable as its weakest link. The design of a multimodal freight transportation network that is composed of a connected set of single-mode networks and service layers plays a critical role in determining the system functionality. For example hub-and-spoke networks are more susceptible to vulnerability than random networks because when the link between hub and spoke is targeted, the spoke can easily be disconnected from the main network (Grubesic and Murray 2007). The connectors (transfer points) of single-mode networks (transfer edges and transfer nodes) are of interest explaining key aspects of multimodal freight transportation network design. At transfer locations (such as ports), goods are transferred from one mode to another. The process may include storage, packaging, consolidation, or technical services. In addition to transfer of goods between single-mode networks, changes in the vehicles (i.e., ship to trucks at a port) and personnel (i.e., ship crew to truck drivers at a port) occur.

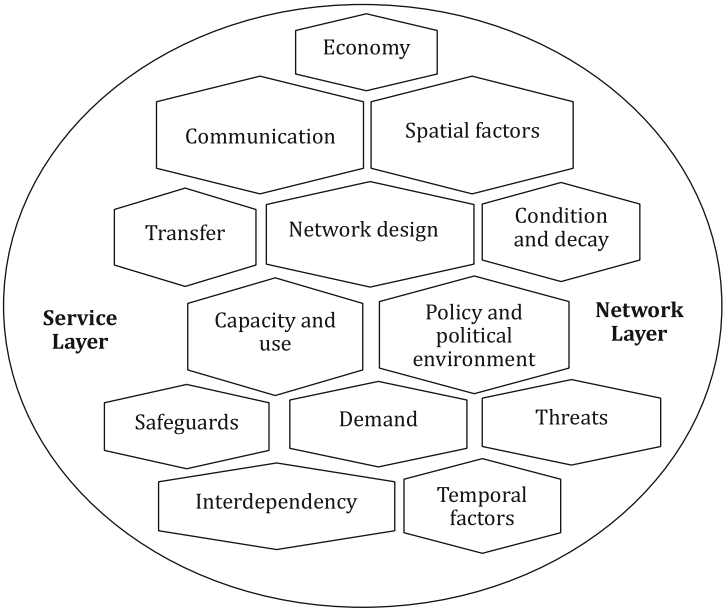


Fig. 7.3 Dimensions of vulnerability, multimodal freight transportation systems

Decision-makers may also differ as when the custody of the goods is transferred from the shipping company to the trucking company at a port. As a result, planning, coordination, and handling of these operations play a significant role, and communication is a key component.

The multimodal freight transportation systems, like other utility services, require substantial investment, continuous maintenance, and timely expansion as the demand for the services grows. Parallel to investments and expansion, demand grows and the cycle continues. However, there are limitations, such as resources, time, or budget. As a result, systems degrade and become more susceptible to extreme events. Therefore, when looking into vulnerability, the current condition of the system needs to be analyzed. When the system or a component of the system is obsolete, failure is inevitable. In addition, where demand grows faster than the available capacity, there is less slack (redundancy) to incorporate the unexpected events. This lack of capacity again increases the vulnerability. Redundancy, generally introduced during the design phase in order to handle some of the variation in daily traffic, may not be sufficient under extreme conditions.

System functionality and its vulnerability are influenced by location and topology; for instance, soil and weather conditions affect the system functionality and may increase the vulnerability of the system. For example, the Gulf Coast is susceptible to hurricanes and tropical storms, whereas southern California is susceptible to wildfires and earthquakes (Schmidtlein et al. 2008). The widespread multimodal freight transportation networks may be subject to different environmental threats, as well as other extreme events. Another aspect is the proximity or interdependency of systems,

which may trigger cascading failures (i.e., 2003 Northeast blackout). Making use of safeguards in design or in addition to the design may decrease vulnerability. Policies and political environments can influence communication between agents to elevate the collaboration and introduce benefits.

The timing of the threat also plays an important role in the resulting vulnerability. Rush-hour traffic would carry a high number of cars on the transportation network, and in the case of a failure, vulnerability is higher than, for example, at 3:00 in the morning. Duration of the threat, such as the time an earthquake lasts, directly influences the vulnerability of the system. If a segment of the transportation network is not being used for freight transportation, then we will categorize it as not important and hence not vulnerable.

Each of the facets such as timing and duration of the threat, and importance of the affected segment of the network on freight transportation contributes to the system vulnerability in a positive or a negative way. Assume each of these dimensions is expressed as a variable x_i. The *vulnerability* of a system can be defined based on the change in these facets:

$$Vulnerability = f(\Delta x_1, \Delta x_2, \ldots, \Delta x_{12}, \Delta x_{13}). \tag{7.4}$$

System functionality is the actual result of the inherent dynamic states of the system and we assume that the vulnerability is the change in the system's functionality

$$V = (f_n - f_m) \tag{7.5}$$

where, f_m is the lowest system function value reached after an extreme event. In this study, we assume that the system's functionality reaches the state f_m as a result of an extreme event. In general, the system may reach a level of f_k with a probability p_k and we can find the expected value of vulnerability, $E(V)$:

$$E(V) = f_n - \sum p_k f_k. \tag{7.6}$$

Acknowledging the multifaceted nature of our multimodal transportation system, one should pay close attention to the reliability, vulnerability, and risk associated with a given freight transportation network. We will next illustrate how vulnerability, reliability, and risk come into play in the case of an extreme event impacting the freight transportation network. Two new measures, resilience and recovery, are also introduced to discuss the impact of an extreme event on the functionality of the transportation system which in turn will lead to a new definition of risk.

Vulnerability, Recovery, and Resilience

Figure 7.4 is an illustration of the impact of an extreme event on the system functionality (performance) as a function of time. The figure assumes three possible scenarios associated with the extreme event. The system recovery depends on the

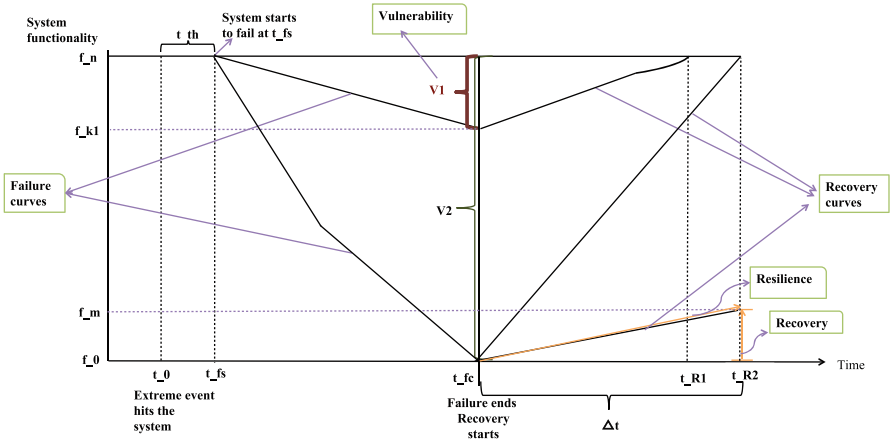


Fig. 7.4 Vulnerability, resilience, and failure recovery behavior of systems

scenario. This figure will form the basis of our discussion for the remaining sections of this chapter.

Prior to an extreme event, a transportation system is assumed to function normally at the desired (acceptable) performance level, f_n. So, let f_n be the predisaster system functionality level as seen in Fig. 7.4. When an extreme event hits the system, the system can either fail or cope with the situation. The system may withstand the pressure of the event for some time; this interval is referred to as the *system failure threshold*, t_{th}. For instance, only a single bridge failed in the 2002 I-40 bridge collapse (OK). The time between the start and completion points of the bridge failure was negligible (instantaneous failure). In the case of an M3 hurricane, such as Hurricane Katrina (2005), many bridges and roadways incrementally failed due to various primary reasons ranging from high winds, heavy rain, and flooding. The damage moved from south to north. Hence, t_{th} was not negligible. When the threshold value is nonzero as in Hurricane Katrina, the decision-maker may initiate immediate protection protocols during this time period minimizing the impact of the event. Hence, the magnitude of t_{th} is important for the recovery period. Let

$$t_{th} = t_{fc} - t_{fs}, \tag{7.7}$$

where t_{fs} is the failure start time and t_{fc} is the failure completion time. Although we do not elaborate on the threshold value any further in this chapter, it is an important variable that decision-makers must consider while studying impacts of extreme events.

The period between the time that a system starts to fail and reaches a complete failure may follow different paths (see example failure curves in Fig. 7.4). In case of a hub-spoke network, when a hub fails, the system degrades and performance decreases. The failure property is defined based on the network attributes such as the network topology. However, system performance does not diminish completely because the hub is highly connected. On the other hand, when a spoke is cut, complete

failure is highly probable due to its low connectivity and hence, high vulnerability. On another note, structural failures are immediate, and congestion behavior changes in time may take a linear or nonlinear form. We represent different discrete levels of system degradation for a specific case under study via lines for clarity purposes. The differences in failure curves of each case indicate the sensitivity of the performance measure to the extreme event and may be valuable to the decision-makers in terms of determining what actions to take during that threshold time.

When a system cannot handle the impact, the system fails. In other words, it reaches the lowest performance level, f_m, where $f_m = f_0 = 0$ or may be equal to f_k a degraded system functionality level k. We can then compute the *vulnerability, V*, of the system as a measure of how much the system performance has degraded due to the extreme event (which is a composite result of the system's inherent dynamic states). In Fig. 7.4, $V_1 = f_n - f_{k1}$and $V_2 = f_n - f_0$ represent the vulnerability of the system at two different degradation levels.

Recovery

If a system is capable of recovery, then the recovery phase begins. We define two other concepts, *recovery* and *resilience*, which are widely used in vulnerability analysis. Recovery refers to the percent gain in system functionality after the occurrence of the extreme event (disaster). Assume that the system recovers partially to a functionality level f_m from f_0. Then,

$$recovery = (f_m - f_0), \tag{7.8}$$

where f_m may be equal to f_n, meaning that the system recovers completely to its predisaster levels. Similar to vulnerability, recovery depends on the system characteristics. Recovery may involve multiple stages such as no recovery immediately after the disaster, small recovery after main connections are established, and recover to predisaster functionality levels, f_n after some time. The recovery function may not be similar to the failure function. While both provide information on the same system, there are different factors involved in each step, such as external circumstances. Therefore, similar factors in both terms need to be identified in order to eliminate a possible covariance in calculations. Under which conditions failure and recovery curves would be similar, different, and how this information can be helpful for vulnerability analysis is another future research question. Recovery time for different systems may be different—slow or fast recovery may be observed (t_{R1} and t_{R2} in Fig. 7.4). Another possibility is that the system may not reach the predisaster functionality levels in the recovery period (f_m at time t_{R2} in Fig. 7.4).

Resilience

Resilience, re, is used to refer to the *ease of system recovery*, the system's ability to return to a stable functionality level after the extreme event. Here, the time is taken into

consideration to refer how easy it is for the system to return to a designed/operational system performance, f_m. Hence,

$$re = (f_m - f_0) / \Delta t \tag{7.9}$$

or,

$$re = recovery / \Delta t \tag{7.10}$$

where, Δt is the interval between the start and the end of the recovery period, which is $(t_{R4} - t_{R0})$ in Fig. 7.4.

In our context, reliability is the probability that a system functions at a desired level. The risk associated with an extreme event is a term expressing the impact of the event on the performance of the system. Hence, it is a function of system vulnerability and resilience. We rewrite Eq. 7.1 as Eq. 7.10, substituting the consequences of a scenario with vulnerability and resilience of the system. If a system is unable to recover, then the system has a higher risk. The p stands for the probability of the extreme event in the following risk function:

$$risk = f(p, V, re) \tag{7.11}$$

In Fig. 7.4, we gathered vulnerability, resilience, recovery, and risk together to observe the relationship between concepts. In the next section, we will demonstrate how one can relate these concepts to various decision-making phases using Hurricane Katrina as an example.

An Example: The Impact of Hurricane Katrina on Freight Flows on the Roadways A disaster is an event concentrated in time and space in which a relatively self-sufficient subdivision of society undergoes severe danger and incurs losses, resulting in diminished physical and essential functions of the society (Fritz 1961; Peek and Mileti 2002). Hurricanes are one of the costliest and deadliest disasters. Hurricanes deliver high winds, storm surge, and rainfall. The physical size of a hurricane influences the storm surge and the extent of damage. When a vulnerable region faces a hurricane, we observe diminished physical environment and functions of the society who lives in the region in terms of damaged infrastructure, homes, buildings, and even loss of lives. The unfortunate increase in number of extreme events urges researchers, governments, and society to better understand extreme events, preparing, managing, and recovering, given our imperfect and rather unreliable systems.

On the 29 August 2005, Hurricane Katrina hit the land and caused widespread devastation in Louisiana (LA), Mississippi (MS), and Alabama (AL). Categorized as an M3 storm, Katrina hit the New Orleans, LA region, with winds higher than 140 mph and caused 20–25 ft storm surges. Many areas were under water and slight to severe damage was observed in residential and nonresidential homes, government buildings, and infrastructure in three states.

Table 7.2 Hurricane Katrina roadways, failure, and repair timelines

Timeline	Predisaster	Bridge/roadway conditions at the end of						
		August 29	September 5	September 20	1st month	3rd month	6th month	After 6 months
Roadway network conditions	No failures	Hurricane hits the area	Flood I	Flood II	Set I repaired	Set 2 repaired	Set 3 repaired	Set 4 repaired
Scenario name	Sc0.0		Sc1.0	Sc1.1	Sc1.2	Sc1.3	Sc1.4	Sc1.5

1. Impact on Roadway Freight Traffic Flows

According to a report by the American Society of Civil Engineers (ASCE) approximately 45 roadway bridges sustained moderate to major damage, and the transportation system was severely disrupted. In addition, massive flooding left part of the city deserted at least for a month (see DesRoches 2006 for further details of the damage and reasons for failure at roadways and railroads; for further information on damage to the ports and the coasts, see Curtis 2007).

The Appendix lists 24 roadway locations we have selected for the definition of scenarios based on the damage level and the spatial location (Table 7.5). The timeline of the recovery is identified to examine the changes in recovery of the freight supply–demand balance and the network. The damaged roadways and bridges are grouped according to their repair dates. Table 7.2 shows the designed scenario steps: predisaster before the hurricane, two flood cases (dated 5 and 21 August by NGA 2009), and four phases based on the recovery pattern of the infrastructure at the end of the 1st, 3rd, and 6th month, and after the 6th month.

In the flood scenarios the New Orleans area is closed; thus, the area is excluded from the analysis for these specific scenarios by either removing the node from the network or by removing the node from the origin–destination (O–D) matrix so that there is no such node to deliver or ship.

A dynamic freight flow matrix is used for the flood scenario steps, reflecting closure due to floods. Shortest-path (All-or-Nothing, AON) assignments are completed in TransCAD for each scenario step. Based on the travel distance, the percentage change in traffic flow for each scenario step is calculated as shown in Fig. 7.5 and summarized for the states AL, LA, MA, and the USA in Table 7.3.

The impact of the damage caused by Hurricane Katrina on freight flows over the roadways was felt in three states. The closed roadways and the floods cut off the supply routes from the three-state region (AL, LA, MS) to other states (blue lines on Fig. 7.5). Freight flow that normally passes through these regions, particularly through the New Orleans area, was rerouted towards north (orange lines on Fig. 7.5).

If the system functionality for freight flow during the predisaster period (Sc0.0) is represented as 100 % (current performance level), then as a result of the first and the second flood (Sc1.0 and Sc1.1, respectively) the system functionality decreased to 96.86 % for AL. In other words, only 96.86 % of the predisaster freight flow used AL after the two floods. During the three stages of the recovery period, an increase in

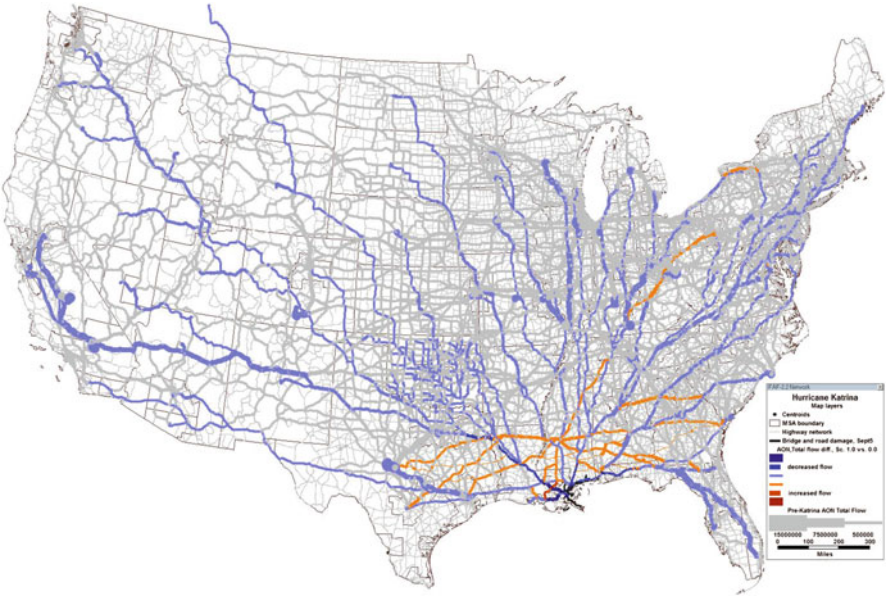


Fig. 7.5 Impact of Hurricane Katrina on roadway freight flows, Sc0.0 versus Sc1.0

Table 7.3 Truck traffic percentage changes due to Hurricane Katrina roadway damages

State	Sc0.0	Sc1.0	Sc1.1	Sc1.2	Sc1.3	Sc1.4	Sc1.5
AL	100	96.86	96.86	100.05	100.43	*100.43*	100
LA	100	73.46	73.46	95.42	100.42	*100.42*	100
MS	100	88.02	88.02	99.31	*100*	100	100
US	100	98.89	98.89	99.87	*100*	100	100

100: fully functional

freight flow was observed (Sc1.2, Sc1.3, and Sc1.4) for AL. The system functionality was above the predisaster period (over 100 %). Decreased system functionality for LA, MA, and the USA were as indicated in Table 7.3. The system functionality above 100 % indicates that more freight was transported than normally would be on the corresponding roadway network; the change in flow in the USA returned to 100 % at the end of the 3rd month (Sc1.3). Likewise LA and AL reached predisaster system levels after the 6th month (Sc1.5). For more details on how freight flow was impacted by Hurricane Katrina, see Aydin et al. (2011).

2. Vulnerability, Recovery, and Resilience

We next demonstrate how vulnerability, recovery, and resilience measures can be calculated for the roadway freight transportation system using the data for Hurricane Katrina. We calculate these measures for three states and for the USA. The relationships among the measures are illustrated in Fig. 7.6.

Assume that the vulnerability of the roadway network can be measured by the change in the truck traffic due to the damage on the roadways. Given the traffic

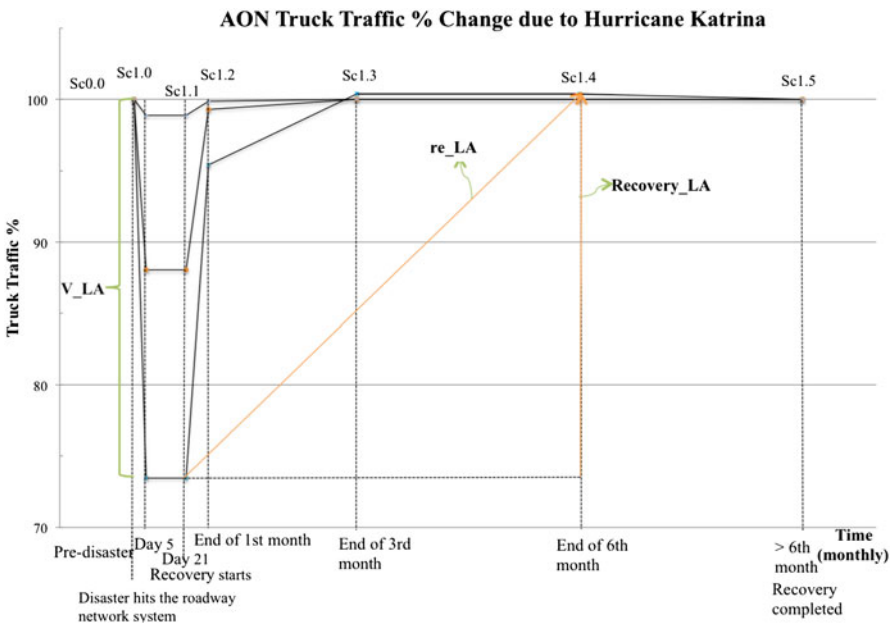


Fig. 7.6 Vulnerability and resilience of roadways to Hurricane Katrina

flow changes on the roadway network, we calculate the vulnerability of the roadway transportation system using Eq. 7.5. Note that we are using the results of Sc1.1 in Fig. 7.3 because the largest impact on truck traffic flow occurred at this particular scenario.

$$V_{AL} = 100 - 96.86 = 3.14$$
$$V_{LA} = 100 - 73.46 = 26.50$$
$$V_{MS} = 100 - 88.02 = 11.98$$
$$V_{US} = 100 - 98.89 = 1.11$$

The state with the highest vulnerability and diminished system performance was LA with a vulnerability value of 26.50, followed by MA and the USA in decreasing order. This information can be used in allocating resources to the highest-need region. For example, Wal-Mart closely observed the path of Hurricane Katrina, and allocated necessary items away from Katrina's path but close enough to satisfy the regions needs faster than the Federal Emergency Management Agency (FEMA) itself.

The recovery periods for MA and the USA are assumed to be at the end of the 3rd month since the system recovered fully at that time. Similarly, performance values are at the end of the 6th month for AL and LA (Table 7.3). By using Eqs. 7.8 and 7.9, recovery and resilience values for the three-state region and the USA can be determined as given in Table 7.4.

Note that LA recovered faster than the other states and was the most vulnerable state for the impact of Hurricane Katrina to freight flow on roadways. In addition, LA

Table 7.4 : Recovery and resilience of the roadway network from Hurricane Katrina

State	Recovery	Resilience
AL	3.57	0.60
LA	26.96	4.49
MS	11.98	3.99
US	1.11	0.37

also has the highest resilience value. Although LA experienced the largest negative change in the roadway functionality, the LA area's resilience was higher than the resilience of the other states and the USA.

Assuming that the probability of a hurricane was same for the regions, we can assess a value for the risk measure as a function of vulnerability and resilience measures using Eq. 7.11 as indicated in Fig. 7.7. Note that lower vulnerability value and higher resilience value lead to a low risk for the system. Hence, a high-vulnerability–low-resilience point on the graph represents the highest-risk situation. The graph is partitioned with red and blue lines to indicate regions with extreme risk and comparatively lower risks, respectively. This information will be useful for decision-makers in assessing state-level vulnerability and resilience of transportation infrastructure for risk management purposes.

Assuming that the probability of a major hurricane hitting each area is the same, based on the vulnerability and resilience values calculated for this example and illustrated in Fig. 7.7, the USA as a whole has the lowest vulnerability-resilience binary value demonstrating the comparatively low vulnerability and resilience of the roadway network. AL, MA, and LA follow in increasing order. The US binary value presents a faster adaptation and rapid recovery, concluding that the USA managed to absorb the extreme event impact. On the other hand, the LA area is highly vulnerable and also expected to be highly resilient to the impact of Hurricane Katrina. If the probability of a hurricane hitting the regions is not constant, then resilience and vulnerability calculations should be modified to reflect this fact.

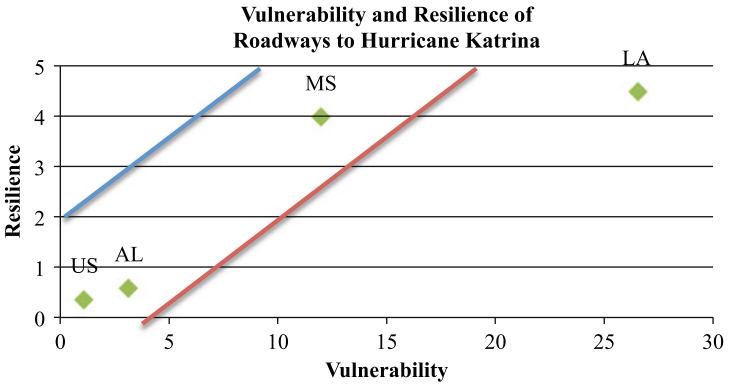


Fig. 7.7 Vulnerability and resilience of roadways to Hurricane Katrina

Conclusion and Suggestions for Future Research

The transportation system is a necessary component of the economy. This vital link is being challenged by the increasing demand on transportation infrastructure and services. Multimodal transportation infrastructure offers cost and service advantages, flexibility, speed, capacity, routing, and scheduling. Hence, it is of utmost importance for the competitive global markets and strong economies. However, the increased number and frequency of extreme events are threatening the aging transportation infrastructure, having a significant impact on the transportation of freight and hence, impacting national and global economies.

Disasters such as the Kobe earthquake in 1995, the Northeast blackout in 2003, and, Hurricane Katrina in 2005 crippled transportation services, damaged infrastructure, and caused social and indirect losses. The need to understand the behavior of systems (why and how systems fail, what happens when an extreme event hits a region, and the region's ability to recover) and systems' vulnerability and resilience to extreme events is vital to minimize (or perhaps eliminate) its impact to the global economy. The study of vulnerability, reliability, resilience, and risk is important to transportation systems. While literature on the application of these terminologies to different problem settings is vast, these measures have not been discussed all together for a given problem setting. However, the relationship among these terms is important and most often depends on the characteristics of the problem setting in question. In this chapter, we define and illustrate the relationships of the terminologies within the context of a multimodal freight flow transportation system. The intent of the discussion is to stimulate further research on the topic and provide valuable insights to the transportation planners and decision-makers as to how vulnerable and resilient the transportation infrastructure is to extreme events.

The roadway damage caused by Hurricane Katrina (2005) is used to demonstrate the above concepts and the impact of the hurricane to the freight flow transportation traffic on the roadway network. Future research can involve the expansion of our concepts to analyze vulnerability of a multimodal transport route. For instance, assume we are given two network layers, G_1 and G_2 are connected via $E_{12} = (n_1, n_2)$. The artificial edge connecting two layers is E_{12} and the transfer points are $n_1 \in N_1, n_2 \in N_2$. Assume origin O is $n_3 \in N_1$ and destination Dis $n_4 \in N_2$. The route r_1 is on N_1starting from n_3 to n_1 and route r_2 is on N_2 starting from n_2 to n_4, where goods from O to D take route r_1, transfers from layers G_1 to G_2 through E_{12} and takes route r_2 through in addition, $(n_1, n_3 \in N_1)$ and $(n_2, n_4 \in N_2)$. Then the complete route becomes $r_1 \cup E_{12} \cup r_2$, and vulnerability of this route can be calculated using multiattribute theory and decision analysis tools by defining the facets influenced by the dynamic system characteristics and estimating the vulnerability and resilience from the change in system functionality in case of an extreme event. Subsequently, we can relate vulnerability of the route to other terms and have a comprehensive view on the multimodal network.

Further research is also necessary to determine how the dimensions listed in Fig. 7.3 contribute to vulnerability and how one can aggregate the impact of each dimension into one vulnerability value. If a weight schema is to be proposed, then how

these weights should be determined will also be the subject of future research. Note that the weights of these dimensions might be different for a planner, a traveler, or a freight company since they represent different utilities for different settings. Research is needed to determine criticality of the artificial edges that connect different modes via transfer points and the transfer characteristics into vulnerability calculations. Moreover, understanding of how a system fails and then recovers provides insights for future design, prevention, and recovery strategies. Connection of the knowledge and complex system analysis, specifically a multimodal transportation system with multiple interdependent networks and service layers with multiple decision-makers, will equip us towards a less vulnerable and more resilient future.

Appendix

Table 7.5 List of bridges and roadways used in the Hurricane Katrina roadway scenarios. (Modified from DesRoches 2006)

Damaged bridge/roadway name	Carried	Damage level	State	Repair time
Bayou La Batre Bridge	Highway 188	Moderate	AL	≤ 1
Cochrane Africatown USA Bridge	US-90	Moderate	AL	≥ 6
Mobile Delta Causeway	I-10 to US90/98	Moderate	AL	≤ 6
Bayou Lafourche @ Leeville	LA-1	Extensive	LA	≤ 1
Bonfouca	LA-433	Extensive	LA	≤ 6
Caminada Bay	LA-1	Extensive	LA	≤ 1
Chef Menteur	US-90	Extensive	LA	≤ 3
Claiborne	LA-39	Moderate	LA	≤ 3
East Pearl River	US-90	Moderate	LA	≤ 1
Inner Harbour Navigation Channel	Florida Avenue	Extensive	LA	≥ 6
Lake Pontchartrain	I-10	Complete	LA	≤ 6
Bayou Barataria–Jefferson	LA302	Moderate	LA	
Pontchartrain Causeway	LA Causeway	Complete	LA	≤ 1
Rigolets Pass	US-90	Extensive	LA	≤ 1
David V. LaRosa Bridge	W. Witman Road	Moderate	MA	
Popps Ferry Bridge	Popps Ferry Road	Significant	MA	
Tchefuncte River Madisonville Bridge	LA-22	Moderate	LA	≤ 1
US-11@ Lake Ponchartrain	US-11	Extensive	LA	≤ 1
West Pearl River	US-90	Moderate	LA	≤ 1
Yscloskey	LA-46	Extensive	LA	≥ 6
Biloxi Back Bay Bridge	I-110	Extensive	MA	≤ 3
Biloxi-Ocean Springs Bridge	US-90	Complete	MA	≥ 6
I-10 Pascagoula River Bridge	I-10	Extensive	MA	≤ 1
US-90 Bay St. Louis Bridge	US-90	Complete	MA	≥ 6
US-90 Henderson Point Bridges	US-90	Complete	MA	≤ 6
US-90 roadway between Pass Christian and Biloxi-Ocean Springs Bridge	US-90	Extensive	MA	≤ 6

References

Aydin SG, Pulat PS, Shen Q (2011). A framework to analyze extreme events with case studies, Working Paper, The University of Oklahoma, Norman, OK

Ball MO, Golden BL, Vohra RV (1989) Finding the most vital arcs in a network. Operations Research Letters 8:73–76

Bell MGH (2000) A game theory approach to measure the performance reliability of transport networks. Transportation Research Part B 34(6):533–545

Bell MGH, Iida Y (1997) *Transportation network analysis*. Wiley, Chichester

Berdica K (2002) An introduction to road vulnerability: What has been done, is done and should be done. Transp Policy 9:117–127

Chen A, Yang H, Lo HK, Tang WH (2002) Capacity reliability of a road network: An assessment methodology and numerical results. Transportation Research Part B 36:225–252

Clark SD, Watling DP (2005) Modeling network travel time reliability under stochastic demand. Transportation Research Part B 39:119–140

Curtis SA. (2007). *Hurricane Katrina damage assessment: Louisiana, Alabama, and Mississippi Ports and Coasts*. Reston, VA: American Society of Civil Engineers

DesRoches R (2006). *Hurricane Katrina: Performance of transportation systems*. Reston: ASCE Technical Council on Lifeline Earthquake Engineering

D'Este GM, Taylor MAP (2003) Network vulnerability: An approach to reliability analysis at the level of national strategic transport networks. In: Bell MGH, Iida Y (Eds.) *The network reliability of transport*. Pergamon-Elsevier, Oxford, pp 23–44

EuroStat. (2011). EU transport in figures: Statistical pocketbook 2001. http://ec.europa.eu/transport/publications/statistics/pocketbook-2011_en.htm.

Fritz C (1961) Disaster. In: Merton RK, Nisbet RA (Eds.) *Contemporary social problems*. Harcourt Press, New York, pp 651–694

Garrison WL (1960) Connectivity of the interstate highway system. Pap Reg Sci 6:121–137

Jenelius E, Petersen T, Mattsson LG (2006) Importance and exposure in road network vulnerability analysis. Transportation Research Part A: Policy Practice 40(7):537–560

Knoop V, Zuylen H van, Hoogendoorn S (2008) The influence of spill back modeling when assessing consequences of blockings in a road network. European Journal of Transportation Infrastructure Research 8:287–300

Kurauchi F, Uno N, Sumalee A, Seto Y (2009) Network evaluation based on connectivity vulnerability. *Transportation and Traffic Theory 2009: Golden Jubilee: Papers selected for presentation at ISTTT18*: 637–49. Springer, ISTTT Series. Dordrecht and New York

Matisziw TC, Murray AT (2009) Modeling s–t path availability to support disaster vulnerability assessment of network infrastructure. Comput Oper Res 36:16–26

Murray AT, Grubesic TH (2007) Reliability and Vulnerability in Critical Infrastructure: A quantitative geographic perspective. Springer, Berlin Heidelberg

Murray AT, Matisziw TC, Grubesic TH (2008) A methodological overview of network vulnerability analysis. Growth Change 39(4):573–592

MW (Meriam-Webster) Dictionary (2008)

NERC (2004) *Final report on the August 14, 2003 Blackout in the United States and Canada: Causes and Recommendations*, North American Electric Reliability Council (NERC)

Nicholson A, Du Z-P (1994) "Improving transportation system reliability: A framework", 17th ARRB Conference, Gold Coast, Queensland, pp. 1–17

Nicholson A (2003). Transport network reliability measurement and analysis. *Transportes, XI*, 49–62

NGA. (2009). Hurricane Katrina remote sensing data. FEMA Mapping and Analysis Center. http://www.gismaps.fema.gov/2005pages/rsdrkatrina.shtm. Accessed 10 Mar 2009

Peek LA, Mileti DS (2002) The history and future of disaster research. In: Bechtel RB, Churchman A (Eds.) *Handbook of environmental psychology*. Wiley, New York, pp 511–524

Ratliff HD, Sicilia GT, Lubore SH (1975) Finding the n most vital links in flow networks. Manage Sci 21:531–539

Renesys C (2004) Impact of 2003 blackouts on internet communications. Retrieved from www.renesys.com/wp-content/uploads/.../Renesys_BlackoutReport.pdf

Schmidtlein MC, Deutsch RC, Piegorsch WW, Cutter SL (2008) "A sensitivity analysis of the social vulnerability index". *Risk Analysis 28*(4):1099–1114

Szeto WY, O'Brien L, O'Mahony M (2007) Generalization of risk averse traffic assignment. In: Allsop RE, Bell MGH, Heydecker BG (Eds.) *Transportation and traffic theory*. Elsevier Science, Amsterdam, pp 127–153

van Nes R (2002) Design of multimodal transport network, Civil Engineering. Delft Technical University, Delft, p. 304

Vos F, Rodriguez J, Below R, Guha-Sapir D (2010) *Annual disaster statistical review 2009: The numbers and trends*. CRED, Brussels

USDOT BTS (2004) *Transportation statistics annual report*. U.S. Department of Transportation (USDOT), Bureau of Transportation Statistics (BTS), Washington, DC

USDOT BTS (2009) *Transportation statistics annual report*. U.S. Department of Transportation (USDOT), Bureau of Transportation Statistics (BTS), Washington, DC

Wakabayashi H, Iida Y (1992) Upper and lower bounds of terminal reliability of road networks: an efficient method with Boolean algebra. Journal of Natural Disaster Science 14(1):29–44

WEF (2011) Global risks 2011—a global risk network report. World Economic Forum. Retrieved from http://reports.weforum.org/global-risks-2011/

Yang H, Bell MGH, Meng Q (2000) Modeling the capacity and level of service of urban transportation networks. Transportation Research Board B 34:255–275

Zhang P, Peeta S (2011) A generalized modeling framework to analyze interdependencies among infrastructure systems. Transportation Research Part B 45:553–579

Chapter 8
Scheduling and Financial Planning in Stochastic Activity Networks

Bajis M. Dodin and Abdelghani A. Elimam

Introduction

Stochastic Activity Networks (SANs) deal with projects where the required information for managing the project is not known with certainty. In most cases, information related to duration or resources of some or all activities are given as random variables (r.v.) characterized by probability distribution functions (pdfs). Examples of these projects are ample; they include most of high technology projects, new product development projects, behavioral and service oriented projects, among others. Management of SAN projects raises important issues that are emanating from the stochastic variations of the project (see Chaps. 4 and 5 of Elmaghraby 1977; Herroelen and Leus 2005). One of these issues, which has been heavily investigated, is the determination of the project schedule and project completion time. Another issue is the determination of the project budget and financial plan; in spite of its importance, this issue is yet to receive the proper attention (see Wiesemann et al. 2010; Chap. 3 of Demeulemeester and Herroelen 2002; Dayanand and Padman 1998).

Random variations cover various aspects of stochastic projects. In most cases, these variations emanate from the need to develop or discover the required innovations or technology for achieving the project objectives. These may lead to variations in the structure of the project network, the duration of activities, the amount of resources needed, and the prices paid to acquire these resources. All of these variations lead to changes in project schedule, duration, and budget. For instance, network structure may be hard to finalize at the initial stage. It also may be altered at later stages of the project due to unforeseen conditions, change in technology, or quality

B. M. Dodin (✉)
College of Business, Alfaisal University,
P.O. Box 50927, Riyadh 11533, Saudi Arabia
e-mail: bdodin@alfaisal.edu

A. A. Elimam
School of Science and Engineering, Mechanical Engineering Department,
American University, Cairo, Egypt
e-mail: aelimam@aucegypt.edu

P. S. Pulat et al. (eds.), *Essays in Production, Project Planning and Scheduling,*
International Series in Operations Research & Management Science 200,
DOI 10.1007/978-1-4614-9056-2_8,

audit results. As project work proceeds, new conditions might be uncovered that would necessitate adding or deleting activities leading to a change in the network structure. The results of completed project work quality audits might require undertaking additional activities for repair or rework. Most work on SANs assume that network structure is always given, and stays that way throughout the project management cycle (again see Demeulemeester and Herroelen 2002). In this chapter we also assume that the structure of the activity network (AN) is given.

In addition to the potential change in the network structure, managers and researchers are always faced with the challenging task of estimating the required resources and duration for the activity. The quantity of resource(s) required to complete an activity may be expressed as a random variable(s) due to not precisely knowing the details of the activity work. Consider for instance the number of programming hours a software engineer will consume to develop a certain module of a larger program; or the number of experiments required before a certain compound or medicine is developed. The resources can be of two types: Renewable such as operators and machines/tools, and nonrenewable such as all consumables, typical of which are money, and material consumed. The skill of the renewable resource or the manner of its deployment may affect the duration of the activity. An example of this is often found in service projects, such as those in health care or in audit staff scheduling (see Dodin and Elimam 1997), where the duration of the activity depends on the skill level. Hence, duration of the activity may depend on the amount of the renewable resources required; consequently, the duration of the activity may be expressed as a function of the required renewable resources. By contrast, the duration of the activity may be independent of the amount of nonrenewable resources required or the mode of deploying these resources. Finally, the prices paid for some or all of these resources may also vary, particularly for projects with long durations, or in times of economic volatility.

Based on the above, one can see that the combined effect of the stochastic variations in the network structure, duration of activities, amounts of resources, and the price of these resources would have a profound impact on the project budget, its distribution over the various activities, and on its schedule. These variations have been a major source of difficulty for budgeting and managing projects with high degree of uncertainty. In spite of the need to develop such pioneering projects at the least possible cost, not much work has been published in the area of budgeting and financial planning, and scheduling for stochastic projects. This is different from the work that has been completed on maximizing net present value of stochastic projects such as that of Wiesemann et al. (2010); Sobel et al. (2009); Benati (2006).

Recognition of the need to manage projects with some of the above variations started with the work of Elmaghraby (1964) on generalized activity networks. He attempted to handle the issues of scheduling and project duration emanating from structural changes in the AN. He introduced a methodology that combines elements of the Project Evaluation and Review Technique (PERT) and the Critical Path Method (CPM) with those of decision nodes/analysis. This work was expanded by Pritsker and Whitehouse (1966) through the development of Graphical Review and Evaluation Technique (GERT) to include cost elements. The difficulty in solving GERT

models led to developing GERT Simulation by Pritsker and Sigal (1974). In GERT Simulation the duration of the activity is specified as a r.v., independent of the required resource(s), and the cost of the activity is expressed as a linear function of its duration. For a survey of research related to GERT see Neumann (1999). Since then not much has been published on the area of financial planning in stochastic projects. In this chapter we assume that for a given project the renewable and nonrenewable resources are given as random variables with specified pdfs. The duration of the activity is also a r.v. written as a function of the renewable resource(s) required. Given these relations, the managerial questions that remain to be answered are:

1. What is the bidding price or planned budget (PB) for the project?
2. Given the PB, what is the financial plan for the project?
3. What is the optimal duration and schedule of the project for the PB?
4. How do the variations in these relations affect the above three measures (PB, financial plan, and project duration and schedule)?

This chapter deals with the above questions. It develops practical and accurate analytical procedures to answer these questions. This procedure can be used to explore the relationship between the probability of completing the project at a given time and the amount of resources to be used as well as its corresponding budget, i.e., we establish the time–cost trade off curve for the stochastic project or any of its subprojects/stages.

The chapter is organized as follows: In the section "Determining the Probability Distribution Function of the Project Cost," a procedure is developed to calculate the pdf of the project cost. A PB for the project can be based on the pdf of the project cost. Then, in the section "Determining the Probability Distribution Function of the Project Duration," another procedure is developed to calculate the pdf of project duration. In the section "Calculating the Project Financial Plan," linear programming is used to determine the financial plan that yields the optimal project duration for the given PB. An example is provided in the section "Illustrative Example." Concluding remarks are given in the section "Conclusions and Extensions." The following symbols will be used in the presentation of the chapter, and Fig. 8.1 provides its outline:

Parameters

A	Set of activities in the project
|A|	Number of activities
N	Set of nodes and index of the last node in the project
$B(j)$	Set of nodes preceding node j and directly connecting to it by an activity (i,j)
D(i,j)	Random variable denoting the duration of activity (i,j) ε A; it is a function of R(i,j)
R(i,j)	Random variable representing the cost/quantity of the renewable resource requirements for activity (i,j), US$/unit
NR(i,j)	Random variable representing the cost/quantity of the nonrenewable resource requirements for activity (i,j), US$/unit
T(j)	Random variable representing the duration of the project up to node j
$TC(i,j)$	Random variable denoting the total cost of activity (i,j).
$NTC(j)$	Random variable denoting the cost of the project up to node j

PB	Planned budget derived from the pdf of the NTC(N)
F_{ij} (r)	$= \Pr(TC(i,j) \leq r)$, cumulative probability distribution of the total cost of activity (i,j)
F(t)	$= \Pr(T(N) \leq t)$, cumulative probability distribution of project completion time
CL_{ij}	Lower bound on activity (i,j) renewable resource cost
CU_{ij}	Upper bound on activity (i,j) renewable resource cost
DL_{ij}	Lower bound on activity (i,j) duration
DU_{ij}	Upper bound on activity (i,j) duration
w_{ij}	The conversion factor between the renewable resource cost of activity (i,j) and its duration, i.e., the slope of the renewable resources cost and duration function
NRC	Total cost of all nonrenewable resources for project activities in the PB

Decision Variables

y_{ij}	Duration of activity (i,j) ε A
q_{ij}	Amount of renewable resource funds allocated to activity (i,j)
$t(j)$	Realization for the duration of the project up to node j

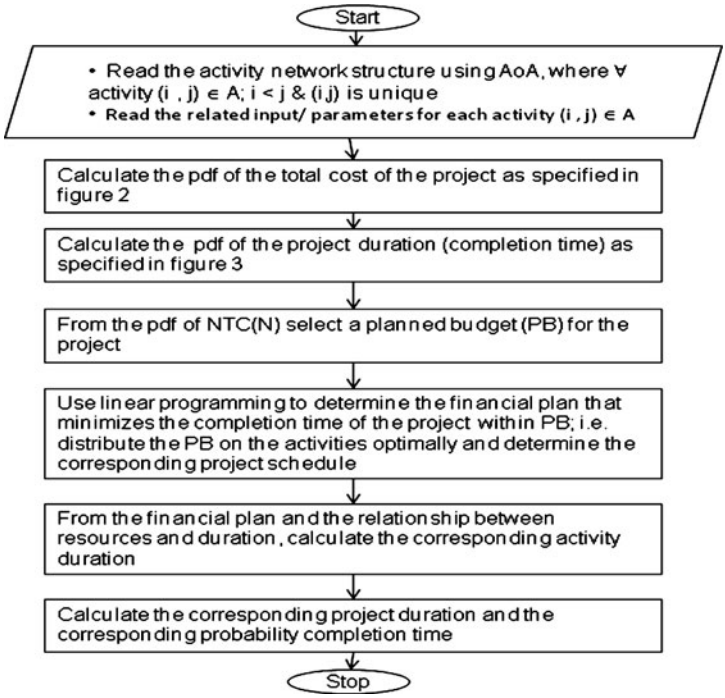


Fig. 8.1 Scheduling and financial planning for projects represented by SANs

Determining the Probability Distribution Function of the Project Cost

In this chapter the project is represented by an activity-on-arc network. The arcs of the network represent the activities, and the nodes represent the events. The events can be numbered from 1 to N where 1 is the unique starting node, and N is the unique

ending node; an activity $(i,j)\ \varepsilon\ A$, where A is the set of all activities in the AN, $i < j$ where i is the start of the activity, and j is the end of the activity and the ordered pair is (i,j) is unique. Since some or the entire project activities are not well defined in the sense of the activity content, the duration and cost of such activities are given as random variables. It is also assumed that the cost of the activity consists of the following two elements:

- *Cost of the nonrenewable (NR)resources:* The quantity of the NR resources required by the activity is a r.v. independent of the activity duration; hence the cost of the NR resources is a r.v. with a given pdf. $NR(i,j)$ can be stated as quantity or as cost, as it is assumed that the cost/unit of the nonrenewable resource is fixed.
- *Cost of the renewable (R) resources:* The quantity of the R resources required for each activity is also a r.v. with a given pdf. It is also observed that the R resources required by the activity determine the duration of the activity, i.e., the duration is expressed as a function of the renewable resource(s). $R(i,j)$ can also be expressed as quantity or cost as the cost/unit is fixed.

As a result of the above assumptions, the cost of each activity is the sum of the above two random variables; it is a r.v. $TC(i, j) = NR(i, j) + R(i, j)$ with a pdf that can be calculated by convoluting the above two probability distributions. In this case if $f(r) = \Pr(R(i, j) = r)$ and $g(s) = \Pr(NR(i, j) = s)$, then

$$F(c) = \Pr(TC(i,j) \le c) = \sum\nolimits_{r=0}^{c} f(r)g(c - r).$$

The pdf of the activity cost is independent of its duration or schedule. The project cost or any of its segments can be calculated by summing the individual activities costs in the project or in its subprojects. Hence, in principle, the cost of the project can be calculated by performing $(2\,|A| - 1)$ convolution operations where $|A|$ is the number of activities. This may be theoretically possible for some simple pdfs, and a small size $|A|$. However, for all practical purposes $|A|$ is not small, and it is not possible to convolute this many pdfs, especially when some of the individual pdfs are hard to convolute. Furthermore, as two pdfs are convoluted, the outcome is a more complicated pdf; when this is convoluted with a third pdf, the resulting pdf is messier, and so on. The convolution process reaches to a point where it cannot be carried out. Consequently, it is very important to develop a practical and accurate procedure to carry out the $(2\,|A| - 1)$ convolution operations ending with the pdf of the project cost in a reasonable computing time. This is the subject of this section.

The above difficulty in carrying out the convolution operations is valid whether we have continuous or discrete distribution functions. It is more valid in the case of continuous pdfs, where the difficulty is apparent. To illustrate this difficulty in case of discrete pdfs, suppose we have 20 activities, the cost of each is a r.v., characterized by a discrete pdf, where each r.v. has five realizations/outcomes. The convolution of two of these will have between 9 and 25 unique realizations; it depends on the similarity between the realizations of the two convoluted distributions. It will have nine unique realizations if the two convoluted pdfs have the same realizations; and it could reach to 25 unique realizations if the two convoluted distributions have

different realizations. Similarly if the resulting pdf is convoluted with a third of the original distributions; the new distribution will have unique realizations ranging from 13 to 125; and so on. Therefore, in this case the final distribution at node N may have unique realizations ranging from 81 to 5^20 (which is a very large number). The following procedure is used to calculate the pdf of the cost at each node in the project ending with node N designating the end of the project.

Procedure for Calculating the pdf of the Project Cost

This procedure is summarized in Fig. 8.2. It starts at node 1, where it has a cost of zero with probability of 1, and then advances sequentially to node 2, then to 3, and so on until node N, ending with the realizations and pdf of the project cost. At each node j it calculates the cost of the subproject ending in that node.

For instance, in the AN of Fig. 8.3, at each node ($j > 1$) we first calculate the pdf of the cost of each activity incident into the node by convoluting the two random variables $R(i,j)$ and $NR(i,j)$. Then, to calculate the cost up to node j, we convolute the pdf of the cost up to node (j–1) with the pdf of cost of the activities incident into node j. So at node $j = 2$, it is simply equal to the cost of activity (1,2). Hence, only one convolution operation is performed to calculate the pdf of $TC(1,2)$. Then, to calculate the cost up to node 3, four convolution operations are performed. The first two are for calculating TC(1,3) and TC(2,3). Then the last two are for convoluting the cost of node 2 with TC(1,3); then the outcome is convoluted with the TC(2,3). The process moves to node 4, where, similarly, four convolution operations are required. Hence, in this case, a total of nine convolution operations are performed. The proposed sequential procedure is stated as follows:

1. Initialization:
   a. Input the AN structure using activity-on-arc mode of representation, and number the nodes from 1 (unique starting node/event of the project) to N (the unique completion node of the project) such that for any activity/arc (i,j) $i < j$, and the pair (i,j) is unique.
   b. Input the pdf of the renewable and nonrenewable costs of each activity $(i,j) \varepsilon A$.
   c. Start at node $j = 1$, and set its cost to 0 with probability of 1; then set $j = j + 1$;
2. Calculating the pdf of the project cost up to node j:
   a. Determine the set $B(j)$, which is the set of activities ending in node j; If node $j - 1$ is not connected to node j, then connect them by adding the dummy activity $(j - 1, j)$ to the set $B(j)$ with a cost of 0 and probability 1. Rank the activities in the set $B(j)$ in an increasing order of their starting nodes.
   b. For each activity $(i,j) \varepsilon B(j)$, calculate the pdf, $F_{ij}(r)$, of the total cost of the activity, denoted by $TC(i, j) = R(i, j) + NR(i, j)$, by convoluting the pdf of the cost for its renewable resources with that of its nonrenewable resources as defined above. Please note that if the pdf of $TC(i,j)$ is already calculated/given, then go to the next activity in the set $B(j)$.

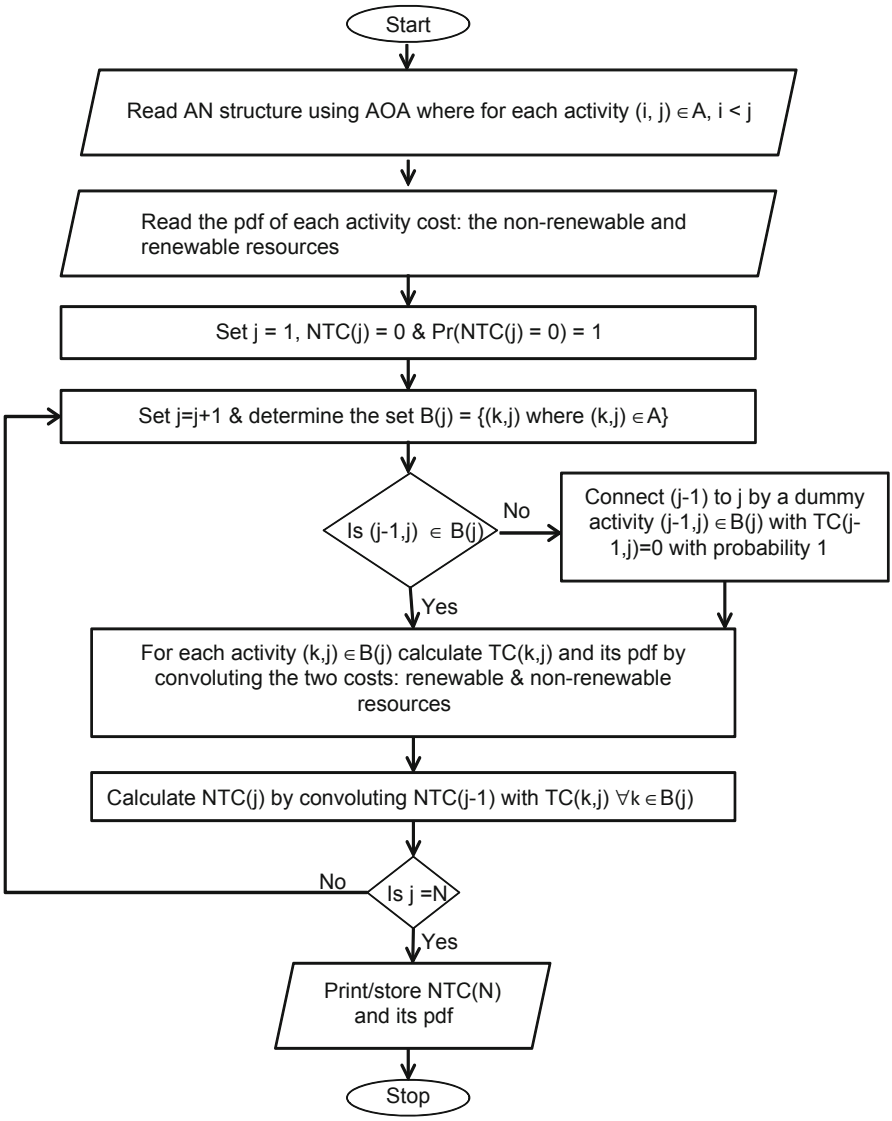


Fig. 8.2 Calculating the probability distribution function of stochastic project cost

c. Let K be the number of realizations for the r.v. $TC(i,j)$. If K is greater than a desired number of realizations the analyst would like to have for the r.v., such as $k < K$, then the pdf of $TC(i,j)$ of the activity is approximated by another pdf with only k realizations. This is done where the full range of $TC(i,j)$ is preserved. Therefore, the maximum and minimum realizations of $TC(i,j)$ are maintained with their respective probabilities.

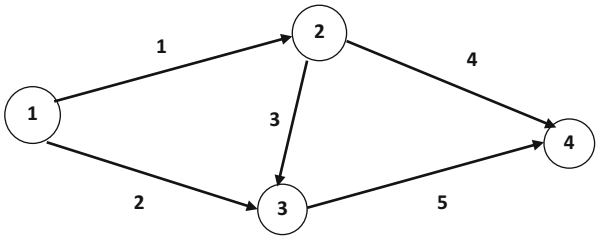


Fig. 8.3 Sample activity network

The remaining $K - 2$ realizations are mapped into $K - 2$ realizations using the same rules as in Dodin (1985).

d. To calculate the cost of the project up to node *j,* denoted by $NTC(j)$, convolute the pdf of $NTC(j - 1)$ with the pdf of the cost of the first activity in the set $B(j)$. The resulting pdf is then convoluted with the pdf of the cost of the second activity in $B(j)$; and so on until convoluting with the pdf of the cost of the last activity in the set $B(j)$. After each convolution operation if $K > k$, do the operation presented in 2.a.

3. **Termination**: Set $j = j + 1$. If $j < N$ go to 2, otherwise record the pfd of $NTC(N)$ and all its statistics.

The convolution operation of two pdfs is carried out as it is the case in Dodin (1985) and it will not be repeated in this chapter. It should be noted that the convolution operation presented in Dodin (1985) assumes that all random variables are characterized by discrete pdfs. Consequently, if an activity in SAN has a cost with a continuous pdf, it should first be discretized. This can be done by applying the discretizing procedure developed also in Dodin (1985).

From the above distribution for $NTC(N)$, designating the cost of the project, we can calculate all of its statistics (mean, mode, median, min., max., skewness, and quintiles). This is done without any reference or reliance on the activity duration or project schedule.

Given that most projects consist of many activities (more than ten), the central limit theorem implies that the pdf of the project cost, $NTC(N)$, converges to a normal distribution with mean value μ_p equaling the sum of the mean values of the cost for all activities, and project cost variance σ_p^2 equaling the sum of the variances for all activities. This may allow us to establish bidding price forecasts with certain confidence limits, establish upper and lower bounds on the project cost, and assess the risks attached to each bidding price or PB.

Determining the Probability Distribution Function of the Project Duration

As stated above, the duration of the activity is given as a function of its renewable resource requirements. In most instances the duration, expressed as a r.v., is negatively correlated with the quantity/cost of the renewable resource requirements: the larger

the quantity of renewable resources, the lower the duration, and vice versa. Hence the lower end for the cost distribution of the renewable resources of an activity matches the upper end of the distribution of its duration. The pdf of the activity duration is calculated using the inverse of the above function. In case the pdf of activity (i,j) duration, denoted by $D(i,j)$, is given as a conditional probability of that of the $R(i,j)$, then the pdf of $D(i,j)$ can be calculated first from the conditional probability.

Once the pdf of the duration for each activity is determined, then the pdf of the project duration, denoted by $T(N)$, or any of its subprojects ending in node j, $T(j)$, can be calculated using any of the available methods. However, as it is stated in Demeulemeester and Herroelen (2002); Dodin (1985); Elmaghraby (1977), calculating the exact pdf of the project duration for nontrivial projects is not possible. Consequently, and also for practical reasons, we rely on using one of the approximating procedures to calculate the pdf of the project duration. In this regard, we can use the sequential approximating procedure developed in Dodin (1985). This procedure can be applied for projects of any size, independent of the activity underlying probability distributions. However, it may be more practical in approximating the pdf of $T(N)$ to rely on characterizing it. It was shown in Dodin and Servanci (1990) that such a pdf can be approximated by either a normal distribution or by an extreme value (EV) distribution. In both cases, what is required is to simply calculate the corresponding mean and variance of such a distribution, then it will be easy to calculate $F(t) = \Pr(T(N) \leq t)$ for any $t > 0$.

Determining if the pdf of $T(N)$ converges to a normal or to an EV distribution depends on the number of paths competing to be the longest path (in duration) in SAN. The pdf converges to either of the following:

- A normal distribution, exactly as it is the case in the PERT method, if there is a path in SAN that dominates all other paths in the sense that its probability of being the longest path is higher than it is for any other path and with a reasonable margin
- An EV distribution if there are several paths that have similar probabilities for being the longest path

In performing the above test, we use Dodin (1984) to identify the n most critical paths, where n can be any positive integer; but for practical purposes $n = 3$ or 4 can be satisfactory for the normality test. If one path dominates, in probability, all others and emerges to be the longest path, then the project duration is normally distributed and its parameters are approximated exactly as it is in PERT. In this case, the mean duration of the project and its variance are given by:

$$\mu(P) = \sum_{(i,j)\in CP} \mu_{ij} \quad \text{and} \quad \sigma^2(P) = \sum_{(i,j)\in CP} \sigma_{ij}^2,$$

And $F(t) = \Pr(T(N) \leq t)$ for any $t > 0$ can be easily calculated from the standard normal tables. If otherwise, then the pdf of project duration is approximated by an EV distribution, where its parameters are calculated as in Dodin and Servanci (1990).

In this case:

$$\mu(EV) = a_n + \frac{0.577}{b_n} \quad \text{and} \quad \sigma^2(EV) = \frac{\pi^2}{6\,b_n^2}$$

where $n =$ the number of dominating paths (close in length) determined in the above test,

$$a_n = \mu(P) + \sigma(P)\left[\sqrt{2\log n} - \frac{(\log\log n + \log 4\pi)}{2\sqrt{2\log n}}\right]$$

$$b_n = \frac{\sqrt{2\log n}}{\sigma(P)} \quad \text{and}$$

$$F(t) = \Pr\left(T(N) \le t\right) = \exp\left[-e^{-b_n(t-a_n)}\right]$$

The process of determining the pdf of T(N) is summarized in the flowchart of Fig. 8.4.

The pdf of project duration depends on the pdfs of the activity durations. But these depend on the pdf of the corresponding activity renewable costs. Hence, it is expected that the pdf of $T(N)$ is negatively correlated with the pdf of $NTC(N)$ derived in the section "Determining the Probability Distribution Function of the Project Cost." In this case the high end of $NTC(N)$ matches the low end of $T(N)$, and the low end of $NTC(N)$ matches the high end of $T(N)$. The question now is how to use both distributions to establish a project financial plan, and a schedule for all activities. For a given budget or bidding price, as derived in the section "Determining the Probability Distribution Function of the Project Cost," how is the budget distributed over the individual activities? What is the duration of each activity? What is the corresponding project completion time? Can the budget be distributed over the activities to achieve the least completion time for the project? These issues are the subject of the next section.

Calculating the Project Financial Plan

This section deals with distributing the PB on the activities in an optimal manner, where optimality is defined as seeking to allocate more funds to the activities that affect project duration the most; hence, achieving the least possible project completion time for a given budget. Also, it is important to discuss how sensitive the project completion time is to changes in the budget and vice versa? That is, how much should the budget increase to achieve a given completion time, hence, increasing the probability of completing the project within the given time? If PB represents the 90 % realization in the pdf of $NTC(N)$, then how should this amount be distributed among all the activities? Should we distribute these in a uniform manner such as 90 % funding for each activity? Would this yield a project duration realization $t(N)$ where

$$F(t) = \Pr\left(T(N) \le t(N)\right) = 0.90$$

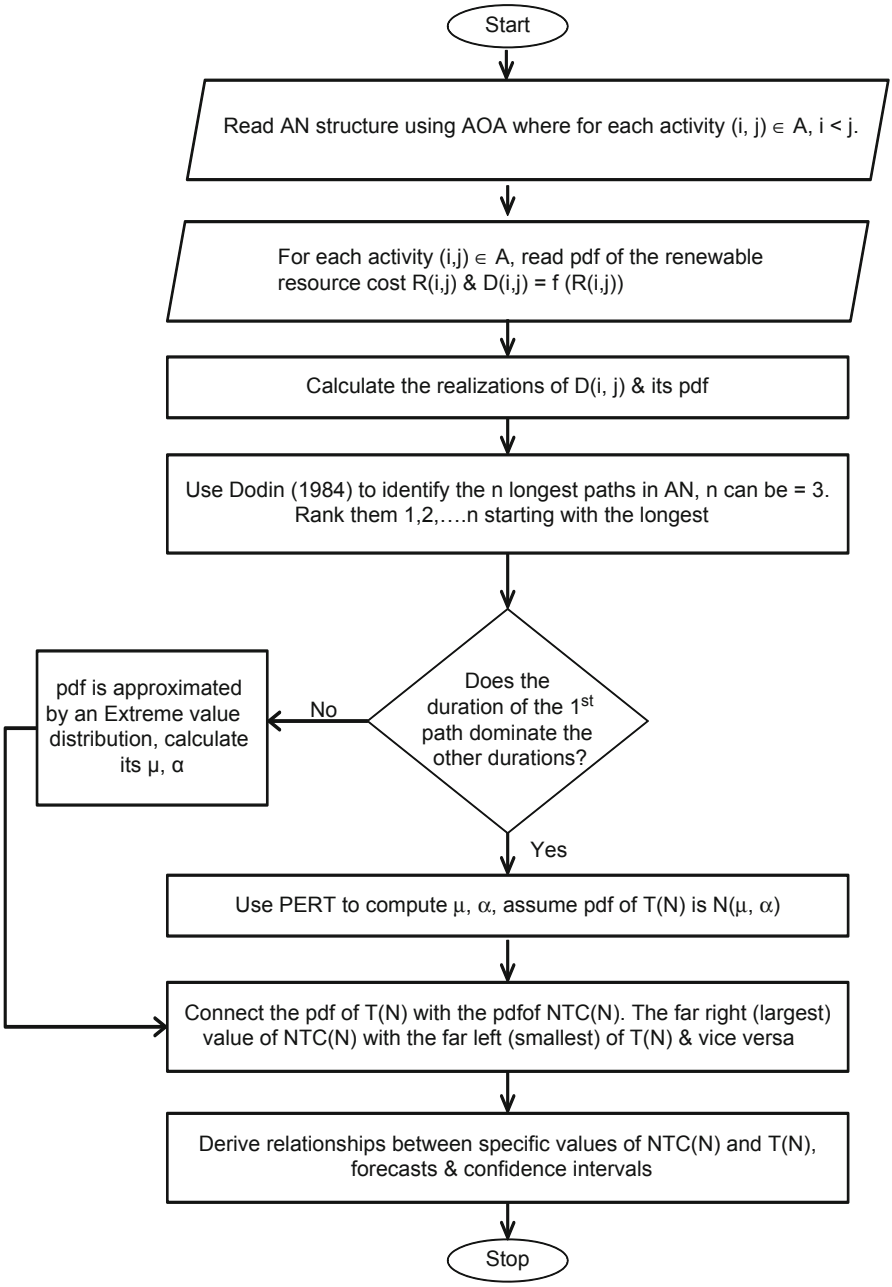


Fig. 8.4 Flowchart of determining the pdf of project duration

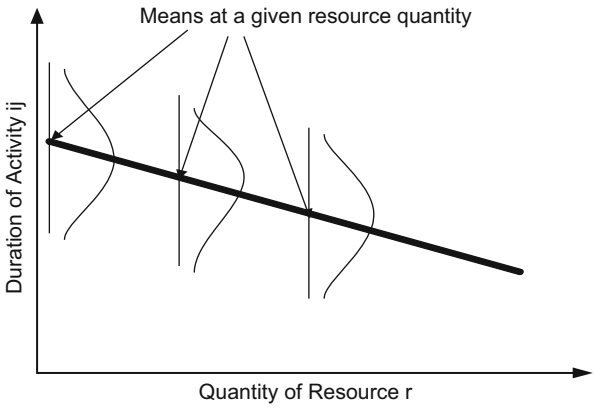


Fig. 8.5 Renewable resource quantity—activity duration

Alternatively, should some activities be funded at a higher level while others at a lower level? How will these be selected, and what is the impact on the project completion time? These questions imply that different allocations of the PB provide for different project schedules and project completion times, $t(N)$. Given the reverse relationship between the activity renewable resources cost and the activity duration, the more funds are allocated to the activity, the lower is its duration (see Fig. 8.5). Hence, if the budget allows for funding each activity at its highest level, then each activity can be processed at the least/crash duration; and the project may be completed at minimum time. Conversely, if the budget is short, and each activity is funded at its lowest level, then this may lead to completing the project at maximum time. Consequently, from the activity cost–duration relation we can determine the maximum project budget required, and hence the minimum project duration; and the minimum budget and the maximum duration. In managing real world projects, one rarely adopts either of these two extremes. Managers try to select the least necessary budget, and use it to achieve the shortest project completion time, i.e., the maximum probability of completing the project within budget.

The above criterion can be used to guide the distribution of the PB over the activities in an optimal manner, where optimality is defined by obtaining the least project duration; which maximizes the probability completion time of the project for the given budget. In deterministic project management, distribution of funds is guided by the rational used in the Ford–Fulkerson (1962) algorithm. In this case, we wish to distribute the funds on the critical activities with the least cost first; hence, achieving maximum reduction in project duration for a given budget. In this case, the problem is formulated as a linear program (LP) to determine the minimum project completion time emanating from a specified budget (see Elmaghraby 1977). In SANs, the concept of critical activity does not work as in deterministic ANs. The corresponding concept in SANs is known as activity criticality index (CI). The CI of an activity is the sum of the CIs of the paths that contain this activity, i.e., the activity is a constituent of the path, where the CI of a path is the probability that the path is the longest in the SAN. If the CI of each activity is determined, then we can use a similar formulation to distribute the funds on the activities with the highest CIs

and lowest costs. However, calculating the CI for each activity in SAN is a problem by itself (see Dodin and Elmaghraby 1985). Consequently we are satisfied by using the relation presented in Fig. 8.5. This expresses the activity average duration as a function of the activity renewable resource cost. Such a relation is used in the following LP formulation to optimally distribute the PB over all activities. In this formulation the project completion time $t(N)$ is minimized subject to:

- The precedence relations between the project activities represented by constraint set number 1
- Upper and lower limits on the activity duration y_{ij} represented by constraint set number 2
- Relationships between the renewable resources cost q_{ij} to be allocated to an activity (i,j) bounded from below by $CL(i,j)$ and the activity duration y_{ij} (as in Fig. 8.5); these are represented by constraint set number 3
- The funds allocated to activity (i,j), q_{ij}, does not exceed CU_{ij}; these are represented by constraint set number 4
- Total distributed funds do not exceed the renewable resources share in the PB as shown in constraint number 5

$$\textit{Min.}\ \mathrm{t}(N)$$

$$\textit{Subject to}$$

$$t(1) = 0$$

$$t(j) \geq t(i) + y_{ij} \qquad \forall j \in N \ \text{ and } \ \forall\, i \ \in B(j) \tag{8.1}$$

$$DL_{ij} \leq y_{ij} \leq DU_{ij} \quad \forall j \in N \ \text{ and } \ \forall\, i \in B(j) \tag{8.2}$$

$$y_{ij} = DU_{ij} - w_{ij}(q_{ij} - CL_{ij}) \quad \forall\, j \in N \ \text{and} \ \forall i \in B(j) \tag{8.3}$$

$$q_{ij} \leq CU_{ij} \quad \forall j \in N \ \text{ and } \ \forall i \in B(j) \tag{8.4}$$

$$\sum_{(i,j)\in A} q_{ij} \leq PB - NCR \tag{8.5}$$

The decision variables in the above LP are the $\{\ y_{ij}\}$ and the $\{\ q_{ij}\}$, where we have $|A|$ variables of each, and $t(j)$, where we have $(N-1)$ variables. Solving this LP provides the financial plan, which is the optimal allocation of the funds over the activities, represented by the values of $\{\ q_{ij}\}$. It also provides the corresponding activity durations $\{\ y_{ij}\}$, and the corresponding project completion time $t(N)$. From the activity durations and event realizations $t(j)$, a project schedule such as that of the latest start time schedule for each activity can be constructed. From the pdf of $T(N)$, we can calculate $\Pr\,(T(N) \geq t(N))$ which is the largest project completion probability within the PB.

The above formulation can also be used to establish the cost–time response curve. For each value of PB we can establish a financial plan, a project schedule, and completion time along with the corresponding completion time probability. This curve provides a menu to chose from for PB, $t(N)$, and $F(t(N)) = \Pr\,(T(N) \leq t(N))$. In the following section an illustrative example is provided.

Table 8.1 Probability distribution of activity cost in US$ 1,000.00

Activity cost	$TC(1,2)$	$TC(1,3)$	$TC(2,3)$	$TC(2,4)$	$TC(3,4)$
Realization j	$(c_j, p(c_j))$	$(c_j, p(c_j))$	$(c_j, p(c_j))$	$(c_j, p(c_j))$	$(c_j, p(c_j))$
1	2, 0.2	3, 0.2	2, 0.5	2, 0.5	3, 0.3
2	3, 0.3	4, 0.5	3, 0.5	4, 0.4	4, 0.3
3	4, 0.3	5, 0.3		6, 0.1	5, 0.3
4	5, 0.2				6, 0.1
$E(TC(I,j))$	3.5	4.1	2.5	3.2	4.2
$\sigma(TC(I,j))$	1.025	0.70	0.50	1.76	0.98

Table 8.2 Probability distribution function of the project cost up to node *j*

Realization	$NTC(1)$	$NTC(2)$	$NTC(3)$	$NTC(4)$	Cumulative probability For $NTC(4)$
	$(c_j, p(c_j))$	$(c_j, p(c_j))$	$(c_j, p(c_j))$	$(c_j, p(c_j))$	
1	0, 1.0	2, 0.2	7, 0.020	12, 0.003	0.003
2		3, 0.3	8, 0.100	13, 0.018	0.021
3		4, 0.3	9, 0.215	14, 0.05265	0.077
4		5, 0.2	10, 0.275	15, 0.1039	0.178
5			11, 0.235	16, 0.15455	0.332
6			12, 0.125	17, 0.1812	0.513
7			13, 0.030	18, 0.1733	0.687
8				19, 0.1377	0.824
9				20, 0.0913	0.916
10				21, 0.05065	0.966
11				22, 0.02305	0.989
12				23, 0.0082	0.998
13				24, 0.00215	0.999
14				25, 0.0003	1.000
Mean	0.00	3.50	10.10	17.5	
Standard Deviation	0.00	1.025	1.34	2.12	

Illustrative Example

Consider the AN of Fig. 8.3. The renewable and nonrenewable costs for each activity have been added resulting in the cost distribution specified in Table 8.1. To calculate the pdf of $NTC(j)$, the cost of the project up to node *j,* we use the sequential procedure presented in the section "Determining the Probability Distribution Function of the Project Cost." It starts at node one with a cost distribution $NTC(1) = (0,1)$. Then the process moves to node 2 where the pdf of $NTC(2)$ is equal to the pdf of $TC(1,2)$. The process moves to node 3, where the pdf of $NTC(3)$ is obtained by convoluting $NTC(2)$ with $TC(1,3)$; then the outcome is convoluted with $TC(2,3)$. Finally, the project cost, represented by $NTC(4)$, is obtained by convoluting $NTC(3)$ with $TC(2,4)$, and the outcome is convoluted with $TC(3,4)$. The pdf of the project cost up to node *j* for all $j \in N$ is presented in Table 8.2. The means and standard deviations are presented in the last two rows.

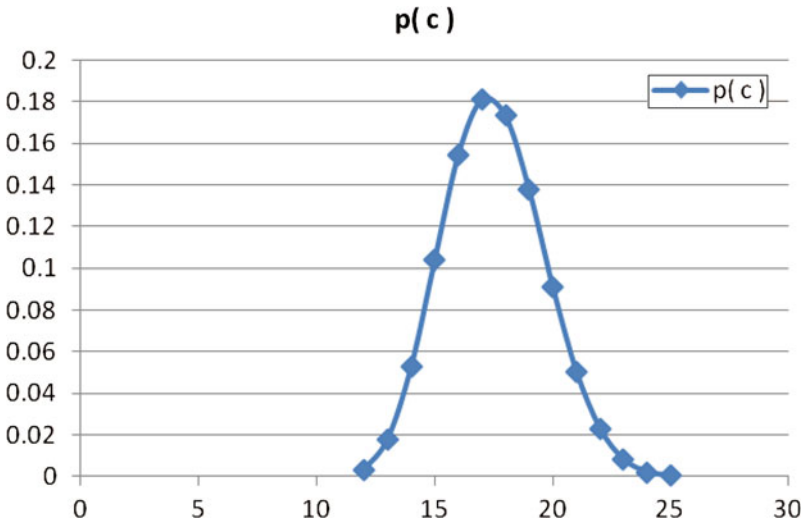


Fig. 8.6 Probability distribution function of the project cost

The project cost is presented in Table 8.2 by the ordered pairs $\{(c_j, p(c_j)\}$ in the column headed by $NTC(4)$. It is clear from these ordered pairs and their plot in Fig. 8.6 that the project cost is almost normally distributed with mean value $= 17.5$, and standard deviation $= 2.12$. Realizations of $NTC(4)$ represented by $\{c_j\}$ in column 5, and its corresponding cumulative probability distribution presented in the last column of Table 8.2 can be used to select a PB or a bidding price, and to assess the probability of completing the project within the selected budget. For instance, if the selected budget is US$ 20,000, then the probability of not completing the project within this budget is 8.4 %. If this level of risk is not acceptable, then increasing the budget by US$ 1,000.00 reduces the risk to 3.4 %.

To calculate the probability of project completion time, $F(t)$ for the PB, we first need to determine $D(i,j)$ and its pdf for each activity (i,j) ε A. The pdf of $D(i,j)$ is derived from the relation between the activity renewable resource cost and duration. It can be provided as input exactly as that of $R(i,j)$. Table 8.3 has the pdf for all $D(i,j)$. Then $F(t)$ is determined as prescribed in the section "Determining the Probability Distribution Function of the Project Duration." Due to the small size of the SAN of Fig. 8.3, the pdf for project duration is calculated exactly. It is presented in column 5 of Table 8.4; its cumulative pdf is presented in column 6 of Table 8.4. Please note that $F(t)$ is not affected by the size of the PB or its distribution over the five activities; it is affected by $\{R(i,j)\}$. The project average completion time $E(T(N)) = 9.12$, where $E(T(N))$ obtained by the PERT method is 8.80.

The distribution of the project completion time, $F(t)$, can be used to determine the project's largest probability completion time emanating from any PB. First the financial plan is derived using the method presented in the section "Calculating the Project Financial Plan," yielding the duration for each activity and the least project

Table 8.3 Probability distribution of activity duration

Activity duration	$D(1,2)$	$D(1,3)$	$D(2,3)$	$D(2,4)$	$D(3,4)$
Realization j	$(d_j, p(d_j))$	$(d_j, p(d_j))$	$(d_j, p(d_j))$	$(d_j, p(d_j))$	$(d_j, p(d_j))$
1	4, 0.2	3, 0.2	3, 0.5	6, 0.5	5, 0.3
2	3, 0.3	2, 0.5	2, 0.5	4, 0.4	4, 0.3
3	2, 0.3	1, 0.3		3, 0.1	3, 0.3
4	1, 0.2				2, 0.1
$E(D(i,j))$	2.50	1.90	2.5	4.80	3.80
$\sigma(D(i,j))$	1.025	0.70	0.50	1.327	0.98

Table 8.4 Probability distribution function of project completion time

Realization	$T(1)$	$T(2)$	$T(3)$	$T(4)$	Cumulative probability of $T(4)$
	$(t_j, p(t_j))$	$(t_j, p(t_j))$	$(t_j, p(t_j))$	$(t_j, p(t_j))$	
1	0, 1.0	4, 0.2	7, 0.10	12, 0.03	01.00
2		3, 0.3	6, 0.25	11, 0.105	0.970
3		2, 0.3	5, 0.30	10, 0.262	0.865
4		1, 0.2	4 0.25	9, 0.288	0.603
5			3, 0.10	8, 0.211	0.315
6				7, 0.0858	0.1040
7				6, 0.0169	0.0182
8				5, 0.0013	0.0013
Mean	0.00	3.50	5.00	9.12	PERT $\mu = 8.8$
Standard deviation	0.00	1.025	1.14	1.29	PERT $\sigma = 1.5$

completion time. The latest start time schedule is applied to these realizations to determine a project schedule. Then, the corresponding $F(t(N))$ is determined from the exact pdf presented in Table 8.4.

Given a PB of US$ 20,000.00, and suppose it is the total renewable resources cost. The PB does not allow for maximum funding of each of the project activities; hence, it is not possible to process each activity at minimum/crash time. How would this budget be distributed over the five activities? As discussed in the section "Calculating the Project Financial Plan," different budget allocations may lead to different activity durations, which result in different project schedule and completion time. We wish to determine the financial plan that yields the minimum completions time, which maximizes the probability of completing the project within the given budget. We use the above LP model to determine the optimal financial plan, and corresponding activity durations. In this instance the LP model consists of 13 variables and 26 constraints. In specifying constraint number 3, it is assumed that the relationship between renewable resources cost and activity duration depicted in Fig. 8.5 is continuous. In this case the slope w_{ij} is calculated from the two ordered pairs, (maximum q_{ij}, minimum y_{ij}) and (minimum q_{ij}, maximum y_{ij}) presented in the input Tables 1 and 3. Solving the LP model results in the optimal financial plan $\{q_{ij}\}$ and the corresponding activity durations $\{y_{ij}\}$. These are presented in Table 8.5 along with the probability of the level of funding for each activity.

Table 8.5 Financial plan (in US$ 1,000) and resulting activity durations

Activity	(1,2)	(1,3)	(2,3)	(2,4)	(3,4)
Activity cost, q_{ij}	5	3	3	3.715	5.285
Activity duration, y_{ij}	1	3	2	4.714	2.710
$\Pr(R(i,j) \leq q_{ij})$	1.00	0.20	1.00	0.843	0.929

Contrasting the data in rows 2 and 3 of Table 8.5 with the criticality of each activity, we notice that activities that are critical most of the time, i.e., have higher criticality indices, received either maximum or close to maximum funding; while less critical activities received less funding. Analyzing the criticality of the five activities using the duration input data of Table 8.3, we notice that project completion time is dominated by path 1-2-3-4 with an average value of 9.12 periods, and activities (1,2), (2,3), and (3,4) are the most critical; where activity (1,3) is the least critical followed by activity (2,4), which is more critical. Solution of the LP model resulted in distributing the PB of US$ 20,000 as follows: Activities (1,2) and (2,3) received maximum funding, hence have minimum/crash durations, and activity (3,4) received close to maximum (92.9 %) funding with duration close to the minimum (2.71 periods). However, activity (1,3) received minimum (20 %) funding, hence with maximum duration, where activity (2,4) received higher (84.3 %) funding and have higher duration (4.714 periods).

The optimal financial plan resulted in a realization for the project duration, $T(N) = 5.714$ periods. This corresponds to a $\Pr(T(N) \geq 5.714) = 0.981$, as shown in Table 8.4. Consequently, this financial plan provides a 98 % chance of completing the project within time, and the PB provides a 90 % of completing the project within budget.

Conclusions and Extensions

In this chapter, the problem of scheduling and financial planning in SAN projects is considered. First, issues of uncertainty in project environment are discussed. These include AN structure, renewable and nonrenewable resource requirements, price/cost of the resources, and duration of activities. Dependence of activity duration on the renewable and nonrenewable resources is explored. We conclude that while activity duration is independent of the nonrenewable resource requirements, it can be expressed as a function of the renewable resource requirements. This relationship allows connecting the project schedule and duration to project budget and financial plan.

Starting with a project network structure where the cost of renewable and nonrenewable resources are given as random variables characterized by different pdfs, an accurate and practical procedure is developed to calculate the pdf of the project cost. The procedure can be applied to all SAN sizes regardless of the underlying activity cost probability distributions. The cost pdf allows project managers to select a budget for the project and assess the risks of completing the work of the project within the

selected budget. The question then becomes how to distribute the selected budget over the activities of the project? The budget can be distributed over the activities in many different ways—each may lead to a different project schedule and completion time. What is the best distribution plan of funds over the activities? In answering this question, we rely on the relation between the two random variables: activity renewable resource requirements and activity duration. First, this relation is reversed providing the pdf for the activity duration. These probability distributions are then used to calculate the pdf of project completion time. Also activity pdfs are used with the given budget to determine the optimal financial plan. Optimality is defined by realizing the least project completion time for the given budget. This problem is formulated as an LP with the objective of minimizing the project completion time for the given budget.

Solution of the LP yields the financial plan which specifies the share of each activity in the budget, the corresponding activity duration, and the least project completion time. The resulting durations are used to construct a project schedule such as the early or late start time schedules. The pdf of project completion time can be used to assess the probability of completing the project within the budget and its corresponding project completion time.

The procedures developed above for calculating pdf of $NTC(N)$ and the optimal financial plan are yet to be applied to large SANs. It will be tested in future work. Also other random variations, such as the price of the renewable and nonrenewable resources that affect the management of stochastic projects are not investigated in this chapter. We hope to investigate the impact of these variations on project budget, financial plan, and project duration in future work.

References

Benati, S. (2006). An optimization model for stochastic project network with cash flows. *Computational Management Science, 3*(4), 271–284.

Dayanand, N., & Padman, R. (1998). On payment schedules in contractor client negotiations in projects: An overview and research issues. In Weglarz J. (Ed.), *Project scheduling: Recent models, algorithms and applications*. Boston: Kluwer Academic Publishers.

Demeulemeester, E. L., & Herroelen, W. S. (2002). *Project scheduling: A research handbook*. Boston: Kluwers's International Series; Kluwer Academic Publishers.

Dodin, B. (1984). Determining the K most critical paths in PERT networks. *Operations Research, 32*(4), 859–877.

Dodin, B. (1985). Approximating the distribution function in stochastic networks. *Computers & Operations Research, 12*(3), 251–264.

Dodin, B., & Elimam, A. A. (1997). Audit scheduling with set-up costs and overlapping relationships. *European Journal of Operational Research, 97,* 22–33.

Dodin, B., & Elmaghraby, S. E. (1985). Approximating the criticality indices of the activities in PERT networks. *Management Science, 13*(2), 207–223.

Dodin, B., & Servanci, M. (1990). Stochastic networks and the extreme value distribution. *Computers & Operations Research, 17*(4), 397–409.

Elmaghraby, E. E. (1964). An algebra for the analysis of generalized activity networks. *Management Science, 10*(3), 494–914.

Elmaghraby, S. E. (1977). *Activity networks: Project planning and control by network models*. New York: Wiley- Interscience.

Ford, L. R. Jr., & Fulkerson, D. R. (1962). *Flows in networks*. New Jersey: Princeton U. P., Princeton.

Herroelen, W., & Leus, W. S. (2005). Project scheduling under uncertainty: Survey and research potential. *European Journal of Operational Research, 165*(2), 289–306.

Neumann, K. (1999). Scheduling of projects with stochastic evolution structure. In Weglarz J. (Ed.), *Project scheduling—recent models, algorithms and applications*, (pp. 309–332). Boston: Kluwer Academic Publishers (Chapter 14).

Pritsker, A. A. B., & Sigel, C. E. (1974). The GERT IIIZ user's manual, Pritsker and Associates, Inc. 1201 Wiley Drive, West Lafayette, Indiana.

Pritsker, A. A. B., & Whitehouse, G. E. (1966). GERT: Graphical evaluation and review technique, Part II, probabilistic and industrial engineering applications. *Journal of Industrial and Engineering, 17*(6), 293–301.

Sobel, M. J., Szmerekovsky, J. G., & Tilson, V. (2009). Scheduling projects with stochastic activity duration to maximize net present value. *European Journal of Operational Research, 198*(3), 697–707.

Wallace, S. W. (1989). Bounding the expected time- cost curve for stochastic pert networks from below. *Operations Research Letters, 8*(2), 89–94.

Wiesemann, W., Huhn, D., & Rustem, B. (2010). Maximizing the net present value of a project under uncertainty. *European Journal of Operational Research, 202*(2), 356–367.

Wollmer, R. D. (1985). Critical Path Planning Under Uncertainty. *Mathematical Programming Study, 25,* 164–171.

Chapter 9
A Risk Integrated Methodology for Project Planning Under Uncertainty

Willy Herroelen

Introduction

Project management involves the planning, scheduling, and control of project activities to achieve performance, cost, and time objectives for a given scope of work while using resources in an efficient and effective manner (Demeulemeester and Herroelen 2002; Demeulemeester et al. 2007). Project scheduling and control has been the subject of extensive research efforts leading to an impressive body of literature (Demeulemeester and Herroelen 2002; Elmaghraby 1977) while a wide variety of commercialized software packages have been released and put to use in practical project settings. Despite all these efforts, numerous publications have documented projects that went severely over budget or dragged on long past their originally scheduled completion date (see, e.g., Flyvbjerg et al. 2003; Standish Group 2004).

The planning, scheduling, and control of projects under stochastic conditions is indeed a complex and challenging task, involving decisions at the strategic, tactical, and operational levels (Leus et al. 2007). At the *strategic level*, the long-term strategic resource planning decisions to be made by top management include the selection of projects of major strategic importance, major resource investments, and project financing. The *tactical decisions* have to do with project selection/acceptance, the allocation of company capacity, and reliable due date setting. Detailed scheduling and resource allocation decisions have to be made at the *operational level*. Success in managing a project requires a complete and realistic project baseline schedule that represents the project plan. Project control implies the deployment of corrective actions when the project baseline schedule is rendered infeasible by the disruptive events that may occur during actual project execution.

W. Herroelen (✉)
Research Center for Operations Management, Department of Decision Sciences and Information Management, Faculty of Business and Economics, KU Leuven, Leuven, Belgium
e-mail: Willy.Herroelen@econ.kuleuven.be

P. S. Pulat et al. (eds.), *Essays in Production, Project Planning and Scheduling,* International Series in Operations Research & Management Science 200,
DOI 10.1007/978-1-4614-9056-2_9,

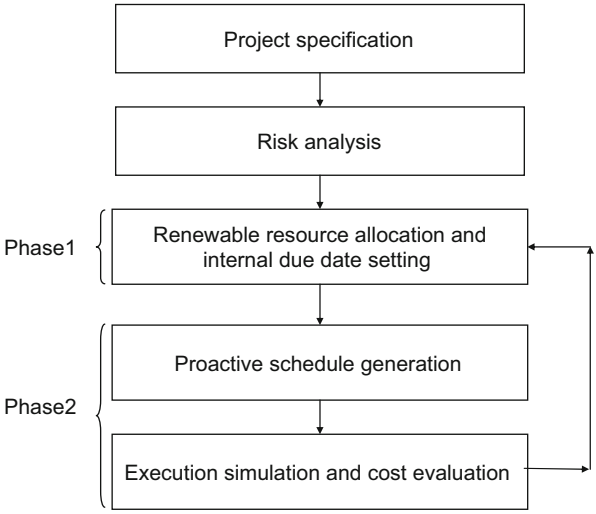


Fig. 9.1 Iterative two-phase project planning procedure

In this chapter, we describe a risk integrated methodology for tactical and operational project planning under uncertainty. The methodology integrates *quantitative risk analysis* with reliable *proactive/reactive project scheduling* procedures.

Project risk management aims to provide insight into the risk profile of a project so as to facilitate the mitigation of the impact of risks on project objectives such as budget and time. Effective risk management requires a risk analysis process that is scientifically sound and that is supported by reliable quantitative techniques. In this chapter, we consider risks that impact both the duration of project activities and the availability of renewable resources. The traditional practice of *quantitative risk analysis* assumes that the duration of a project activity captures all uncertainty that originates from the occurrence of risks; i.e., uncertainty is commonly placed on activity durations using three-point estimates of low, most likely, and long activity durations and selecting appropriate probability distributions (Hulett 2009). Contrary to this *activity-based approach*, we opt for a *risk-driven approach* in which the impact of each identified risk is assessed individually and is subsequently mapped to the duration of an activity. The probability distribution for a project activity is developed based on the probability and impact of all the risks that are assigned to it and their impact on its duration if they do occur. In doing so, the uncertainty is associated with each risk, not with the project activity that is affected by risks (Creemers et al. 2010; Schatteman et al. 2008). Quantitative project risk analysis is the subject of the Section "The Need for Quantitative Risk Analysis." It provides crucial input for the generation of robust baseline schedules that are adequately protected against possible disruptions that may occur during project execution.

The risk integrated methodology we describe in this chapter consists of two iterative phases (Fig. 9.1).

The *project specification* input of phase 1 consists of an activity-on-the-node network for the project $G(N, A)$ and the externally imposed customer project due

date δ. For each activity i in the project network, we assume that the project planning team can come up with a single point estimate d_i of the activity duration, an estimate of the per period requirement r_{ik} for each renewable resource type $k \in K$, and a so-called inflexibility weight w_i. The inflexibility weight represents the marginal cost of deviating from the planned starting time s_i of an activity during the execution of the project (Leus and Herroelen 2004). A small activity weight reflects high scheduling flexibility: it does not "cost" that much if the actually realized activity starting time during schedule execution differs from the planned activity starting time in the baseline schedule. Activities that are to be executed by resources with ample availability, for example, will be given a relatively small flexibility weight, reflecting project management's view that their rescheduling cost is relatively small. A heavy activity weight denotes low scheduling flexibility, reflecting management's view that deviations from planned activity starting times are deemed very costly for the organization due, for example, to the high penalty costs that are incurred when individual milestones or the project due date are not met. Activities that use scarce resources or rely on subcontractors that are in a strong bargaining position will also receive a heavy weight as it is preferable for the starting time of these activities to be kept fixed in time as much as possible.

The project should be the subject of a *qualitative and quantitative risk analysis* allowing for the identification of the major project risks and a quantification of both their probability of occurrence and their impact.

During *phase 1* decisions have to be made about the number of regular renewable resource units a_k of type $k \in K$ to be allocated to the project and the so-called internal project due date δ'. These decisions will serve as input for the robust baseline schedule generation and schedule execution problem that is solved in the second phase. Up to the internal due date δ', activities can be performed using the allocated regular renewable resource units. In case the project takes longer than δ', we assume that irregular emergency resource capacity can be hired at a cost. The internal due date δ' is bounded from below by the critical path length CP of the project and bounded from above by the externally imposed customer due date δ, $\mathrm{CP} \leq \delta' \leq \delta$. The decisions to be made in phase 1 can then be represented by means of a $(|K| + 1)$ vector $sol = (a_1, ..., a_K, \delta')$, corresponding to $|K|$ resource allocation decisions and one internal due date decision. An effective procedure for setting the renewable resource levels and the internal due date is introduced in the Section "Resource Allocation and Internal Due Date Setting."

Phase 2 of the integrated procedure implements a proactive/reactive schedule generation methodology, whose components have been heavily researched over the past few years (Herroelen 2007). A proactive baseline schedule can be generated using a combination of resource buffering, minimal makespan scheduling, and time buffering. The proactive baseline scheduling system we propose aims at generating a baseline schedule that is precedence and resource feasible and that is effectively protected (using time and/or resource buffers) in an effort to achieve timely project completion and schedule stability. The proactive schedule is generated using a two-step procedure. In the first step, a precedence and resource feasible schedule is generated with acceptable project duration. In a second step, the schedule is to be

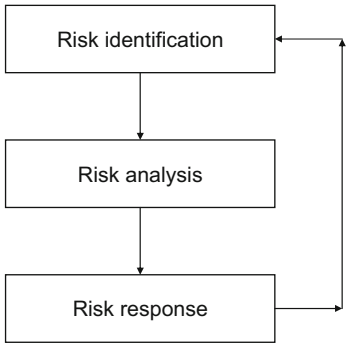


Fig. 9.2 Iterative risk management process

protected against disruptions that may occur during the execution of the project. This is done by inserting buffers into the schedule (Demeulemeester and Herroelen 2010). Resource and time buffering form the subject of the Section "Robust Schedule Generation."

The proactive schedule is then to be used as a guideline during project execution. A sufficient number of schedule executions are simulated using the stochastic information about possible resource breakdowns and activity durations. When the built-in protection of the baseline schedule is no longer sufficient and the schedule becomes infeasible, schedule feasibility has to be restored by deploying a proper reactive scheduling procedure. The procedure has to decide whether the schedule is repaired by (a) preempting one or more of the active activities and by rescheduling activities that are planned in the future and that are affected by the preemption through precedence relations or the use of shared resources, or (b) by hiring irregular renewable resource capacity at an additional irregular capacity cost (Deblaere et al. 2011; Van de Vonder et al. 2007). This allows for the calculation of the expected values of the irregular capacity costs and the schedule instability costs. The *feedback loop* shown in Fig. 9.1 then involves the use of the mean-variance function of the schedule execution costs to evaluate the resource and internal due date decisions made in phase 1. Schedule execution and reactive scheduling form the topic of the Section "Schedule Execution and Reactive Scheduling."

The Need for Quantitative Risk Analysis

The need to manage uncertainty is inherent in most projects. The Project Management Institute defines a *project risk* as "an uncertain event or condition that, if it occurs, has a positive or negative effect on at least one project objective" (PMI 2008). The essential purpose of *risk management* is to improve project performance via systematic identification, appraisal, and management of project-related risk (Chapman and Ward 1997). Risk management (PMI 2008) is an iterative process involving risk detection, risk analysis, and risk response (Fig. 9.2).

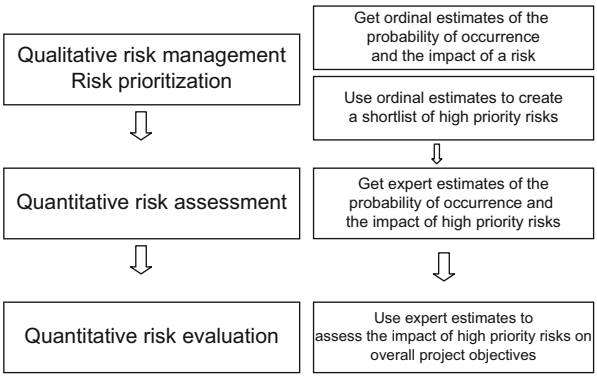


Fig. 9.3 Overview of the risk analysis process. (Creemers et al. 2010)

Risk Identification

The risk identification process involves the identification of the major risks that may affect the project objectives. This implies that the roots of the risks must be identified rather than the risk symptoms. Useful tools in the risk identification process are *risk check lists* containing a structured overview of all the risks that may occur on the project. The Belgian Building Research Institute, for example, maintains a *risk management database* containing standardized risk checklists describing all the risks that have occurred in the past on different types of construction projects (Schatteman et al. 2008; Van de Vonder et al. 2010).

Risk Analysis

Once the risks have been identified, they should be properly analyzed. The risk analysis process consists of three major phases (Fig. 9.3): qualitative risk management and risk prioritization, quantitative risk assessment, and quantitative risk evaluation.

Qualitative risk analysis relies on an ordinal scale to assign a score (for example, 1—low, 2—medium, and 3—high) for both the probability of occurrence of the risk (P) and its impact on the project objectives (I). This allows the risks to be prioritized based on their total score ($P \times I$), the priority risks being the risks with the highest score.

Both the risk identification and the qualitative risk analysis provide the input for the so-called *risk register*. For each identified risk, the risk register contains a clear description of the risk, its probability of occurrence (P), its impact score (I), the total risk score ($P \times I$), as well as the proactive and reactive measures taken to respond to the risk (see "Risk Response"). The high priority risks should then be the subject of a *quantitative risk analysis*.

Quantitative Risk Analysis

Quantitative risk assessment is the procedure by which experts provide detailed estimates of the probability of occurrence and the impact of high priority risks. These estimates are then used in the quantitative risk evaluation procedure to analyze the impact of the short-listed risks on the overall project objectives. In the following, we rely on examples from construction industry practice to clarify the main issues involved.

Activity Groups

Quantitative risk assessment is commonly considered by practitioners to be a rather time-consuming procedure. We follow Schatteman et al. (2008) in suggesting the use of so-called *activity groups*, i.e., groups of activities that share common risks. In a construction project, for example, the activity group "masonry" may contain all the masonry activities that are subject to similar risks (e.g., the risk of weather delay). Obviously, the same risk may have an impact on different activity groups. For example, weather delay may not only affect the activity group "masonry" but may also affect the activity group "roofing." Risks can then be assessed at the activity group level, rather than at the level of each of the many individual project activities.

Risk Impact Types

In this chapter, we are interested in the so-called project schedule risks, the risks that affect the project schedule. We follow Van de Vonder et al. (2010) in identifying different impact types of project scheduling risks: (a) proportional or fixed impact, (b) start time delay, and (c) breakdown.

Proportional or Fixed Activity Duration Impact

Risks may affect the duration of project activities. The risk "bad soil quality," for example, may have a proportional impact on the duration of a construction activity in the activity group "excavation." Proportional risk impacts may be assessed by first asking the risk expert to estimate the probability (percentage) that the risk will impact an activity group (often the impact will be 100 %) and to provide an estimate of the impact on the duration of an activity in the activity group using a pessimistic, most probable, and optimistic estimate.

Similar input is required for risks having a fixed impact on the duration of an activity (for example, the need to perform an additional stability study). The probability percentage of fixed impact risks will rarely be 100 %.

Starting Time Delays

Risks may cause a delay in the planned starting time of activities. The risk "late material delivery," for example, may cause a delay in the start of one or more construction activities in the activity group "reinforcement work." Again the expert can be asked to provide the percentage probability of occurrence of the impact together with three-point starting time delay estimates.

Breakdowns

The risk type "breakdown" may be used to model breakdowns in the use of renewable resources (for example, machine defects). In this case, the expert may be asked to provide an estimate of the mean time to failure (MTTF) and the mean time to repair (MTTR) for the particular renewable resource types.

Risk Response

Having identified the risk exposure and quantified its potential impact, proper actions should be identified to respond to the risks. Risk response may include risk avoidance (for example, performing the activity using an alternative activity execution mode that does not contain the risk), risk reduction (taking actions to reduce the risk by reducing its probability of occurrence and/or its impact), risk transfers (for example, passing on the risk to a third party by outsourcing an activity), and *risk anticipation through proactive/reactive scheduling*. The latter approach constitutes a crucial component of our suggested risk integrated project planning methodology.

Resource Allocation and Internal Due Date Setting

During the first phase of the integrated procedure, a decision has to be made on the level of the renewable resource capacity a_k of resource type $k \in K$ to allocate to the project from the project start time $t = 0$ up to the internal project due date $t = \delta'$. As already mentioned above, the internal due date δ' is bounded from below by the critical path length CP of the project and bounded from above by the externally imposed due date δ, $CP \leq \delta' \leq \delta$. The decisions that need to be made in phase 1 can then be represented by means of a $(|K|+1)$ vector $sol = (a_1, ..., a_K, \delta')$, corresponding to $|K|$ resource allocation decisions and one internal due date decision.

Lambrechts (2007) has developed an effective tabu search procedure to optimize the resource allocation and due date setting decisions in a computationally efficient manner. The author represents a solution to the resource allocation and due date setting problem by means of the vector $sol = (a_1, ..., a_K, \delta')$ and suggests evaluating

the value of the objective function z corresponding to such a vector using the simulation procedure to be described below as a crucial step of phase 2 of the integrated procedure.

The objective function z is defined as z = average(TC), the average value of the total cost function TC obtained over all simulation scenarios, where TC is defined as the sum of the *regular renewable resource costs*, the *expected irregular renewable resource costs*, and the so-called *expected schedule instability costs*. More formally,

$$TC = \delta' \sum_{k} a_k c_k^{reg} + E\left(\sum_{k,t} A_{kt}^{irreg} c_k^{irreg}\right) + \sum_{i \in N} w_i E\,|S_i - s_i| \qquad (9.1)$$

with

δ'	The internal project due date
a_k	The amount of renewable resource type $k \in K$ allocated to the project
c_k^{reg}	the per period cost of a regular unit of renewable resource type k
A_{kt}^{irreg}	The number of irregular units of renewable resource type k hired in period t
c_k^{irreg}	The per period cost of an irregular renewable resource unit of type k
w_i	The activity flexibility weight, i.e., the per period start time deviation cost of activity i
s_i	The planned starting time of activity i in the baseline schedule
S_i	The actually realized starting time of activity i during schedule execution
E	The expectation operator

In other words, we assume that the regular resource capacity a_k is allocated to the project prior to the start of project execution at a regular resource cost c_k^{reg} per period. In case the project takes longer than the internally set due date δ', or when a schedule infeasibility occurs during project execution, we assume that additional irregular emergency capacity A_{kt}^{irreg} can be hired on a per period basis at a cost c_k^{irreg} per period. The last term in Eq. (9.1) represents the so-called schedule instability costs relying on the notion of *schedule stability* or *solution robustness*. Schedule stability refers to the difference between a project baseline schedule and the actually realized schedule during project execution. Leus and Herroelen (2004) suggest measuring schedule stability by the weighted sum of the absolute differences between the planned activity starting times s_i in the baseline schedule and the actually realized activity start times S_i during project execution. As mentioned earlier, the weights w_i represent the activity disruption cost per time unit, i.e., the nonnegative cost per time unit overrun or underrun on the start time of activity i. This disruption cost reflects either the difficulty in shifting the booked time window on the required resources or the difficulty in obtaining the required resources, or the importance of on-time performance of the activity.

The exact evaluation of Eq. (9.1) is unrealistic; computing the expected project duration and the probability that a project without resource constraints is finished by a given time instant, assuming an early start schedule—the Program Evaluation and Review Technique (PERT) problem—is already #P complete (Hagstrom 1988). Hence we opt to obtain the value z = average(TC) through a number of simulation

runs, performed in phase 2, using the same baseline schedule, the same resource allocations, and the same internal due date.

In the *tabu search* procedure of Lambrechts (2007), a move is defined as a unit increase and decrease of every element of *sol*. A move is declared admissible if the combination of the resource allocation and the internal due date resulting from the move are feasible. The admissible move yielding the best value z' obtained in the current iteration of the tabu search is executed if it is not declared tabu. Whenever a move is accepted, the opposite of the move is declared tabu for the next $|K|$ iterations. The tabu restriction can be overridden when the move corresponds to a solution value that improves the best overall solution value z^* found up to the current iteration. The starting value of δ' is set to $\delta' = \frac{CP+\delta}{2}$, where CP denotes the critical path length and the starting value of a_k is set to $a_k = \frac{a_k^{ESS}+a_k^{LEV}}{2}, \forall k$, where $a_k^{ESS} = \max_k(\sum_{i \in B_t} r_{ik})$, with B_t the set of activities in progress during period t in the early start schedule *ESS*, and $a_k^{LEV} = \max_t(\sum_{i \in B_t} r_{ik})$ the corresponding maximum per period resource requirement in a leveled schedule, obtained by a reliable resource leveling procedure (see, e.g., Gather et al. 2010; Neumann and Zimmermann 2000). As such, the starting values are taken in the middle of two extremes for the planned project duration and resource capacity, corresponding to a schedule with smallest possible makespan δ' and high peak resource requirement a_k^{ESS}, and a longer schedule with makespan δ with lower, leveled per period resource usage a_k^{LEV}.

Robust Schedule Generation

Phase 2 of the integrated project planning procedure implements a proactive/reactive schedule generation methodology. The proactive/reactive project scheduling literature (Herroelen 2007; Herroelen and Leus 2005) suggests the generation of a proactive project schedule using a combination of resource buffering, minimal makespan scheduling, and time buffering.

The Generation of a Resource-Buffered Schedule

During project execution, renewable resources may be subject to breakdown causing the planned baseline schedule to become infeasible. The proactive scheduling strategy may involve the use of resource buffers to protect the baseline schedule against resource disruptions.

Resource buffering can be achieved by including so-called *resource slack*. This means that the project is planned using a regular renewable resource availability a_k^{reg*} that is lower than the regular resource availability a_k determined in phase 1 of the integrated project planning procedure. The required size of the resulting resource buffers will depend on the probability distribution of the resource availabilities.

It can be shown (Ross 1983) that a single renewable resource unit of resource type k with independently and identically distributed times between failure X_k and independently and identically distributed repair times Y_k has a stationary availability (the probability that the resource is active at a time in the future) given by

$$A_k = \frac{E(X_k)}{E(X_k) + E(Y_k)}. \tag{9.2}$$

Remember that proper estimates of MTTF $E(X_k)$ and MTTR $E(Y_k)$ are to be obtained from the risk expert during the quantitative risk analysis procedure. Writing $E(X_k) = 1/\lambda_k = MTTF_k$, $E(Y_k) = 1/\mu_k = MTTR_k$, and $\rho_k = MTTR_k/MTTF_k$, we have $A_k = 1/(1 + \rho_k)$. The probability $P(\mathbf{a}_k = j)$ can now be written as

$$P(a_k = j) = \binom{\mathbf{a}_k}{j} (A_k)^j (1 - A_k)^{a_k - j} = \binom{a_k}{j} \frac{\rho_k^{a_k - j}}{(1 + \rho_k)^{a_k}}.$$

The expected value (taking breakdowns into account) of the resource availability in the steady state for renewable resource type $k \in K$ can now be written as

$$E(\mathbf{a}_k) = \left\lfloor \sum_{m=0}^{a_k} m \times P(A_k = m) \right\rfloor. \tag{9.3}$$

This value can be used as the buffered resource availability a_k^{reg*}. In case this buffered availability is smaller than the maximum resource requirement $\max_{i \in N} r_{ik}$, its value is augmented until the activity with the highest resource demand for resource type k can be executed.

The initial project baseline schedule can now be generated using any exact or heuristic procedure for solving the well-known *resource-constrained project scheduling problem* (RCPSP), involving the determination of the activity start times subject to the precedence and renewable resource constraints under the minimal makespan objective (Hartmann and Briskorn 2010; Herroelen 2005). If the resource-buffered schedule violates the internal project due date δ', the most constrained resource type is identified and its availability is progressively increased up to the maximum (original) availability a_k. The schedule generation procedure is then reexecuted until the due date δ' is met. The most constraining resource type is defined as the resource type that leads to the highest decrease in schedule makespan when its buffered availability is increased by one unit. The resource type with the smallest deviation between the expected resource availability and the adjusted buffered availability is used as a tiebreaker.

Time Buffering

Translating Resource Uncertainty into Time Uncertainty

The resource-buffered (minimal makespan) schedule can be the subject of time buffering. Lambrechts et al. (2011) have shown that, under realistic assumptions,

resource availability uncertainty can be effectively translated into activity duration uncertainty. When a resource infeasibility occurs and the decision is made to hire no irregular renewable resource capacity, activities that were in progress at the time of a resource breakdown are preempted. The authors make a distinction between a preempt-repeat and a preempt-resume environment. In a *preempt-repeat environment*, preempted activities have to be restarted from scratch, while in a *preempt-resume environment*, preempted activities may be restarted from the point where execution halted.

Lambrechts et al. (2011) prove that in a *preempt-repeat* environment with fixed resource allocations, the expected activity duration extension due to breakdowns for an activity i with planned duration d_{i} and renewable resource usage r_{ik} of renewable resource type k for which the time to failure of each resource unit is exponentially distributed with parameter λ_{k} and the time to repair is also exponentially distributed with parameter μ_{k}, is given by

$$E[\gamma_i] = \frac{\psi_i}{(1-\psi_i)(\sum_k \lambda_k r_{ik})}\left(1+\sum_k \frac{\lambda_k r_{ik}}{\mu_k}\right) - d_i, \tag{9.4}$$

where $\psi_i = 1 - e^{-d_i \sum_k \lambda_k r_{ik}}$.

For a *preempt-resume* environment, Lambrechts et al. (2011) prove that the expected duration extension due to resource breakdowns for an activity i with planned duration d_{i} and renewable resource usage r_{ik} of renewable resource type k for which the time to failure of each resource unit is exponentially distributed with parameter λ_k and the time to repair is also exponentially distributed with parameter μ_{k} is given by

$$E[\gamma_i] = d_i \sum_k \frac{\lambda_k r_{ik}}{\mu_k}. \tag{9.5}$$

Time Buffering Procedures

The nice thing about the results derived above is that both time and resource uncertainty can now be effectively dealt with by proactive/reactive scheduling procedures that were originally developed to cope with activity duration uncertainty. A wide variety of exact and suboptimal procedures have been developed and evaluated on their effectiveness and efficiency (Van de Vonder et al. 2008). Despite its simplicity, the so-called starting time criticality (STC) heuristic, developed by Van de Vonder et al. (2006) obtains excellent results.

The iterative *STC heuristic* relies on information provided by the activity weights w_{i} and the variance structure of the activity durations. The underlying idea is to take a resource-buffered schedule as input and iteratively create intermediate schedules by adding a one-unit time buffer in front of that activity that is the most starting time critical in the current intermediate schedule. The *starting time criticality* $stc(i)$ of

activity i in the current schedule is defined as

$$stc(i) = P(S_i > s_i) \times w_i = \xi_i \times w_i, \tag{9.6}$$

where ξ_i denotes the probability that activity i cannot be started at its scheduled starting time s_i. Activities are listed in decreasing order of the $stc(i)$, breaking ties arbitrarily. The list is scanned and the time buffer to be placed in front of the current activity from the list is augmented by one time unit. If the resulting schedule does not violate the project due date and results in a lower surrogate stability cost $\sum_i stc(i)$, the schedule serves as input for the next iteration step. If not, the procedure takes the next activity in the list. Whenever the procedure reaches an activity i with $stc(i) = 0$ (by definition, this is the case for all activities j with a planned starting time $s_j = 0$ in the baseline schedule) and no further improvement is found, the procedure terminates with a local optimum.

Schedule Execution and Reactive Scheduling

At the start of project execution, the activity durations are set to the input durations d_i and a regular renewable resource capacity of a_k units per period is allocated to the project from its start at time $t = 0$ up to time $t = \delta'$ (for periods $t > \delta'$, regular resource capacity is set to 0). During the simulation, at the start of every time period t, the real activity duration is updated for the activities starting at time t and the real resource availability a_{kt}^{obs} in time period t becomes known. In case a resource breakdown is of such a magnitude that the real resource availability is insufficient to satisfy the resource requirement of the activities that are active at time t $\left(\exists k : \sum_{i:i\in B_t} r_{ik} > a_{kt}^{obs}\right)$, the following resource conflict resolution procedure is used. For each activity $i \in B_t$, it has to be decided whether to preempt that activity or to keep it at its current starting time. The difference between the total resource requirements of the nonpreempted activities and the observed renewable resource availability a_{kt}^{obs} at time t then needs to be filled by hiring irregular resource capacity:

$$\forall k : a_{kt}^{irreg} = \max(0, \sum_{i\in B_t:\ i \text{ not preempted}} r_{ik} - a_{kt}^{obs}).$$

The resource conflict resolution procedure uses full enumeration to determine which activities have to be preempted, yielding the lowest combination of additional instability costs and additional irregular capacity costs.

Rescheduling may be done using one of the existing reactive scheduling procedures developed in the literature (Lambrechts et al. 2008). The *scheduled order repair heuristic*, for example, is a list scheduling heuristic that reschedules the activities in the order dictated by the baseline schedule (using the lowest activity number as a tiebreaker), while taking into account the reduced resource availabilities. When a

disruption occurs in time period t^*, a priority list L is created including the activities that are not yet completed at t^*, ordered in increasing order of their baseline start time s_i. This priority list is then decoded into a feasible schedule using a serial schedule generation scheme that takes the known resource availabilities a'_{kt} up to the current time period t^* into account. Activities selected from the list are started as soon as possible. For activities $i \in B_{t*}$, the procedure first tries the current time t^*. If this is infeasible, the procedure tries the next time period $(t^* + 1)$ and subsequent time periods if necessary. For the activities not yet started, it is only necessary to consider the earliest precedence feasible starting time.

During each simulation run, the resource capacity costs and the schedule instability costs are calculated. When a sufficient number of simulation runs have been performed, the mean-variance function of the schedule execution cost TC is calculated. A solution is stored if its cost is lower than the best solution obtained so far and the feedback loop to the tabu search procedure of phase 1, shown in Fig. 9.1, can be performed, allowing for an update of the resource availabilities a_k and the due date δ'.

The last generated schedule can then be used as the baseline schedule during actual project execution.

Conclusions

The objective of this chapter was to describe the working principles of an integrated procedure for the planning of projects under time and renewable resource uncertainty. The integrated procedure heavily relies on quantitative schedule risk analysis and involves two phases to be performed iteratively. In phase 1, decisions are made about the amount of regular and irregular renewable resource capacities to be allocated to the project. In phase 2, a robust baseline schedule is constructed based on the decisions made in phase 1 and the output of the quantitative schedule risk analysis. The execution of this robust baseline schedule is then simulated a sufficient number of times for varying uncertainty scenarios allowing for the computation of the schedule execution costs composed of the regular and irregular renewable resource costs and the schedule instability costs. The mean-variance function of the schedule execution costs is then used to evaluate and eventually update the resource and due date factor decisions that were made in phase 1. The final proactive project schedule can then be used as a robust baseline schedule during actual project execution.

Acknowledgement We are much indebted to Olivier Lambrechts, who developed the tabu search procedure for optimizing the resource allocation and due date setting decisions and extensively experimented with the two-phase procedure (Lambrechts 2007). We also benefited from the insights developed during the two research projects on *Risk Management in the Construction Industry* I-II, sponsored by the *Flemish Government Agency for Innovation by Science and Technology*.

References

Chapman, C., & Ward, S. (1997). *Project risk management—Processes, techniques and insights*. New York: Wiley.

Creemers, S., Demeulemeester, E., & Van de Vonder, S. (2010). A new approach for quantitative risk analysis, Research Report KBI 1029, Department of Decision Sciences and Information Management, K.U.Leuven, Belgium.

Deblaere, F., Demeulemeester, E., & Herroelen, W. (2011). Reactive scheduling in the multi-mode RCPSP. *Computers & Operations Research, 38*(1), 63–74.

Demeulemeester, E., Deblaere, F., Herbots, J., Lambrechts, O., & Van de Vonder, S. (2007). A multi-level approach to project management under uncertainty. *Tijdschrift voor Economie en Management, LII*(3), 391–409.

Demeulemeester, E., & Herroelen, W. (2002). *Project scheduling—A research handbook. International Series in Operations Research & Management Science*. New York: Springer-Verlag.

Demeulemeester, E., & Herroelen, W. (2010). Robust project scheduling. *Foundations and Trends in Technology, Information and Operations Management, 3*(3-4), 201–376.

Elmaghraby, S. E. E. (1977). *Activity networks—Project planning and control by network models*. New York: Wiley.

Flyvbjerg, B., Bruzelius, N., & Rothengatter, W. (2003). *Megaprojects and risk: An anatomy of ambition*. Cambridge: Cambridge University Press.

Gather, T., Zimmermann, J., & Bartels, J. H. (2010). Exact methods for the resource leveling problem. *Journal of Scheduling*, to appear.

Hagstrom, J. N. (1988). Computational complexity of PERT problems. *Networks, 18*(2), 139–147.

Hartmann, S., & Briskorn, D. (2010). A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research, 207*(1), 1–14.

Herroelen, W. (2005). Project scheduling-Theory and practice. *Production and Operations Management, 14,* 413–432.

Herroelen, W. (2007). Generating robust project baseline schedules. *Tutorials In Operations Research-OR Tools and Applications: Glimpses of Future Technologies*, Institute for Operations Research and the Management Sciences (INFORMS). *Chapter, 7,* 124–144.

Herroelen, W., & Leus, R. (2005). Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research, 165*(2), 289–306.

Hulett, D. (2009). *Practical schedule risk analysis*. Farnham: Gower Publishing Ltd.

Lambrechts, O. (2007). Robust project scheduling subject to resource breakdowns, Ph. D. thesis, K.U.Leuven.

Lambrechts, O., Demeulemeester, E., & Herroelen, W. (2008). Proactive and reactive strategies for resource-constrained project scheduling with uncertain resource availabilities. *Journal of Scheduling, 11*(2), 121–136.

Lambrechts, O., Demeulemeester, E., & Herroelen, W. (2011). Time-slack based techniques for robust project scheduling subject to resource uncertainty. *Annals of Operations Research, 186*(1), 443–464.

Leus, R., & Herroelen, W. (2004). Stability and resource allocation in project planning. *IIE Transactions, 36*(7), 667–682.

Leus, R., Hans, E., Herroelen, W., & Wullink, G. (2007). A hierarchical approach to multiproject planning under uncertainty. *Omega, 35*(5), 563–577.

Neumann, K., & Zimmermann, J. (2000). Procedures for resource leveling and net present value problems in project scheduling with temporal and resource constraints. *European Journal of Operational Research, 127,* 425–443.

PMI (2008). *A Guide to the Project Management Body of Knowledge* (PMBOK® Guide), Fourth Edition, Project Management Institute.

Ross, S. (1983). *Stochastic processes*. New York: John Wiley.

Schatteman, D., Herroelen, W., Boone, A., & Van de Vonder, S. (2008). A methodology of integrated risk management and proactive scheduling of construction projects. *Journal of Construction Engineering and Management, 134*(11), 885–895.

Standish Group International, 2004, CHAOS Report in year 2004: CHAOS chronicles, http://www.standishgroup.com/sample_research.

Van de Vonder, S., Demeulemeester, E., & Herroelen, W. (2006). Proactive-reactive project scheduling trade-offs and procedures. In Jozefowska, J. & Weglarz, J. (Eds.), *Perspectives in Modern Project Scheduling*, Springer's International Series in Operations Research and Management Science, Chapter 2, 25–51.

Van de Vonder, S., Ballestin, F., Demeulemeester, E., & Herroelen, W. (2007). Heuristic procedures for reactive project scheduling. *Computers and Industrial Engineering, 52*(1), 11–28.

Van de Vonder, S., Demeulemeester, E., & Herroelen, W. (2008). Proactive heuristic procedures for robust project scheduling: An experimental analysis. *European Journal of Operational Research, 189*(3), 723–733.

Van de Vonder, S., Creemers, S., & Meulyzer, E. (2010). *Risicomanagement in de bouw*, WTCB-IWT-K.U.Leuven.

Chapter 10
Dynamic Resource Constrained Multi-Project Scheduling Problem with Weighted Earliness/Tardiness Costs

M. Berke Pamay, Kerem Bülbül and Gündüz Ulusoy

Introduction and Motivation

Building a high-rise building in a business district, or manufacturing a special purpose machine for a customer, or organizing a concert all involve various tasks to be completed in a systematic order to reach a final target. The project management approach can be applied to any of these endeavors as a decision tool to improve efficiency. This wide range of applications makes projects a common structure for organizing work. Besides internal company activities like maintenance or research and development (R&D), project-based companies such as in construction, make-to-order manufacturing, or software development industries all present examples of multi-project management applications. Payne (1995) reports that up to 90 % of the value of all projects occur in a multi-project context. Typically, multiple projects share common resource pools whose capacities are not sufficient to support all project activities at the same time, leading to the resource-constrained multi-project scheduling problem (RCMPSP), which focuses on scheduling multiple projects while using available resource profiles and satisfying the precedence constraints to optimize the desired objective function.

Most project scheduling models are of static nature, where schedules are based on the data that are available before the solution procedure and the effects of unexpected events such as disruptions in projects, arrival of new projects, and changes in resource availability are not considered. Herbots et. al. (2007) point out that static approaches are less realistic and a revision of the existing schedule might be required, especially when dealing with external projects. The main reason behind the

G. Ulusoy (✉) · K. Bülbül · M. B. Pamay
Manufacturing Systems Engineering Program, Sabancı University,
Orhanlı, Tuzla, 34956 Istanbul, Turkey
e-mail: gunduz@sabanciuniv.edu

K. Bülbül
e-mail: bulbul@sabanciuniv.edu

M. B. Pamay
e-mail: mberke@sabanciuniv.edu

P. S. Pulat et al. (eds.), *Essays in Production, Project Planning and Scheduling,*
International Series in Operations Research & Management Science 200,
DOI 10.1007/978-1-4614-9056-2_10, © Springer Science+Business Media New York 2014

dynamic nature of external projects lies in the complex network of business relations between companies. Cooperation with other organizations, subcontractors, and customers is a common way of doing business resulting in a multi-project environment. Anticipating the total project load in the future becomes almost impossible for the companies as their project portfolios change over time. Therefore, models dealing with the dynamic multi-project environments become critical to provide realistic decision instruments. The model presented in this chapter is an attempt to partially fill the need for creating effective decision tools to be employed in dynamic multi-project environments; in particular, if the events have to be handled case by case with low visibility into the future.

Selecting the appropriate performance measure is essential to reflect reality. Minimizing the project completion time is a popular performance measure focusing on the effective usage of resources as well as the responsiveness of a company to its market. However, dynamic decision processes involve progressive schedule generation steps. Therefore, the starting times of the activities as well as the resource allocation decisions in the schedule can change dramatically while minimizing the makespan for the modified data sets. Handling these changes effectively requires organizational responsiveness—a crucial competitive capability. Drastic updates to the schedule and resource commitments may lead to significant organizational overhead and may not be desirable or even possible. Therefore, focusing on deviations from the baseline schedule in subsequent scheduling activities can help absorb any negative ripple effects of the dynamic events in the organization. As a result, punishing both earliness and tardiness, directly or indirectly, forces the companies to schedule all activities on time or as close as possible to their due dates or completion times in the baseline schedule.

No baseline schedule exists for a newly arriving project, and the main concern for such a project is quoting a due date that trades off its potential revenue against the impact of accommodating it in the baseline schedule. Yang and Sum (1997) state that a negotiation procedure between the client (project owner) and the contractor is generally adopted in the decision process to handle this problem. The client wants the project to be completed as soon as possible and might even offer an increased payment for an earlier completion time as an incentive for the contractor. From the perspective of the contractor, the new project generates more revenue if completed earlier; however, the risk of paying late delivery costs for existing commitments has to be mitigated by pushing the new project toward the end of the existing schedule at the expense of forfeiting some of the potential revenue. The mathematical model we propose in this chapter captures the trade-off between the revenue to be collected from a new project and the penalties which may result from not meeting existing delivery and resource commitments for the contractor.

The problem under consideration can be defined as follows. In a multi-project environment with a certain number of available renewable resource types; a processing time, a due date, resource profiles, and associated unit tardiness and earliness costs are assigned to each activity. A baseline schedule exists for this set of projects. At a given point in time, a new project arrives. For the newly arriving project a due date has to be assigned and it has to be incorporated into the baseline schedule resulting

in a new schedule. A cost parameter for the completion time of the new project representing the cost of delaying a new project by one time unit is defined and is referred to as the completion time factor *K*. The objective then becomes the weighted sum of the earliness/tardiness costs of the ongoing projects plus the cost associated with the new project's completion time. Hence, the problem under consideration can be considered as a variant of the RCMPSP with weighted earliness/tardiness penalties (RCMPSPWET) and will be denoted as DRCMPSPWET in reference to the dynamic nature of the decision environment.

Within the context of this problem, the activity due dates and associated penalties are important parameters defining the characteristics of an instance. An applicable due date selection procedure is to convert the planned completion times into due dates. In other words, a baseline schedule, which is accepted by the contractor as well as by the client, is generated, and associated costs are defined to penalize deviations from the baseline plan in the new schedule. This approach can be applied to our deterministic model easily, since each disruption, as explained earlier, provides a new baseline schedule and can be converted into due dates for a potential new event in the future. With this approach, the dynamic problem can be simulated for multiple disruptions. The changes in revenue and deviations in schedules can be observed for multiple project arrivals at different points in time. Another strategy might involve defining some critical progress levels and penalties only for certain milestones of the projects. From a mathematical modeling point of view, defining milestones translates into choosing relatively higher cost parameters for the corresponding activities. Moreover, higher penalties for project completion times can be selected to emphasize the significance of completing projects at their previously scheduled times even if we allow shifting activities within a project. In the extreme case, we may omit the due date costs for all activities except those for the terminal activities of the projects. In summary, by setting the cost parameters associated with the activity due dates properly, we may model the problem with varying levels of flexibility and data requirements.

For any of these options, the following step is to determine the unit tardiness penalty values so that the deviations from the baseline schedule are not ruled out. An important factor for these penalties is the tightness of the due dates. A project with tight due dates has a greater possibility of becoming tardy; so the penalty values for a unit time should be lower than those under loose due dates, where the contractor has a wider time horizon to complete the project on time. In addition, the cost parameters have to be determined in a way that a trade-off between deviations from the baseline schedule and the due date of the new project exists.

In this chapter, the dynamics of the problem are analyzed with respect to the total number of activities, the due date tightness, the due date range, the number of resource types, and the completion time factor. The goal is to design a solution method that rapidly provides near optimal solutions for this problem. Quick solution methods can make rescheduling time and cost feasible in comparison with repair heuristics, which incorporate myopic approaches in most cases. This study makes the following contributions:

- The problem under consideration—DRCMPSPWET—is developed conceptually and a mathematical programming formulation of the problem is provided.
- A local search (LS) heuristic is designed and implemented. It is tested for solution quality and time against exact solutions obtained for a certain number of problem instances.
- A unique data set is generated for investigating the effects of the total number of activities, the due date tightness, the due date range, the number of resource types, and the completion time factor of the newly arriving project on the solution approach.

The chapter is organized as follows: In the Section "Related Literature and Problem Description," the related work in the literature and the problem definition are presented and an integer programming formulation for DRCMPSPWET is given. In the Section "An Iterated Local Search Approach for RCPSPWET," a heuristic approach for DRCMPSPWET is presented. The discussion of the data sets and an evaluation of the results are included in the Section "Computational Study." Conclusions and possible extensions for future work are presented in the Section "Concluding Remarks and Future Work."

Related Literature and Problem Description

Herroelen and Leus (2005) classify the related work on DRCPSP under four categories: reactive scheduling, stochastic rescheduling, fuzzy project scheduling, and proactive scheduling. Note that our problem falls within the scope of the first category. Hence, we will concentrate here only on work in the area of reactive scheduling. Interested readers may refer to a recent review of stochastic project scheduling by Ashtiani et al. (2011). The models focusing on reactive scheduling try to model any unexpected event within a deterministic approach. Instead of executing a full rescheduling process, another option would be trying to minimize the effects of the unexpected event building on a baseline schedule which might or might not be repaired. One such example is the study of Artigues and Roubellat (2000) considering the case of activity insertion to the baseline schedule. The objective is to minimize the maximum lateness in a multi-mode multi-project setting. The multi-project environment is transformed to a resource flow network setting and dominant insertion cuts are used to generate the new schedule. El Sakkout and Wallace (2000) propose a method for minimizing the weighted absolute difference between the starting times of each activity in the baseline and modified schedules. The weighted absolute differences correspond to the earliness/tardiness concepts with symmetric costs, if the finishing times in the baseline schedule are treated as due dates. They propose a repair-based heuristic approach to solve this problem.

Resource-Constrained Project Scheduling Problem with Weighted Earliness/Tardiness Costs

To the best of our knowledge, the existing work on resource-constrained project scheduling problem with weighted earliness/tardiness costs (RCPSPWET) is limited to single projects and no research has been conducted with multiple projects. Moreover, the concept of a baseline schedule is also not included in most of the studies. Neumann et al. (2003) mention an original schedule subject to change as a result of unexpected events. The limited work in the literature includes some exact solution approaches as well as heuristic methods for the problem.

An exact solution procedure for the resource-unconstrained version of the problem is suggested by Vanhoucke et al. (1999). The objective function is composed of the weighted sum of the earliness and tardiness values. This approach is based on a recursive search algorithm and consists of two main steps. First, a schedule is generated by scheduling activities at their due dates or later while considering only precedence relations. As a result, no right shift in the schedule can decrease the objective value. In the second step of the algorithm the set of activities, for which a backward shift can decrease the objective value, are selected by implementing a recursive search. Vanhoucke et al. (2001) extend the model to include resource capacity constraints. Using the exact solution algorithm for the resource-unconstrained version they develop a branch and bound algorithm based on resolving the resource conflicts in a resource-unconstrained solution. Precedence relations are added between activities in process during a period of resource conflict. Each conflict corresponds to a new node in the search tree and feasible solutions are obtained, if all conflicts are resolved. A further extension of the resource constrained model is provided by Vanhoucke (2002). In this study, for each activity, various due date options are offered. Each option differs in the tightness and unit cost values of the due date. That is, if an earlier due date is selected for an activity, the unit earliness and tardiness cost values are lower than those for a later due date. The objective is to select an appropriate due date option for each activity and generate a schedule such that the weighted sum of the earliness and tardiness values is minimized. A double branch and bound algorithm is developed to solve this problem. First, the resource-unconstrained model is solved with the convex due date cost profiles. These profiles are obtained by converting the combination of different due date cost functions for each activity into a convex envelope. Using these convex envelopes a single due date is selected for each activity. However, unit earliness or tardiness costs might change according to the convex envelope profile. The solution yields a lower bound on the cost of the actual due date profile and the first branch and bound is applied while considering the distance between the convex envelope and the original due date profile for each activity completion time. The optimal solution is obtained after applying a second branch and bound procedure in order to resolve the resource conflicts as in Vanhoucke et al. (2001).

Ballestin et al. (2008) develop an iterated LS algorithm for RCPSPWET. A population of feasible solutions is generated and LS procedures are applied to improve the objective function value. Activity lists and a schedule generation scheme are used

to generate corresponding schedules. The activities are scheduled iteratively with respect to a parameter called the simulated due date, which is the completion time of an activity in a randomly generated precedence feasible but resource-unconstrained schedule. Simulated due dates are selected instead of the original due date values in the problem data in order to create diversity in the population. Four different LS procedures are then applied to existing schedules. At this stage, the activity lists are not changed; instead, schedules are modified in order to obtain improved solutions for a particular activity list in the population. To expand the search space, the activity lists are perturbed. The sequence of the activities in the list as well as the simulated due dates are updated using five different perturbation procedures.

Another list-based heuristic approach is proposed by Nanobe and Ibaraki (2006). This work covers a variety of project scheduling problems with convex cost functions including the weighted earliness/tardiness problem. The solution procedure relies on keeping event lists to obtain schedules. Each activity consists of a start- and an end-event, where positions of events in a list define priority relations. Each list can be mapped to an event-on-node network representation, and the dual problem can be solved as a minimum cost network flow problem. Event lists have to be resource and precedence feasible. This is done by checking the total resource demand of activities which are allowed to be processed simultaneously. If necessary, the list is modified and made feasible by changing the positions of events. A neighborhood is defined by moving events in the list backward or forward, and an iterated LS is applied to the solution with the best objective value.

Problem Formulation

The DRCMPSPWET is defined here over an activity-on-node multi-project network with dummy start and finish activities. No precedence relation is assumed among the projects. The precedence relations among the activities are of the finish-to-start type with zero time lag. All activities are of a single mode. Hence, only renewable resources are taken into account. Preemption is not allowed.

A special case of RCMPSPWET with a single project, a single resource of unit capacity, unit resource usage for each activity, no precedence relationships, and zero unit earliness costs reduces to the strongly NP-hard single-machine scheduling problem of minimizing the total-weighted tardiness (Lenstra et al. 1977). Hence, RCMPSPWET is strongly NP-hard since the model presented in this study generalizes RCMPSPWET by incorporating a revenue function for the due date quoted for a new project. The overall objective is then to quote a due date that is as early as possible in order to maximize revenue while constructing a new schedule that minimizes the total-weighted deviation of the activity finishing times from their completion times in the baseline schedule. We define the following notation.

Sets and indices:

T	Set of time periods
I	Set of all projects in the baseline schedule
I^*	Set of all projects including the arriving project
h	$\|I\|$
$h+1$	Index of the arriving project
J_{i}	Set of activities of project i
P_{i}	Set of precedence relations between activities ε of project i
R	Set of renewable resources

Parameters:

W_{rt}	Amount of renewable resource r available in period t
ES_{ij}	Earliest start time of activity j of project i
LS_{ij}	Latest start time of activity j of project i
d_{ij}	Due date of activity j of project i
p_{ij}	Processing time of activity j of project i
w_{ijr}	Renewable resource requirement of activity j of project i of type r per unit time
e_{ij}	Earliness penalty of activity j of project i per unit time
t_{ij}	Lateness penalty of activity j of project i per unit time
K	Completion time factor for the arriving project

The parameters presented above are required to define an instance of DRCMPSP-WET. For each activity, the p_{ij} and w_{ijr} values define the single execution mode. However, there are additional parameters for activities depending on their status in the problem. For activities in the baseline schedule, a due date and unit earliness and tardiness penalties must be specified as well as a completion time factor standing for the cost associated with the completion time of the arriving project. Note that d_{ij} and K are not part of the original problem data in the experimental study. Their values depend on the baseline schedule of the instance. We elaborate on this issue further in the Sections "Due Date Generation," "Due Date Range," "Due Date Tightness," and "Completion Time Factor." Finally, the available capacities of the renewable resources are required. Note that the earliest and latest start times (LSTs) of activities can be calculated for a given time horizon $|T|$ using the conventional forward and backward pass algorithms of the critical path method (see, e.g., Badiru and Pulat 1995). The objective function under consideration is non-regular, and delaying activities may decrease the total cost. Therefore, an optimal schedule may contain unforced idle time; however, no activity will complete at a time later than $|T|$ in an optimal schedule, where $|T|$ is set to the sum of the maximum due date and the sum of the processing times of all activities of the arriving project.

Decision Variables

A 0–1 decision variable x_{ijt} is defined for each activity in the multi-project network including the dummy start and finish activities. For the activities in the baseline schedule, a finishing time, earliness and tardiness values have to be determined. For the arriving project, a due date is quoted as the finishing time of the dummy finish activity of the arriving project.

x_{ijt} {1, if activity j of project i starts at time period t; 0, otherwise.
f_{ij} Finishing time of activity j of project i
d_{h+1} Due date of the arriving project
E_{ij} Earliness of activity j of project i
T_{ij} Tardiness of activity j of project i

Mathematical Model DRCMPSPWET:

$$\min \sum_{i \in I} \sum_{j \in J_i} \left(e_{ij} \cdot E_{ij} + t_{ij} \cdot T_{ij}\right) + K \cdot d_{h+1} \tag{10.1}$$

$$f_{il} - f_{ik} \geq p_{il} \qquad \forall i \in I^*, \forall (k,l) \in P_i \tag{10.2}$$

$$f_{ij} = \sum_{t=ES_{ij}}^{LS_{ij}} x_{ijt} \cdot t + p_{ij} \qquad \forall i \in I^*, \forall j \in J_i \tag{10.3}$$

$$E_{ij} \geq d_{ij} - f_{ij} \qquad \forall i \in I, \forall j \in J_i \tag{10.4}$$

$$T_{ij} \geq f_{ij} - d_{ij} \qquad \forall i \in I, \forall j \in J_i \tag{10.5}$$

$$d_{h+1} \geq f_{h+1_j} \qquad \forall j \in J_{h+1} \tag{10.6}$$

$$\sum_{i \in I^*} \sum_{j \in J_i} \sum_{\theta = \max\{ES_{ij}, t - p_{ij} + 1\}}^{t} x_{ij\theta} \cdot w_{ijr} \leq W_{rt} \qquad \forall r \in R, \forall t \in T \tag{10.7}$$

$$\sum_{t=ES_{ij}}^{LS_{ij}} x_{ijt} = 1 \qquad \forall i \in I^*, \forall j \in J_i \tag{10.8}$$

$$x_{ijt} \in \{0,1\} \qquad \forall i \in I^*, \forall j \in J_i, \forall t \in ES_{ij}, \ldots, LS_{ij} \tag{10.9}$$

$$d_{h+1}, f_{h+1\,j} \geq 0 \qquad \forall j \in J_{h+1} \tag{10.10}$$

$$E_{ij}, T_{ij}, f_{ij} \geq 0 \qquad \forall i \in I, \forall j \in J_i \tag{10.11}$$

The objective function Eq. (10.1) consists of the weighted sum of the earliness and tardiness values of the activities in the baseline schedule and the completion time cost of the new project. Constraint Eq. (10.2) defines the precedence relationships among the activity pairs. The finishing times of the activities are determined in constraint

Eq. (10.3). Constraints Eqs. (10.4) and (10.5) determine the earliness and tardiness values, respectively. The quoted due date value, i.e., the completion time of the newly arriving project, is set by constraint Eq. (10.6). The total renewable resource usage in each time period is restricted to the maximum available amount in constraint Eq. (10.7). Finally, constraint Eq. (10.8) ensures that each activity is executed once and constraints Eqs. (10.9), (10.10), and (10.11) define the domains of the decision variables.

This problem formulation above differs from the single project static RCPSWET problem formulation given by Vanhoucke et al. (2001) in that it reflects a multi-project dynamic decision environment. The dynamic nature of the problem is incorporated into the formulation through the second term in the objective function Eq. (10.1) and the additional decision variables and associated constraints. Being the product of the completion time factor K and the quoted due date for the new project the second term represents an implicit cost of due date quotation and hence introduces into the formulation the trade-off between the stability of the activity finish times of the existing projects and the quoted due date for the new project.

An Iterated Local Search Approach for RCPSPWET

Heuristic procedures have been developed for RCPSPWET in single project environments as discussed in the Section "Related Literature and Problem Description." List-based heuristics reported by Ballestin and Trautman (2008) and Nanobe and Ibaraki (2006) perform well both in terms of solution quality as well as computation times. Moreover, neighborhoods can easily be defined for the schedules represented by the lists, and the associated schedule generation procedures are simple and efficient. Therefore, a population-based LS procedure is suggested to solve the problem at hand. The general flow of the solution algorithm is presented in Fig. 10.1.

The heuristic method starts by generating an initial population of activity lists. Three different improving steps are applied to this initial population iteratively in order to improve the activity lists. These steps replace the sequencing and optimal timing procedures commonly used in the machine scheduling literature for weighted earliness/tardiness problems. (Kanet and Sridharan (2000) give an overview of different optimal timing algorithms in the machine scheduling domain.) First, a list-position-based neighborhood search is performed to improve the sequencing in each activity list. An optimal timing-based neighborhood search is then applied to move chains of activities earlier in time. Finally, for all resource types in an instance, the associated arcs that prevent resource conflicts are added to the network and the resulting optimal timing problem is formulated and solved as a linear program (LP).

Activity Lists and Schedule Generation

An activity list in the population is used to represent a schedule. Each activity is assigned to a position in the list. In a precedence feasible activity list, each activity is

```
input  : An instance of DRCMPSPWET.
output: A feasible solution for the instance.
1  begin
2      Initialization;
3      Create the initial population;
4      foreach activity list in the initial population do
5          Apply List Positional Neighborhood Search;
6          Apply Timing-Based Neighborhood Search;
           /* The following lines are performed on some activity lists only.
              See text.                                                    */
7          Construct an extended precedence graph that prevents resource infeasibilities
           based on the current best schedule associated with the activity list;
8          Solve the optimal timing problem for this extended precedence graph as an LP;
9      end
10     Report the best schedule identified;
11 end
```

Fig. 10.1 Flow of the LS heuristic

positioned after its predecessors and before its successors. Given a precedence feasible activity list, a locally optimal schedule is generated by scheduling each activity in the list to start at its locally optimal position. For an activity in the baseline schedule, a locally optimal position is defined as the one which minimizes (earliness + tardiness) cost for this activity without shifting the activities already scheduled. The activities of the newly arriving project are scheduled as early as possible because the associated cost component in the objective function is increasing in the completion time of this project.

Initial Population Generation

An initial population is generated to apply the neighborhood search procedures. Each member of the population is a precedence feasible activity list. To ensure the diversity of the initial population and explore a larger portion of the search space, activity lists are constructed by applying two different priority rules and adapting a shifting bottleneck (SB)-based heuristic originally developed for job shop scheduling problems with non-regular objectives by Bulbul and Kaminsky (2010) to our problem, in addition to randomly generating precedence feasible activity lists.

To create activity lists the most total successors (MTS) and minimum LST priority rules are employed by selecting the activity with the best value among the precedence

feasible candidates. These are network and critical path-based priority rules, respectively (Demeulemeester and Herroelen 2002). The basic idea behind the selection of these dispatching rules is to increase the possibility of adding a larger number of precedence feasible activities to the candidate list earlier and thereby improving their chance of on-time scheduling as well as achieving higher resource utilization. Biased sampling versions of these priority rules are also used to increase the size of the population. That is, candidate activities are assigned probabilities proportional to their respective priorities, and the next activity in the list is picked randomly based on these selection probabilities.

The SB heuristic is a well-known machine-based decomposition method in the machine scheduling literature (Adams et al. 1988). In the application of the SB framework to job shop scheduling problems, the machine capacity constraints are initially all relaxed and are then added back to the problem sequentially by solving a series of single-machine scheduling subproblems. The objective function value of a single-machine subproblem provides an estimate of the effect of the capacity restrictions of the machine under consideration on the overall schedule. The currently unscheduled machine with the highest subproblem objective value is referred to as the bottleneck machine, and the sequence of operations on this machine is fixed first before those of the remaining unscheduled machines. The SB approach was originally developed for the classical job shop scheduling problem of minimizing the makespan by Adams et al. (1988), and it was later extended to job shop scheduling problems with maximum lateness (Demirkol et al. 1997) and total-weighted tardiness minimization objectives (Pinedo and Singer 1999; Singer 2001; Mason et al. 2002) among others. Recently, Bulbul and Kaminsky (2010) extended this framework to job shop scheduling problems with any objective function whose associated optimal timing problem can be expressed as an LP. Their approach is particularly effective, if the individual completion times are associated with explicit costs as in our problem. Based on this observation, we adapted the SB algorithm of Bulbul and Kaminsky (2010) for our purposes. Initially, a schedule is obtained by relaxing all resource capacities and solving the resulting model as an LP. The SB heuristic then resolves the resource conflicts present in the optimal solution of this relaxation iteratively by solving a set of single-resource-weighted earliness/tardiness scheduling subproblems with precedence constraints. The unit earliness and tardiness costs in the subproblems are estimated using LP sensitivity analysis as in the original paper. The subproblem is a generalization of the NP-hard single-machine-weighted earliness/tardiness problem, and the iterated LS approach we design for the overall problem is also used to solve the subproblems of the SB heuristic with some minor modifications and simplifications. These details are discussed in Pamay (2011). The solution of a subproblem introduces new precedence relationships based on the concept of resource flows (e.g., see Artigues and Roubellat 2000). These new precedence constraints are incorporated into the optimal timing LP and ensure that the capacity of the resource under consideration is no longer violated. These steps are repeated until all resource conflicts are removed and a feasible solution to the original problem is obtained. This basic algorithm is enhanced by executing a restricted tree search over all possible orders of resolving the resource conflicts and results in several feasible

solutions for the original problem. A standalone application of this SB heuristic does not produce high quality solutions; however, it provides us with a tool to diversify the initial population. The schedules constructed by the SB heuristic are converted to activity lists based on the activity start times and added to the initial population. In our computational study, we report the results of the iterated LS algorithm both with and without the initial solutions from the SB heuristic and demonstrate a significant added value from their inclusion in the initial population.

List Positional Neighborhood Search

Once the initial population has been generated, the first neighborhood search procedure starts. This process is applied to each member of the initial population separately and if an improvement is observed the activity list is replaced and the search for better schedules continues with the new activity list. First, all activities in an activity list are sorted in non-increasing order of their contributions to the objective function. For the activities of the new project this contribution is zero unless they belong to the critical path of the project. A critical activity of the new project is assigned a cost of (K $f_{h+1,|Jh+1|}$), where $f_{h+1,|Jh+1|}$ is the completion time of the new project. The neighborhood search proceeds by processing each activity in the list in the order specified above. The activity under consideration may be moved to an earlier position in the list while preserving precedence feasibility. Consequently, it can be scheduled at earlier stages of the schedule generation process and has a greater chance of incurring a lower cost. A selected activity may be moved anywhere between its predecessor with the latest position in the list and its current position. Each of these possible moves is evaluated by removing the activity from its current position and inserting it at the required spot in the list. For each position, the objective function value is determined by using the locally optimal scheduling scheme. If the objective value can be improved, this change is applied to the activity list. If the evaluated moves for the current activity are non-improving, the activity with the next highest cost contribution is selected and the procedure is repeated until a limited number of non-improving steps is reached. If no improvement can be observed until reaching this threshold level, the best non-improving move is applied, and the move is added to a tabu list to track forbidden moves. In general, the neighborhood search for an activity list terminates if either a prespecified maximum number of neighborhood search moves or a prespecified maximum number of non-improving steps is reached first.

Timing-Based Neighborhood Search

To check for further possible improvements a timing-based LS is applied. The locally optimal scheduler places an activity in its locally optimal position without shifting activities already scheduled. Therefore, the total objective function value may be

reduced by moving a single activity earlier or later in time. This can be done by modifying the due dates of the activities temporarily such that the locally optimal positions of the activities are changed for the same sequence. To this end, we first determine chains of activities in the precedence graph which are processed without idle time in between in the current schedule and then calculate the total cost contribution of each chain. The chain with the maximum cost is selected, and the due date of the first activity in this chain is decreased by a single time unit. This due date value is used while scheduling the activity locally optimally, but the objective function is still calculated with the original problem data. By decreasing the due date of an activity, other members of the chain can move earlier in time, and the objective function value may be improved. If this is the case, then we identify an improved schedule associated with the current activity list. The procedure is repeated for other chains in non-increasing order of their contributions to the objective function. The search is terminated if either the prespecified maximum number of non-improving steps or the maximum number of neighborhood search steps is reached first.

LP-Based Optimal Timing

A final improvement step is applied to a limited number of activity lists in the initial population. We insert additional arcs into the precedence graph which avoid resource infeasibilities based on the current feasible schedule associated with the activity list (e.g., see Artigues and Roubellat 2000). This allows us to formulate an LP which yields a resource-feasible optimal schedule for the given extended precedence graph. In the LP formulation below, the set of extended precedence relationships $\bar{P}$ includes the original precedence relationships on top of the precedence relationships derived from the resource flows. In essence, the concept of resource flows allows us to convert conditions on resource feasibility into temporal relationships. In our presentation, $(i,k,j,l) \in \bar{P}$ if there is either a precedence relationship or a resource flow between activities (i,k) and (j,l):

$$\min \sum_{i \in I} \sum_{j \in J_i} \left(e_{ij} \cdot E_{ij} + t_{ij} \cdot T_{ij} \right) + K \cdot d_{h+1}$$

$$f_{jl} - f_{ik} \geq p_{jl} \qquad \forall (i,k,j,l) \in \overline{P} \tag{10.12}$$

$$(4), (5), (6), (10), (11)$$

Note that the structure of the LP above is similar to the mathematical model of DRCMPSPWET, except that the binary variables x_{ijt} and the related constraints are replaced by constraints Eq. (10.12) under the presence of extended precedence relationships.

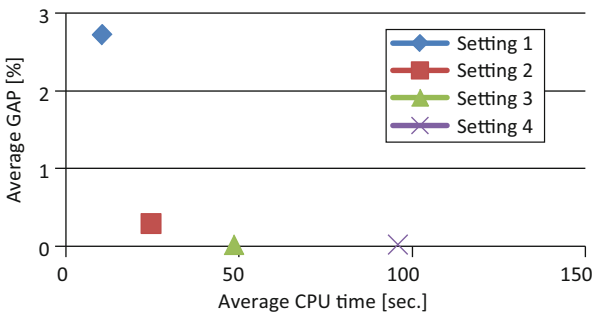


Fig. 10.2 Parameter fine-tuning results

The number of activity lists to which the LP-based improvement step is applied is referred to here as the number of LP-based search steps. The search for the best-performing values of this parameter together with the maximum number of neighborhood search steps and the maximum number of non-improving steps for both positional and neighborhood searches are the subject of the next section.

Parameter Fine-Tuning

In order to select the best-performing parameter settings, a fine-tuning procedure is applied. Twenty different instances with 200 activities are tested. Six different parameters are adjusted: the maximum number of steps for the positional neighborhood search, the maximum number of steps for the timing-based neighborhood search, two different parameters for the maximum number of non-improving steps of these neighborhoods, the number of LP-based search iterations, and the size of the tabu list in the positional neighborhood search. A preliminary analysis revealed that the solution quality and time are insensitive to the size of the tabu list. This parameter has therefore been fixed at 5 in the rest of our study. The different values selected for each setting and the results are presented in Table 10.1 and Fig. 10.2, respectively.

Figure 10.2 shows the average gap between the best solution and the solution found by each setting and the average CPU times. Setting 3 attains the best trade-off in terms of solution quality and CPU times. Therefore, setting 3 is selected for the solution procedure(s) applied.

Computational Study

All solution approaches were implemented in Visual C#. IBM ILOG CPLEX Optimization Studio 12.1 is used as the engine for solving the LP models. A data set of 800 unique instances is generated to test the performance of the suggested methods.

Table 10.1 Parameter selection settings

	Max number of positional neighborhood search steps	Max number of timing-based neighborhood search steps	Max number of non-improving steps for the positional neighborhood	Max number of non-improving steps for the timing-based neighborhood	Number of LP-based search steps	Size of the tabu list
Setting 1	20	30	10	20	5	5
Setting 2	50	50	20	40	5	5
Setting 3	100	100	30	70	10	5
Setting 4	200	200	100	150	10	5

Table 10.2 Parameter settings for the data set generated

Total number of activities	20, 40, 50, 100, 150, or 200
Due date range	Clustered or distributed
Due data tightness	Tight or loose
Number of resource types	2 or 5
Completion time factor	High or low

The experiments were conducted on a single core of an HP Compaq DX 7400 Microtower with a 2.33 GHz Intel Core 2 Quad CPU Q8200 processor and 3.46 GB of RAM.

Experimental Data

As stated before, the related work in the literature focuses on the single project version of RCPSPWET. Moreover, existing benchmark instances do not always investigate the effects of different problem parameters on the performance of the proposed solution approaches. Therefore, a new data set is generated. Each instance of the problem set consists of a group of projects present in a baseline schedule with activity-based due dates, unit earliness and tardiness costs. A newly arriving project is also included with a completion time factor K. The parameter settings for the entire data set are given in Table 10.2. The rationale behind adopting each of these parameters will be discussed in the upcoming subsections.

Project Pool Generation

Since our problem is a multi-project scheduling problem, each instance in the test problem data set consists of a group of projects. For this reason, a project pool is generated first, which will later be used to create the multi-project instances. Various random project generation procedures have been discussed in the literature. ProGen is developed by Kolisch et al. (1995) for RCPSP and its multi-mode extension. ProGen/max developed by Schwindt (1995) is an upgraded version of ProGen for

minimal and maximal time lag extensions of generalized precedence relations. A more recent project generator, called RanGen, has been developed by Vanhoucke et al. (2003). We use this generator because RanGen enables the user to select predefined complexity measures for generated networks, which is important for differentiating the instances.

Four parameters have to be specified in RanGen to obtain different project networks. The first parameter is the order strength (OS), which is defined as the number of precedence relations including the transitive ones but not including those arcs incident from or into the dummy start and end activities, respectively, divided by the maximum number of precedence relations $n(n-1)/2$, where n denotes the number of non-dummy activities in the network (Mastor 1970). RanGen is able to generate unique networks with the prespecified OS values. Three different OS values (0.25, 0.50, and 0.75) are selected. For each project, five types of renewable resources are defined. Two different resource-usage-related parameters are included. The first resource-related measure is the resource density (RU) defined as in Eq. (10.13) below:

$$RU = RU_i = \sum_{r=1}^{R} \begin{matrix} 1 & \text{if } w_{ir}0, \\ 0 & \text{otherwise} \end{matrix} \tag{10.13}$$

This parameter specifies the number of resource types used by an activity *i, RUi,* in the network. RU is preferred to another resource-related measure referred to as the resource factor (RF) introduced by Alvarez-Valdes and Tamarit (1989), because RF might yield networks in which some activities do not use any resources at all. Another resource measure, the resource constrainedness (RC), is defined as the ratio between the available capacity of a resource type (W_r) and the average usage of activities ($\overline{w}_r$) of this particular resource in Eq. (10.14). The RU and RC values are selected as 4, 5, and 0.25, 0.50, respectively. The number of the activities in a project is taken as an input data as well. To achieve the required number of activities for each RCMPSPWET instance, projects with 5, 10, 20, and 30 activities are generated.

$$RC_i = \frac{\overline{w}_r}{W_r} \tag{10.14}$$

All parameter settings are summarized in Table 10.3. The project pool for each n, except for $n = 5$, consists of 50 different projects. A total of 32 projects with five activities is used because the generator is not able to generate 50 unique networks with the specified OS values due to the small number of nodes in the network.

The data set can be obtained by sending a request to the corresponding author.

Total Number of Activities

The total number of activities in an instance is an important measure of the size as well as the difficulty of the instance. As presented in Table 10.2, for a given instance the number of activities is ranging from 20 to 200 activities, excluding the dummy activities. Note that we solve instances with up to 200 activities while

Table 10.3 Settings for the project pool generation

Number of activities	OS	RU	RC	Number of unique projects
5	0.25	4	0.25	3
	0.50	4	0.50	10
	0.75	5	0.50	9
	0.50	5	0.25	10
10	0.25	4	0.25	10
	0.50	5	0.25	10
	0.75	5	0.25	10
	0.75	4	0.50	10
	0.25	5	0.50	10
20	0.25	5	0.25	10
	0.50	5	0.25	10
	0.75	5	0.25	10
	0.25	4	0.50	10
	0.75	4	0.50	10
30	0.25	5	0.25	10
	0.50	5	0.25	10
	0.75	5	0.25	10
	0.25	4	0.50	10
	0.75	4	0.50	10

OS order strength, *RU* resource density, *RC* resource constrainedness

the maximum number of activities considered in the literature on earliness/tardiness project scheduling problems is 100 (Ballestin and Trautman 2008, Vanhoucke et al. 2001, and Neumann et al. 2003).

Project Combinations

For each setting of the total number of activities, different combinations of projects are selected from the project pool to create an instance of DRCMPSPWET with the required number of activities. For example, in order to generate an instance of DRCMPSPWET with 30 activities, a combination of three projects with ten activities each is selected as one of the combinations. In this scenario, for two of these three projects, due dates, earliness and tardiness costs are generated. The third project is defined as the newly arriving one, and a completion time factor is determined for it. Another combination uses a project portfolio of six projects with five activities, where one of these projects is designated as the new arrival. For each value of the total number of activities in Table 10.2, up to three different combinations are selected. These combinations differ in the total number of projects in an instance. For each combination, five different master instances are generated. These master instances provide the information about which projects in the pool are added to the project portfolio. This is accomplished by selecting projects from the pool with the desired number of activities randomly. Master instances are then used to create unique

instances by adding the data about the due dates, the unit earliness and tardiness costs, and the completion time factors depending on the values of the remaining data generation parameters. The unit earliness and tardiness costs are drawn from uniform distributions in the range 0–10, and the generation of the due dates and the completion time factors are detailed in the Sections "Due Date Generation," "Due Date Range," "Due Date Tightness," and "Completion Time Factor." All the project combination schemes are summarized in Table 10.8 in the Appendix.

Due Date Generation

Due dates are generated in this study based on a baseline schedule. All projects in an instance, except for the new arrival, have an associated existing schedule constructed by the scheduling routine described next. In Ballestin and Trautman (2008), Vanhoucke et al. (2001), and Neumann et al. (2003), on the other hand, the data sets are generated by considering the critical paths and the earliest start time values of the activities in the network.

The method used to obtain the baseline schedule is quite important for the effective utilization of the resources. Therefore, makespan minimization is selected as the objective for generating the baseline schedule. There are many heuristic approaches in the literature developed for makespan minimization. We decided to use a scheduling scheme with an effective dispatching rule in order to generate schedules with good makespan values within reasonable computation times. In his review paper about the performance of different dispatching rules for makespan minimization, Kolisch (1996) states that the LST rule shows the best performance. Therefore, the LST rule is used here together with the serial scheduling scheme (SSS) for generating the baseline schedule. At each iteration, SSS selects the activity with the minimum LST among the ones whose predecessors are already scheduled and schedules it at the earliest feasible point in time leading to an active schedule. The LST values are calculated using the backward pass algorithm of the critical path method. However, we implement the LST rule slightly differently depending on the desired range of the due dates as discussed next.

Due Date Range

The range of the due dates over the time horizon in the baseline schedule is important for flexibility in scheduling. That is, if the due dates of a project are spread over the entire planning horizon, activities can be moved forward or backward more freely in time while scheduling the arriving project. Clustered due dates, on the other hand, reduce the flexibility of the projects and constrain them to move only within shorter time windows provided in the baseline schedule. The difference between these two settings is visualized in Figs. 10.3 and 10.4. In the first case, all projects are active

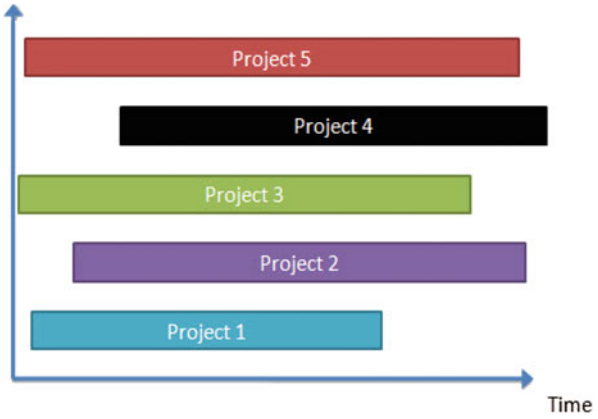


Fig. 10.3 Distributed due date windows

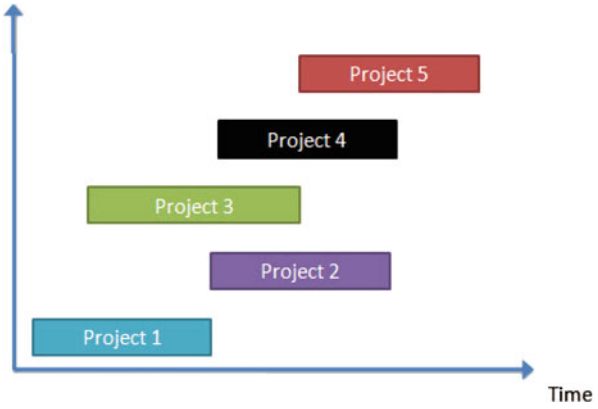


Fig. 10.4 Clustered due date windows

during most of the schedule timeline whereas in the second case only a few projects are active within a given time interval.

In order to obtain schedules with these two different characteristics, the basic schedule generation scheme based on the LST rule is modified. For the distributed due date generation, we keep track of the progress levels of the projects while scheduling activities iteratively. In other words, the activity with the lowest LST value is picked among the activities of the project with the minimum progress level. With this approach, the projects are kept active along the entire timeline of the baseline schedule. The clustered due date range is obtained by randomly selecting a project and scheduling all activities of this particular project one by one according to the LST rule in order to complete the selected project as soon as possible after it is started. The process continues by selecting another unscheduled project randomly until all the projects are scheduled. In order to observe the effects of this parameter setting, distributed and clustered due date generation schemes are applied to project combinations with a relatively high number of projects. Otherwise, only the distributed

due date generation scheme is employed. The details are provided in Table 10.8 in the Appendix.

Due Date Tightness

An additional parameter controls the tightness of the due dates. Tight due dates values are closer to the starting time of the schedule and offer less flexibility for meeting the due date. Loose due dates, on the other hand, allow delays to activities without affecting successor activities or incurring additional cost. In other words, there is a higher possibility of meeting loose due dates compared to tight ones. Due date tightness in the related articles in the literature is manipulated by multiplying the individual due dates (or an average due date) by a tightness factor. We use a different approach. In order to reflect these tightness and looseness considerations in our data generation scheme, we change the available resource capacities in the baseline schedule. For setting loose due dates, we only allow a resource to be used at 80 % of its available capacity. By creating a baseline schedule for the current project portfolio by utilizing the resources at less than full availability, slack resource capacities can be used to schedule a newly arriving project without causing significant deviations in the new schedule. As a consequence, the makespan of the baseline schedule is increased but additional resource capacity is allocated to schedule the newly arriving project. We would expect that the number of activities scheduled on time would increase and for the same master instance a lower objective value can be obtained. By a similar argument, a baseline schedule constructed with fully available resources would result in tight due dates. For each unique instance loose and tight due dates settings are present in the data set.

Number of Resources

The number of resource types is another complicating factor in a project scheduling problem. In general, instances with more resources are more challenging. In our data set, the number of resource types is either two or five. Initially, all instances are created with five resource types, and the last three resource types are simply dropped from instances with 2 resource types.

Completion Time Factor

As one of the contributions of this study, the effects of the completion time factor K on the schedules will be studied, and we need to set a completion time factor value for each instance in the data set. As stated before, a trade-off between the earliness/tardiness costs of the activities in the baseline schedule and the completion-time-related cost of the new project must exist in order to obtain a reasonable problem

setting. Otherwise, scheduling the newly arriving project at the beginning or at the end of the schedule, depending on the dominant cost component, might yield good solutions for most of the instances. Therefore, we implemented another pre-scheduling step, similar to that in the due date generation process, to obtain the completion time factors. For two due date range settings, different approaches are used. For the distributed due date setting, we generate a new schedule employing the LST rule after adding the new project to the set of projects present in the baseline schedule of the instance and then calculate the total earliness/tardiness cost for the projects in the baseline schedule. The completion time factor is obtained by dividing the total earliness/tardiness cost by the completion time of the newly arriving project. For the clustered due date range, recall that projects are added to the schedule one by one in some sequence. The new project is inserted into each possible position in this sequence, and the LST rule is invoked for the resulting order. We obtain different cost values for the same instance depending on the position of the new project in the sequence and then take the average of these cost values and also compute the average completion time of the new project. The ratio of these two average values yields the completion time factor of the instance. Thus, we generated completion time factors specific to the instance data instead of selecting the same factor for all instances.

These completion time factor values are scaled in order to provide different parameter settings. In the "high" setting, the scaling factor is 1 and reflects that we expect the contribution of the new project to the overall objective function to be roughly the same as the total earliness/tardiness cost of the projects in the baseline schedule. In the "low" setting, the scaling constant is 0.5.

Results

We had two primary goals in mind while designing our computational study. First, we demonstrated that the LS method provides solutions of high quality in reasonable computation times. Second, we explored the effects of various problem parameters detailed in the previous sections on solution quality. We implemented two variants of our iterated LS algorithm as discussed in the Section "Initial Population Generation." In one variant (LS), the initial population consists of randomly generated lists in addition to activity lists produced by dispatch rules. In the second variant (LS-SB), the initial population is enhanced by activity lists retrieved from the SB heuristic mentioned in the Section "Initial Population Generation." Detailed results are available in Pamay (2011).

In the first part of our computational study, we benchmark the proposed LS method against the mixed integer programming (MIP) formulation presented in the Section "Problem Formulation" solved by ILOG CPLEX 12.1 which can only handle instances with up to 40 activities. Larger instances require excessive computation times. The time limit imposed on CPLEX is 1 h for instances with 20 and 30 activities, and 2 h for instances with 40 activities. If CPLEX does not terminate with an optimal solution within the allotted time (39 and 42 instances with 30 and 40

activities, respectively), then we report the best integer solution identified during the optimization. Therefore, all gaps are computed with respect to the best solution available. The results in Table 10.4 are grouped by the number of activities and resource types (indicated in the first two columns), and the number of instances in the group is given in the third column. The results in Table 10.4 attest to the competitiveness of the iterated LS heuristic. All optimal solutions are available for 40 instances with 20 activities, where LS attains the optimal solution in 24 cases with an optimality gap of 3.87 % on average. When the initial population is extended with activity lists from the SB heuristic, the number of optimal solutions identified increases to 29 with an average optimality gap of 1.12 %. LS attains better solutions than MIP in 11 and 17 cases for instances with 30 and 40 instances, respectively. The corresponding numbers for LS-SB are 11 and 18. Both LS and LS-SB match the best solution obtained by MIP in 19 and 10 cases for instances with 30 and 40 activities, respectively. For instances with 40 activities, LS and MIP perform on a par, and LS-SB is superior to MIP; however, both LS and LS-SB take a fraction of the effort required by CPLEX. The diversification effect of the activity lists retrieved from the SB heuristic manifests itself in both the average and the maximum gaps. For instances with 50 or more activities the differences in the maximum gaps are particularly significant.

It is not possible to identify a uniform pattern regarding the effect of the due date tightness on the solution quality from the data in Table 10.5. The results for instances with 40 or less activities suggest that instances with loose due dates are somewhat easier.

Next, we investigate the impact of the distributed and clustered due dates on the iterated LS heuristic. Recall that instances with up to 40 activities are all generated with the "distributed" option; therefore, no MIP result is available for this analysis. Results presented in Table 10.6 suggest that the added value of the extended initial population is more critical when the due dates are distributed.

Finally, Table 10.7 explores the sensitivity of our results to the completion time factor of the new project. It is evident that LS and LS-SB return solutions of high quality under both the "low" and "high" settings of the completion time factor. The effect of the extended initial population is more pronounced for smaller values of the completion time factor.

In summary, the proposed iterated LS heuristic delivers solutions of high quality. Instances with up to 200 activities are solved in short CPU times given that our problem is not an operational problem and does not need to be solved frequently. Furthermore, the performance of our algorithm is robust under various data generation settings; in particular, if we opt for using an enhanced initial population as described in the Section "Initial Population Generation."

Table 10.4 Comparison of the LS method against CPLEX 12.1

Number of activities	Number of resource types	Number of instances	Avg. gap (%)			Max. gap (%)			Avg. CPU time (s)		
			MIP	LS	LS-SB	MIP	LS	LS-SB	MIP	LS	LS-SB
20	2	20	0.00	2.53	2.09	0.00	16.71	16.71	74	3	4
	5	20	0.00	5.21	0.15	0.00	35.05	1.28	476	6	27
30	2	40	0.10	9.78	8.74	3.39	40.10	40.10	1,241	5	7
	5	40	1.55	5.05	3.24	15.07	22.91	20.21	4,138	9	45
40	2	40	12.40	9.87	8.88	456.88	41.16	39.67	796	6	8
	5	40	7.87	10.60	7.74	55.85	70.40	59.20	5,121	12	62
50	2	60		1.37	0.00		20.25	0.00		7	10
	5	60		4.77	0.00		56.14	0.00		14	57
100	2	80		0.49	0.00		28.10	0.00		14	20
	5	80		2.20	0.00		39.42	0.00		26	138
150	2	80		0.47	0.00		21.24	0.00		24	35
	5	80		1.20	0.00		57.13	0.00		46	231
200	2	80		0.04	0.00		3.40	0.00		35	49
	5	80		0.17	0.00		12.54	0.00		72	304

Table 10.5 Effect of the due date tightness on the solution quality

Number of activities	Due date tightness	Number of instances	Avg. gap (%)			Max. gap (%)		
			MIP	LS	LS-SB	MIP	LS	LS-SB
20	Tight	20	0.00	4.22	1.35	0.00	35.05	16.71
	Loose	20	0.00	3.52	0.88	0.00	29.41	9.09
30	Tight	40	0.62	6.59	5.51	15.07	36.84	36.84
	Loose	40	1.04	8.25	6.47	14.55	40.10	40.10
40	Tight	40	14.41	12.59	10.66	456.88	53.34	44.42
	Loose	40	5.85	7.88	5.96	55.85	70.40	59.20
50	Tight	60		1.01	0.00		17.11	0.00
	Loose	60		5.12	0.00		56.14	0.00
100	Tight	80		1.06	0.00		28.10	0.00
	Loose	80		1.63	0.00		39.42	0.00
150	Tight	80		1.45	0.00		57.13	0.00
	Loose	80		0.22	0.00		14.56	0.00
200	Tight	80		0.00	0.00		0.00	0.00
	Loose	80		0.21	0.00		12.54	0.00

Table 10.6 Effects of the due date range on the solution quality

Number of activities	Due date range	Number of instances	Avg. gap (%)		Max. gap (%)	
			LS	LS-SB	LS	LS-SB
50	Distributed	80	4.27	0.00	56.14	0.00
	Clustered	40	0.65	0.00	12.50	0.00
100	Distributed	120	1.68	0.00	39.42	0.00
	Clustered	40	0.37	0.00	8.80	0.00
150	Distributed	120	0.16	0.00	14.56	0.00
	Clustered	40	2.87	0.00	57.13	0.00
200	Distributed	120	0.14	0.00	12.54	0.00
	Clustered	40	0.00	0.00	0.00	0.00

Concluding Remarks and Future Work

The purpose of this work is to study the dynamic project scheduling environments. In that problem setting, a project arrives on top of an existing project portfolio, and a due date has to be quoted for the new project while keeping the costs related to changes in the schedule at a minimum. The objective function consists of the weighted earliness/tardiness costs of the activities of the existing projects in the current schedule in addition to a term that increases linearly with the anticipated completion time of the new project. An iterated LS heuristic is developed to solve large instances of this problem. In order to analyze the performance of the proposed method, a new multi-project data set is created by controlling the due date tightness, the due date range, the number of resource types, the completion time factor, and the total number of activities in an instance. A series of computational experiments were

Table 10.7 Effects of the completion time factor on the solution quality

Number of activities	Completion time factor	Number of instances	Avg. gap (%)			Max. gap (%)		
			MIP	LS	LS-SB	MIP	LS	LS-SB
20	Low	20	0.00	5.07	1.89	0.00	35.05	16.71
	High	20	0.00	2.67	0.34	0.00	29.41	5.00
30	Low	40	1.58	7.61	5.43	15.07	36.84	36.84
	High	40	0.08	7.23	6.55	1.25	40.10	40.10
40	Low	40	5.09	10.47	8.02	55.85	53.34	38.42
	High	40	15.18	10.00	8.60	456.88	70.40	59.20
50	Low	60		3.86	0.00		56.14	0.00
	High	60		2.27	0.00		42.40	0.00
100	Low	80		2.23	0.00		39.42	0.00
	High	80		0.47	0.00		13.76	0.00
150	Low	80		0.44	0.00		16.08	0.00
	High	80		1.23	0.00		57.13	0.00
200	Low	80		0.21	0.00		12.54	0.00
	High	80		0.00	0.00		0.00	0.00

carried out to test the performance of the LS approach. Moreover, exact solutions for the small instances are provided. The results indicate that the proposed LS heuristic performs well in terms of both solution quality and solution time. The value of an extended initial population is also demonstrated.

Several interesting extensions of this work are listed below:

- Precedence relations between projects can also be included considering that in practice some projects need to precede others due to technological factors, e.g., in R&D environments.
- Arrival of multiple projects at a time or at different points in time may be studied.
- A multi-mode extension is clearly an important research direction we may pursue in the future.

To the best of our knowledge, the proposed work is the first study of the multi-project dynamic version of RCPSPWET, namely, DRCMPSPWET. The relative scarcity of the literature on this problem suggests that static and dynamic RCMPSPWET constitute a rich topic for further research activities. Moreover, the practical relevance of this problem for companies, which have to manage their project portfolio in dynamic environments, offers a wide range of implementation options in the business context.

Acknowledgments We gratefully acknowledge the support given by The Scientific and Technological Research Council of Turkey (TUBITAK) through Project Number MAG 109M571.

Appendix

Table 10.8 Details of the data set generated for the computational study

Number of activities	Combinations	ID	Number of MI	Due date range	Due date	Number of resource types	*K*	E/T Cost values	Number of instances
20	(5 A × 3P + 5A x 1P)	A20_1	5	Distributed	Loose or tight	2 or 5	High or low	U(0,10)	40
30	(10 A × 2P + 10 A × 1P)	A30_1	5	Distributed	Loose or tight	2 or 5	High or low	U(0,10)	40
30	(5 A × 5P + 5 A × 1P)	A30_2	5	Distributed	Loose or tight	2 or 5	High or low	U(0,10)	40
40	(10 A × 3P + 10 A × 1P)	A40_1	5	Distributed	Loose or tight	2 or 5	High or low	U(0,10)	40
40	(10 A × 3P 5A × 1P + 5 A x 1P)	A40_2	5	Distributed	Loose or tight	2 or 5	High or low	U(0,10)	40
50	(10 A × 4P + 10 A × 1P)	A50_1	5	Distributed	Loose or tight	2 or 5	High or low	U(0,10)	40
50	(5A × 8P + 10 A × 1P)	A50_2	5	Clustered or distributed	Loose or tight	2 or 5	High or low	U(0,10)	80
100	(30 A × 3P + 10 A × 1P)	A100_1	5	Distributed	Loose or tight	2 or 5	High or low	U(0,10)	40
100	(10 A × 2P 20A × 3P + 20 A × 1P)	A100_2	5	Distributed	Loose or tight	2 or 5	High or low	U(0,10)	40
100	(10 A × 9P + 10 A × 1P)	A100_3	5	Clustered or distributed	Loose or tight	2 or 5	High or low	U(0,10)	80

Table 10.8 (continued)

Number of activities	Combinations	ID	Number of MI	Due date range	Due date	Number of resource types	*K*	E/T Cost values	Number of instances
150	(30 A × 4P + 30 A × 1P)	A150_1	5	Distributed	Loose or tight	2 or 5	High or low	U(0,10)	40
150	(20 A × 6P + 30 A × 1P)	A150_2	5	Distributed	Loose or tight	2 or 5	High or low	U(0,10)	40
150	(10 A × 10P 30 A × 1P + 20 A × 1P)	A150_3	5	Clustered or distributed	Loose or tight	2 or 5	High or low	U(0,10)	80
200	(30 A × 6P + 20 A × 1P)	A200_1	5	Distributed	Loose or tight	2 or 5	High or low	U(0,10)	40
200	(20 A × 8P 10A × 1P + 30 A × 1P)	A200_2	5	Distributed	Loose or tight	2 or 5	High or low	U(0,10)	40
200	(5 A × 10P 10A × 12P + 30 A × 1P)	A200_3	5	Clustered or distributed	Loose or tight	2 or 5	High or low	U(0,10)	80

References

Adams, J., Balas, E., & Zawack, D. (1988). The shifting bottleneck procedure for job shop scheduling. *Management Science*, *34*, 391–401.

Alvarez-Valdes, R., & Tamarit, J. M. (1989). Heuristic algorithms for resource constrained project scheduling: A review and empirical analysis. In Slowinski and Weglarz (Eds.), *Advances in project scheduling* (pp. 113–134). The Netherlands: Elsevier.

Artigues, C., & Roubellat, F. (2000). A polynomial activity insertion algorithm in a multi-resource schedule with cumulative constraints and multiple modes. *European Journal of Operational Research*, *127*(2), 297–316.

Ashtiani, B., Leus, R., & Aryanezhad, M. (2011). New competitive results for the stochastic resource-constrained project scheduling problem: exploring the benefits of pre-processing. *Journal of Scheduling*, *14*, 157–171.

Ballestin, F., & Trautman, N. (2008). An iterated-local-search heuristic for the resource-constrained weighted earliness-tardiness project scheduling problem. *International Journal of Production Research*, *46*, 6231–6249.

Badiru, A. B., & Pulat, P. S. (1995). *Comprehensive project management*. New Jersey: Prentice Hall PTR.

Bulbul, K., & Kaminsky, P. (2010). A linear programming-based general method for job shop scheduling. *Journal of Scheduling*, in press. http://dx.doi.org/10.1007/s10951-012-0270–4.

Demeulemeester, E., & Herroelen, W. (2002). *Project scheduling. A research handbook*. Dordrecht: Kluwer.

Demirkol, E., Mehta, S., & Uzsoy, R. (1997). A computational study of shifting bottleneck procedures for shop scheduling problems. *Journal of Heuristics*, *3*(2), 111–137.

El Sakkout, H., & Wallace, M. (2000). Probe backtrack search for minimal perturbation in dynamic scheduling. *Constraints*, *5*(4), 359–388.

Herbots, J., Herroelen, W., & Leus, R. (2007). Dynamic order acceptance and capacity planning on a single bottleneck resource. *Naval Research Logistics*, *54*(8), 874–889.

Herroelen, W., & Leus, R. (2005) Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research*, *165*, 289–306.

Kanet, J., & Sridharan, V. (2000). Scheduling with inserted idle time: problem taxonomy and literature review. *Operations Research*, *48*(1), 99–110.

Kolisch, R. (1996). Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation. *European Journal of Operational Research*, *90*(2), 320–333.

Kolisch, R., Sprecher, A., & Drexl, A. (1995). Characterization and generation of a general class of resource-constrained project scheduling problems. *Management Science*, *41*, 1693–1703.

Lenstra, J., Rinnooy Kan, A., & Brucker, P. (1977). Complexity of machine scheduling problems. *Annals of Discrete Mathematics*, *1*, 343–362.

Mason, S., Fowler, J., & Carlyle, W. (2002). A modified shifting bottleneck heuristic for minimizing total weighted tardiness in complex job shops. *Journal of Scheduling*, *5*(3), 247–262.

Mastor, A. (1970). An experimental investigation and comparative evaluation of production line balancing techniques. *Management Science*, *16*(11), 728–746, 1970.

Nanobe, K., & Ibaraki, T. (2006). A metaheuristic approach to the resource constrained project scheduling with variable activity durations and convex cost functions. In Jozefowska and Weglarz (Eds.), *Perspectives in Modern Project Scheduling*, International Series in Operations Research & Management Science, 92, Springer, Berlin, pp. 225–248.

Neumann, K., Schwindt, C., & Zimmermann, J. (2003). *Project Scheduling with Time Windows and Scarce Resources*, 2nd ed.. Berlin: Springer Verlag.

Pamay, M. B. (2011). A linear programming based method for the resource constrained multi-project scheduling problem with weighted earliness/tardiness costs, MSc thesis, Sabanci University, Turkey, 2011. http://research.sabanciuniv.edu/17694/.

Payne, J. H. (1995). Management of multiple simultaneous projects: A state-of-the-art review. *International Journal of Project Management*, *13*, 163–168.

Pinedo, M., & Singer, M. A (1999). Shifting bottleneck heuristic for minimizing the total weighted tardiness in a job shop. *Naval Research Logistics*, *46*(1), 1–17.

Schwindt, C. (1995). ProGen/max: A new problem generator for different resource-constrained project scheduling problems with minimal and maximal time lags, Technical Report WIOR 449, University of Karlsruhe.

Singer, M. (2001). Decomposition methods for large job shops. *Computers & Operations Research*, *28*(3), 193–207.

Vanhoucke, M., Demeulemeester, E., & Herroelen, W. (1999). An exact procedure for the unconstrained weighted earliness tardiness project scheduling problem, Research Report 9907, Department of Applied Economics, Katholieke Universiteit Leuven, no. 9907.

Vanhoucke, M., Demeulemeester, E., & Herroelen, W. (2001). An exact procedure for the resource-constrained weighted earliness tardiness project scheduling problem. *Annals of Operations Research*, *102*, 179–196.

Vanhoucke, M. (2002). Optimal due date assignment in project scheduling, Working Paper, Ghent University and Vleric Luevent Gent Management School, no. 159.

Vanhoucke, M., Demeulemeester, E., & Herroelen, W. (2003). RanGen: A random network generator for activity-on-the-node networks, Tech. Rep., Department of Applied Economics, Katholieke Universiteit Leuven.

Yang, K. K., & Sum, C. (1997). An evaluation of due date, resource allocation, project release, and activity scheduling rules in a multi-project environment. *European Journal of Operational Research*, *13*, 139–154.

Chapter 11
Multimode Resource-Constrained Project Scheduling Problem Including Multiskill Labor (MRCPSP-MS) Model and a Solution Method

Mónica A. Santos and Anabela P. Tereso

Introduction

The resource-constrained project scheduling problem (RCPSP) that we address in this chapter involves multilevel activities where each activity can be processed using one of several modes that are available for each resource, with each mode of a resource belonging to a different skill level and incurring different cost and duration. This class of problems is an extension of RCPSP, which has been shown to be nondeterministic polynomial-time (NP)-hard (Blazewicz et al. 1983). Some of the earlier methods proposed for the solution of the project scheduling problem include: critical path method (CPM; Kelley and Walker 1959) and program evaluation and review technique (PERT; MacCrimmon and Ryavec 1964; Clark 1962), resource allocation method (Davis 1966), resource leveling procedures (Bandelloni et al. 1994; Zimmermann and Engelhardt 1998), Monte-Carlo simulation-based methods (Metropolis et al. 1953; Ragsdale 1989), and those based on criticality indices (Dodin and Elmaghraby 1985). Ever since, there have been mathematical models proposed for more complex problems. The search for optimal solutions for the RCPSP has focused on the use of integer programming (IP; Pritsker et al. 1969; Berthold et al. 2010; Nemhauser and Wolsey 1988), dynamic programming (DP; Bellman and Dreyfus 1959, Elmaghraby et al. 1992), and branch and bound (B&B) techniques.

The presence of binary variables in a problem has led researchers to develop B&B-based procedures for its solution. The success of this technique depends on the branching scheme and on the tightness of the bound used. Kis (2005) explored the scheduling problem where the need for resources for each activity varies in proportion to the intensity of the activity itself. To formalize the problem, they used an integer

M. A. Santos (✉) · A. P. Tereso
University of Minho, 4710–057 Braga, Portugal
e-mail: monica.lasalete@gmail.com

A. P. Tereso
e-mail: anabelat@dps.uminho.pt

P. S. Pulat et al. (eds.), *Essays in Production, Project Planning and Scheduling,*
International Series in Operations Research & Management Science 200,
DOI 10.1007/978-1-4614-9056-2_11, © Springer Science+Business Media New York 2014

linear programming model and proposed a B&B-based algorithm to find an optimal solution. However, B&B procedures are inadequate for real-size problems, despite their efficiency relative to a frontal attack on the discrete optimization problem. The need to solve real problems in reasonable central processing unit (CPU) times, have led researchers to develop heuristic procedures.

The heuristics used belong to one of two classes: priority rule-based methods or metaheuristic approaches. The priority rule-based methods build a plan by selecting activities from a range of activities available successively so that all activities are sequenced (Boctor 1993; Dean et al. 1992; Heilmann 2000). The metaheuristic-based methods begin with an initial solution and then search for its improvement by defining an appropriate neighborhood. There are still two types of heuristics—series heuristics where the priority of the activities is predetermined and remains fixed, and parallel heuristics where the priority is updated each time an activity is scheduled for processing. Other types of heuristics found in the literature, considered as a subfield of metaheuristics, are tabu search (TS; Arroub et al. 2010), simulated annealing (SA; Mika et al. 2005), and genetic algorithms (GA; Gonçalves et al. 2004). Tseng (2008) has also discussed the use of GA applied to the multiproject, multimode RCPSP.

For the multimode RCPSP, it has been shown that for highly resource-constrained projects with more than 20 activities and three modes in each activity it is difficult to find optimal schedules (Hartmann 2001). The heuristic methods are simple to understand, easy to apply, and are capable of clarifying the scheduling process. Typically, heuristic methods consider each activity's impact on a specific objective by sorting the activities competing for resources and allocating only some of them the resources needed for scheduling in a period. Besides, metaheuristic make few or no assumptions about the problem and have the advantage of performance consistency and the ability to determine global optimal solutions. However, the search for an optimal solution within feasible solutions makes metaheuristic methods spend more computational time than heuristic methods (Zhang et al. 2006).

The objective of our study is to minimize the total project cost given a due date that includes a bonus for early completion or a penalty for tardiness. In several resource-constrained scheduling problem models found in the literature, there are two important aspects present in any model: the objective and the constraints. The objective may be based on time, such as minimizing the project duration (Boctor 1990; Heilmann 2000; Basnet et al. 2001), or on economic aspects, such as minimizing the project cost (Tereso et al. 2004, 2006; Mika et al. 2005). However, time-based objectives are often in conflict with cost-based objectives. A recurrent situation encountered in practice is the need to complete a project by its due date and maximize profit. Ozdmar and Ulusoy (1995) reported in their survey of the literature studies where the net present value (NPV) is maximized while the due date is a "hard" constraint (Patterson et al. 1989, 1990). There are several other multiobjective studies in the literature where efficient solutions regarding time and cost targets are generated. Guldemond et al. (2008) presented a study related to the problem of scheduling projects with strict deadline jobs, defined as a time-constrained project scheduling problem (*TC*PSP) where a nonregular objective function is used. The original RCPSP uses regular objective functions, like minimizing the makespan, but several

nonregular objectives have become popular like maximizing NPV. Vanhoucke et al. (2000) define regular and nonregular measures of performance: "A regular measure of performance is a nondecreasing function of the activity completion times, while for a nonregular measure of performance the above definition does not hold."

Kazaz and Sepil (1996) have presented a mixed integer linear program (MIP) formulation with Benders decomposition for a project scheduling problem where the cash flows do not occur at some event realization times, but as progress payments at the end of some time periods. In Sepil and Ortaç (1997), three different heuristic rules were developed to solve the same problem, extended with renewable resource constraints. Padman and Dayanand (1997) allow the decision-maker to set progress payment points and Etgar et al. (1997) incorporate elements of bonus/penalty structures. As the costs incurred depend on the activities in progress, while scheduling is based on noncost-related considerations, the researchers explicitly included cash flows resources constraints in their formulations. Elmaghraby and Herroelen (1990) employed the following property of an optimal solution that maximizes the NPV: the activities with positive cash flows should be scheduled as soon as possible and those with negative cash flow as late as possible. They argue that the faster completion of the project is not necessarily the optimal solution with regard to maximizing the NPV. In the study by Mika et al. (2005), a positive flow is associated with each activity. The objective is to maximize the NPV of all cash flows of the project. They used two metaheuristics that are widely used in research: SA and TS. Our problem objective is cost-based and we have a bonus/penalty structure, but we do not consider the NPV objective.

In Ulusoy and Cebelli's (2000) approach to payment scheduling problem (using a multimode RCPSP), the amount and timing of the payments made by the client and received by the contractor are determined so as to achieve an equitable solution. An equitable solution is defined as one where both the contractor and the client deviate from their respective ideal solutions by an equal percentage. The ideal solutions for the contractor and the client result from having a lump sum payment at the start and the end of the project, respectively. A double-loop GA is proposed to solve an equitable solution. The outer loop represents the client and the inner loop the contractor. The inner loop corresponds to a multimode RCPSP with the objective of maximizing the contractor's NPV for a given payment distribution. When searching for an equitable solution, information flows between the outer and inner loops regarding the payment distribution over the event nodes and the timing of these payments.

Willis (1985) described requirements for modeling realistic resources. These requirements include the variable need of resources according to the duration of the activity, variable availability of resources over the period of the project, and different operational modes for the activities. A discrete time/resource function implies the representation of an activity in different modes of operation. Each mode of operation has its own duration and amount of renewable and nonrenewable resource requirements.

The number of activities in a project determines the size of the project, and also, it contributes to the complexity of a scheduling problem. Another fact contributing to the complexity of a project is the precedence relations among the activities. As

shown by Elmaghraby and Herroelen (1980), projects with the same number of activities and the same number of activity relationships can have varying degrees of difficulty. So, there is not a straightforward association between project complexity and problem difficulty. Meanwhile, many project complexity measures have been proposed (Herroelen 2006).

Boctor (1993) presented a heuristic procedure for the scheduling of nonpreemptive resource-constrained projects where the resource is renewable from period to period. Each activity is assumed to have a set of possible durations and resource requirements. The objective is to minimize the project duration. The heuristic used belongs to the class of priority rule-based methods. This class builds a plan by selecting activities from a range of activities available successively so that all activities are sequenced. A general framework to solve large-scale problems was suggested. The heuristic rules that can be used in this framework were evaluated, and a strategy to solve these problems efficiently was designed. Heilmann (2000) also worked with the multimode case in order to minimize the duration of the project. In his work, besides the different modes of execution of each activity, a maximum and minimum delay between activities is specified. He presented a priority rule-based heuristic. Basnet et al. (2001) presented a "filtered beam" search (FBS) technique to generate makespan minimizing schedules for multimode single resource-constrained projects where there is a single renewable resource to consider and the multimode consists essentially of how many people can be employed to finish an activity.

The problem presented here also belongs to the class of project scheduling problems with multilevel (or multimode) activities, with each activity being processed by resources operating at appropriate modes where each mode belongs to a different resource skill level, which implies different cost and duration. Usually, multimode RCPSP defines an explicit set of modes for each activity, with a specific activity duration and resource requirements. Our approach, however, defines a set of resource levels. Each activity may elect a level for each one of the resources required. The combination of all possible levels of each resource, required for the execution of the activity, provides the alternative modes of execution for each activity.

Santos and Tereso (2010) presented a formal multimode resource-constrained project scheduling problem including multiskill labor (MRCPSP-MS) model and a breadth-first search (BFS) procedure. Consequently, Santos and Tereso (2011a, b) presented an adaptation of an FBS scheme to this problem and reported results of a preliminary computational investigation. In this chapter, we present further analysis on the results obtained for networks of different sizes.

In a BFS scheme, all the nodes (partial solutions) in the search tree are evaluated at each stage before going any deeper, subsequently realizing an exhaustive search that visits all nodes of the search tree. The B&B search technique can be seen as a polished BFS, since it applies some criteria in order to reduce the BFS complexity. Usually, it consists of keeping track of the best solution found so far, discarding a node if it cannot offer a better solution. FBS is a heuristic B&B procedure that uses BFS, but only the top best nodes are kept. At each stage of the tree, all successors for the selected nodes at the current stage are generated, but it only stores a preset number of descendent nodes at each stage, called the beam width.

Table 11.1 Problem characteristics with three levels

Resources	Processing time (*j*,*r*,*l*)		
Activities	$1(b_1)$	.....	$\|R\| = \rho\ (b_\rho)$
1	(1,1,1) (1,1,2) (1,1,3)	...	$(1,\rho,1)\ (1,\rho,2)\ (1,\rho,3)$
2	(2,1,1) (2,1,2) (2,1,3)	...	$(2,\rho,1)\ (2,\rho,2)\ (2,\rho,3)$
⋮	⋮	⋮	⋮
$\|A\| = m$	(*m*, 1,1) (*m*, 1,2) (*m*, 1,3)	...	$(2,\rho,1)\ (2,\rho,2)\ (2,\rho,3)$

b_r the number of units available of resource r, (j,r,l) the processing time $p(j,r,l)$ for activity j of resource r at level l

The B&B and the beam search (BS) procedures have been typically applied to the RCPSP (Basnet et al. 2001; Demeulemeester and Herroelen 1996; Kis 2005). The differentiating aspects of our approach are: (a) the definition of a set of states followed by the activities, (b) the priority rules used to solve resource conflicts, and (c) the alternative evaluation rules used to discard undesirable "branches."

Our approach allows determination of a project solution using the BFS scheme or the FBS scheme. We implemented the proposed approach using C# language.

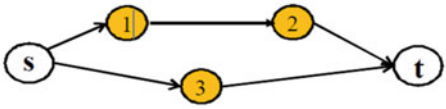
Problem Description

Consider a project network in the activity-on-arc (AoA) representation: $G = (N, A)$, with $|N| = n$ (representing the events) and $|A| = m$ (representing the activities). Each activity may require the simultaneous use of several resources with their consumption dictated by the selected execution mode—each resource may be deployed at a different level. The objective is to determine the optimal allocation of resources to the activities in order to minimize the total cost incurred (due to resources + penalty for tardiness + bonus for earliness). We follow the dictum that an activity should be initiated as soon as it is sequence feasible.

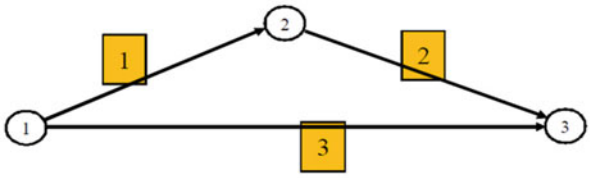
There are $|R| = \rho$ resources. A resource has a capacity of several units (say workers or machines) and may be used at different levels, such as a "resource" of electricians of different skill levels, or a "resource" of milling machines but of different capacities and ages. A level might be the power of usage of a machine: high, medium, or low, or the amount of hours used by a resource: half-time, normal time, or extra-time. Another example would be a "resource" of routes where the levels could be: easy, short, fast, or economic.

The different levels of a resource $r \in R$ may be represented as $L(r) = \{r_1, \ldots r_{L(r)}\}$; the number of levels varies with the resource. If resource r is utilized at level l for activity j then the processing time shall be denoted by $p(j,r,l)$. An activity normally requires the simultaneous utilization of more than one resource for its execution, but each activity must be allocated exactly one unit of each resource.

To better visualize the problem, one can summarize its characteristics in a matrix format as shown in Table 11.1. For illustrative purposes, we assume that any resource



Project with 3 activities. AoN



AoA representation.

Fig. 11.1 A project with three activities

may have at most three levels: low (level 1), average or normal (level 2), and high (level 3). A cell entry in the matrix is the processing time $p(j,r,l)$ for activity j of resource r at level l. Due to space limitations, Table 11.1 exhibits the information as (j,r,l); the symbol "p" is forfeited. If an activity does not require a resource, it is indicated in the matrix by the symbol $\emptyset$ (null). The symbol b_r gives the number of units available of resource r.

The processing time of an activity is given by the maximum of the durations that would result from a specific allocation of various resources. To better understand this representation, consider the miniscule project of Fig. 11.1 with three activities. Assume that the project requires the utilization of four resources.

Let $\gamma(r,l)$ be the unitary cost of resource r at level l. Then, in each cell, there shall be a $2 \times |L(j,r)|$ matrix in which the first row gives the duration $p(j,l)$ and the second row the cost $c(j,r,l)$. The total cost of a resource, at each level, is obtained as the product of the resource unitary cost with the activity processing time.

For brevity, Table 11.2 gives only the processing times and costs at each level of the resources; the identity of the activity and the resource are suppressed.

The top row indicates that there are four resources with varying availability: resources 1 and 4 have $b1 = b4 = 2$ units each, resource 2 has only one unit ($b2 = 1$), and resource 3 has $b3 = 3$ units. The following row specifies a cost rate for each level of the resource. A positive entry in the row corresponding to activity j indicates the resource(s) required by the activity—each activity must be allocated exactly one unit of each resource. However, the resource may be utilized at any of its specified levels. For instance, resource 2 has only one unit available, which can be utilized at any of its three levels: if, for activity 2, it is utilized at level 1 (the lowest level), then the processing time is 7 (i.e., $p(2,2,1) = 7$) and the cost is 14 (i.e., $c(2,2,1) = 14$); if it is utilized at level 2 (the intermediate level), then the processing time is 5 and the cost is 25; finally, if it is utilized at level 3 (the highest level), then the processing time is 3 and the cost is 30; etc.

Table 11.2 Project resource requirements, processing times, and resource costs

↓Act	Resources (availability) → Resource unit cost for each level	1 (2) (1,3)	2 (1) (2,5,10)	3 (3) (1,3,5)	4 (2) (2,4,6)	η_j
1	$p(j,r,l)$	(14,6)	Ø	(12,8,5)	(18,12,7)	3
	$c(j,r,l)$	(14,18)	(12,24,25)	(36,48,42)		
2	$p(j,r,l)$	Ø	(7,5,3)	Ø	(8,5,4)	2
	$c(j,r,l)$	(14,25,30)	(16,20,24)			
3	$p(j,r,l)$	(20,12)	(22,16,10)	Ø	Ø	2
	$c(j,r,l)$	(20,36)	(44,80,100)			

$p(j,r,l)$ the processing time $p(j,r,l)$ for activity j of resource r at level l, $c(j,r,l)$ the total cost of activity j of resource r at level l, η_j count of resources required for activity j

Suppose that all four resources are allocated at their respective level 2 (normal intensity). Letting p_j denote the duration of activity j and c_j the total cost of activity j, we get:

- Activity 1 shall take $p1 = max\{6, 8, 12\} = 12$ time units; $c_j = 18+24+48 = 90$ monetary units
- Activity 2 shall take $p2 = max\{5, 5\} = 5$ time units; $c_j = 25+20 = 45$ monetary units
- Activity 3 shall take $p3 = max\{12, 16\} = 16$ time units; $c_j = 36 + 80 = 116$ monetary units

And the project shall consume $max\{12+5, 16\} = 17$ time units to complete. However, due to resource restrictions, activities 2 and 3 cannot be executed at the same time since resource 2 has only one unit which must be allocated to either activity. So, if allocated first to activity 2, the project should consume $max\{12 + 5, 17 + 16\} = 33$ time units; if allocated first to activity 3, the project should consume $max\{16 + 5, 16\} = 21$ time units. The latter decision should be the preferred one. So, the total resource cost of the project shall be 251 monetary units and will be finished at time $T = 21$ (assuming the project started at time $T = 0$). We also consider lateness costs and earliness gains (negative costs). If the project's specified due date is $Ts = 24$, the project will finish early. If the unitary cost for earliness is equal to -10, the total cost will be $251 - 10*3 = 221$.

Mathematical Model

Briefly, the constraints of this problem are:

- Precedence relationships among the activities.
- A unit of a resource is allocated to at most one activity at any time (the unit of the resource may be idle during an interval) at one level.
- Capacity of each resource: the number of units allocated for processing at any time should not exceed the capacity of the resource to which these units belong.

- An activity can be started only when it is sequence feasible and all the requisite resources are available, each perhaps at its own level, and must continue at that level for all the resources without interruption or preemption.

The objective is to find an optimal solution that minimizes the overall project cost, while respecting a specified delivery date. A penalty is incurred for tardiness beyond the specified delivery date, or a reward is secured for early completion.

Consider the following variables:

Input variables

$G(N, A)$: project network in AoA representation with a set N of nodes and a set A of activities.
n: number of nodes; $n = |N|$.
m: number of arcs or number of activities; $m = |A|$.
(i, j): activity, represented by arc (i, j).
r: resource $r \in R$.
L_r: set of levels for resource r.
$\eta_{i,j}$: the count of resources required by activity (i, j).
ρ: number of resources, $\rho = |R|$.
b_r: capacity of resource r.
$\gamma(r, l)$: marginal cost of resource $r$at level l (US$/period).
γ_E: marginal gain from early completion of the project (US$/period).
γ_L: marginal loss (penalty) from late completion of the project (US$/period).
$p(i, j, r, l)$: the processing time of activity (i, j) when resource r is allocated at level l (time period).
T_S: target completion time of the project (time period).

State variables

C^k: the kth uniformly directed cutset (udc) of the project network that is traversed by the project progression (i.e., a set of ongoing activities); $k = 1, \ldots, K$.
t_i: time of realization of node i (AoA representation) where node 1 is the "start node" of the project and node n its "end node" (time period).

Decision variables

$x(i, j, r, l)$: a binary variable, of value 1 if resource r is allocated to activity (i, j) at level l, and 0 otherwise. l is the level at which a resource is applied to an activity $l \in L_r$.

Output variables

c_E: earliness cost (US$).
c_T: tardiness cost (US$).
c_{ET}: earliness-tardiness cost (US$).
c_R: total resource cost for all project activities (US$).
TC: total cost of the project (US$).

Next, we present the relevant constraints.

We begin by defining the processing time of an activity as the maximum of the processing times imposed by the different resources. These processing times will be a function of the levels at which the resources required by the activity are allocated, and an activity cannot start before all the preceding activities have finished; we have

$$t_j - t_i \geq \max\{p(i,j,r,l) * x(i,j,r,l)\}, \forall i,j \in N, \forall r \in R, \forall l \in L_r \tag{11.1}$$

The total units of a resource allocated at any time to all the activities should not exceed the capacity of the resource to which these units belong. This restriction is applicable to the activities that are concurrently active (i.e., ongoing), which must lie in the same *udc*.

The total allocation of resource r to the active activities in the "current" *udc* C^k cannot exceed its available capacity

$$\sum\nolimits_{i,j \in C^k} x(i,j,r,l) \leq b_r, \forall r \in R, \forall l \in L_r, k = 1, \ldots, K \tag{11.2}$$

A unit of a resource is allocated to an activity at only one level (the unit of the resource may be idle during an interval of time):

$$\sum\nolimits_{l \in L_r} x(i,j,r,l) = 1, \forall i,j \in N, \forall r \in R \tag{11.3}$$

An activity must be allocated all the resources it needs at some level, at which time it can be started and must continue at the same level for all the resources without interruption or preemption. This requirement is represented as follows:

$$\eta_{i,j} - \sum\nolimits_{r \in R} \sum\nolimits_{l \in L_r} x(i,j,r,l) = 0, \forall i,j \in C^K \tag{11.4}$$

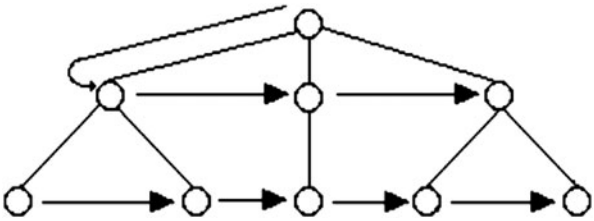
The difficulty in implementing this constraint set stems from the fact that we do not know a priori the identity of the udcs that shall be traversed during the execution of the project since that depends on resource allocation. The allocation of the resources is bounded by their availabilities at each udc, but the latter cannot be known until after the allocation of resources has been determined. Enumerating all the udcs and constraining the resources' usage at each one would overconstrain the problem (see Ramachandra and Elmaghraby 2006). There are several ways to resolve this difficulty, formal as well as heuristic. The formal ones are of the IP genre, which, when combined with the nonlinear mathematical programming model presented above, present a formidable computing burden. On the other hand, the heuristic approaches are more amenable to computing.

The objective function is composed of two parts: the cost of use of the resources, and the gain or loss due to earliness or tardiness, respectively, of the project completion time (t_n) relative to its due date.

Earliness and tardiness (delay) are defined by:

$$e \geq T_s - t_n \tag{11.5}$$

Fig. 11.2 BFS traversal



$$d \geq t_n - T_s \tag{11.6}$$

$$e, d \geq 0 \tag{11.7}$$

The costs may be evaluated as follows:

1. The cost of resource utilization in the selected level for each activity is:

$$c_R(i,j) = \sum\nolimits_{r \in R} \sum\nolimits_{l \in L_r} c(i,j,r,l) * x(i,j,r,l) \tag{11.8}$$

$$c(i,j,r,l) = \gamma(r,l) * p(i,j,r,l) \tag{11.9}$$

2. The earliness/tardiness costs are:

$$c_{ET} = c_E + c_T = \gamma_E.e + \gamma_L.d \tag{11.10}$$

3. Total resources cost for all activities of the project:

$$c_R = \sum\nolimits_{i,j \in N} c_R(i,j) \tag{11.11}$$

4. Total cost of the project:

$$TC = C_R + C_{ET} \tag{11.12}$$

The desired objective function may be written simply as:

$$\min TC \tag{11.13}$$

Solution Method

Our initial approach to solve the problem on hand relies on a BFS scheme. In the BFS scheme, all the nodes (partial solutions) in the search tree are evaluated at each stage before going any deeper (Fig. 11.2), subsequently realizing an exhaustive search that visits all nodes of the search tree. The B&B search technique can be seen as a

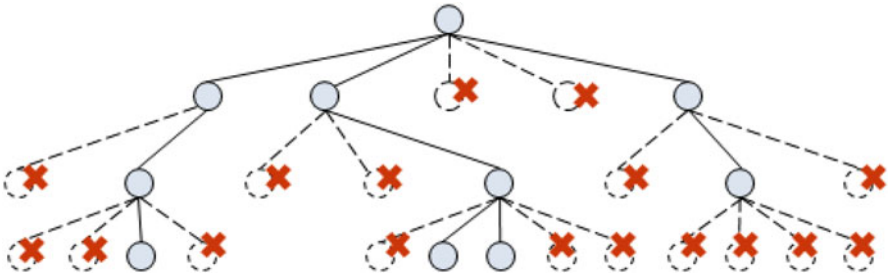


Fig. 11.3 A beam search tree with beam width = 3

polished BFS, since it applies some criteria in order to reduce the complexity of the BFS scheme. Usually, it consists of keeping track of the best solution found so far and checking if the solution given by that node is better than the best known solution. So, if that node cannot offer a better solution than the solution obtained so far, the node is fathomed. The B&B approach is more efficient if the bounds are tight.

The B&B process consists of two procedures:

1. Subset generation
2. Subset elimination

The former (subset generation) is accomplished by branching where a set of descendent nodes is generated, thereby creating a tree-like structure. The latter (subset elimination) is realized through either bounding where upper and lower bounds are evaluated at each node, or via feasibility checking where the extension of a partial solution is deemed infeasible and the branch is fathomed. A bound can be strong, which is usually harder to calculate but it accelerates finding an optimal solution, or it can be weak, which is easier to calculate but makes it slower to find the desired solution.

In our case, the objective is to minimize the total cost encountered that is based on the bonus achieved or the penalty cost incurred while respecting or exceeding the specified due date, respectively. As a result, finding a bound depends on the following three project parameters cited: the penalty cost, bonus cost, and due date. As noted above, a bound helps in reducing the search while not discarding potentially desirable branches.

FBS is a heuristic B&B procedure that uses BFS but only the top "best" nodes are kept. At each stage of the tree, it generates all successors for the selected nodes at the current stage, but only stores a predetermined number of descendent nodes at each stage, called the beam width (Fig. 11.3).

In the proposed procedure, we have the option of using either the BFS or the FBS scheme. For the latter, we need to specify an appropriated beam width.

We consider the activities to be in one of four states: "to begin," "pending," "active" (i.e., ongoing), and "finished." To get the first activities with which to initiate the process, we search all activities that do not have any predecessors. These activities are set to the state "to begin." All others are set to the state "pending."

Activities in the state "to begin" are analyzed in order to check resources availability. If we have enough resources, all activities in the state "to begin" are modified to the state "active"; otherwise, we apply in sequence the following rules until the resources conflicts are resolved:

1. Give priority to activities that are precedents to a larger number of "pending" activities.
2. Give priority to activities that use fewer resources.
3. Give priority to activities in sequence of arrival to the state "to begin."

An "event" represents the starting time of one or more activities, and the project begins at event 0 in which no activity has started yet. Each activity must be allocated exactly one unit of each resource. For each active activity, we calculate all the possible combinations of levels of resources. Then, we aggregate all these combinations to get the initial combinations of allocation modes for all "active" activities. These initial combinations form branches through which we will get possible solutions for the project. All combinations have a copy of the resource availability information and activities' current state.

If the FBS scheme is selected to obtain a solution, then:

1. If the number of combinations is less than the beam width value, all combinations are kept.
2. Otherwise, the set of combinations must be reduced to the beam width value; so, some combinations need to be discarded. To evaluate the best combinations, we may pick one of the following rules:
   Select the top best combinations that have:
   a. Minimum duration
   b. Minimum cost
   c. Minimum cost/duration

   Not all combinations of the set can be directly compared because the number of activities that have been scheduled in each combination may differ. So, the combinations are grouped by the number of activities that have already been scheduled.

   Then, the combinations are compared with the others that belong to the same group. The final set is composed by a share of combinations of each group formed before.

   The ratio of each group in the final combinations set is calculated by:

$$\text{ratio} = \frac{\text{groupcount}}{\text{totalcombinations}} \tag{11.14}$$

In either case, we continue applying the following procedure to each combination:

3. To all activities in progress, we find the ones that will be finished first, and set that time as the next event.
4. We update activities found in step 1 to state "finished" and release all the resources being used by them.

5. For all activities in the state "to begin," we seek the ones that can begin, the same way we did when initiating the project. Activities in the state "to begin" are analyzed in order to check resource availability. If no resource conflict exists, all activities in the state "to begin" are set to state "active" and resources are set as being used; otherwise, we apply in sequence the rules described above.
6. For all activities in the state "pending," we check for precedence relationships. For all activities that are precedence feasible, their state is updated to state "to begin." These activities are not combined with the previous set of "to begin" activities to give priority to activities that entered first in this state.
7. If there are resources available and any pending activities were set "to begin," we apply step 5 again.
8. For all new "active" activities, we set their start time to the next event found in step 3 and determine all the possible combinations of its resource levels. Then we join all found combinations for these activities, getting new combinations to add to the actual combination being analyzed. This generates new branches for investigation.
9. We continue by applying step 1 (or 3) to each new combination until all activities are set to state "finished."
10. When all activities reach the "finished" state, we obtain a valid solution for the problem.

We evaluate the project completion time and the total cost incurred for all complete solutions, and choose the best among them.

A flowchart of the proposed solution method is shown in Fig. 11.4.

Computational Results and Analysis

The proposed solution method was implemented in C#, an object-oriented programming (OOP) language, using Visual Studio 2010. To construct the project network (in AoN), we used Graph#, an open source library for. Net/WPF applications that is based on a previous library QuickGraph. These libraries support GraphML that is an XML-based file format for graphs although we defined our own particular xml format.

The following computational tests were performed on an Intel® Pentium® M @1.20 GHz 1.25 GB RAM.

Three-Activity Network

Consider a three-activity network for four resources, one with two levels and the others with three different levels. Assume the following parameter values for earliness and lateness costs: $\gamma_E = -10$, $\gamma_L = 20$, and the due date $T_S = 24$ (Table 11.3).

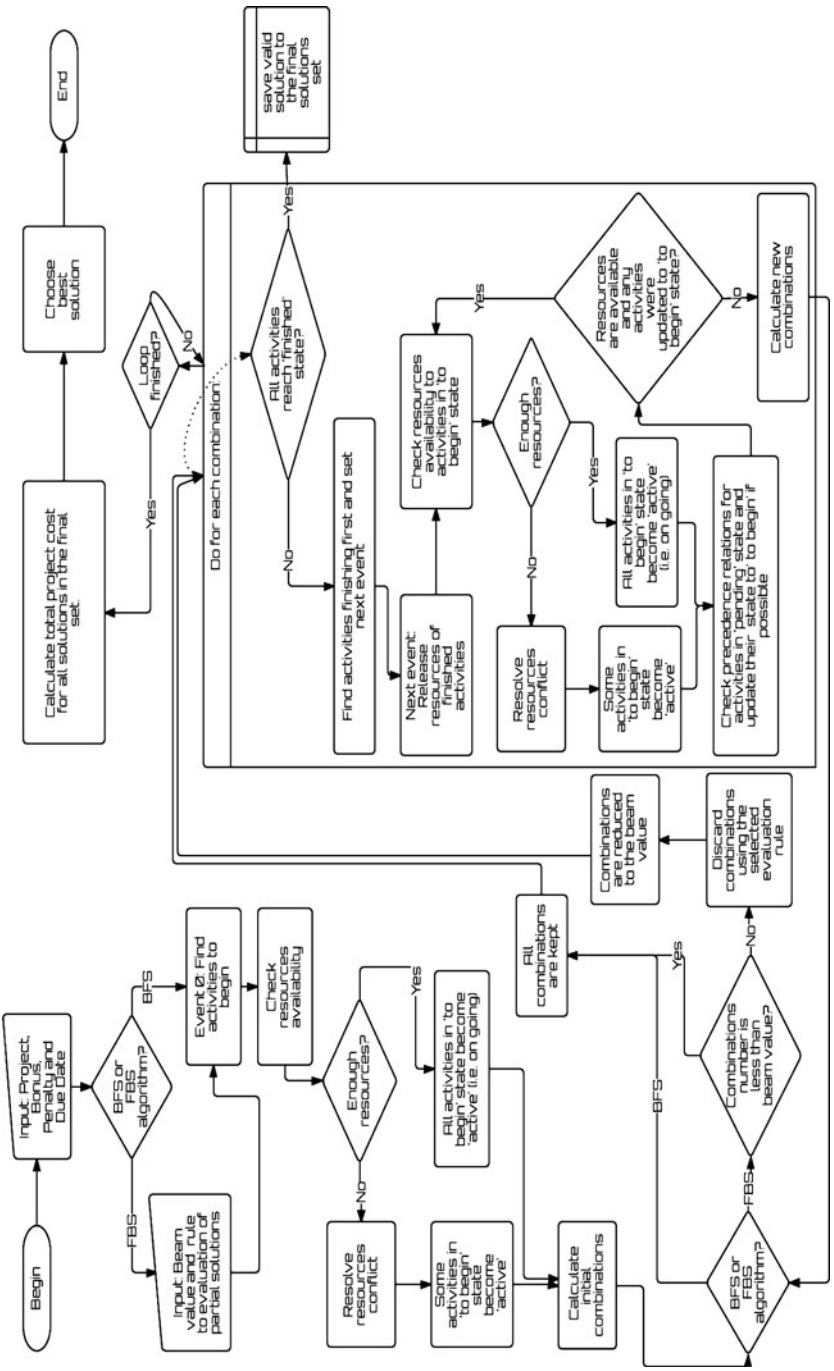


Fig. 11.4 A flowchart of the proposed solution method

Table 11.3 Three-activity network solution values obtained using the BFS scheme

t_n	C_E	C_T	C_R	TC	Runtime (ms)
16	80	0	230	150	44

The BFS scheme generates 972 combinations for the three-activity network. We used a beam width between 20 and 900. As we can see by the results exhibited in Table 11.4, the "duration" evaluation type was the best for this network, achieving the same result that was obtained for the BFS scheme, even for the smaller beam width. The evaluation type "cost" performed better than the "cost/duration," which did not achieve the best solution even with a beam width of 900. However, "cost/duration" evaluation gave the lowest project cost when the bonus or penalty $C_R = 193$ was not considered.

As can be observed from Fig. 11.5, the quality of the solutions achieved increases, i.e., the value of the TC decreases or remains equal with increase in the beam width value. This does not happen for the project cost C_R. On the contrary, this variable is highest for the best solutions. The reason for this difference has to do with the bonus and due date specified. These values make us achieve best solutions with unprofitable C_R values, because these complete the project earlier. If the earliness and lateness costs where: $\gamma_E = 0$, $\gamma_L = 0$, the best solution would be $C_R = 189$, $TC = 189$, and $t_n = 30$.

Five-Activity Network

Consider a five-activity network with the same resources as the three-activity network above. Assume the following parameter values for earliness and lateness costs: $\gamma_E = -10$, $\gamma_L = 20$, and the due date $T_S = 30$ (Table 11.5).

The BFS scheme generates 104,976 combinations for the five-activity network. We varied beam width between 50 and 100,000. As we can see by the results exhibited in Table 11.6, the "duration" evaluation type was faster in reaching results similar to the ones obtained for the BFS scheme. The other evaluation types are far from the solution obtained for the BFS algorithm using lowest beam widths but achieved better C_R (project cost without bonus or penalty) values, $C_R = 323$ for "cost" and $C_R = 315$ for "cost/duration" type (Fig. 11.6).

Again, we obtain better TC for worst C_R values, but in this case, the bonus is never materialized. For this project, our method could not find a solution with a completion time less than or equal to the due date 30; the earliest time for completion is 36. If the earliness and lateness costs where $\gamma_E = 0$, $\gamma_L = 0$, the best solution would be $C_R = 315$, $TC = 315$, and $t_n = 68$.

A plot of the TC and C_R values obtained for the BFS scheme, and $\gamma_E = -10$, $\gamma_L = 20$, against several due dates, is shown in Fig. 11.7.

Note that starting from $T_S = 36$, the total project cost takes advantage of the bonus independent of the particular bonus and penalty cost parameter values. So,

Table 11.4 Three-activity network solution values obtained using the FBS scheme

Beam width	Evaluation type																	
	Cost						Duration						Cost/duration					
	t_n	C_E	C_T	C_R	TC	Runtime (ms)	t_n	C_E	C_T	C_R	TC	Runtime (ms)	t_n	C_E	C_T	C_R	TC	Runtime (ms)
20	25	0	20	200	220	8	16	80	0	230	150	2	29	0	100	196	296	11
50	24	0	0	205	205	14	16	80	0	230	150	4	28	0	80	193	273	22
200	20	40	0	218	178	51	16	80	0	230	150	68	26	0	40	201	241	61
450	16	80	0	230	150	198	16	80	0	230	150	319	24	0	0	205	205	177
700	16	80	0	230	150	162	16	80	0	230	150	112	22	20	0	213	193	116
900	16	80	0	230	150	431	16	80	0	230	150	134	19	50	0	225	175	197

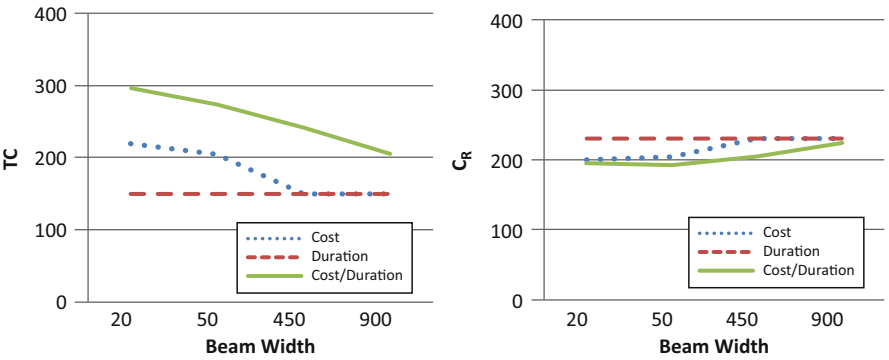


Fig. 11.5 Variations of TC and C_R values versus beam width for evaluation types cost, duration, and cost/duration

Table 11.5 Five-activity network solution values obtained using the BFS scheme

t_n	C_E	C_T	C_R	TC	Runtime (s)
36	0	120	400	520	18

for this project, it would be profitable to adjust the due date specified first to a value higher than 36.

Ten-Activity Network

Now, consider a ten-activity network for five different resources, three of them with two possible levels, one having five levels, and the last one with three elective levels. Assume the following rates for earliness and lateness costs: $\gamma_E = -15$, $\gamma_L = 20$, and the due date $T_S = 36$ (Table 11.7).

Table 11.6 Five-activity network solution values obtained using the FBS scheme

Beam width	Evaluation type																	
	Cost						Duration						Cost/duration					
	t_n	C_E	C_T	C_R	TC	Runtime (s)	t_n	C_E	C_T	C_R	TC	Runtime (s)	t_n	C_E	C_T	C_R	TC	Runtime (s)
50	63	0	660	323	983	0.03	36	0	120	400	520	0.02	68	0	760	315	1,075	0.03
500	58	0	560	333	893	0.3	36	0	120	400	520	0.31	65	0	700	319	1,019	0.24
5,000	52	0	440	353	793	3.6	36	0	120	400	520	4.4	60	0	600	329	929	3.71
10,000	48	0	360	363	723	5.3	36	0	120	400	520	8.0	58	0	560	333	893	6.2
50,000	37	0	140	395	535	17.0	36	0	120	400	520	17.8	51	0	420	360	780	14.7
100,000	36	0	120	400	520	19.3	36	0	120	400	520	25.3	42	0	240	383	623	20.0

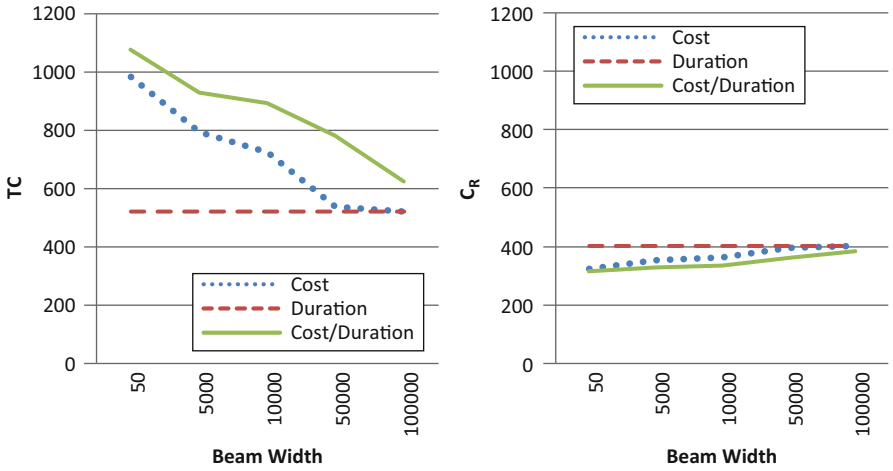


Fig. 11.6 Variations of TC and C_R values versus beam width for evaluation types cost, duration, and cost/duration

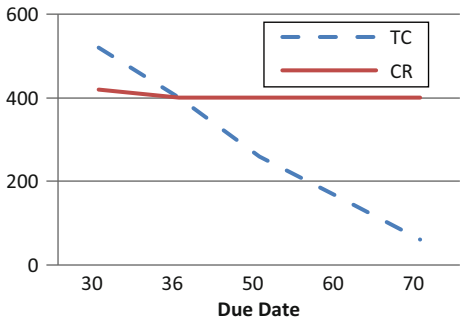


Fig. 11.7 Variation of TC and C_R values versus due dates for the BFS scheme

A solution could not be achieved in a reasonable time for the BFS scheme.

We observed a performance decrement in runtime values. This project, besides having more activities than the previous ones, also has more resources and resources levels. Therefore, it is of higher complexity than the previous ones. The evaluation type "cost" provides the best solutions, with a $TC = 360$ for a beam width of 50,000. We achieved reasonable solutions for "duration." However, weak solutions were obtained for "cost/duration."

As can be seen in Fig. 11.8, there is no significant difference between the TC and the C_R values. However, the "cost/duration" values are worse as the penalty cost for all beam width values was applied to this one.

Let us analyze the TC and C_R, obtained for different due dates, using the same γ_E, and γ_L. The beam width analyzed is the 50,000, for the "cost" and the "duration" evaluation types (Fig. 11.9).

Table 11.7 Ten-activity network solution values obtained using the FBS scheme

Beam width	Evaluation type																	
	Cost						Duration						Cost/duration					
	t_n	C_E	C_T	C_R	TC	Runtime (s)	t_n	C_E	C_T	C_R	TC	Runtime (s)	t_n	C_E	C_T	C_R	TC	Runtime (s)
50	29	15	0	406	391	0.2	24	90	0	508	418	0.2	45	0	340	456	796	0.16
500	29	15	0	406	391	3.2	26	60	0	468	408	4.1	45	0	300	444	744	3.3
1,000	29	15	0	406	391	4.9	26	60	0	468	408	6.6	45	0	300	444	744	5.6
5,000	27	45	0	417	372	17.2	26	60	0	468	408	31.3	44	0	320	447	727	19.4
10,000	27	45	0	417	372	40.5	24	90	0	490	400	50.8	44	0	280	440	720	38.1
50,000	27	45	0	405	360	723	23	90	0	480	375	635	42	0	250	451	691	355

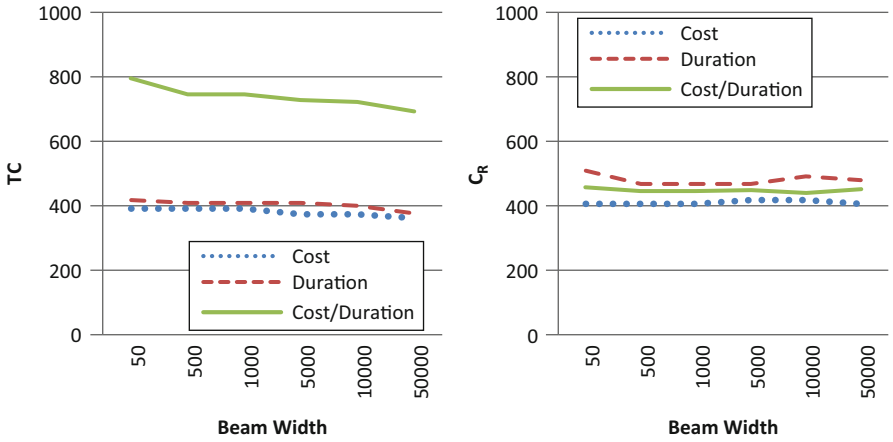


Fig. 11.8 Variations of TC and C_R values versus beam width

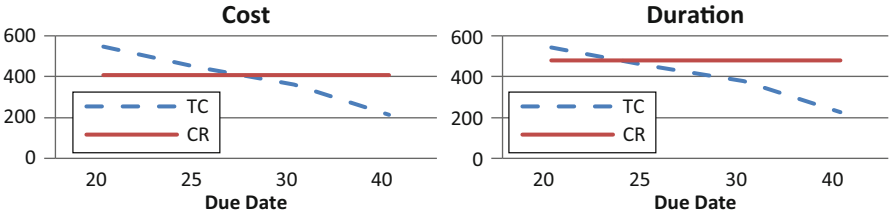


Fig. 11.9 Variation of TC and C_R values versus due dates for evaluation types cost and duration

For both evaluation types, there is a due date from which the total project cost takes advantage from the bonus; this due date is greater than 27 for "cost" and greater than 23 for "duration."

Twenty-Activity Network

Consider a twenty-activity network for the four resources with three different levels each, with the following parameter values for earliness and lateness costs: $\gamma_E = -10$, $\gamma_L = 20$, and the due date $T_S = 70$ (Table 11.8).

The proposed method could not achieve a solution using the "duration" evaluation type and beam widths of 5,000 and 10,000 because of computing memory limitations. However, the best solutions were achieved by the "duration" evaluation type $TC = 1479$. Once again, the "cost/duration" type performed poorly, and the "cost" type, even with a twenty times higher beam width, did not achieve a better solution than the one obtained by a beam width of 500. The bad performance of the "cost" type is directly related to the due date specified 70. For a due date $T_S = 135$, we would get a $TC = 846$, which is identical to the C_R obtained for a beam width of 50.

Table 11.8 Twenty-activity network solution totals obtained using FBS scheme

Beam width	Evaluation type																	
	Cost						Duration						Cost/duration					
	t_n	C_E	C_T	C_R	TC	Runtime (s)	t_n	C_E	C_T	C_R	TC	Runtime (s)	t_n	C_E	C_T	C_R	TC	Runtime (s)
50	135	0	1,300	846	2,146	2.34	70	0	0	1,502	1,502	1.9	135	0	1,300	846	2,146	1.34
500	110	0	800	902	1,702	3.17	70	0	0	1,502	1,502	10.4	129	0	1,180	865	2,045	4.34
1,000	110	0	800	902	1,702	15.9	70	0	0	1,492	1,492	26.2	129	0	1,180	865	2,045	13.7
2,000	110	0	800	902	1,702	25.0	70	0	0	1,479	1,479	164	129	0	1,180	865	2,045	23.2
3,000	110	0	800	902	1,702	31.9	70	0	0	1,479	1,479	257	129	0	1,180	864	2,044	35.4
5,000	110	0	800	902	1,702	57.3	–	–	–	–	–	–	129	0	1,180	864	2,044	51.7
10,000	110	0	800	902	1,702	79.0	–	–	–	–	–	–	129	0	1,180	864	2,044	84.2

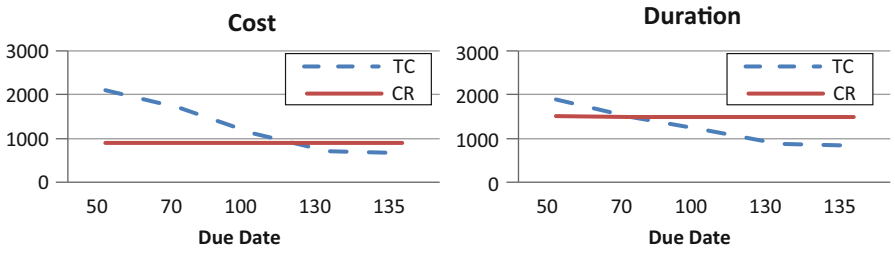


Fig. 11.10 Variations of TC and C_R versus different due dates for evaluation types cost and duration

In Fig. 11.10, we depict variations of the TC and C_R values over different due dates using the same γ_E, and γ_L values. The beam width used is 3,000 for the "cost" and the "duration" evaluation types.

The due dates, for which the total project cost takes advantage of the bonus, are 110 and 70 for "cost" type and "duration" type evaluations, respectively.

For $\gamma_E = 0$, $\gamma_L = 0$, the solution obtained for the "cost" evaluation type is $TC = C_R = 846$, and $t_n = 135$. For the "duration" evaluation type we have $TC = C_R = 1479$ and $t_n = 70$. So, according to the penalty and bonus values, there is the possibility for adjusting the due date in order to get the advantage of the best of these solutions.

Thirty-Activity Network

Next, we considered a thirty-activity network for four different resources, three of them with two possible levels and one having four levels. Assume the following rates for earliness and lateness costs: $\gamma_E = -10$, $\gamma_L = 10$, and the due date $T_S = 100$ (Table 11.9).

The best solution found for this network was $TC = 557$ with the evaluation type "cost." In the solutions achieved using the "duration" type, we obtain a better solution for the minimum beam width than for the higher ones. For a beam width of 50, the algorithm was able to preserve lowest cost solutions even with a small range of branches in the search tree. The higher beam width values gave best solutions in terms of project duration, but they were inferior in terms of total project cost calculations. Again, the "cost/duration" type achieved worse solutions than the other two evaluation types. The variations of TC and C_R values over different beam widths for each of the evaluation types are depicted in Fig. 11.11.

With $\gamma_E = 0$, $\gamma_L = 0$, the best solution for "cost" is $TC = C_R = 668$ and $t_n = 94$. For the "duration" evaluation type we have $TC = C_R = 746$ and $t_n = 87$.

Observing the graphics of the TC and C_R values obtained for different due dates using $\gamma_E = -10$, $\gamma_L = 10$, and beam width of 5,000, note that they both have the same due date from which a bonus or a penalty is applied $T_S = 87$ (Fig. 11.12).

Table 11.9 Thirty-activity network solution values obtained using the FBS scheme

Beam width	Evaluation type																	
	Cost						Duration						Cost/duration					
	t_n	C_E	C_T	C_R	TC	Runtime (s)	t_n	C_E	C_T	C_R	TC	Runtime (s)	t_n	C_E	C_T	C_R	TC	Runtime (s)
50	87	130	0	696	566	1.2	94	60	0	668	608	1.2	101	0	10	731	741	1.3
500	87	130	0	687	557	4.2	87	130	0	746	616	2.6	100	0	0	725	725	3.3
1,000	87	130	0	687	557	8.0	87	130	0	746	616	12.9	100	0	0	725	725	6.8
2,000	87	130	0	687	557	47.6	87	130	0	746	616	51.8	99	10	0	688	678	4.9
5,000	87	130	0	687	557	160	87	130	0	746	616	163	95	50	0	719	669	135
10,000	87	130	0	687	557	242	87	130	0	746	616	346	95	50	0	697	647	178

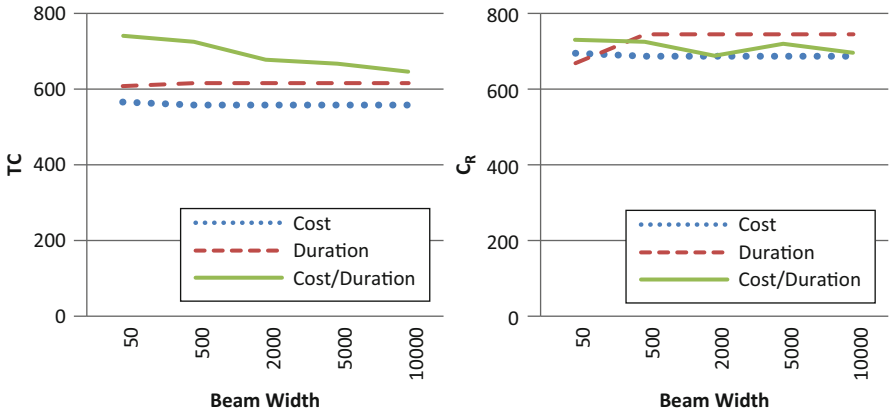


Fig. 11.11 Variation of TC and C_R versus beam width for evaluation types cost, duration, and cost/duration

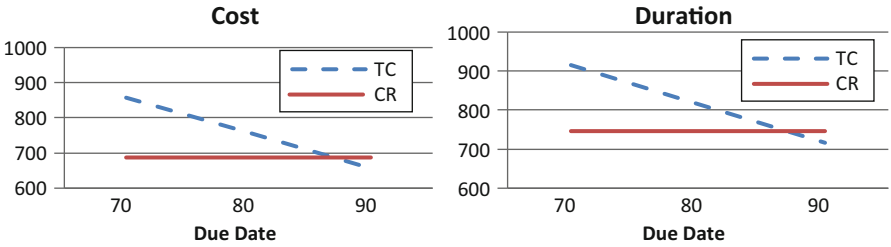


Fig. 11.12 Variation of TC and C_R versus different due dates for evaluation types cost and duration

Remark

The performance of an evaluation type seems to be intrinsically reliant on project characteristics and especially on the defined values of bonus, penalty, and due date. For complex project networks, an increase in beam width until it is computationally feasible to obtain a solution does not offer, necessarily, better solutions. For each project, there exists a specific due date beyond which the bonus or the penalty is realized. Knowing the recommended solution for a project (obtained using different evaluations types) without considering the bonus, penalty, or due date can be useful, especially when there is a possibility of negotiating their values.

Conclusions and Further Research

The RCPSP belongs to the class of NP-hard problems (Blazewicz et al. 1983). However, this problem becomes more difficult to solve when practical issues such as multiobjective, multimode, and multiproject ones are included. A heuristic-based approach is the best approach to use in such a case.

In this chapter, we have addressed a RCPSP with multiple resource modes available at different levels. Given a due date, the objective is to allocate resources to all activities of the project so as to minimize the total cost encountered because of resource utilization, plus the net gain (bonus) accrued from finishing the project earlier or the penalty incurred for finishing the project late. We have presented a mathematical formulation for this problem and have developed an FBS-based method for its solution. This is essentially a B&B-based method except for a limited number of branches that are kept at a node.

Different criteria were used to evaluate a node, namely, cost, duration, and cost/duration. A cost-based criterion was found to generate better solutions, as expected. However, we observed that even the duration-based criterion generated good solutions (lowest cost values) for smaller beam width. The cost/duration evaluation criterion was found to always give inferior results. Although we have developed an effective procedure for the solution of this problem, yet it requires further investigation to study the problem's inherent properties, which can further aid in obtaining solutions of better quality in reasonable CPU times.

References

Arroub, M., Kadrou, Y., & Najid, N. (2010). An efficient algorithm for the multi-mode resource constrained project scheduling problem with resource flexibility. *International Journal of Mathematics in Operational Research, 2*(6), 748–761.

Bandelloni, M., Tucci, M., & Rinaldi, R. (1994). Optimal resource leveling using non-serial dynamic programming. *European Journal of Operational Research, 78*(2), 162–177.

Basnet, C., Tang, G., & Yamaguchi, T. (2001). A beam search heuristic for multi-mode single resource constrained project scheduling. In proceedings of 36th Annual Conference of the Operational Research Society of New Zealand, Christchurch, NZ, Nov-Dec, 1–8

Bellman, R., & Dreyfus, S. (1959). Functional approximations and dynamic programming. *Mathematical Tables and Other Aids to Computation, 13*, 247–251.

Berthold, T., Heinz, S., Lübbecke, M. E., Möhring, R. H., & Schulz, J. (2010). A constraint integer programming approach for resource-constrained project scheduling. In proceedings of CPAIOR 2010, LNCS, June, Andrea Lodi, Michela Milano, Paolo Toth (Eds.), Springer, 6140, pp. 51–55.

Blazewicz, J., Lenstra, J. K., & Rinnooy Kan, A. H. G. (1983). Scheduling subject to resource constraints: Classification and complexity. *Discrete Applied Mathematics, 5*(1), 11–24.

Boctor, F. F. (1990). Some efficient multi-heuristic procedures for resource constrained project scheduling. *European Journal of Operational Research, 49*, 3–13.

Boctor, F. F. (1993). Heuristics for scheduling projects with resource restrictions and several resource-duration modes. *International Journal of Production Research, 31*, 2547–2558.

Clark, C. E. (1962). The PERT model for the distribution of an activity time. *Operations Research, 10*(3), 405–406.

Davis, E. W. (1966). Resource allocation in project network models—A survey. *Journal of Industrial Engineering, 17*(4), 177–188.

Dean, B. V., Denzler, D. R., & Watkins, J. J. (1992). Multiproject staff scheduling with variable resource constraints. *IEEE Transactions on Engineering Management, 39*, 59–72.

Demeulemeester, E. L., & Herroelen, W. S. (1996). An efficient optimal solution procedure for the preemptive resource-constrained scheduling problem. *European Journal of Operational Research, 90*, 334–348.

Dodin, B. M., & Elmaghraby, S. E. (1985). Approximating the criticality indices in the activities in PERT networks. *Management Science, 31*, 207–223.

Elmaghraby, S. E. (1992). Resource allocation via dynamic programming in activity networks. *European Journal of Operational Research, 88*, 50–86.

Elmaghraby, S. E., & Herroelen, W. S. (1980). On the measurement of complexity in activity networks. *European Journal of Operational Research, 5*(4), 223–234.

Elmaghraby, S. E., & Herroelen, W. S. (1990). The scheduling of activities to maximize the net present value of projects. *European Journal of Operational Research, 49*, 35–40.

Etgar, R., Shtub, A., & LeBlanc, L. J. (1997). Scheduling projects to maximize net present value—The case of time-dependent, contingent cash flows. *European Journal of Operational Research, 96*, 90–96.

Gonçalves, J. F., Mendes, J. J. M., & Resende, M. G. C. (2004). A genetic algorithm for the resource constrained multi-project scheduling problem. Technical ReportTD-668LM4, AT&T Labs Research

Guldemond, T., Hurink, J., Paulus, J., & Schutten, J. (2008). Time-constrained project scheduling. *Journal of Scheduling, 11*(2), 137–148.

Hartmann, S. (2001). Project scheduling with multiple modes: A genetic algorithm. *Annals of Operational Research, 102*, 111–135.

Heilmann, R. (2000). Resource-constrained project scheduling: A heuristic for the multi-mode case. *OR Spektrum, 23*, 335–357.

Herroelen, W. (2006). Project scheduling-theory and practice. *Production and Operations Management, 14*(4), 413–432.

Kazaz, B., & Sepil, C. (1996). Project scheduling with discounted cash flows and progress payments. *Journal of the Operational Research Society, 47*, 1262–1272.

Kelley, J. E., & Walker, M. R. (1959). Critical path planning and scheduling. In proceedings of Eastern Joint Computer Conference, Boston, December 1-3, 1959, NY 1960, pp. 160–173

Kis, T. (2005). A branch-and-cut algorithm for scheduling of projects with variable-intensity activities. *Mathematical Programming, 103*(3), 515–539.

MacCrimmon, K. R., & Ryavec, C. A. (1964). An analytical study of the PERT assumptions. *Operations Research, 12*(1), 16–37.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., & Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics, 21*, 1087–1092.

Mika, M., Waligora, G., & Weglarz, G. (2005). Simulated annealing and tabu search for multi-mode resource-constrained project scheduling with positive discounted cash flows and different payment models. *European Journal of Operational Research, 164*(3), 639–668.

Nemhauser, G. L., & Wolsey, L. A. (1988). *Integer and combinatorial optimization*. Wiley-Interscience, Hoboken, NJ, USA

Ozdamar, L., & Ulusoy, G. (1995). A survey on the resource-constrained project scheduling problem. *IIE Transactions, 27*, 574–586.

Padman, R., & Dayanand, N. (1997). On modelling payments in projects. *Journal of the Operational Research Society, 48*, 906–918.

Patterson, J. H., Slowinski, R., Talbot, F. B., & Weglarz, J. (1989). An algorithm for a general class of precedence and resource constrained scheduling problems. In R. Slowinski & J. Weglarz (Eds.), *Advances in project scheduling* (pp. 3–28). Amsterdam: Elsevier.

Patterson, J. H., Slowinski, R., Talbot, F. B., & Weglarz, J. (1990). Computational experience with a backtracking algorithm for solving a general class of precedence and resource constrained scheduling problems. *European Journal of Operational Research, 49*, 68–67.

Pritsker, A., Watters, L., & Wolfe, P. (1969). Multi-project scheduling with limited resources: A zero-one programming approach. *Management Science, 16*, 93–108.

Ragsdale, C. (1989). The current state of network simulation in project management theory and practice. *Omega: The International Journal of Management Science, 17*, 21–25.

Ramachandra, G., & Elmaghraby, S. E. (2006). Sequencing precedence-related jobs on two machines to minimize the weighted completion time. *International Journal of Production Economics, 100*(1), 44–58.

Santos, M. A., & Tereso, A. P. (2010). On the multi-mode, multi-skill resource constraint project scheduling problem (MRCPSP-MS). In proceedings of 2nd International Conference on Engineering Optimization (EngOpt 2010), Lisbon, Portugal, September 6–9

Santos, M. A., & Tereso, A. P. (2011a). On the multi-mode, multi-skill resource constrained project scheduling problem—computational results. In proceedings of ICOPEV—International Conference on Project Economic Evaluation, Guimarães, Portugal, April 28–29

Santos, M. A., & Tereso, A. P. (2011b). On the multi-mode, multi-skill resource constrained project scheduling problem—A software application. In A. GasparCunha, R. Takahashi, G. Schaefer, & L. Costa (Eds.), *Soft Computing in Industrial Applications*, *96*, 239–248

Sepil, C., & Ortaç, N. (1997). Performance of the heuristic procedures for constrained projects with progress payments. *Journal of the Operational Research Society, 48*, 1123–1130.

Tereso, A.P., Araújo, M.M., Elmaghraby, S.E. (2004). Adaptive resource allocation in multimodal activity networks. *International Journal of Production Economics, 92*, 1–10.

Tereso, A.P., Mota, J.R., Lameiro, R.J. (2006). Adaptive resource allocation technique to stochastic multimodal projects: A distributed platform implementation in JAVA. *Control Cybern, 35*, 661–686.

Tseng, C. (2008). Two heuristic algorithms for a multi-mode resource-constrained multi-project scheduling problem. *Journal of Science and Engineering Technology, 4*(2), 63–74.

Ulusoy, G., & Cebelli, S. (2000). An equitable approach to the payment scheduling problem in project management. *European Journal of Operational Research, 127*, 262–278.

Vanhoucke, M., Demeulemeester, E., & Herroelen, W. (2000). An exact procedure for the resource-constrained weighted earliness-tardiness project scheduling problem. *Annals of Operations Research, 102*, 179–196.

Willis, R. J. (1985). Critical path analysis and resource constrained project scheduling theory and practice. *European Journal of Operational Research, 21*, 149–155.

Zhang, H., Li, H., & Tarn, C. M. (2006). Heuristic scheduling of resource-constrained, multiple-mode and repetitive projects. *Construction Management and Economics, 24,* 159–169.

Zimmermann, J., & Engelhardt, H. (1998). Lower bounds and exact algorithms for resource levelling problems. Technical Report WIOR-517, University of Karlsruhe

Chapter 12
Hybrid Flow Shop Scheduling with Availability Constraints

Hamid Allaoui and Abdelhakim Artiba

Introduction

The rapid evolution and highly competitive nature of today's global markets give the operation–production function a role of first importance in the global competitiveness of companies. Nowadays manufacturers are facing an economy where competition is based on products of high quality offered at lower prices while respecting due dates. In this context the reduction of costs and the improvement of quality become the principal concerns of those who seek to improve performance.

In the field of industrial production, the current tendencies indicate that the powerful manufacturing systems must be quickly adapted to fluctuations of the market (random requests) and to internal disturbances (breakdowns of the machines). The machines must be simultaneously able to manufacture several types of products in small quantities. In such a context, the optimal planning of production, reduction of the production cycle time, and the control of these machines increasingly become the main objectives for both investors and producers. Under these conditions, the joint determination of the production rate, the maintenance policy, and the scheduling rule which minimize the operations costs of these systems is an important research problem in the production systems area.

Production and maintenance are two important interrelated functions in any industrial process. In the past, production and maintenance have been treated as two separate functions. Nowadays because of their widely recognized interdependence, there is an increasing interest in developing optimization models that take into consideration the integration of the two functions.

One area where integration between maintenance and production functions can be advantageous is scheduling. Both production and maintenance scheduling must recognize the realities of the other's needs. A classic question is: under what

H. Allaoui (✉)
Université de Lille Nord de France, Université d'Artois, Arras, France
e-mail: hamid.allaoui@univ-artois.fr

A. Artiba
Université de Lille Nord de France, Université de Valenciennes, Valenciennes, France
e-mail: abdelhakim.artiba@univ-valenciennes.fr

P. S. Pulat et al. (eds.), *Essays in Production, Project Planning and Scheduling,*
International Series in Operations Research & Management Science 200,
DOI 10.1007/978-1-4614-9056-2_12,

circumstances does production schedule around maintenance versus maintenance scheduling around production? The integration of the two scheduling systems gives both functions the ability to accomplish their scheduling more productively, leading to greater plant utilization.

Most of the literature on scheduling assumes that machines are available at all times. However, due to maintenance activities machines cannot operate continuously. In general, maintenance activities can be classified into two categories: preventive and corrective. For preventive maintenance the machine is checked, repaired, and recalibrated before failure. On the other hand, for corrective maintenance the machine is repaired following breakdown.

In the case of preventive maintenance the question is whether the maintenance decision will be done separately or jointly with the job scheduling decision. If the maintenance decision is made separately in advance, the nonavailability periods will then become a constraint for job scheduling. On the other hand, if the maintenance decision is made jointly with the job scheduling the start time and duration of the machine nonavailability interval is a decision variable. In the case of corrective maintenance the question for a partial failure is whether we should stop the machine now and repair it immediately or repair it later. If we do not repair it, the machine can still operate at a less-efficient speed. On the other hand, for a complete failure which can happen at any time, the machine should be repaired to be back to normal speed.

Scheduling under maintenance constraints has attracted much attention recently. The nonavailability consideration adds complexity to any scheduling problem. In this research we deal with the hybrid flow shop scheduling problem supposing that the decision of preventive maintenance is already made and the corrective maintenance is not considered.

Three directions have always been important in the investigation of scheduling problems. The first is the investigation of computational complexity. The second is the search for exact algorithms that give optimal solution. If the time needed by these algorithms is excessive, heuristics to give approximate solutions will be investigated.

In this chapter, exact methods will be investigated. We first study the two machine flow shop scheduling problem to minimize the makespan under availability constraints in the nonresumable case. The two-machine flow shop without availability constraint is a polynomial problem (Johnson 1954). However if we consider only one nonavailability period on either the first or the second machine the problem becomes Non-deterministic Polynomial-time hard (NP-hard) (Lee et al. 1997). To solve this problem optimally we propose a dynamic programming model whose computational time is independent of the processing times of jobs. We then propose a branch and bound algorithm to schedule a two-stage hybrid flow shop with only one machine at the first stage and m identical machines at the second stage under availability constraints, in nonresumable case, to minimize the makespan which is NP-hard in the strong sense. The time required by this algorithm is still reasonable for small size instances.

Problem Statement

A common manufacturing environment in many industries (such as the glass, steel paper, textile, and semiconductor) is a hybrid flow shop. This scheduling system is a combination of the serial and parallel shop organizations. In this chapter we deal with hybrid flow shop scheduling problems assuming that machines may become unavailable during certain periods. Indeed most of the literature on scheduling problems assumes that the machines are always available during the scheduling period. However in most industrial settings a machine can be unavailable for maintenance activities, such as unforeseen breakdowns (stochastic unavailability) or scheduled preventive maintenance where the periods of unavailability (also called gaps) are known in advance (deterministic unavailability). Under such circumstances, special consideration is needed in order to obtain optimal solutions.

Hybrid Flow Shop

A hybrid flow shop (HFS), also called a flexible flow shop, is a multistage production system consisting of a set of u stages with each stage k ($1 \leq k \leq u$) having m_k parallel machines such that all jobs have to be processed through all the stages in the same order: there are n jobs J_i ($1 \leq i \leq n$), each job consisting of a chain of u operations $O_{i1}, \ldots, O_{iu}$ that have to be executed in this order. In the classical flow shop each stage contains only one machine. At any time each job can be processed by at most one machine and each machine can process at most one job. The assumptions are as follows: (i) all N jobs are independent and available for processing at time $t = 0$ (i.e., $\forall i : r_i = 0$), (ii) machines of the same stage are identical, (iii) the processing time $p_{i,k}$ of each job i on stage k is known, (iv) preemption and splitting of job is not allowed: a job, once started on a machine, continues in processing until it is completed; (v) jobs are allowed to wait between two stages, and the storage is unlimited. The objective function is to minimize the maximum completion time $C_{\max}$.

Maintenance Constraints

Most of the literature on scheduling assumes that machines are available at all times. However, due to various reasons, machines may not be always available in many realistic situations. Therefore, a more realistic scheduling model should take into account the following machine maintenance activities:

- Preventive maintenance: where maintenance is performed on a scheduled basis with scheduled intervals often based on manufacturers' recommendations and past experience with the equipment. This may involve replacement or repair, or both.

- Corrective maintenance: replacement or repair is performed only at the time of failure. This may be the appropriate strategy in some cases, such as when the hazard rate is constant and/or when the failure has no serious cost or safety consequence or it is low on the priority list.

The problem of integrating production and preventive maintenance has been approached in the literature in two different ways. Some authors approached this problem by determining the optimal preventive maintenance schedule in the production system and others by taking maintenance as a constraint to the production system (Ben Daya 1999; 1998a) and (Ben Daya and Makhdoom 1998b). Thus the question is whether the preventive maintenance scheduling decision will be done separately or jointly with the job scheduling decision. In this chapter we assume that the preventive maintenance for each machine is done in the specified window), and do not consider breakdown maintenance. There are three types of machine unavailability (Lee et al. 1997; Lee 1997; Lee 1996) discussed in the literature:

- *Resumable* : A machine is called resumable if a job that cannot be finished before a down period of a machine can be continued without any penalty after the machine becomes available again.
- *Nonresumable* : A machine is called nonresumable if the job that cannot be completed before a period of machine nonavailability must be totally restarted rather than continuing after the machine is brought back on line.
- *Semiresumable* : A machine is called semiresumable if the nonfinished job before a period of machine nonavailability must be partially restarted. There are two types of semiresumability: In *Type−I*, in addition to processing the nonfinished part, the machine needs to process extra work that is proportional to the finished part of that job. In *Type−II*, if a job is not processed to completion before the machine is stopped for maintenance, an additional setup is necessary when the processing is resumed.

Notation

In order to be able to refer the problem under study, we use a three field notation $\alpha \mid \beta \mid \gamma$ presented in (Kubiak et al. 2002; Lee 1997) taking into account the availability constraints:

- $F2(P), a_{kj,l} \mid r \mid Cmax$: Minimizing the makespan in a hybrid flowshop with a *resumable* availability constraint and an arbitrary number of gaps (l) on each machine (j) on each stage (k);
- $F2(P), a_{kj,l} \mid nr \mid Cmax$: Minimizing the makespan in the two-machine flowshop with a *nonresumable* availability constraint and an arbitrary number of gaps (l) on each machine (j) on each stage (k);
- $F2(P), a_{kj,l} \mid sr \mid Cmax$: Minimizing the makespan in the two-machine flowshop with a *semiresumable* availability constraint and an arbitrary number of gaps (l) on each machine (j) on each stage (k);

For example $F2(P), h_{11,l}|\ (m_1 = 1, m_2 = 2), nr\ |Cmax$ represents the problem of minimizing makespan in the two stages hybrid flow shop with one machine at the first stage and two at the second stage under a *nonresumable* availability constraint and an arbitrary number of gaps at the first stage and no gaps at the second stage.

For the classical two machine flow shop problem it is more convenient to use Lee's notation (Lee et al. 1997).

- $F2|r - a(M_j)|Cmax$: Minimizing the makespan in the two-machine flow shop with a *resumable* availability constraint on the machine M_j.
- $F2|nr - a(M_j)|Cmax$: Minimizing the makespan in the two-machine flow shop with a *nonresumable* availability constraint on the machine M_j.

Literature Review

The hybrid flow shop is a nontrivial problem. Most of the works explore three different issues: computational complexity, modeling criteria, and constraints and solution methods. In terms of complexity hybrid flow shop scheduling problems can be roughly grouped into three categories: (1) the two-stage hybrid flow shop, (2) the three-stage hybrid flow shop, and (3) k-stage hybrid flow shop. Most theoretical research on HFS scheduling deals with single criterion problems, among which the minimization of makespan is the most common. Other objectives considered include the minimization of the maximum tardiness, the total flow time and the sum of completion times. Researchers have recently give more attention to maintenance or availability constraints in a single machine, flow shop, and parallel machines problems but not yet for hybrid flow shop problems with more than one stage and more than one machine at each stage. The HFS scheduling problems will be reviewed from the point of view of complexity, exact methods, and heuristics.

Hybrid Flow Shop

Complexity Most of the work in the hybrid flow shop scheduling literature that addresses complexity is done on the single and two-stage hybrid flow shop. While interesting results have been obtained for these cases, there has been less work on the k-stage ($k \geq 3$). For the single stage hybrid flow shop which is the parallel machine problem, Karp (1972) has proved that the problem of minimizing the makespan with only two machines without preemption $P2|\ |C_{\max}$ is NP-hard in the ordinary sense. The only variant of the hybrid flow shop problem solved in polynomial time is the classical two-machine flow shop $F2\,|.|\,C_{\max}$ solved by (Johnson 1954). Garey et al (1976) have shown that the problem $F3\,|.|\,C_{\max}$ is NP-hard in the strong sense. It has been shown by Hoogeveen et al. (1996) that the problem $F2(P)\,|.|\,C_{\max}$ is NP-hard in the strong sense even if there is only one machine at the first stage and two machines at the second stage under both preemption and non-preemption.

Exact Methods Branch and Bound methods have been widely used in hybrid flow shop for finding an optimal solution. The Branch and Bound (B&B) algorithms of Salvador (1973), and Rajendran and Chaudhuri (1992) can be used to solve two-stage HFS with two parallel identical machines at the first stage and only one machine at the second. Brah and Hunsucker (1991) give B&B algorithm to solve k-stage ($k \geq 3$) HFS problem which is modified later by Portman et al. (1998). The B&B approaches have not been used in real world application because of their high computational requirements.

Two-Stage HFS We will review works in the two-stage hybrid flow shop literature according to six categories. The first category involves a single machine at the first stage and parallel identical machines at the second stage ($m_1 = 1, m_{2(I)} = m > 1$). Several studies have been reported for the second category with parallel identical machines at the first stage and one machine at the second stage ($m_{1(I)} = m > 1, m_2 = 1$). The third category consists of one machine at the first stage and nonidentical machines at the second stage ($m_1 = 1, m_2 = m > 1$). The fourth category involves parallel nonidentical machines at the first stage and only one machine at the second stage ($m_1 = m > 1, m_2 = 1$). In the fifth category we find identical machines at both stages ($m_{1(I)} > 1, m_{2(I)} > 1$). In the last category the problem studied is dealing with nonidentical machines at both stages ($m_1 > 1, m_2 > 1$).

In the first category Sriskandarajah and Sethi (1989) deal with the problem of minimizing the makespan. They show that an arbitrary sequence has a worst case performance (*wcpb*) ratio of $3 - \frac{1}{m}$. They develop a heuristic based on Johnson's rule and show that its *wcpb* is equal to $3 - \frac{3}{m} + \frac{1}{m^2}$. Gupta et al. (1991) develop two heuristics to approximately minimize the makespan. In the second category Chen et al. (1998) considered Gupta's heuristics for the two-stage hybrid flow shop with a single machine at the second stage and determines its *wcpb* to be of $3 - \frac{2}{m}$. Gupta (1988) develops an efficient heuristic algorithm for finding an approximate solution. He proposes using Johnson's rule to first sequence the jobs using only one machine at each stage, then assigning jobs to the machines at the first stage in order to minimize the additional idle time at the second stage. His computational experiments are limited to only two machines at the first stage. Chen et al. (1998) classifies the heuristics for this problem into three classes and performs some empirical comparisons among selected heuristics in three classes.

In the third category Narasimhan and Panwalkar (1984) develop a heuristic known as the cumulative minimum deviation (CMD) rule. It is proved to be better than Shortest Processing Time (SPT) and Longest Processing Time (LPT) in decreasing the machine idle time and in-process waiting at the second stage. In the fourth category Elmaghraby and Soewandi (2001) propose a new heuristic based on viewing the second stage as sequencing jobs with "ready times" that are obtained by processing the jobs optimally at the first stage. Gupta (1988) proposes four heuristics. The two first heuristics H_1 and H_2 have a *wcpb* of 2, and the two last heuristics have a *wcpb* of $2 - \frac{1}{m}$.

Among works in the fifth category we can cite the work of Sriskandarajah and Sethi (1989). These authors consider the problem when $m_1 = m_2$.They show that Johnson's sequence has a *wcpb* $3 - \frac{1}{m}$ and their heuristic has an error bound between

$\frac{7}{3} - \frac{2}{3m}$ and $3 - \frac{1}{m}$. Langston (1987) first shows that if a random sequence has a *wcpb* bounded by $3 - \frac{1}{m}$. He also shows that if the jobs are sequenced according to the SPT rule with respect to $p_{i,1}$, then the *wcpb* is bounded by $\frac{5}{2}$, and that the *wcpb* will be improved to 2 if the sequence is arranged in nondecreasing order of $p_{i,2}$. Lee and Vairaktarakis (1994) develop a heuristic H that has an error bound of $2 - \frac{1}{\max(m_1, m_2)}$. This bound extends a recent bound for the case $(m_1 = 1, m_2 = m > 1)$ and significantly improves all other results for some special cases of the two-stage hybrid flow shop.

The general case of the last category involving multiple machines at both stages has been addressed by Narasimhan and Mangiameli (1987) and Paul (1979). Paul examines production scheduling problems in the glass container industry in which there are four stages with several unrelated machines in each stage. The author chooses the first two stages to develop the scheduling and formulates it as a job shop problem. The objectives are to minimize the number of tardy jobs and the average tardiness. To find a solution, a simulation method is adopted to test dispatching rules.

Three-Stage HFS Adler et al. (1993) consider the three-stage problem with parallel nonidentical machines at stages 1, 2, and 3 $(m_1 > 1, m_2 > 1, m_3 > 1)$ in the Bagpak Production Scheduling System (BPSS), a scheduling support system for plants that produce paper bags. The production process consists of three stages, but not all orders have to go through all three stages. They try to achieve a compromise among three objectives of minimizing: (i) the sum of tardiness; (ii) the sum of setup times; and (iii) the work-in-process inventory. Riane et al. (1998) focuses on a three-stage hybrid flow shop to minimize the makespan with one machine at stages 1 and 3, and two machines at the second stage. The two machines at the second stage are dedicated (i.e., it is known beforehand on which of the two machines each job will be processed). They propose two heuristics, one based on dynamic programming and the second one on a branch and bound algorithm. They evaluate the performance of these heuristics by an experimental study. Jin et al. (2002) treat a three-stage HFS for the production of printed circuit boards. They propose a global procedure that utilizes a genetic algorithm and three subproblems to minimize the and evaluate their performance using computational experiments.

***k*-Stage HFS** The k-stage problem with parallel identical machines at each stage $(m_{1(I)} > 1, m_{2(I)} > 1, \ldots, m_{k(I)} > 1)$ is studied by Wittrock (1985, 1988). He presents an algorithm that schedules the loading of parts. The objective is primarily to minimize the makespan and secondarily to minimize queueing. Zhou et al. (1996) consider the k-stage problem with parallel nonidentical machines at each stage $(m_1 > 1, m_2 > l, \ldots, m_k > 1)$. They propose a concept of synthetic knowledge and a heuristic node-path intelligent search method. The objective they use is to minimize the number of changeovers on all machines. The most general heuristic for the k-stage HFS problem with parallel identical machines is provided by Lee and Vairaktarakis (1994) who show that the *wcpb* r of their heuristic is:

$$r \leq k - \frac{1}{\max\{m_1, m_2\}} - \frac{1}{\max\{m_3, m_4\}} - \ldots - \frac{1}{\max\{m_{k-1}, m_k\}}.$$

For some special cases of the k-stage HFS problem, this bound improves significantly all the previous results.

Approximation Schemes Schuurman and Woeginger (2000), Hochbaum and Shmoys (1987), Hall (1995), and Williamson et al. (1997) investigate the approximation behavior of the $F(P) \mid \mid Cmax$ problem. Their known results on the approximation of the hybrid flow shop problem are summarized in Table 12.1.

Table 12.1 Approximation schemes

		Number of machines per stage		
		=1	Constant	Arbitrary
k	=1	Trivial	FPTAS	PTAS
stages	=2	Polynomial	PTAS	PTAS
	Constant ≥ 3	PTAS	PTAS	Open
	Arbitrary	∄PTAS	∄PTAS	∄PTAS

The single-stage flow shop is the ordinary multiprocessor scheduling problem $P \mid \mid Cmax$. When the number of machines is constant, Sahni (1976) proves that the problem possesses a pseudopolynomial solution algorithm that can be used to construct a Fully Polynomial-Time Approximation Scheme (FPTAS). When the number of machines is part of the input, the problem is strongly NP-hard, and hence the Polynomial-Time Approximation Scheme (PTAS) of Hochbaum and Shmoys (1987) is the best possible approximation result unless P=NP. The two-stage flow shop can be solved in polynomial time if $m_1 = m_2 = 1$ Johnson (1954). For the cases where the number k of stages and the number of machines per stage all are constants, Hall (1995) constructs a PTAS. Schuurman and Woeginger (2000) construct a PTAS for the problem of two stages and an arbitrary number of machines per stage. Determining the approximation behavior of the k-stage HFS $F(P) \mid \mid Cmax$ where $k > 3$ is a constant and where the number of machines is part of the input, is still an open Hall (1995) question in the area of hybrid flow shop. Finally, for the case where the number of stages is part of the input, Williamson et al. (1997) prove that the existence of a polynomial time approximation algorithm with worst-case ratio less than $\frac{5}{4}$ would imply $P = NP$.

Scheduling with Availability Constraints

Generally the limited availability of machines is due to the maintenance constraints (preventive and breakdowns). It may also result from other reasons. An example is the case of preschedules which exist because most real world resource planning problems are dynamic. Thus machine unavailability can arise when machines continue to process unfinished jobs scheduled in the previous planning period at the beginning of the new planning period. In this section we review all works related to this area under two categories: the first contains deterministic problems which involve preventive maintenance and the second stochastic problems taking into consideration breakdowns.

Machine scheduling with availability constraints is an important topic in scheduling (see, for example, Blazewicz (1988) and Pinedo (2002)) and has attracted much attention recently (see the surveys by Lee et al. (1998) and Schmidt (2000)). The nonavailability consideration adds complexity to any scheduling problem. In this section we focus only on works concerning flow shop scheduling with availability constraints.

Deterministic Case For the resumable case, Lee et al. 1997 shows that both $F2|r-a(M_1)|C_{\max}$ and $F2|r-a(M_2)|C_{\max}$ are NP-hard, where $r-a(M_1)$ and $r-a(M_2)$ indicate that there is only one nonavailability interval on each of machine 1 and machine 2, respectively. If $s_1 = s_2 = 0$ then Johnson's algorithm is optimal for $F2|r-a|C_{\max}$. He also shows that for $F2|r-a(M_1)|C_{\max}$, the *wcpb* of Johnson's heuristic is equal to 2. He provides two heuristics with error bound of $\frac{C_{H1}}{c^*} \leq \frac{3}{2}$ and $\frac{C_{H2}}{c^*} \leq \frac{4}{3}$. He notes that the problem is irreversible, an important characteristic that is distinct from the classical flow shop problem. Cheng and Wang (2000) provide an algorithm for the $F2|r-a(M_1)|C_{\max}$ problem with an improved error bound of $\frac{C_H}{c^*} \leq \frac{4}{3}$. Kubiak et al. (2002) show that no polynomial-time algorithm with a fixed worst case performance ratio exists unless $P = NP$. They construct a branch and bound algorithm to solve the problem optimally with multiple nonavailability intervals. Blazewicz et al. (2001) use local search based heuristics for the $F2|r-a(M_1)|C_{\max}$ problem. Braun et al. (2002) derive sufficient conditions for the optimality of Johnson's permutation in the case of one or more nonavailability intervals. They show that usually Johnson's permutation remains optimal in the case of nonavailability intervals.

For the nonresumable and semiresumable cases, Lee (1999) studies the $F2|sr1-a|C_{\max}$ where an availability constraint is imposed on only one machine or on both machines. The problem is clearly NP-hard because a special case, $1|nr-a|C_{\max}$, is already NP-hard. Furthermore, the problem with multiple nonavailability intervals is strongly NP-hard as its single machine special case is already strongly NP-hard (2001). Lee (1999) provides a pseudo-polynomial dynamic programming algorithm to solve the $F2|sr1-a(M_1)|C_{\max}$ problem optimally in which the semiresumable availability constraint is imposed on machine 1. He shows that *wcpb* of Johnson's algorithm is equal to 2. Thus, both the resumable and nonresumable cases have the same error bound which is independent of the value of α. On the other hand, if we apply Johnson's algorithm to the $F2|sr1-a(M_2)|C_{\max}$ then $\frac{C_{JA}}{C^*} \leq \max\{\frac{3}{2}, 1+\alpha\}$ and the bound is tight. He develops an improved heuristic with $wcpb \leq \frac{3}{2}$. It is also shown by Lee and Vairaktarakis (1994) that for the problem where an availability constraint is imposed on both machines, $F2|sr1-a(M_1,M_2)|C_{\max}$, is NP-hard even if $s_1 = s_2 = s$, and $t_1 = t_2 = t$. For such a special case, $s_1 = s_2 = s$, and $t_1 = t_2 = t$ we have $\frac{C_{JA}}{C^*} \leq 1+\alpha$.

Stochastic Case Allahverdi and Mittenhal (1998) address the problem of minimizing makespan in a two-machine flowshop when the machines are subject to random breakdowns. They first show that it is sufficient to consider the same sequence of the jobs on each machine. After providing an elimination criterion for minimizing

makespan with probability 1, they show that under appropriate conditions Johnson's algorithm stochastically minimizes makespan.

Allahverdi and Mittenhal (1998) consider a two-machine fowshop scheduling problem where machines suffer random breakdowns and processing times are constant with respect to both makespan and maximum lateness objectives. They provide an elimination criterion for a two machine flow shop when both machines are subject to random breakdowns. They show that the LPT and SPT orders are optimal with respect to both criteria in a two machine flow shop when the first or the second machine, respectively, suffers stochastic breakdowns.

From this brief literature review, we first conclude that the problem of two-machine flow shop with availability constraints is widely studied. The most important work is presented by (Lee et al. (1997)). In this chapter, we try to give an alternative model to solve the problem for optimality. We also conclude that few of works are dedicated to hybrid flow shop scheduling with availability constraints. This observation motivates us to propose a modified branch and bound algorithm to optimize the makespan in his shop environment.

Scheduling the Two-Machine Flow Shop With Availability Constraints

This section is concerned with the problem of scheduling n immediately available jobs in a flow shop composed of two machines to minimize the makespan, when it is known that there shall be only one interruption in machine availability either on the first machine or on the second machine. For convenience we call the period of unavailability a "gap" (Allaoui et al. 2006). The two machine flow shop (without a gap) with the objective of minimizing the makespan is perhaps the first "multi machine" scenario ever treated by researchers in the field. Its optimal solution is due to Johnson (1954). Since that date it has witnessed several extensions and variations; see Lee (1997) and Lee et al. (1997) for a comprehensive review of earlier contributions. Lee proves that the problem with one gap on only one machine is NP hard in the ordinary sense.

The processing times of job i are given by $p_{i,1}$ and $p_{i,2}$ on the first and second machine; respectively. The start time of the gap on machine j is denoted by s_j, its duration by g, and its termination by $t_j = s_j + g$; ($j = 1, 2$). We assume that machine j, $j = 1, 2$, is unavailable during the period from s_j to t_j $\left(0 \leq s_j \leq t_j\right)$ while the other is always available. First, we characterize the problem instances in which Johnson's rule gives the optimum solution. We then propose a dynamic programming to solve the first problem whose computational time is independent of processing times but exponential in the number of jobs, if the gap is on the first machine. This reduces the computational burden of the search for optimality drastically.

For simplicity, we refer to *Johnson's rule* (*JR*) for optimizing the makespan in the absence of the gap by *JR*, and to the order of jobs (the sequence) resulting from

applying the rule by Johnson order (JO). The presence of the gap may render the JO nonoptimal. Still, we shall adopt the JR as our heuristic.

Johnson's rule (JR) : Divide the n-job set into two disjoint subsets, S_1 and S_2, where $S_1 = \left\{J_i : p_{i,1} \leq p_{i,2}\right\}$ and $S_2 = \left\{J_i : p_{i,1} > p_{i,2}\right\}$. Order the jobs in S_1 in the nondecreasing order of $p_{i,1}$ and the jobs in S_2 in the nonincreasing order of $p_{i,2}$. Sequence jobs in S_1first, followed by S_2.

We adopt the following notations.

A :	the set of jobs processed after the gap. A^{JR} denotes the jobs in $A$when following JR.
B :	the set of jobs processed before the gap. B^{JR} denotes the jobs in $B$when following JR.
$\emptyset$:	indicates the absence of the gap.
G :	indicates the *presence*of a gap.
$C^{JR}_{\max}(\emptyset)$:	the makespan following JR when there is *no gap,* it is optimal $C^{JR}_{\max}(\emptyset) = C^{*}_{\max}(\emptyset)$.
$C^{JR}_{\max}(G)$:	the makespan following JR when *there is a gap,* it may not be optimal.
$C^{*}_{\max}(G)$:	the *optimal*makespan when *there is a gap.*
$C_{i,j}(S_k)$:	the completion time of job i on machine j, $i \in S_k$.
$N_k(B)$:	the subset of k jobs that are to be scheduled before the gap; with $N_k(A) = N - N_k(B)$ denoting the subset of $n-k$ jobs that are to be scheduled after the gap.
$p_{i,j}$:	the processing time of job i on machine j, $j = 1, 2$.
s_j :	the start time of the gap on machine j, $j = 1, 2$.
r :	the resumable case.
nr :	the nonresumable case.
t_j :	the end of the gap, $t_j = s_j + g$, $j = 1, 2$.
S_1	$\left\{J_i : p_{i,1} \leq p_{i,2}\right\}$. In JO the jobs in this set are ordered in nondecreasing order of $p_{i,1}$
S_2	$\left\{J_i : p_{i,1} \succ p_{i,2}\right\}$. In JO the jobs in this set are ordered in nonincreasing order of $p_{i,2}$

Lee (1997) proves an important lemma which characterizes the optimal solution either when the gap is on the first machine or on the second machine.

Lemma (Lee's (1997) Lemma 2, page 132.) There exists an optimal sequence such that the jobs in the set B and the jobs in the set A are sequenced in JO.

Proof This assertion is proved by switching the order of two jobs in either B or A. Such a switch can only increase the makespan by the optimality of the JO of either sets.

Let us consider the gap on the first machine. In the two-machine flowshop with availability constraints problem there are two types of idle time. The first is due to the sequence of jobs, denoted by I_{seq}, and the second is due to the gap, denoted by I_{gap}. It is known that JO minimizes the sum $\sum I_{seq}$ (see Johnson (1954)). Hence its optimality for the two-machine flowshop with a gap depends on wether it minimizes I_{gap}. Let J^A_{JO} denote the first job completed after the gap on the first machine when the jobs are sequenced in JO.

Dynamic Programming Model

It is easy to determine the maximal number of jobs that "fit" in the interval $[0, s_1]$: Simply order the jobs in nondecreasing order of $p_{i,1}$ and select the first n_1 jobs for which $\sum_{i=1}^{n_1} p_{i,1} \leq s_1$ but $\sum_{i=1}^{n_1+1} p_{i,1} > s_1$. Then we know that no more than n_1 jobs can be processed on the first machine before the gap. The issue now becomes: which jobs? This may be answered in one of two ways. (It is this determination that gives this problem its NP-hardness character.) One may solve a knapsack problem and enumerate all its alternate optima, or one enumerates all subsets of cardinality $\leq n_1$, of which there are $\binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{n_1} \leq 2^n - \sum_{r=0}^{n/2-1} \binom{n}{r}$. The inequality is because if $n_1 > n/2$ then we need to consider the jobs allocated to the set A.

Consider a subset of k jobs, $k \leq n_1$ that are to be scheduled before the gap. Denote the subset by $N_k(B)$, and denote their processing time on the first machine by $P_1(N_k(B))$. That is,

$$P_1(N_k(B)) = \sum_{i \in N_k(B)} p_{i,1}.$$

Note that if $P_1(N_k(B)) > s_1$ then this subset of jobs is deleted since it is not eligible as candidate. This implies that we may end up evaluating fewer subsets than the upper bound given above. For the remainder of this discussion we assume that attention is limited to subsets with $P_1(N_k(B)) \leq s_1$. Clearly, if the jobs in $N_k(B)$ are the only jobs executed before the gap, then the remaining jobs, denoted by $N_k(A) = N - N_k(B)$ must start after the gap since

$$N = N_k(B) + N_k(A).$$

By Lemma there exists a minimal makespan schedule such that the jobs in $N_k(B)$ are ordered in *JO* and the those in $N_k(A)$ are also ordered in *JO*. Denote the makespan obtained by such ordering of the jobs by $C_{\max}(N_k(B))$. Clearly $C_{\max}(N_k(B))$ is an upper bound (*u.b.*) on the value of the optimum, since it is a feasible schedule. Let $C_{i,1}(N_k(B))$ denote the completion time of job i on the first machine in subset $N_k(B)$; and define $C_{i,2}(N_k(B))$ similarly for the second machine. Denote the completion time of the jobs in $N_k(B)$ on the second machine by $C_2(N_k(B))$[1]. Identify the first job after the gap generically by j_a. Its start time on the first machine is t_1. Denote its start time on the second machine by $s_{j_a,2} \geq t_1 + p_{j_a,1}$. The notation is explained in Fig. 12.1. In the figure there is only one line representing the first machine, but there are two lines representing the second machine, depending on the completion time of the jobs in $N_k(B)$ on the second machine relative to the completion time of job j_a on the first machine. The first line of the second machine depicts the case in which $s_{j_a,2} = t_1 + p_{j_a,1}$. The second line depicts the case in which $s_{j_a,2}$ is strictly greater than $t_1 + p_{j_a,1}$.

[1] Thus $C_2(N_k(B)) = C_{2,l}(N_k(B))$ where l is the last job in the set B.

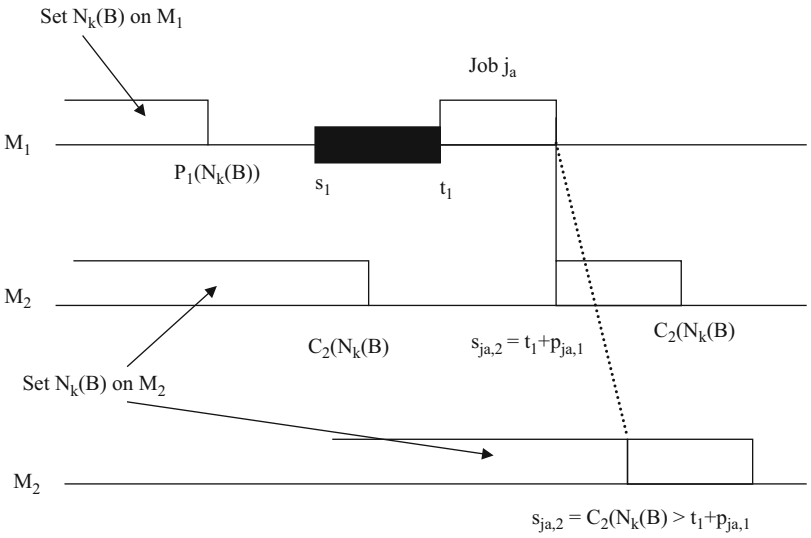


Fig. 12.1 Two cases of completion time of set $N_k\,(B)$ on the second machine

Suppose that the set $N_k\,(B)$ is augmented by job h which is currently in the set $N_k\,(A)$ so that we now have the set

$$N_{k+1}\,(B) = N_k\,(B) \cup \{h\}$$

that is supposed to be processed before s_1. Job h is accepted as an augmentation to $N_k\,(B)$ if $P_1\,(N_{k+1}\,(B)) \leq s_1$; otherwise it is rejected. The issue is to relate the minimal makespan $C_{\max}\,(N_{k+1}\,(B))$ to the old makespan $C_{\max}\,(N_k\,(B))$; which would enable us to write the Dynamic Programming (DP) extremal equation. (Recall that to achieve the minimal makespan under $N_k\,(B)$ for any $k \geq 2$ the jobs in $N_k\,(B)$ must be ordered in JO, and so are the jobs in $N_k\,(A)$.)

Denote the job immediately preceding job h in $N_k\,(A)$ generically by $h-1$, and denote the job immediately succeeding it in $N_k\,(A)$ generically by $h+1$. It is easy to deduce that if $N_k\,(B)$ is augmented by job h which is currently in the set $N_k\,(A)$ then the gain in the makespan, if any, is given by

$$v = s_{h+1,2}\,(N_k\,(A)) - \max\left\{C_{h-1,1}\,(N_k\,(A)) + p_{h+1,1}, C_{h-1,2}\,(N_k\,(A))\right\}.$$

Let $f\,(N_k\,(B))$ denote the minimal makespan under set $N_k\,(B)$. Let

$$N_{k-1}\,(B) = N_k\,(B) - \{h\}\,.$$

Then the DP extremal equation is given by

$$f\,(N_k\,(B)) = \min_{h \in N_k(B)}\{f\,(N_{k-1}\,(B)) - v\}\,.$$

Iteration is initiated at $k = 1$ (*i.e.,* subsets containing exactly a single job, j) for which the makespan is determined following *JO* for the jobs in $N - \{j\}$ which constitutes the set A. Iteration is terminated when all subsets of size n_1 have been considered. The unconditional optimum is given by

$$\min_{k=1,\cdots,n_1} \{f\,(N_k\,(B))\}\,. \tag{12.1}$$

The worst case complexity of the procedure is $nlogn + 2^{n_1}$. The advantage of this model over others is that it's complexity is independent of the processing times $\left\{p_{i,j}\right\},\ j = 1, 2.$

Example *Consider the following set of jobs*

Job i	**1**	**2**	**3**	**4**	**5**	**6**
$p_{i,1}$:	2	6	10	4	10	4
$p_{i,2}$:	1	7	3	3	1	5

$s_1 = 9$ *and* $g = 3$.

JR would yield the sequence: 6,2,3,4,1,5 with makespan equal to 45. It is easy to deduce $n_1 = 2$. *The following table gives the dynamic programming iterations. The set of jobs to be inserted before the gap is* $\{1, 2, 4, 6\}$. *Observe that we enumerated only* $\binom{4}{1} + \binom{4}{2} = 10\ \left(\ll 2^6 = 64\right)$ *subsets, and evaluated the makespan for only 8 subsets because two were infeasible. This procedure is a simple case of the procedure given by* Held and Karp (1962) *due to the limited number of subsets that need to be examined.*

Stage	**Set** B **in** *JO*	**Set** A **in** *JO*	$C_{\max}$
1	{1}	{6, 2, 3, 4, 5}	47
	{2}	{6, 3, 4, 1, 5}	43
	{4}	{6, 2, 3, 1, 5}	45
	{6}	{2, 3, 4, 1, 5}	45
2	{2, 1}	{6, 3, 4, 5}	41
	{4, 1}	{6, 2, 3, 5}	43
	{6, 1}	{2, 3, 4, 5}	43
	{6, 4}	{2, 3, 1, 5}	41

The makespan given by dynamic programming procedure is 41 and the optimal sequences are 2,1,6,3,4,5 or 6,4,2,1,3,5.

Combinatorial Approach

This approach may be viewed as a "shortcut" for the DP formalism discussed above. Observe that once the set $N_k\,(B)$ has been defined its *optimal* sequence is known, as is the optimal sequence of the complementary set $N_k\,(A)$. There is no need for optimization! The makespan is easily determined, and it is retained as the "best in hand" if it is smaller than the last best in hand, else it is discarded.

A pertinent observation in this regard is that we are interested in "packing" as many jobs as possible before the time s_1, whence enumeration of subsets should start from the largest to the smallest. The superposition of simple elimination rules due to dominance (as in the branch-and-bound approach) should speed up the calculations.

Example *Consider the previous example. We knew that* $n_1 = 2$ *and that the set of jobs which can be in Bis* $\{1, 2, 4, 6\}$. *Therefore we start with subsets of cardinality 2.*

$\{2, 1\}$	feasible	$C_{\max} = 41$
$\{4, 1\}$	feasible	$C_{\max} = 43$
$\{6, 1\}$	feasible	$C_{\max} = 43$
$\{4, 2\}$	infeasible	$\times$
$\{6, 2\}$	infeasible	$\times$
$\{6, 4\}$	feasible	$C_{\max} = 41$

Hence the optimal solution given by this approach is the sequence 2,1,6,3,4,5 or 6,4,2,1,3,5 with makespan equal to 41, as secured above. This procedure has the same worst case complexity as the proposed DP.

The two machine flow shop scheduling is a special case of the two stage hybrid flow shop. In the next section we focus on the two stage hybrid flow shop with more than one machine in the second stage.

Scheduling the Two Stage Hybrid Flow Shop With Availability Constraints

In this section we investigate the two-stage hybrid flow shop scheduling problem with only one machine at the first stage and m identical machines at the second stage to minimize the makespan Allaoui et al. (2006). At any time, every job can be processed by at most one machine and every machine can process at most one job. Jobs can wait between the two stages in unlimited storage. We assume that each machine is subject to at most one unavailability period. The start and end time of each period are known in advance and only the *nonresumable* case is studied. This problem will be denoted as $F2(P)|(m_1 = 1, m_2 = m), nr - a|C_{\max}$.

Notations

We adopt the following notations.

J_i :	Job i, $i = 1, \ldots, n$.
M_j :	Machine j at the second stage, $j = 1, \ldots, m$.
$p_{i,1}$:	Processing time of J_i at the first stage.
$p_{i,2}$:	Processing time of J_i at the second stage.

$MS1$: The sum of processing time at the first stage ($\sum_{i=1}^{n} p_{i,1}$).
$MS2$: The sum of processing time at the second stage ($\sum_{i=1}^{n} p_{i,2}$).
$C_{i,1}$: The completion time of J_i at the first stage.
$C_{i,2}$: The completion time of J_i at the second stage.
$C_{\max}$: The makespan.
s_1 : The start time of the gap on machine in the first stage.
t_1 : The finish time of the gap on machine in the first stage.
$s_{j,2}$: The start time of the gap on M_j in the second stage.
$t_{j,2}$: The finish time of the gap on M_j in the second stage.

Complexity Analysis

The only variant of the flow shop problem solved in polynomial time is the two-machine flow shop $F2\,|.|\,C_{\max}$ (see Johnson(1954)). The problem $F3\,|.|\,C_{\max}$ is NP-hard in the strong sense (see Garey and Johnson (1979)). The problem $F2(P)\,|.|\,C_{\max}$ is NP-hard in the strong sense even if there is only one machine at the first stage and two machines at the second stage (see Hoogveen (1996)).

Thus this proposition stands obviously.

Proposition 1 *The problem* $F2(P)|(m_1 = 1, m_2 = m), nr - a|C_{\max}$ *is NP-hard in the strong sense*.

Lee (1991) studies the problem $Pm|\ |C_{\max}$ in which each machine has at most one gap and shows that if there is no machine which is always available then no polynomial approximation scheme exists unless $P = NP$. We can generalize this result for the two-stage hybrid flow shop problem under the same assumption (there is no machine which is always available) for the second stage. Therefore we assume in the following that there is at least one machine at the second stage which is always available.

To solve the $k-stage$ hybrid flowshop scheduling problems, the only exact method available is the branch and bound of Brah and Hunsucker (1991). In this chapter we use the concept of B&B to solve the two-stage hybrid flow shop with availability constraints when there is only one machine at the first stage and m machines at the second stage. The branch and bound algorithm consists of three steps: bounding, branching, and node elimination.

Determination of Lower Bounds

We will use the approach of Brah and Hunsucker (1991) to calculate a lower bound for the branch and bound. Our contribution is to update their bounds by integrating the availability constraints into the calculations in order to have tight bounds.

We first introduce the following notation:

N :	The set of all jobs
S :	A subset of jobs such that $S \subseteq N$
S' :	A subset of jobs on S and an other job J_q such that $J_q \notin S$ and $S' = S \cup \{J_q\}$
$\sigma_1(S)$:	The partial schedule of the subset jobs S at the first stage
$\sigma_2(S)$:	The partial schedule of the subset jobs S at the second stage
$C_0(\sigma_1(S))$:	The completion time of the partial schedule $\sigma_1(S)$ at the first stage
$C_j(\sigma_2(S))$:	The completion time of the partial schedule $\sigma_2(S)$ on machine M_j $(j = 1, \ldots, m)$ in the second stage

Thus the partial schedule $\sigma_k(S')$ $(k = 1, 2)$ is obtained by adding the job J_q to the partial schedule $\sigma_k(S)$. The makespan can be expressed as:

$$C_{\max} = \max_{1 \le j \le m} C_j(\sigma_2(N)) \tag{12.2}$$

The completion times at the first stage and on every machine at the second stage are given by:

$$C_0(\sigma_1(S')) = \left\{ \begin{array}{ll} t_1 + p_{q,1} & \text{if } (C_0(\sigma_1(S)) \le s_1 < C_0(\sigma_1(S)) + p_{q,1}) \\ C_0(\sigma_1(S)) + p_{q,1} & \text{otherwise} \end{array} \right\} \tag{12.3}$$

We can calculate $C_j(\sigma_2(S'))$ in two cases:

- If J_q is assigned to M_j then

$$C_j(\sigma_2(S')) = \left\{ \begin{array}{l} t_{j2} + p_{q,2} \quad \text{if } (Max(C_j(\sigma_2(S)), C_0(\sigma_1(S'))) \le s_{j,2} \\ < Max\{C_j(\sigma_2(S)), C_0(\sigma_1(S'))\} + p_{q,2} \\ Max(C_j(\sigma_2(S)), C_0(\sigma_1(S'))) + p_{q,2} \quad \text{otherwise} \end{array} \right\} \tag{12.4}$$

- If J_q is not assigned to M_j then

$$C_j(\sigma_2(S')) = C_j(\sigma_2(S)) \tag{12.5}$$

Machine Based Bounds We use the unprocessed work load at any of the two stages to give a lower bound on the value of the optimal makespan at that stage. For any given partial schedule $\sigma_1(S')$ we denote the maximum completion time for this partial schedule at the first stage by:

$$MCT(\sigma_1(S')) = C_0(\sigma_1(S')) \tag{12.6}$$

The completion time and processing requirement for this partial schedule at the first stage are given by:

$$ACT(\sigma_1(S')) = C_0(\sigma_1(S')) + \sum_{i \in N - S'} p_{i,1} \tag{12.7}$$

For the second stage, the maximum completion time and the average completion time for any given partial schedule $\sigma_2(S')$ can be expressed as:

$$MCT(\sigma_2(S')) = \underset{1\leq j\leq m}{Max}\,(C_j(\sigma_2(S'))) \tag{12.8}$$

$$ACT(\sigma_2(S')) = \frac{\sum_{j=1}^{m} C_j(\sigma_2(S'))}{m} + \frac{\sum_{i\in N-S'} p_{i,2}}{m} \tag{12.9}$$

For the problem studied in this chapter, we have only one machine at the first stage, implying that the following relation always holds due to Eqs. 12.6 and 12.7:

$$MCT(\sigma_1(S')) \leq ACT(\sigma_1(S')) \tag{12.10}$$

Then the machine-based lower bound for the branching node for the first stage can be given as:

$$LBM[\sigma_1(S')] = \left\{ \begin{array}{l} ACT(\sigma_1(S')) + \underset{i\in N-S'}{Min}\left\{p_{i,2}\right\} \text{ if} \\ (C_0(\sigma_1(S')) > t_1) \vee ((C_0(\sigma_1(S')) + \sum_{i\in N-S'} p_{i,1}) \leq s_1) \\ ACT(\sigma_1(S')) + (t_1 - s_1) + \varepsilon_{\min} + \underset{i\in N-S'}{Min}\left\{p_{i,2}\right\} \quad \text{otherwise} \end{array} \right\} \tag{12.11}$$

The amount $\varepsilon_{\min}$ is obtained at each node by resolving the *knapsack* problem:

$$Max(\, Y = \sum_{i\in N-S'} x_i \times p_{i,1})$$

$$\text{Such that : } C_0(\sigma_1(S')) + Y \leq s_1$$

$$x_i = \left\{ \begin{array}{ll} 1 \text{ if } J_i \text{is assigned before the gap} \\ 0 \quad \text{otherwise} \end{array} \right\}$$

Then

$$\varepsilon_{\min} = s_1 - Y^*$$

The complexity of the calculations at each node can be at most $O(\left|N - S'\right| \times (s_1 - C_0(\sigma_1(S'))))$. Thus there is a tradeoff between the time of calculations and the quality of bounds.

For the second stage the lower bound for the branching node is:

$$LBM[\sigma_2(S')] = Max(MCT(\sigma_2(S')), ACT(\sigma_2(S'))) \tag{12.12}$$

The effect of the availability constraint is introduced in the calculations of $C_j(\sigma_2(S'))$.

Job-Based Bounds The calculations for a job based bound focuses on the remaining processing required by each unscheduled job at each stage k ($k = 1, 2$). The job-based bound for a hybrid flow shop can not be strong since there are alternate routes for the other jobs in the set which is not the case in the classical flow shop. It is easy to see that the job based lower bound at the first stage is given by:

$$LBJ[\sigma_1(S')] = C_0(\sigma_1(S')) + \underset{i \in N-S'}{Max} \left\{ p_{i,1} + p_{i,2} \right\} \tag{12.13}$$

And the job lower bound at the second stage is given by:

$$LBJ[\sigma_2(S')] = \underset{j}{Min}(C_j(\sigma_2(S'))) + \underset{i \in N-S'}{Max} \left\{ p_{i,2} \right\} \tag{12.14}$$

Thus the composite lower bound for the two-stage hybrid flow shop with availability constraints for the branching node at stage $k(k = 1, 2)$ is as follows:

$$LBC[\sigma_k(S')] = Max \left\{ LBM[\sigma_k(S')], LBJ[\sigma_k(S')] \right\} \tag{12.15}$$

Branching Strategy

We propose a new branching strategy for the two-stage hybrid flow shop problem which is different from that of Brah and Hunssucker. Thus at the first stage the decision is the sequence of jobs, and at the second stage it is the assignment of jobs to a specific machine M_j among the m parallel machines. The branching procedure will be represented by a search tree. Two types of nodes will be used:

- A square node indicates that job J_i is sequenced in position r at the first stage.
- A circular node that job J_i is assigned to machine M_j at the second stage.

In order to develop the algorithm based on this branching strategy we give the following rules:

1. The first level after the dummy node concerns a sequencing decision.
2. On the branching tree we use two types of levels. The first is the level of sequencing decision and the second is the level of assignment decision. Thus we use the notation L_{ed} to distinguish each level. If the nodes concern the first stage (i.e., sequencing decision) $d = 1$. However if the node concerns the second stage (i.e., assignment decision) $d = 2$. The number of the level in the branching tree if $d = 1$ or $d = 2$ is indicated by e such that $1 \leq e \leq n$.
3. Every sequencing decision must be followed by an assignment decision, such that we can not find a path in the branching tree with two successive nodes of the same kind of decision (i.e., sequencing or assignment).
4. The number of nodes generated at level L_{e1} is at most equal to $n - e + 1$, and at level L_{e2} is at most equal to m.
5. No path can be considered as a feasible solution if the number of square nodes or circle nodes is less than the number of jobs n.

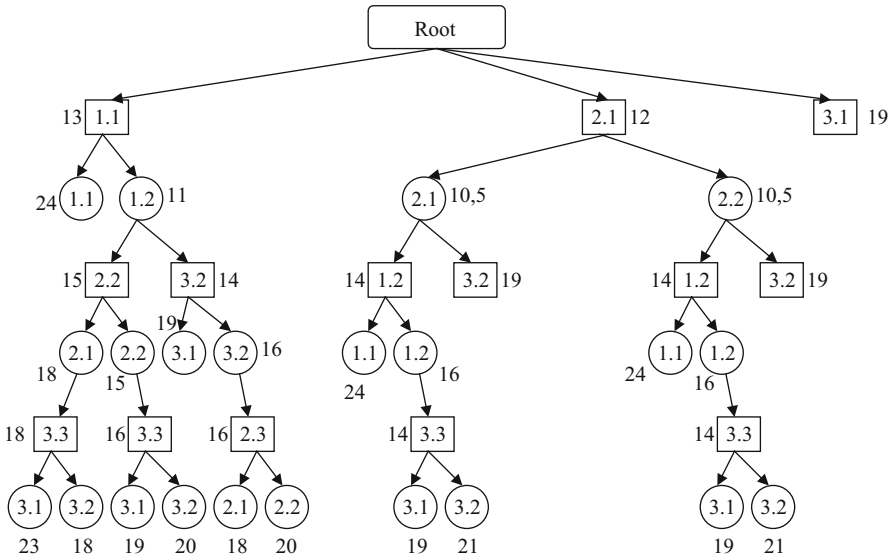


Fig. 12.2 The branch and bound search tree

Adding to the decision taken, each node is represented by these three elements:

- The partial schedule $\sigma_1(S')$ for the square nodes and $\sigma_2(S')$ for the circle nodes such that S' is the subset of jobs.
- The job q added after each decision. It is the left number on each node.
- The composite lower bound $LBC[\sigma_k(S')]$ $(k = 1, 2)$.

Example *To illustrate this B&B procedure we consider two-stage hybrid flow shop with one machine in the first stage and two machines in the second stage. Three jobs have to be scheduled to minimize the makespan.*

Job i	**1**	**2**	**3**
$p_{i,1}$:	1	2	3
$p_{i,2}$:	10	4	5

$(s_1 = 2$ *and* $t_1 = 5)$; $(s_{1,2} = 8$ *and* $t_{1,2} = 14)$.
In Fig. 12.2 *we present the search tree of the branch and bound algorithm described above applied to this instance, with the value of the composite bound for each node. According to the next section the makespan given by a heuristic is equal to 18. Hence we can take the* $ub = 18$.

The branch and bound procedure gives two optimal solutions with a makespan equal to 18. In one of these two solutions the jobs are sequenced in the first stage in the order 1,3,2. Job 2 is assigned to the first machine at the second stage and jobs 1 and 3 are assigned to the second machine.

The computational burden of this procedure becomes excessive for instances with more then ten jobs.

Conclusion

Motivated by the idea of integrating production and maintenance, and by a scheduling environment commonly found in many industrial applications which is the hybrid flow shop, this chapter studies the hybrid flow shop scheduling with availability constraints. The problem of machine scheduling with availability constraints has attracted much attention in the scheduling field recently. In this chapter, we first present a detailed survey of hybrid flow shop scheduling and scheduling with availability constraints. Then we solved to optimality two special cases of HFS with availability constraints: the two-machine flow shop problem and the two-stage hybrid flow shop problem with only one machine at the first stage and m machines ($m \geq 1$) at the second stage. We have proposed a dynamic programming to solve the first problem whose computational time is independent of processing times but exponential in the number of jobs. The second problem was solved by a branch and bound algorithm. Only small size instances could be solved in reasonable computational time. In this chapter, exact methods were investigated. Heuristics and metaheuristics should be addressed in future works to solve large instances of this problem.

References

Adler, L., Fraiman, N., Kobacker, E., Pinedo, M., Plotnicoff, J. C. & Wu, T. P. (1993). BPSS: A scheduling support system for the packaging industry. *Journal Operations Research, 41*, 641–648.

Allahverdi, A. (1996). Two-machine proportionate flowshop scheduling with breakdowns to minimize maximum lateness. *Computer Operations Research,* (23-10) 909–916.

Allahverdi, A., & Mittenthal, J. (1998). Dual criteria scheduling on a two-machine flowshop subject to random breakdowns. *International Transaction Operational Research,* (5–4) 317–324.

Allaoui, H., & Artiba, A. (2004). Integrating simulation and optimization to schedule a hybrid flow shop with maintenance constraints. *Computers Industrial Engineering, 47,* 431–450.

Allaoui, H., Artiba, A. (2006). Two stage hybrid flow shop scheduling with availability constraints. *Computers Operations Research, 33*(5), 1399–1419.

Allaoui, H., Artiba, A., Elmaghraby, S. E., & Riane, F. (2006). Scheduling two machine flow shop with an availability constraint on the first machine. *Int J Product Econ, 99*(1–2), 16–27.

Artiba, A., Emelyanov, V., Iaasinovski, S. (1998). *Introduction to Intelligent Simulation: The RAO Language*. Kluwer Academic Publishers: Dordercht.

Avramidis, A. N., & Wilson, J. R. (1996). Integrated variance reduction strategies for simulation. *Operations Research, 44*(2), 327–346.

Ben Daya, M. (1999). Integrated Production, Maintenance, and Quality Model for Imperfect Processes. *IIE Transactions, 31,* 491–501.

Ben Daya, M., & Hariga, M. (1998). A Maintenance Inspection Model: Optimal and Heuristic Solutions. *Inter J Quality Reliability Management, 5,* 481–488.

Ben Daya, M., Makhdoom, M. (1998). Integrated Production and Quality Model Under Various Preventive Maintenance Policies. *Journal of the Operational Research Society, 49,* 840–853.

Blazewicz, J., Finke, G., Haupt, R., & Schmidt, G. (1988). New trends in scheduling theory. *European Journal of Operational Research, 37,* 303–317.

Blazewicz, J., Breit, J., Formanowicz, P., Kubiak, W., & Schmidt, G. (2001) Heuristic algorithms for two-machine flowshop with limited machine availability. *Omega, 29,* 599–608.

Brah, S. A., Hunsucker, J. L. (1991). Branch and bound algorithm for the flow shop with multiple processors. *European Journal of Operational Research, 51*(1), 88–99.

Braun, O., Lai, T. C., Schmidt, G., & Sotskov, Y. N. (2002). Stability of Johnson's schedule with respect to limited machine availability. *International Journal of Production Research, 40,* 4381–4400.

Chen, B., Potts, C. N., & Woeginger, G. J. (1998). A review of machine scheduling: Complexity, algorithms and approximability. Handbook of Combinatorial Optimization (Volume 3). In D.-Z. Du, & P. Pardalos (Eds), Kluwer Academic Publishers. 21–169.

Cheng, T. C. E. & Wang, G. (2000). An improved heuristic for two-machine flowshop scheduling with an availability constraint. *Operations Research Letters, 26,* 223–229.

Elmaghrabi, S. E., & Soewandi, H. (2001). Sequencing jobs on two-stage hybrid flowshop with identical machines to minimize makespan. *IIE Trans, 33,* 985–993.

Garey, M. R. Johnson, D. S., & Sethi, R. (1976). The complexity of flow shop and job shop scheduling. *Math Oper Res, 1,* 117–129.

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

Grangeon, N. Métaheuristiques et modèles d'évaluation pour le problème du flow shop hybride hi érarchisé: Contexte déterministe et contexte stochastique, Thesis at Université Blaise Pascal.

Gupta, J. N. D., Hariri, A. M. A., & Potts, C. N. (1994). Scheduling a two-stage hybrid flow shop with parallel machines at the first stage. *Annals of Operations Research, 69,* 171–191.

Gupta, J. N. D. (1988). Two-stage, hybrid flowshop scheduling problem. *The Journal of the Operational Research Society, 38,* 359–364.

Gupta, J. N. D., & Tunc, E. A. (1991). Schedules for a two stage hybrid flowshop with parallel machines at the second stage. *International Journal of Production Research, 29,* 1489–1502.

Hall, L. A. (1995). Approximability of flow shop scheduling. *Proc. 36th IEEE Symp. on Foundations of Computer Science,* 82–91.

Held, M., & Karp, R. M. (1962). A dynamic programming Approach to Sequencing Problems. *Journal of the SIAM, 10*, 196–210.

Hochbaum, D. S., Shmoys, D. B. (1987). Using dual approximation algorithms for scheduling problems: theoretical and practical results.*Journal of the ACM*, *34*, 144–162.

Hoogeveen, J. A., Lenstra, J. K. & Veltman, B. (1996). Preemptive scheduling in a two-stage multiprocessor flow shop is NP-hard. *European Journal of Operational Research, 89,* 172–175.

Jin, Z. H., Ohno, K., Ito, T., Elmaghraby, & S. E. (2002). Scheduling hybrid flowshops in printed circuit board assembly lines. *POM 11,* 216–230.

Johnson, S. M. (1954). Optimal two- and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly, 1,* 61–68.

Karp, R. M., (1972). Reducibility among combinatorial problems. In R. E. Miller,\& J. W. Thatcher (eds.), Complexity of Computer Computations, Plenum Press, 85–103.

Kubiak, W., Blazewicz, J., Formanowicz, P., Schmidt, G. (2002). Two-machine flowshop with limited machine availability. *European Journal of Operational Research, 136,* 528–540.

Langston, M. A. (1987). Interstage transportation planning in the deterministic flow-shop environment. *Operations Research, 35*(4), 556–564.

Lee, C.-Y., (1991). Parallel machine scheduling with nonsimultaneous machine availabile time. *Discrete Applied Mathematics, 30,* 53–61.

Lee, C.-Y. (1996). Machine scheduling with an availability constraint. *J.Global Optimization.9*, 363–382

Lee, C-Y, Cheng, T. C. E. & Lin, B. M. (1993). Minimizing the Makespan in the 3-Machine Assembly-Type Flowshop Scheduling Problem. *Management Science, 39*(5), 616–625.

Lee, C.-Y. & Vairaktarakis, G. L. (1994). Minimizing makespan in hybrid flowshops. *Operations Research Letters 16*, 149–158.

Lee, C.-Y., Lei, L. & Pinedo, M. (1997). Current trends in deterministic scheduling. *Annals of Operations Research, 70,* 1–41.

Lee, C.-Y (1997) Minimizing the makespan in the two-machine flowshop scheduling problem with an availability constraint. *Operations Research Letters, 20,* 129–139.

Lee, C.-Y., & Vairaktarakis, G. (1998). Performance Comparison of Some Classes of Flexible Flowshops and Job Shops. *International Journal of Flexible Manufacturing Systems,* (Special Issue on Manufacturing Flexibility) *10,* 379–405.

Lee, C.-Y. (1999). Two-Machine Flowshop Scheduling with Availability Constraints. *European Journal of Operational Research, 114*(2), 198–207.

Linn R, & Zhang W (1999) Hybrid flow shop scheduling: a survey". *Computers & Industrial Engineering, 37*, 57–61.

Mittenthall, J., & Ragavachari, M. (1993). Stochastic flowshops, *Operations Research, 30,* 148–162.

Narasimhan, S. L., Panwalkar, S. S. (1984). Scheduling in a two-stage manufacturing process. *International Journal of Production Research, 22,* 555–564.

Narasimhan, S., & Mangiameli, P. (1987). A comparison of sequencing rules for a two-stage hybrid flow shop. *Decisions Science, 18,* 250–265.

Paul, R. J. (1979). A production scheduling in the glass container industry. *Operations Research, 22,* 290–302.

Pinedo, M. (2002). Scheduling: Theory, Algorithms and Systems (Second Edin), Prentice-Hall.

Portman MC, Vignier A, Dardilhac D, & Dezalay D., (1998). Branch and bound crossed with GA to solve hybrid flowshops. *European Journal of Operational Research, 107,* 389–400.

Rajendran, C., & Chaudhuri, D. (1992). Scheduling in n-job, m-stage flowshop with parallel processors to minimize makespan. *International Journal of Production Economics 27,* 137–143.

Riane, F., Artiba, A., & Elmaghraby, S. E. (1998). A hybrid three-stage flowshop problem: Efficient heuristics to minimize makespan. *European Journal of Operational Research 109,*(2), 321–329.

Sahni, S. (1976). Algorithms for scheduling independent tasks. *Journal of the ACM*, *23,* 116–127.

Salvador, M. S., (1973). A solution to a special case of flow shop scheduling problems. In Symposium of the Theory of Scheduling and Applications, Elmaghraby, ed, 83–91.

Schmidt, G. (2000). Scheduling with limited machine availability. *European Journal of Operational Research, 121,* 1–15.

Schuurman, P., & Woeginger, G. J. (2000). A polynomial time approximation scheme for the two-stage multiprocessor flow shop problem, *Theo Computer Science, 237,* 105–122.

Sriskandarajah, C., & Sethi, S. P. (1989). Scheduling algorithms for flexible flow shops: worst and average case performance. *European Journal of Operational Research, 43,* 143–160.

Wittrock, R. J. (1985). Scheduling algorithms for flexible flow lines. *IBM Journal Research Development, 29,* 401–412.

Wittrock, R. J. (1988). An adaptable scheduling algorithm for flexible flow lines. *Journal Operations Research, 36,* 445–53.

Williamson, D. P., Hall, L. A., Hoogeveen. J. A., Hurkens, C. A. J., Lenstra, J. K., Sevastianov, S. V., & Shmoys, D. B. (1997). Short shop schedules. *Operations Research, 45,* 288–294

Zhou, J. R., Huang, J. R., & Jiang, W. S. (1996). Optimization production scheduling of multi-stage interrelated discrete system via synthetic knowledge. *Proceedings of the American Control Conference,* 724–728.

Chapter 13
A Probabilistic Characterization of Allocation Performance in a Worker-Constrained Job Shop

Benjamin J. Lobo, Kristin A. Thoney, Thom J. Hodgson, Russell E. King and James R. Wilson

We analyze a dual resource constrained (DRC) job shop in which both machines and workers are limited, and we seek to minimize $L_{\max}$, the maximum job lateness. An allocation of workers to machine groups is required to generate a schedule, and determining a schedule that minimizes $L_{\max}$ is NP-hard. This chapter details a probabilistic method for evaluating the quality of a specific (but arbitrary) allocation in a given DRC job shop scheduling problem based on two recent articles by Lobo et al. The first article (Lobo et al. 2013a) describes a lower bound on $L_{\max}$ given an allocation, and an algorithm to find an allocation yielding the smallest such lower bound, while the second article (Lobo et al. 2013b) establishes criteria for verifying the optimality of an allocation. For situations where the optimality criteria are not satisfied, Lobo et al. (2013b) presents HSP, a heuristic search procedure to find allocations enabling the Virtual Factory (a heuristic scheduler developed by Hodgson et al. in 1998) to generate schedules with smaller $L_{\max}$ than can be achieved with allocations yielding article 1's final lower bound. From simulation replications of the given DRC job shop scheduling problem, we estimate the distribution of the difference between (a) the "best" (smallest) $L_{\max}$ value achievable with a Virtual Factory–generated schedule, taken over all feasible allocations; and (b) the final lower bound of Lobo et al. (2013a). To evaluate the quality of a specific allocation for

B. J. Lobo (✉) · T. J. Hodgson · R. E. King · J. R. Wilson
Edward P. Fitts Department of Industrial and Systems Engineering,
North Carolina State University, Campus Box 7906, Raleigh, NC 27695–7906, USA
e-mail: bjlobo@gmail.com

K. A. Thoney
Department of Textile and Apparel, Technology and Management, North Carolina State University, Campus Box 8301, Raleigh, NC 27695–8301, USA
e-mail: kristin_thoney@ncsu.edu

T. J. Hodgson,
e-mail: hodgson@ncsu.edu

R. E. King
e-mail: king@ncsu.edu

J. R. Wilson
e-mail: jwilson@ncsu.edu

P. S. Pulat et al. (eds.), *Essays in Production, Project Planning and Scheduling,*
International Series in Operations Research & Management Science 200,
DOI 10.1007/978-1-4614-9056-2_13, © Springer Science+Business Media New York 2014

the given problem, we compute the difference between $L_{\max}$ for the Virtual Factory–generated schedule based on the specific allocation, and the associated lower bound (b) for the problem; then we refer this difference to the estimated distribution to judge the likelihood that the specific allocation yields the Virtual Factory's "best" schedule (a) for the given problem. We present several examples illustrating the usefulness of our approach, and summarize the lessons learned in this work.

Introduction

In conventional job shop scheduling, system operation is constrained only by the number of machines that are available to process jobs. This approach does not account for the additional constraint imposed by the number of workers that are available to operate the machines. Dual resource constrained (DRC) systems (Treleven and Elvers 1985) are subject to limits on the availability of both machines and manpower. Lobo et al. (2013a, b) addressed the problem of minimizing $L_{\max}$, the maximum job lateness, in a DRC job shop. Because the job shop scheduling problem with even a single constraint is NP-Hard (Lenstra and Rinnooy Kan 1979), the approach of Lobo et al. (2013a, b) involved the following: (a) seeking the most promising (hopefully optimal) allocation of the available workers to the machine groups (departments) in the job shop; and (b) seeking the best achievable schedule for the job shop based on this allocation.

Given an allocation ϑ of workers to machine groups, Lobo et al. (2013a) derived a lower bound LB_{ϑ} on $L_{\max}$ for all schedules based on that allocation. They also developed an algorithm to find an allocation ϑ^* that yields the smallest value of LB_{ϑ} over all feasible values of ϑ. The authors' final lower bound on $L_{\max}$ is $\mathrm{LB}_{\vartheta^*}$, and it provides a benchmark for evaluating heuristic solutions to the DRC job shop scheduling problem. Lobo et al. (2013b) established criteria for verifying that an allocation is optimal—i.e., the allocation corresponds to a feasible schedule that minimizes $L_{\max}$. For situations in which ϑ^* does not satisfy the optimality criteria, the authors developed HSP, a heuristic search procedure designed to find allocations enabling the Virtual Factory (a heuristic scheduler developed by Hodgson et al. (1998)) to generate schedules with smaller values of $L_{\max}$ than can be achieved with allocation ϑ^*. The Virtual Factory was chosen as the heuristic scheduler because of its proven track record in successfully generating optimal or near-optimal schedules in job shop scheduling problems for which the primary objective is to minimize $L_{\max}$ (Hodgson et al. 2000, 2004; Schultz et al. 2004; Thoney et al. 2002; Weintraub et al. 1999; Zozom et al. 2003), but Lobo et al. (2013a, b) note that their approach would work with other heuristic schedulers.

The use of heuristics (in this case, HSP and the Virtual Factory) introduces uncertainty into the properties of the delivered solution: if there is a substantial gap between $L_{\max}$ for the Virtual Factory–generated schedule based on an HSP-delivered allocation ϑ^{HSP} and the lower bound $\mathrm{LB}_{\vartheta^*}$, then we must have some reliable method for

evaluating the quality (closeness to optimality) of the allocation ϑ^{HSP}. Enumeration of the allocation search space involves using the Virtual Factory to generate a schedule for every feasible allocation. However, because the size of the allocation search space grows exponentially with an increase in either the number of machine groups or the number of workers, enumerating the set of feasible allocations is generally impractical given the usual constraints on the user's time and computing budget; see Lobo et al. (2013b) for experimental results supporting this conclusion.

A "VF-best" allocation ϑ^{VFB} enables the Virtual Factory to generate a schedule whose $L_{\max}$ value, denoted $\mathrm{VF}_{\vartheta^{\mathrm{VFB}}}$, is the smallest $L_{\max}$ that can be achieved by the Virtual Factory for a given DRC job shop scheduling problem, taken over all feasible worker allocations. To gauge the likelihood that a specific (but arbitrary) allocation is VF-best for the given problem, we consider simulation replications (i.e., randomly generated instances) of the given DRC job shop scheduling problem that are similar to the given problem in the following key respects:

- It has the same general pattern of symmetric or asymmetric loading of the machine groups;
- It has the same level of staffing—i.e., the ratio of the number of workers to the number of machines expressed as a percentage between 0 % and 100 %; and
- It has the same due-date range.

Each simulation replication involves randomly sampling the processing times and routes for all the jobs in the given DRC job shop scheduling problem as detailed in Lobo et al. (2013a); and from these replications we estimate the corresponding probability distribution of the difference $\mathrm{VF}_{\vartheta^{\mathrm{VFB}}}$–$\mathrm{LB}_{\vartheta^*}$. Now the given DRC job shop scheduling problem has fixed processing times and routes for all its jobs; thus the given problem has its own fixed values of the lower bound $\mathrm{LB}_{\vartheta^*}$, the HSP-delivered allocation ϑ^{HSP}, and the Virtual Factory–generated schedule based on ϑ^{HSP} with $L_{\max} = \mathrm{VF}_{\vartheta^{\mathrm{HSP}}}$. Insofar as the observed difference $\mathrm{VF}_{\vartheta^{\mathrm{HSP}}} - \mathrm{LB}_{\vartheta^*}$ for the given problem can be viewed as another sample from the population of differences of the form $\mathrm{VF}_{\vartheta^{\mathrm{VFB}}} - \mathrm{LB}_{\vartheta^*}$ that could be generated by simulation replications of the given problem, we can refer the specific realized value of $\mathrm{VF}_{\vartheta^{\mathrm{HSP}}} - \mathrm{LB}_{\vartheta^*}$ to the estimated distribution of differences in order to judge the likelihood that ϑ^{HSP} is a VF-best allocation for the given problem.

This chapter provides a comprehensive examination of the probabilistic analysis that was briefly introduced by Lobo et al. (2013b). In the second section we review the relevant literature. The third section contains essential background information on the DRC job shop scheduling problem and the overall solution approach used by Lobo et al. (2013a, b). In the fourth section we detail the rationale underlying our proposed method for evaluating the quality of worker allocations in a DRC job shop as well as the results of applying the method to 64 data sets. In the fifth section we present a variety of examples illustrating the usefulness of our method. The sixth section documents the issues encountered and lessons learned during the development and experimental evaluation of the method. The main conclusions of this work and directions for future research are summarized in the final section.

Literature Review

This work spans two main areas: dual resource constrained systems, and statistical performance evaluation of heuristics for planning the operation of those systems. A brief review of the relevant literature follows.

Dual Resource Constrained Systems

The operation of a dual resource constrained (DRC) system is subject to limitations on the availability of both equipment and manpower (Treleven 1989). According to Treleven and Elvers (1985), a DRC shop is "one in which shop capacity may be constrained by machine or labor capacity or both. This situation exists in shops that have equipment that is not fully staffed and machine operators who are capable of operating more than one piece of equipment. . . . [Workers] may be transferred from one work centre to another (subject to skills restrictions) as the demand dictates." Gargeya and Deane (1996) note that part of the complexity of scheduling in DRC systems stems from the need for an effective method to assign manpower to the machines.

Nelson (1967) documents one of the earliest studies of DRC systems. Treleven (1989) reviews the literature on DRC systems, summarizing the various models used, the parameters of the systems investigated, the job dispatching and worker allocation rules employed, and the different criteria used to evaluate system performance. Hottenstein and Bowman (1998) identify two main questions regarding the allocation of workers:

- When should workers move from one machine group to another?
- Where (to which machine group) should the workers move?

To address these questions, Hottenstein and Bowman summarize research concerning worker flexibility, centralization of control, worker allocation rules, queue discipline, and the cost of transferring workers. More recent work focuses on the effects of the following factors on various performance criteria for DRC job shops: (a) cross-training workers to operate machines in different departments (machine groups); and (b) incorporating more-realistic assumptions about worker behavior—e.g., learning, fatigue, and forgetfulness (Jaber and Neumann 2010; Kher 2000; Kher et al. 1999; Malhotra et al. 1993).

Felan et al. (1993) examine the effects of labor flexibility and staffing levels on several job shop performance measures. The authors consider a homogeneous workforce whose flexibility is based on each individual's ability to work in a given number of departments in the job shop; moreover they assume that for a given staffing level, each department has the same number of workers assigned to it. They measure the effects of labor flexibility and staffing level on work in progress (WIP), due-date performance, and cost criteria. According to Felan et al. (1993), these criteria "represent the primary drivers for measuring manufacturing performance

in many organizations." They find that for a given level of workforce flexibility, an increase in the staffing level yields an increase in system cost, a decrease in the WIP, and an improvement in the due-date performance. For a given staffing level, an increase in the worker flexibility level yields an increase in system cost, a decrease in the WIP, and an improvement in the due-date performance. The authors conclude that based on the diminishing returns apparent in the WIP and due-date performance, job shop performance is optimized with a staffing level of 60 % and a workforce that has a medium level of flexibility. Although Felan et al. (1993) allow workers to transfer between departments, they focus on jointly optimizing the levels of staffing and workforce flexibility. In contrast to this approach, Lobo et al. (2013a, b) focus on optimizing the allocation of workers to departments for a *given* staffing level.

The shifting bottleneck heuristic proposed by Adams et al. (1988) is an iterative procedure for finding a job shop schedule that minimizes the makespan (i.e., the completion time of the last job to leave the system). The set M_0 consists of all machines in the job shop for which a schedule has been generated so that initially $M_0 = \emptyset$; and the procedure terminates when all machines belong to M_0. On each iteration of the heuristic, every machine not belonging to M_0 is considered as a separate $1|r_j|L_{\max}$ scheduling problem that involves only the sequencing constraints for the machines currently belonging to M_0 and that yields its own minimum value of $L_{\max}$. The bottleneck machine corresponds to the largest such value of $L_{\max}$ for the machines not belonging to M_0. The bottleneck machine is added to M_0; and then each machine already belonging to M_0 is again considered as a separate $1|r_j|L_{\max}$ scheduling problem that involves only the sequencing constraints for the *other* machines currently belonging to M_0, including the newly added machine. If other machines still do not belong to M_0 at the end of an iteration of the heuristic, then a new iteration is performed to schedule the next bottleneck machine, add that machine to M_0, and finally reschedule all the other machines currently belonging to M_0. A more complete explanation of the shifting bottleneck heuristic can be found in Chap. 7 of Pinedo (2012).

The methodology of Lobo et al. (2013a, b) differs from the shifting bottleneck procedure in several significant respects. First, Lobo et al. consider a DRC system that is constrained by the availability not only of machines, but also of the workers needed to operate the machines. Second, the authors' method for computing $\mathrm{LB}_{\vartheta^*}$ allows preemption when determining a bottleneck machine group; and this relaxation of the original problem enables rapid computation of an effective lower bound on $L_{\max}$ for a specific allocation of workers to machine groups. Finally, each iteration of the heuristic scheduler (i.e., the Virtual Factory) uses estimated queuing times of *all* jobs in the job shop to reoptimize the schedule for *all* machines simultaneously. By contrast, each iteration of the shifting bottleneck heuristic uses only the job-sequencing constraints for the machines currently belonging to the set M_0.

While there has been extensive research on DRC systems, to our knowledge there has not been any published work on the problem addressed in this chapter—namely, the allocation of homogeneous, fully cross-trained workers in a DRC job shop so as to minimize $L_{\max}$.

Statistical Optimum Estimation

Purely statistical techniques based on extreme value theory can also be used to evaluate the performance of a heuristic procedure for determining the worker allocation in a DRC job shop scheduling problem (Derigs 1985; Golden and Alt 1979; Wilson et al. 2004). In particular if we seek to estimate the (finite) lower endpoint ω on the range of feasible values of $L_{\max}$ for a DRC job shop scheduling problem and if we have a mechanism for taking a random sample $\{X_i : i = 1, \ldots, n\}$ of size n from this population of feasible values, then it is natural to use the sample minimum,

$$Y_n = \min\{X_i : i = 1, \ldots, n\},$$

in deriving a useful estimator of ω. We seek a suitably standardized version of the sample minimum,

$$Z_n = (Y_n - \psi_n)/\zeta_n \quad \text{for } n = 1, 2, \ldots,$$

based on appropriate sequences $\{\psi_n : n = 1, 2, \ldots\}$ and $\{\zeta_n > 0 : n = 1, 2, \ldots\}$ of constants that stabilize the location and scale of Z_n as n increases. If the standardized sample minimum Z_n has a nondegenerate limiting cumulative distribution function (c.d.f.),

$$\lim_{n \to \infty} \Pr\{Z_n \le z\} = H(z) \quad \text{for all } z, \tag{13.1}$$

then the Extremal Types Theorem for Minima (Leadbetter et al. 1983, p. 29) ensures that the limiting c.d.f. in Eq. (13.1) must be a three-parameter Weibull distribution,

$$H(z) = \begin{cases} 0, & \text{if } z < \omega, \\ 1 - \exp\left[-\left(\dfrac{z - \omega}{\beta}\right)^{\alpha}\right], & \text{if } z \ge \omega, \end{cases} \tag{13.2}$$

where ω is the distribution's lower endpoint, α is a shape parameter, β is a scale parameter, and both α and β are positive.

In applications for which the condition (13.1) is satisfied, the methods of maximum likelihood or least squares can be used to estimate ω, α, and β from a random sample $\{X_i : i = 1, \ldots, mn\}$ that has been partitioned into m subsamples each of size n, where both m and n are sufficiently large. If $Y_{n,j}$ denotes the minimum observed in the jth subsample of size n for $j = 1, \ldots, m$, then we can fit a three-parameter Weibull distribution to the sample minima $\{Y_{n,j} : j = 1, \ldots, m\}$ as detailed in Coles (2001, Sect. 3.1.3), Golden and Alt (1979, pp. 73–74), or Wilson et al. (2004, Sect. 5). Unfortunately in applications to DRC job scheduling, we have encountered many situations in which the condition (13.1) is not satisfied because, for example, the original random variables $\{X_i\}$ (that is, the values of $L_{\max}$ delivered by our simulation-based procedures) have a discrete distribution or merely a nonzero probability mass concentrated at the lower endpoint ω as discussed in the fourth section below; and then the three-parameter Weibull model in Eqs. (13.1) and (13.2) breaks down.

Even in situations where condition (13.1) is satisfied, there is the challenge of taking sufficiently large values of m and n so that fitting a three-parameter Weibull distribution to the sample minima $\{Y_{n,j} : j = 1, \ldots, m\}$ will yield accurate and reliable point and confidence interval (CI) estimators of the desired lower endpoint ω. Wilson et al. (2004) find that for many job shop scheduling problems with relatively large values of m and n, the CI estimator of Golden and Alt (1979) with high coverage probability is often proved to be invalid when follow-up experimentation with a more-effective heuristic optimization procedure applied to the problem at hand yields values of $L_{\max}$ that lie below the lower limit of the delivered CI estimator for ω. By contrast, $\mathrm{LB}_{\vartheta^*}$ is guaranteed to be a valid lower bound on $L_{\max}$ for the DRC job shop scheduling problem. Moreover, the results in Lobo et al. (2013b) and in the fourth and fifth section of this chapter provide good evidence that $\mathrm{LB}_{\vartheta^*}$ effectively exploits system-specific information about each individual problem so as to provide a lower bound on $L_{\max}$ that can be useful in practice. In summary, we have found that $\mathrm{LB}_{\vartheta^*}$ consistently outperforms purely statistical estimators of the lower endpoint ω of the range of feasible values of $L_{\max}$ for the DRC job shop scheduling problem.

Background on the DRC Job Shop

The DRC job shop scheduling problem is denoted by $J_M|W|L_{\max}$, where M is the number of machines, W is the number of workers, and the objective is to minimize $L_{\max}$ (Pinedo 2012). Each job has a routing through the job shop, a processing time on each machine visited, an initial release time, and a due-date. The physical layout of a job shop naturally leads to organizing the machines into machine groups, where machines that perform the same or similar operations are located in close proximity to each other. Moreover, the machines in the same group require the same set of worker skills for operating each of those machines. Because of the limited availability of workers, an allocation ϑ for the DRC job shop specifies the number of workers assigned to each machine group. The work force is assumed to be homogeneous, so that each worker is able to operate each machine with equal skill and efficiency. Once a worker allocation ϑ has been specified, a schedule can be generated for the job shop based on that allocation. A schedule prescribes a specific order for processing each of the jobs assigned to each machine in the job shop.

To explore fully the properties of allocation ϑ^* and the associated lower bound $\mathrm{LB}_{\vartheta^*}$ on $L_{\max}$ for the DRC job shop scheduling problem, Lobo et al. (2013a, b) performed a large-scale simulation experiment on a system in which there are $M = 80$ machines, $S = 10$ machine groups, and W workers, where $W < M$. The experimental design encompassed systematic variation of the following factors: (a) the type of job shop, as reflected in symmetric (balanced) or asymmetric (unbalanced) loading of the machine groups; (b) the due-date range of the jobs; and (c) the staffing level expressed as $100(W/M)\%$. For the asymmetric (unbalanced) type of job shop, each machine group received, on average, the percentage of the total workload given in Table 13.1. The due-date range of the jobs varied from 200 up to 3,000 in increments

Table 13.1 Percentage of total job shop workload, on average, seen by each machine group.

Machine group	1	2	3	4	5	6	7	8	9	10
Percentage of workload seen, on average	14	14	14	10	8	8	8	8	8	8

of 400. Four staffing levels were considered: 60, 70, 80, and 90 %. Since there were 2 types of job shops, 8 due-date ranges, and 4 staffing levels, there were 64 different designated DRC job shop scheduling problems in the overall simulation experiment (i.e., 64 different combinations of factors (a), (b), and (c)).

For a designated DRC job shop scheduling problem, each simulation replication had 1,200 jobs to be processed; and the key characteristics of each job were assigned as follows.

(i) The job's due date was sampled from the discrete uniform distribution with the lower endpoint equal to zero and the upper endpoint equal to the given due-date range;
(ii) The job's number of operations was sampled from the discrete uniform distribution on the integers from 6 to 10;
(iii) The machine group for each operation was sampled from the appropriate discrete distribution on the integers from 1 to 10 (namely, the discrete uniform distribution for a symmetric job shop, and the discrete distribution in Table 13.1 for an asymmetric job shop), subject to the condition that each job could have at most three operations in the same machine group, but those operations could be nonconsecutive; and
(iv) The processing time for each operation was sampled from the discrete uniform distribution on the integers from 1 to 40.

To drive the sampling operations (i)–(iv), the authors used the random number generator of Park and Miller (1988). Moreover, the authors exploited the method of common random numbers (Law 2007, pp. 578–594) to ensure that for each combination of job shop type and due-date range, 200 independent simulation replications of that scenario were obtained via sampling operations (i)–(iv) so as to ensure that for $j = 1, \ldots, 200$, the jth replication of a given scenario is *exactly the same* for each of the four selected staffing levels. This approach enabled the authors to make a much sharper comparison of the differences in system performance for each of the staffing levels.

Although $\mathrm{LB}_{\vartheta^*}$ is guaranteed to be the smallest value of LB_{ϑ} taken over all feasible values of the worker allocation ϑ, the experimental results of Lobo et al. (2013b) showed that allocation ϑ^* and the corresponding Virtual Factory-generated schedule with $L_{\max} = \mathrm{VF}_{\vartheta^*}$ did not necessarily satisfy either of the authors' optimality criteria. Furthermore, allocation ϑ^* was not necessarily even a VF-best allocation. Therefore the authors developed a heuristic search procedure (HSP) to seek a VF-best allocation. Three distinct search heuristics compose HSP: the Local Neighborhood Search Strategy, and Queuing Time Search Strategies 1 and 2. These search heuristics are performed consecutively, where each search heuristic in turn makes use of the allocation information obtained by the preceding search heuristic(s). Taking the best of the allocations found by its three constituent search strategies, HSP finally

delivers allocation ϑ^{HSP}; then based on that allocation, the Virtual Factory generates a schedule with $L_{\max} = \text{VF}_{\vartheta^{\text{HSP}}}$. An essential complement to the development of HSP is a reliable method for evaluating our degree of confidence that ϑ^{HSP} is a VF-best allocation for the problem at hand.

A Probability Distribution for Performance to the Lower Bound

To gauge the user's confidence that the HSP-delivered allocation ϑ^{HSP} is in fact a VF-best allocation for a given DRC job shop scheduling problem, we estimate the distribution of the difference $\text{VF}_{\vartheta^{\text{VFB}}} - \text{LB}_{\vartheta^*}$ for simulation replications of the designated DRC job shop scheduling problem. To identify a VF-best allocation for each simulation replication of the given DRC job shop scheduling problem, we employ the partial enumeration strategy of Lobo et al. (2013b). In the following two sections, we detail a probabilistic method for evaluating the performance of VF-best allocations for a designated DRC job shop scheduling problem by using the estimated probability distribution of $\text{VF}_{\vartheta^{\text{VFB}}} - \text{LB}_{\vartheta^*}$ for the designated problem.

A Probabilistic Characterization of Performance to the Lower Bound

An Empirical Distribution Describing PLB(ϑ)

For each designated DRC job shop scheduling problem, we generate Q independent simulation replications of the problem, and then we restrict our attention to the Q' problem instances (where $0 \leq Q' \leq Q$) in which allocation ϑ^* did not satisfy either of the optimality criteria in Theorems 1 and 2 of Lobo et al. (2013b). For a particular simulation replication of the designated DRC job shop scheduling problem, we define the performance of an allocation ϑ relative to the lower bound on $L_{\max}$ as the difference

$$\text{PLB}(\vartheta) \equiv \text{VF}_{\vartheta} - \text{LB}_{\vartheta^*}. \tag{13.3}$$

For the ith simulation replication ($i = 1, \ldots, Q'$) whose corresponding allocation ϑ_i^* does not satisfy the optimality criteria of Lobo et al. (2013b), we use the authors' partial enumeration strategy to find a VF-best allocation ϑ_i^{VFB} whose performance relative to the lower bound $\text{LB}_{\vartheta_i^*}$ is measured by $\text{PLB}(\vartheta_i^{\text{VFB}})$. A histogram is then constructed from the resulting data set

$$\{\text{PLB}(\vartheta_i^{\text{VFB}}) : i = 1, \ldots, Q'\}. \tag{13.4}$$

The histogram's horizontal axis represents the range of possible values of the random variable $\text{PLB}(\vartheta^{\text{VFB}})$; and a segment of the horizontal axis encompassing the observed values (13.4) is partitioned into equal-width bins on which rectangles are erected to depict the relative frequencies with which observations fall in the associated bins. The

area of each rectangle in the histogram is equal to the proportion of the observations falling in the bin at the base of the rectangle. Scott's (1979) rule is used to determine the bin width and the number of bins in the histogram. For each of the designated DRC job shop scheduling problems considered in this chapter, we examine the histogram and the associated empirical c.d.f. describing the data set (13.4) obtained through simulation replications of that designated DRC job shop scheduling problem.

A Theoretical Probability Distribution Describing PLB(ϑ^{VFB})

For a given DRC job shop scheduling problem with a specific allocation ϑ of workers to machines for that problem (perhaps ϑ is the HSP-delivered allocation ϑ^{HSP}), we seek to estimate our level of confidence that ϑ is a VF-best allocation, given that ϑ does not satisfy the optimality criteria of Lobo et al. (2013b). Recall that each simulation replication of the given DRC job shop scheduling problem and the given DRC job shop scheduling problem itself share the following key properties: (a) the job shop type (symmetric or asymmetric); (b) the staffing level (ratio of workers to machines, expressed as a percentage); and (c) the due-date range. Because of these similarities, we have some reason to expect that in computing the difference

$$\text{PLB}(\vartheta^{\text{VFB}}) = \text{VF}_{\vartheta^{\text{VFB}}} - \text{LB}_{\vartheta^*} \tag{13.5}$$

for either a simulation replication of the given problem or for the given problem itself, the nuisance effects arising from unique characteristics of the underlying problem will be common to both terms $\text{VF}_{\vartheta^{\text{VFB}}}$ and LB_{ϑ^*} in Eq. (13.5) so that the nuisance effects will cancel out; and the remaining component of the difference (13.5) will depend mainly on the properties of the VF-best allocation ϑ^{VFB} delivered by the Virtual Factory and on the key characteristics (a)–(c) that are shared by the simulation replication and the given DRC job shop scheduling problem.

If allocation ϑ for the given problem is in fact a VF-best allocation, then we may regard the difference $\text{PLB}(\vartheta) = \text{VF}_{\vartheta} - \text{LB}_{\vartheta^*}$ defined by Eq. (13.3) as another observation from the population of differences of the form (13.5). In the spirit of statistical hypothesis testing, we may therefore test the null hypothesis that ϑ is a VF-best allocation for the given problem by estimating the probability that for a simulation replication, the resulting random variable $\text{PLB}(\vartheta^{\text{VFB}})$ in Eq. (13.5) will be greater than the fixed value $\text{PLB}(\vartheta)$ for the given DRC job shop scheduling problem. This upper tail probability can be viewed as the p-value (significance probability) for the test of the null hypothesis that ϑ is a VF-best allocation for the designated problem (Bickel and Doksum 2007, pp. 221–223). From a different perspective, we can interpret this upper tail probability as our level of confidence that ϑ is a VF-best allocation for the given problem. For example, if approximately 90 % of the observations in the data set (13.4) are larger than $\text{PLB}(\vartheta)$ for the given problem, then we can conclude that allocation ϑ is better than approximately 90 % of the VF-best allocations for simulation replications of the given problem; and thus we can be 90 % confident that allocation ϑ is in fact a VF-best allocation for the given DRC job shop scheduling problem.

Fitting a Theoretical Distribution to Random Samples of **PLB**(ϑ^{VFB})

To facilitate their use in practice, we sought to approximate the histogram and the empirical c.d.f. based on the data set (13.4) for a designated DRC job shop scheduling problem by fitting an appropriate standard probability distribution to that data set. We used the Stat::Fit software (Geer Mountain Software Corp. 2001) for this purpose. Thus we obtained the following visual representations of the resulting fit:

- A graph of the histogram of the data set (13.4) superimposed on the fitted probability density function (p.d.f.); and
- A graph of the empirical c.d.f. of the data set (13.4) superimposed on the fitted c.d.f.

In addition, a p-value for the chi-squared goodness-of-fit test was provided for each fitted distribution.

Because Q' varied substantially across the different designated DRC job shop scheduling problems, care was taken when using the p-values as indicators of the goodness-of-fit. When Q' is very large, "practically insignificant discrepancies between the empirical and theoretical distributions often appear statistically significant" (Kuhl et al. 2010). On the other hand, very small values of Q' often result in relatively large p-values because standard goodness-of-fit statistics have low power to distinguish between different distributions based on small samples. These considerations indicate that the p-value cannot be relied on as the sole goodness-of-fit criterion. The p-value from the chi-squared goodness-of-fit test and graphs of both the fitted p.d.f. and the fitted c.d.f. were used to decide which distribution best characterized a given data set.

If the data set (13.4) could be adequately modeled by a standard continuous distribution, then we fitted the associated p.d.f. to the data set as detailed in the following section below. In some situations, the data set (13.4) exhibited a substantial percentage of observations at zero; and in those situations, we used a mixed distribution with nonzero probability mass at the origin and with a continuous right-hand tail having a standard functional form as described below. In other situations we were forced to use a discrete probability mass function (p.m.f.) to describe the data set (13.4). The latter situation arose in cases where there were relatively few distinct nonzero values of PLB(ϑ^{VFB}) in the data set. In the fits that follow, Q, the number of simulation replications for each designated DRC job shop scheduling problem, is 500.

Continuous distributions

For the data set (13.4) associated with each designated DRC job shop scheduling problem, initially we used Stat::Fit to seek the best-fitting continuous distribution. When adequate fits were obtained, the following distributions were used:

- the generalized Beta distribution, denoted as BETA(α, β, a, b), where α and β are the two shape parameters, a is the lower limit, and b is the upper limit;
- the shifted Gamma distribution, denoted as GAMMA(α, β, a), where α is the shape parameter, β is the scale parameter, and a is the lower limit;
- the bounded Johnson distribution, denoted as JOHNSON-SB$(\gamma, \delta, \lambda, \xi)$, where γ and δ are shape parameters, λ is the scale parameter representing the range of possible values of the corresponding random variable, and ξ is the location parameter representing the lower limit of the distribution (Kuhl et al. (2010));
- the unbounded Johnson distribution, denoted as JOHNSON-SU$(\gamma, \delta, \lambda, \xi)$, where γ and δ are shape parameters, λ is the scale parameter (but not the range of possible values of the corresponding random variable, which is infinite in both directions), and ξ is a scale parameter (but not the lower limit of the distribution, which is $-\infty$; see Kuhl et al. (2010));
- the shifted Lognormal distribution, denoted as LOGNORMAL(μ, σ, a), where μ and σ are the mean and standard deviation of the corresponding random variable respectively, and a is the lower limit; and
- the shifted (three-parameter) Weibull distribution, denoted as WEIBULL(α, λ, a), where α is the shape parameter, λ is the scale parameter, and a is the lower limit.

Figure 13.1 depicts four examples of continuous distributions that were fitted to data sets of the form (13.4). The Appendix contains similar graphs for all relevant designated DRC job shop scheduling problems. Tables 13.2 and 13.3 summarize the "best fits" obtained for the relevant designated DRC job shop scheduling problems corresponding to asymmetric and symmetric job shops, respectively. Note that in the corresponding graphs for data sets fitted with a continuous distribution, the legend "Datapoints" identifies the size of the data set, Q'. Both the graphical evidence and the p-values for the goodness-of-fit tests indicated that in each designated DRC job shop scheduling problem for which a continuous distribution was used to approximate the data set, the resulting model was adequate.

Mixed Distributions

If there was a nonnegligible probability mass at PLB$(\vartheta^{\text{VFB}}) = 0$, then it was not possible to fit the data set with a standard continuous distribution. In these cases, the data sets were fitted using a mixed c.d.f. of the form

$$F(x) = p_0 F_0(x) + (1 - p_0) F_c(x) \qquad \text{for } -\infty < x < \infty,$$

where: (a) p_0 is the proportion of observations of PLB(ϑ^{VFB}) that are equal to zero; (b) $F_0(x)$ is the c.d.f. of the degenerate distribution with unit probability mass at the origin so that

$$F_0(x) = \begin{cases} 0, & \text{for } x < 0, \\ 1, & \text{for } x \geq 0; \end{cases}$$

and (c) $F_c(x)$ is the conditional c.d.f. of PLB(ϑ^{VFB}) given that PLB $(\vartheta^{\text{VFB}}) > 0$, where we may take $F_c(x)$ to be the c.d.f. of any of the distributions listed previously.

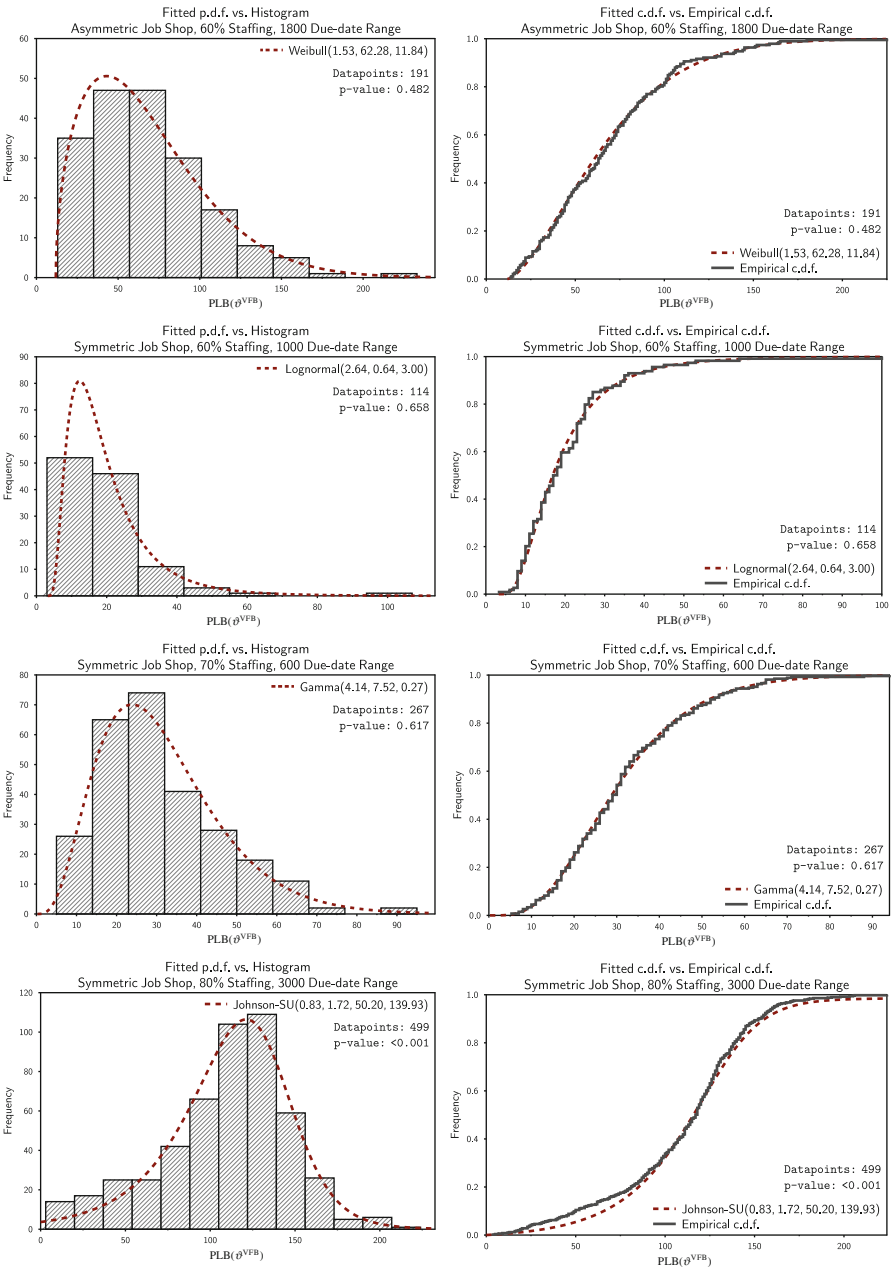


Fig. 13.1 Probability distribution fitting, continuous distribution fits

Table 13.2 All asymmetric job shop probability distribution fits

Due-date Range	Best Fit	p-value	Sample Size Q'
60 % Staffing			
200	GAMMA(2.68, 19.03, 8.00)	0.818	159
600	WEIBULL(1.53, 53.58, 7.41)	0.930	151
1000	WEIBULL(1.51, 55.31, 8.00)	0.522	159
1400	WEIBULL(1.36, 55.42, 11.49)	0.857	176
1800	WEIBULL(1.53, 62.28, 11.84)	0.482	191
2200	GAMMA(2.28, 26.79, 12.00)	0.581	207
2600	GAMMA(3.88, 18.22, 3.17)	0.901	219
3000	GAMMA(3.37, 21.60, 8.25)	0.802	242
70 % Staffing			
200	$0.342F_0(x) + 0.658F_c(x)$, where $F_c(\cdot) \sim$ WEIBULL(1.49, 48.60, 1.00)	0.484	219
600	$0.391F_0(x) + 0.609F_c(x)$, where $F_c(\cdot) \sim$ WEIBULL(1.50, 49.14, 0.43)	0.875	258
1000	$0.347F_0(x) + 0.653F_c(x)$, where $F_c(\cdot) \sim$ WEIBULL(1.29, 47.88, 0.48)	0.643	291
1400	$0.338F_0(x) + 0.662F_c(x)$, where $F_c(\cdot) \sim$ WEIBULL(1.25, 48.01, 0.63)	0.333	311
1800	$0.311F_0(x) + 0.689F_c(x)$, where $F_c(\cdot) \sim$ BETA(1.12, 2.91, 1.00, 168.74)	0.405	318
2200	$0.348F_0(x) + 0.652F_c(x)$, where $F_c(\cdot) \sim$ BETA(0.99, 2.20, 1.00, 160.45)	0.614	345
2600	$0.258F_0(x) + 0.742F_c(x)$, where $F_c(\cdot) \sim$ BETA(0.87, 2.26, 1.00, 175.00)	0.104	361
3000	$0.210F_0(x) + 0.790F_c(x)$, where $F_c(\cdot) \sim$ BETA(0.87, 2.59, 1.00, 186.00)	0.002	377
80 % Staffing			
200	DISCRETE PROBABILITY MASS FUNCTION	–	–
600	DISCRETE PROBABILITY MASS FUNCTION	–	–
1000	DISCRETE PROBABILITY MASS FUNCTION	–	–
1400	DISCRETE PROBABILITY MASS FUNCTION	–	–
1800	DISCRETE PROBABILITY MASS FUNCTION	–	–
2200	DISCRETE PROBABILITY MASS FUNCTION	–	–
2600	$0.462F_0(x) + 0.538F_c(x)$, where $F_c(\cdot) \sim$ LOGNORMAL(1.43, 0.78, 1.00)	0.601	143
3000	$0.399F_0(x) + 0.601F_c(x)$, where $F_c(\cdot) \sim$ LOGNORMAL(1.52, 1.00, 0.29)	0.777	188
90 % Staffing			
200	DISCRETE PROBABILITY MASS FUNCTION	–	–
600	DISCRETE PROBABILITY MASS FUNCTION	–	–
1000	DISCRETE PROBABILITY MASS FUNCTION	–	–
1400	DISCRETE PROBABILITY MASS FUNCTION	–	–
1800	DISCRETE PROBABILITY MASS FUNCTION	–	–
2200	DISCRETE PROBABILITY MASS FUNCTION	–	–
2600	$0.433F_0(x) + 0.567F_c(x)$, where $F_c(\cdot) \sim$ LOGNORMAL(1.43, 0.82, 1.00)	0.311	141
3000	$0.387F_0(x) + 0.613F_c(x)$, where $F_c(\cdot) \sim$ LOGNORMAL(1.46, 1.03, 0.31)	0.095	191

The following are examples of designated DRC job shop scheduling problems fitted with a mixed distribution: (a) the asymmetric job shop with 70 % staffing and a due-date range of 1800; and (b) the symmetric job shop with 90 % staffing and a due-date range of 2600. For each of these designated problems, Fig. 13.2 displays the continuous distribution that provided the "best fit" to the associated subsample consisting of the nonzero values of PLB(ϑ^{VFB}). Figure 13.2 also shows $Q'(1 - p_0)$, the size of the associated subsample (labeled "Datapoints"), and the p-value for the chi-squared goodness-of-fit test. The Appendix contains similar graphs for all relevant designated

Table 13.3 All symmetric job shop probability distribution fits

Due-date Range	Best Fit	p-value	Sample Size Q'
60 % Staffing			
200	GAMMA(2.27, 10.33, 2.00)	0.994	143
600	GAMMA(2.77, 7.30, 1.49)	0.807	121
1000	LOGNORMAL(2.64, 0.64, 3.00)	0.658	114
1400	GAMMA(2.87, 6.64, 1.00)	0.203	126
1800	LOGNORMAL(2.73, 0.65, 2.00)	0.873	128
2200	GAMMA(2.91, 5.73, 5.00)	0.785	119
2600	WEIBULL(1.99, 23.90, 3.35)	0.086	132
3000	JOHNSON-SU(-0.90, 1.51, 12.45, 18.70)	0.498	151
70 % Staffing			
200	GAMMA(2.61, 11.42, 5.00)	0.381	282
600	GAMMA(4.14, 7.52, 0.27)	0.617	267
1000	GAMMA(2.90, 9.62, 5.00)	0.445	267
1400	GAMMA(5.11, 7.06, 0.00)	0.187	304
1800	GAMMA(6.10, 6.43, 0.00)	0.192	313
2200	GAMMA(7.37, 6.13, 0.00)	0.475	349
2600	GAMMA(8.48, 6.07, 0.00)	0.808	370
3000	WEIBULL(2.47, 47.46, 18.17)	0.171	403
80 % Staffing			
200	BETA(3.48, 9.20, 0.00, 194.71)	0.029	409
600	JOHNSON-SU(-0.89, 2.00, 39.47, 35.83)	0.713	421
1000	JOHNSON-SB(2.11, 1.79, 233.70, 6.27)	0.193	436
1400	JOHNSON-SU(-0.35, 1.51, 31.21, 62.22)	0.522	458
1800	JOHNSON-SU(-0.07, 1.50, 32.27, 77.52)	0.602	467
2200	JOHNSON-SU(-0.45, 4.40, 116.97, 75.00)	0.296	472
2600	JOHNSON-SU(0.22, 1.24, 27.01, 106.23)	0.440	486
3000	JOHNSON-SU(0.83, 1.72, 50.20, 139.93)	<0.001	499
90 % Staffing			
200	$0.429F_0(x) + 0.571F_c(x)$, where $F_c(\cdot) \sim$ WEIBULL(1.30, 38.26, 0.26)	0.244	175
600	$0.378F_0(x) + 0.622F_c(x)$, where $F_c(\cdot) \sim$ GAMMA(1.00, 35.91, 1.00)	0.736	217
1000	$0.314F_0(x) + 0.686F_c(x)$, where $F_c(\cdot) \sim$ GAMMA(1.12, 31.43, 1.00)	0.124	264
1400	$0.252F_0(x) + 0.748F_c(x)$, where $F_c(\cdot) \sim$ GAMMA(1.00, 35.38, 1.00)	0.609	294
1800	$0.223F_0(x) + 0.777F_c(x)$, where $F_c(\cdot) \sim$ GAMMA(1.00, 39.40, 1.00)	0.304	323
2200	$0.163F_0(x) + 0.837F_c(x)$, where $F_c(\cdot) \sim$ GAMMA(1.00, 40.55, 1.00)	0.150	375
2600	$0.111F_0(x) + 0.889F_c(x)$, where $F_c(\cdot) \sim$ JOHNSON-SB(0.99, 0.77, 158.79, -0.33)	0.727	423
3000	BETA(1.35, 2.77, 0.00, 110.91)	0.164	483

DRC job shop scheduling problems. The "best fits" for all relevant designated DRC job shop scheduling problems are summarized in Tables 13.2 and 13.3 for asymmetric and symmetric job shops, respectively. Both the graphical evidence and the p-values for the goodness-of-fit tests indicated that in every designated DRC job shop scheduling problem for which a mixed distribution was used to approximate the data set, the resulting model was adequate.

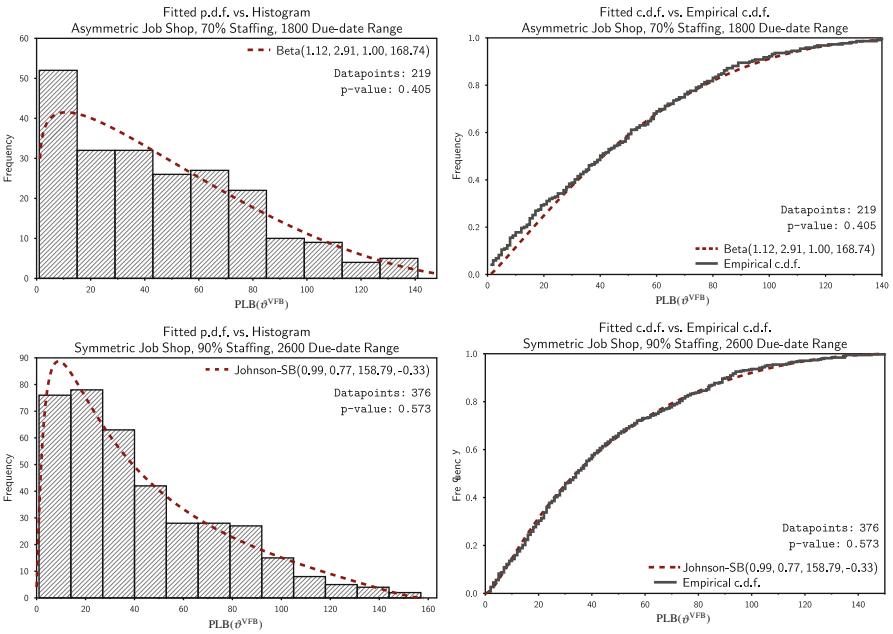


Fig. 13.2 Probability distribution fitting, part $F_c(\cdot)$ of the mixed distribution fits

Discrete Distributions

For a number of designated DRC job shop scheduling problems (e.g., the asymmetric job shop with 90 % staffing and a due-date range of 2200), there were so few values of $\mathrm{PLB}(\vartheta^{\mathrm{VFB}})$ distinct from zero in the data set that a continuous distribution could not be reliably fitted to the remaining nonzero values. In this case, we simply used a discrete distribution on the observed values in the data set, with each distinct value weighted by the relative frequency of its occurrence in the data set. Table 13.4 shows the discrete p.m.f. for an asymmetric job shop with 90 % staffing and a due-date range of 2200. Of the $Q = 500$ simulation replications, there were only $Q' = 97$ instances in which allocation ϑ^* did not satisfy either optimality criterion. The data set contained 54 instances for which $\mathrm{PLB}(\vartheta^{\mathrm{VFB}}) = 0$ and 43 nonzero values of $\mathrm{PLB}(\vartheta^{\mathrm{VFB}})$, which made it an ideal candidate for fitting with a discrete distribution. All the designated DRC job shop scheduling problems fitted with a discrete distribution can be found in Tables 13.2 and 13.3.

Fitting a Distribution in General

Recall that the overall experimental design included 64 designated DRC job shop scheduling problems. In this section we discuss the results of applying our distribution-fitting procedures to all 64 associated data sets of the form (13.4). More

Table 13.4 Discrete empirical distribution, asymmetric job shop, 2200 due-date range, 90 % staffing

Value, x	Frequency	Probability
0	54	0.557
1	4	0.041
2	4	0.041
3	5	0.052
4	4	0.041
5	7	0.072
6	3	0.031
7	1	0.010
8	1	0.010
9	0	0.000
10	1	0.010
11	1	0.010
12	2	0.021
13	2	0.021
14	1	0.010
15	1	0.010
16	2	0.021
17	0	0.000
18	1	0.010
19	0	0.000
20	0	0.000
21	1	0.010
22	0	0.000
⋮	⋮	⋮
26	0	0.000
27	1	0.010
28	0	0.000
29	0	0.000
30	0	0.000
31	1	0.010
$x > 31$	0	0.000

than 10 % of the observed values of PLB(ϑ^{VFB}) were equal to zero for each of the following cases: (a) the symmetric job shop with 90 % staffing and a due-date range of 200 through 2600; (b) the asymmetric job shop with 70 % staffing; (c) the asymmetric job shop with 80 % staffing and a due-date range of 2600 and 3000; and (d) the asymmetric job shop with 90 % staffing and a due-date range of 2600 and 3000. In these cases, a conditional p.d.f. was fitted to the data set composed of the nonzero values of PLB(ϑ^{VFB}). Less than 50 nonzero values of PLB(ϑ^{VFB}) were observed in each of the designated problems: (i) the asymmetric job shop with 80 % staffing and a due-date range of 200 through 2200; and (ii) the asymmetric job shop with 90 % staffing and a due-date range of 200 through 2200. In cases (i) and (ii), the observed values in each data set were used to define a discrete probability distribution.

We found that, in general, the generalized Beta distribution, the shifted Gamma distribution, the shifted Lognormal distribution, and the shifted Weibull distribution could be used to characterize the majority of the data sets. The exception to this was the symmetric job shop with 80 % staffing, where the unbounded Johnson distribution approximated the left-hand tail of the associated data sets substantially better than

any of the other standard continuous distributions. (Kuhl et al. (2010) discuss several diverse engineering applications in which unbounded Johnson distributions yield superior fits to nonstandard tail behavior.) In the symmetric job shop with 80 % staffing and a due-date range of 3000 the small p-value was not a concern when the value of Q' and the graphs of the fitted p.d.f. and c.d.f. (see Fig. 13.1) were taken into account.

The practical applicability of our distribution-fitting approach is clearly demonstrated by the diversity of designated DRC job shop scheduling problems for which the associated data set (13.4) can be adequately modeled by these six different probability distributions in a straightforward manner.

Using the Fitted Distributions to Evaluate Allocation Quality

In the next section we present four examples illustrating the use of the fitted distributions to estimate the user's degree of confidence that a specific allocation ϑ is in fact a VF-best allocation for a given DRC job shop scheduling problem. In the following section we propose an application of the fitted distributions to design a new probabilistic stopping rule for worker-allocation search heuristics such as HSP.

Using the Theoretical Probability Distribution

One of the major advantages of having fitted the data sets of the form (13.4) using standard probability distributions is that evaluating the quality of an allocation ϑ is straightforward. In most of the examples that follow, exact answers can be obtained analytically; and when exact answers are unavailable in a convenient closed form, accurate numerical approximations to the desired tail probabilities can be readily obtained from public-domain mathematical and statistical software packages, commercial spreadsheets, etc. For each of the given DRC job shop scheduling problems described in the sections below, the simulation replications were generated as detailed previously so that the mean processing time for each operation was 20.5 time units (this is the same configuration used in all our other examples).

Symmetric Job Shop with 60 % Staffing and Due-Date Range of 200

The first example involves a symmetric job shop with 60 % staffing and a due-date range of 200. From Table 13.3, the fitted c.d.f. corresponding to this given DRC job shop scheduling problem is $F(x) \sim \text{GAMMA}(2.27, 10.33, 2.00)$. The allocation ϑ^* returned by LBSA (Lobo et al. 2013a) yielded the following results: $\text{LB}_{\vartheta^*} = 4656$; $\text{VF}_{\vartheta^*} = 4727$; $\text{PLB}(\vartheta^*) = 71$; and ϑ^* did not satisfy the optimality criteria of Lobo et al. (2013b). In light of these results, we evaluated

$$\begin{aligned}\Pr\left\{\text{PLB}(\vartheta^{\text{VFB}}) \leq \text{PLB}(\vartheta^*)\right\} &= F(71)\\ &= 0.985;\end{aligned}$$

therefore we were only 1.5 % confident that allocation ϑ^* was a VF-best allocation. Because of the low level of confidence associated with ϑ^*, we used HSP to search for a better allocation. The HSP-delivered allocation ϑ^{HSP} yielded $\text{PLB}(\vartheta^{\text{HSP}}) = 17$. From this result we obtained

$$\Pr\{\text{PLB}(\vartheta^{\text{VFB}}) \leq \text{PLB}(\vartheta^{\text{HSP}})\} = F(17) = 0.346;$$

therefore we were 65.4 % confident that ϑ^{HSP} was a VF-best allocation. We concluded that ϑ^{HSP} was a substantially better allocation than ϑ^* in this case.

Asymmetric Job Shop with 70 % Staffing and Due-Date Range of 1000

The second example involves an asymmetric job shop with 70 % staffing and a due-date range of 1000. From Table 13.2, the fitted c.d.f. corresponding to this given DRC job shop scheduling problem is $F(x) = 0.347F_0(x) + 0.653F_c(x)$, where $F_c(\cdot) \sim \text{WEIBULL}(1.29, 47.88, 0.48)$. Allocation ϑ^* yielded the following results: $\text{LB}_{\vartheta^*} = 3347$; $\text{VF}_{\vartheta^*} = 3370$; $\text{PLB}(\vartheta^*) = 23$; and ϑ^* did not satisfy the optimality criteria. Consequently we evaluated

$$\begin{aligned}\Pr\{\text{PLB}(\vartheta^{\text{VFB}}) \leq \text{PLB}(\vartheta^*)\} &= F(23)\\ &= 0.347F_0(23) + 0.653F_c(23)\\ &= 0.347 \cdot 1 + 0.653 \cdot \left[1 - \exp\left\{-\left(\frac{23 - 0.48}{47.88}\right)^{1.29}\right\}\right]\\ &= 0.553;\end{aligned}$$

therefore we were 44.7 % confident that allocation ϑ^* was a VF-best allocation. Procedure HSP was again employed; and in this case the HSP-delivered allocation ϑ^{HSP} yielded $\text{PLB}(\vartheta^{\text{HSP}}) = 0$. For this simulation replication allocation ϑ^{HSP} was both a VF-best allocation and an optimal allocation.

Symmetric Job Shop with 80 % Staffing and Due-Date Range of 2600

The third example involves a symmetric job shop with 80 % staffing and a due-date range of 2600. From Table 13.3, the fitted c.d.f. corresponding to this given DRC job shop scheduling problem is $F(x) \sim \text{JOHNSON-SU}(0.22, 1.24, 27.01, 106.23)$. Allocation ϑ^* yielded the following results: LB_{ϑ^*}=783; VF_{ϑ^*}= 841; $\text{PLB}(\vartheta^*) = 58$; and ϑ^* did not satisfy the optimality criteria. In view of these results, we evaluated

$$\begin{aligned}\Pr\{\text{PLB}(\vartheta^{\text{VFB}}) \leq \text{PLB}(\vartheta^*)\} &= F(58)\\ &= \Phi\left\{0.22 + 1.24 \ln\left[\left(\frac{58 - 106.23}{27.01}\right)\right.\right.\\ &\qquad \left.\left. + \sqrt{\left(\frac{58 - 106.23}{27.01}\right)^2 + 1}\right]\right\}\\ &= 0.074,\end{aligned}$$

where $\Phi(\cdot)$ denotes the standard normal c.d.f.; consequently we were 92.6 % confident that allocation ϑ^* was a VF-best allocation. Using HSP, we found an allocation ϑ^{HSP} that yielded the results $\mathrm{PLB}(\vartheta^{\mathrm{HSP}}) = 47$ and

$$\Pr\left\{\mathrm{PLB}(\vartheta^{\mathrm{VFB}}) \leq \mathrm{PLB}(\vartheta^{\mathrm{HSP}})\right\} = F(47) = 0.047;$$

therefore we were 95.3 % confident that allocation ϑ^{HSP} was a VF-best allocation. Even though we obtained a high degree of confidence that allocation ϑ^* was a VF-best allocation, HSP was able to find an allocation ϑ^{HSP} with an even higher degree of confidence. In this application, it is arguable whether there was a *practically significant* difference in the performance of ϑ^{HSP} compared with that of ϑ^*.

Symmetric Job Shop with 90 % Staffing and Due-Date Range of 2200

The fourth example involves a symmetric job shop with 90 % staffing and a due-date range of 2200. From Table 13.3, the fitted c.d.f. corresponding to this given DRC job shop scheduling problem is $F(x) = 0.163F_0(x) + 0.837F_{\mathrm{c}}(x)$, where $F_{\mathrm{c}}(\cdot) \sim \textsc{Gamma}(1.00, 40.55, 1.00)$. Allocation ϑ^* yielded the following results: $\mathrm{LB}_{\vartheta^*} = 977$; $\mathrm{VF}_{\vartheta^*} = 1024$; $\mathrm{PLB}(\vartheta^*) = 47$; and ϑ^* did not satisfy the optimality criteria. Consequently we evaluated

$$\begin{aligned}\Pr\left\{\mathrm{PLB}(\vartheta^{\mathrm{VFB}}) \leq \mathrm{PLB}(\vartheta^*)\right\} &= F(47)\\ &= 0.163F_0(47) + 0.837F_{\mathrm{c}}(47)\\ &= 0.163 \cdot 1 + 0.837 \cdot 0.6784 = 0.731;\end{aligned}$$

thus we were only 26.9 % confident that allocation ϑ^* was a VF-best allocation. The HSP-delivered allocation ϑ^{HSP} yielded the results $\mathrm{PLB}(\vartheta^{\mathrm{HSP}}) = 2$ and

$$\Pr\left\{\mathrm{PLB}(\vartheta^{\mathrm{VFB}}) \leq \mathrm{PLB}(\vartheta^{\mathrm{HSP}})\right\} = F(2) = 0.183;$$

consequently we were 81.7 % confident that allocation ϑ^{HSP} was a VF-best allocation. From the perspective of absolute performance, we concluded that *for all practical purposes* ϑ^{HSP} was a VF-best allocation because the (unknown) optimality gap $\mathrm{VF}_{\vartheta^{\mathrm{VFB}}} - \mathrm{LB}_{\vartheta^*}$ for the given problem was bounded above by $\mathrm{VF}_{\vartheta^{\mathrm{HSP}}} - \mathrm{LB}_{\vartheta^*} = 2$ time units, which was less than 10 % of an average operation processing time. Moreover from the perspective of relative performance, there was a *practically significant* improvement in the performance of ϑ^{HSP} compared with that of ϑ^* because we obtained a reduction of $100[(47 - 2)/47]\% = 95.7$ % in the performance-to-lower-bound statistic by using allocation ϑ^{HSP} instead of ϑ^*. Providing additional insights into allocation performance is the main objective of our proposed method for evaluating the user's degree of confidence in having obtained a VF-best allocation—especially in situations for which other methods for absolute or relative performance evaluation yield ambiguous results.

A Probabilistic Stopping Rule

The three different heuristic search strategies that constitute HSP all have the same set of three stopping rules:

(a) An optimal solution has been found (i.e., the allocation satisfies one of the optimality criteria of Theorems 1 and 2 of Lobo et al. (2013b));
(b) All relevant allocations have been searched; or
(c) The execution time limit has been reached.

In addition to rules (a), (b), and (c) above, the fitted probability distributions can be used to formulate a probabilistic stopping rule. Let ω denote a user-specified confidence coefficient such that $0 < \omega < 1$. The proposed probabilistic stopping rule requires a confidence level of at least $100(1-\omega)\%$ that for the DRC job shop scheduling problem at hand, a specific allocation ϑ is a VF-best allocation. If an allocation is encountered during the search process that satisfies this stopping rule, then the search process will terminate even if the maximum search time (c) has not elapsed. Implementation of such a stopping rule should decrease the average execution time needed by the search strategy. On the other hand, we shall see that a probabilistic stopping rule must be carefully adapted to the designated DRC job shop scheduling problem being studied.

For a given value of $\omega \in (0, 1)$, if $F(x)$ denotes the c.d.f. of PLB(ϑ^{VFB}) for a simulation replication of a given DRC job shop scheduling problem, then both the functional form of $F(x)$ and the ω quantile of PLB(ϑ^{VFB}),

$$x_\omega = F^{-1}(\omega) = \min\{x : F(x) \geq \omega\}, \tag{13.6}$$

will in general depend strongly on the problem type, the staffing level, and the due-date range. To illustrate this point, we considered the following three designated DRC job shop scheduling problems:

1. The symmetric job shop with 80 % staffing, a due-date range of 2200, and the c.d.f.

$$F_1(x) \sim \text{JOHNSON-SU}(-0.45, 4.40, 116.97, 75.00);$$

2. The asymmetric job shop with 60 % staffing, a due-date range of 1400, and the c.d.f.

$$F_2(x) \sim \text{WEIBULL}(1.36, 55.42, 11.49);$$

and

3. The asymmetric job shop with 90 % staffing, a due-date range of 2600, and the c.d.f.

$$F_3(x) \sim 0.433 F_0(x) + 0.567 F_c(x),$$

where

$$F_c(x) \sim \text{LOGNORMAL}(1.43, 0.82, 1.00).$$

For each of the above designated DRC job shop scheduling problems, a given problem was generated as detailed previously so that the mean processing time for each operation was 20.5 time units (this is the same configuration used in all our other examples). To satisfy Eq. 13.6 when $\omega = 0.05$, we have the following requirements:

- For the first given DRC job shop scheduling problem, an allocation ϑ_1 is needed such that $\text{PLB}(\vartheta_1) \leq x_\omega = 42$;
- For the second given DRC job shop scheduling problem, an allocation ϑ_2 is needed such that $\text{PLB}(\vartheta_2) \leq x_\omega = 17$; and
- For the third given DRC job shop scheduling problem, an allocation ϑ_3 is needed such that $\text{PLB}(\vartheta_3) = x_\omega = 0$.

Note that for $i = 1$, 2, and 3, the required values of $\text{PLB}(\vartheta_i)$ were determined analytically; and they have been rounded down because $L_{\max}$ is integer-valued in these applications. For the first given problem, the requirement $x_{0.05} = 42$ implies that the corresponding performance-to-lower-bound statistic $\text{VF}_{\vartheta_1} - \text{LB}_{\vartheta^*}$ must not exceed two mean operation processing times. For the second given problem, the requirement $x_{0.05} = 17$ implies that $\text{VF}_{\vartheta_2} - \text{LB}_{\vartheta^*}$ must not exceed 83% of a mean operation processing time. Finally for the third given problem, the requirement $x_{0.05} = 0$ implies that ϑ_3 must be a VF-best allocation. These examples illustrate the potential complications in implementing an effective probabilistic stopping rule for a worker-allocation search heuristic.

Unresolved Problems and Lessons Learned

This section highlights a number of issues faced when implementing the methodology described in the preceding sections, including some unresolved problems and the main strengths and weaknesses of our probabilistic approach.

Unresolved Problems

Because of widespread availability of numerous distribution-fitting software packages, the following questions naturally arise:

- When a user-specified distribution is fitted to a given data set, do these packages deliver similar results?
- When the user requests automatic selection of the distribution yielding the "best fit" to a given data set, do these packages deliver similar results?

We examined the following packages in detail: (a) the Arena Input Analyzer (Kelton et al. 2010); (b) Stat::Fit (Geer Mountain Software Corp. 2001); and (c) ExpertFit (Law 2011). These software packages differ in the following important respects:

- The methods used to set the location (i.e., the upper and lower endpoints) and the number of bins for the histogram depicting the data set to be fitted;
- The methods used to estimate the parameters of each candidate distribution that will be fitted to the data set;
- The methods used to perform the associated goodness-of-fit tests; and
- The algorithms used for automatic selection of the candidate distribution yielding the "best fit" to the given data set.

We found that these differences can lead to substantially different results in practice.

The number of bins in the histogram can be manually specified, and this eliminates one source of the discrepancies between the fits obtained by different software packages when they are applied to the same data set. However, the Arena Input Analyzer does not allow the user to fix the location of the histogram by specifying the lower and upper endpoints of the histogram—that is, the lower endpoint of the first bin and the upper endpoint of the last bin. In many cases the Arena Input Analyzer simply takes the smallest and largest observations in the data set as the lower and upper endpoints of the histogram, respectively; and then it takes the lower endpoint of the histogram as the "known" value of the shift parameter (lower limit) for any shifted distribution that is fitted to the data set. This approach can yield fitted distributions that differ substantially from those delivered by Stat::Fit and ExpertFit. Moreover, this approach can lead to noticeable differences in the degrees of freedom and the resulting p-value of the chi-squared goodness-of-fit test for the final fitted distribution as reported by the Arena Input Analyzer in comparison with the results reported by ExpertFit and Stat::Fit.

We have also observed significant discrepancies between Stat::Fit and ExpertFit in their automatic choice of the best-fitting distribution. This is because their algorithms for ranking the quality of the fits obtained with each candidate distribution are proprietary; thus the user has no basis for judging the reliability of either ranking in a specific application. For example, when we used Stat::Fit to model the data set having the form of Eq. (13.4) for the designated DRC job shop scheduling problem corresponding to a symmetric job shop with 80 % staffing and a due-date range of 2200, the Johnson-SU distribution was recommended as the "best fit." By contrast, ExpertFit declared that all the fits to this data set obtained with a continuous distribution were "bad"; presumably this is because all the observations in the data set are integer-valued. In our judgment, the adequacy of the fitted Johnson-SU distribution is evident from visual inspection of the histogram overlaid with the fitted Johnson-SU p.d.f. and of the empirical c.d.f. overlaid with the fitted Johnson-SU c.d.f. in Fig. 13.14. In our experience, finding a distribution that provides a good fit to a data set is more complex than simply picking the candidate distribution ranked first by any automated fitting procedure—the recommendations provided by such a procedure must be supplemented by careful visual inspection of the fitted distribution as well as the analyst's "feel" for the data and for the ways in which the fitted distribution will ultimately be used.

Strengths and Weaknesses of the Approach

At the start of the fitting process, a contingency table approach was used to determine whether for each combination of job shop type and staffing level, a single probability distribution could be used for all levels of the due-date range. In other words for each combination of job shop type and staffing level, we sought to test the hypothesis that the eight data sets having the form of Eq. (13.4) (corresponding to the eight levels of the due-date range) were all sampled from the same underlying population. The results indicated that the hypothesis is definitely false—in general for each combination of job shop type and staffing level, the data sets of the form of Eq. (13.4) corresponding to different levels of the due-date range were sampled from different populations. This analysis is confirmed by the results of fitting a distribution to each data set separately; see Tables 13.2 and 13.3. Consider, for example, the symmetric job shop with 70 % staffing. A shifted Gamma distribution provides the best fit to each of the eight data sets; however, the distribution's estimated parameters change as the due-date range increases. For a due-date range of 200, the estimates for the shape, scale and shift parameters are $\alpha = 3.00$, $\beta = 11.37$ and $a = 0.89$, respectively; on the other hand for a due-date range of 3000, the corresponding parameter estimates are $\alpha = 10.92$, $\beta = 5.53$, and $a = 0.0$, respectively.

In the context of evaluating a specific worker allocation ϑ for a given DRC job shop scheduling problem, the implication of these results is that a change in the due-date range will require the following:

(a) Generating simulation replications of the given DRC job shop scheduling problem with the new due-date range but with the old values of the job shop type and the staffing level;
(b) Computing the associated data set having the form of Eq. (13.4);
(c) Fitting a new distribution to the new data set obtained in step (b); and
(d) Computing the new level of confidence that ϑ is a VF-best allocation for the given problem.

This complication can greatly increase the work required to use our approach in large-scale applications. On the other hand, so long as the given DRC job shop scheduling problem remains unchanged (i.e., the job shop type, staffing level, and due-date range remain the same), our approach can be used for rapid evaluation of our confidence that different candidate allocations are VF-best for the given DRC job shop scheduling problem.

There are six different probability distributions that were used to characterize the data sets fit using the methodology outlined previously. However, there is a fair amount of subjectivity in deciding which distribution provides the "best fit" to a data set: this decision requires interpreting the p-value for a chi-squared goodness-of-fit test, judging how well the fitted p.d.f. tracks the histogram of the data set, and judging how well the fitted c.d.f. tracks the empirical c.d.f. In addition, when two or more fitted distributions have similar chi-squared goodness-of-fit p-values, and the two aforementioned graphs indicate fits of similar quality for different probability distributions, there are no definitive tie-breaking criteria. In a number of cases documented in this chapter, the characterization provided by a shifted Gamma distribution was

arguably as good as the characterization provided by a shifted Weibull distribution or a generalized Beta distribution. The point of this observation is that there are cases in which two probability distributions can be said to provide a "best fit" to a given data set; and the distribution that is finally selected may be the one that is most familiar to the user or most easily implemented in practice.

Finally, the probability distributions chosen are continuous even though the data sets are integer-valued. We chose to use continuous distributions for the following reasons:

- None of the available parametric discrete distributions provided an adequate fit to any of the data sets, and there was no theoretical basis for formulating new parametric discrete distributions that might be used to obtain acceptable fits to data sets having the form (13.4).
- Although each data set of the form (13.4) is integer-valued, this is an artifact of the experimental design. To achieve tractability with our model of the DRC job shop scheduling problem, it was necessary to assume that processing times were integer-valued; in practice, however, operation processing times are usually continuous.
- In all the situations for which a continuous distribution was used to approximate all or part of a data set of the form (13.4), we judged that the continuous distribution provided an adequate fit to the relevant part of the data set based on visual inspection of the fitted p.d.f. and c.d.f. when they were superimposed on the histogram and the empirical c.d.f., respectively.

Conclusions

The use of heuristics in the solution approach to an NP-Hard problem introduces uncertainty into the solution. The articles by Lobo et al. (2013a, b) address the problem of finding an allocation of workers to machine groups in a DRC job shop that enable a schedule that minimizes $L_{\max}$. In their approach, both the use of HSP to identify promising allocations and the use of the Virtual Factory to generate schedules introduce uncertainty into the solution. The first article finds a lower bound on $L_{\max}$ given an allocation ϑ, and then identifies how to find the allocation ϑ^* yielding the smallest such lower bound. The second article establishes optimality criteria, and in the case that they are not satisfied, presents HSP, a heuristic that seeks an allocation enabling the Virtual Factory to generate a schedule with an $L_{\max}$ value smaller than $\mathrm{VF}_{\vartheta^*}$.

In this article, we use simulation replications of a given DRC job shop scheduling problem to estimate the distribution of the difference $\mathrm{VF}_{\vartheta^{\mathrm{VFB}}} - \mathrm{LB}_{\vartheta^*}$. This distribution can then be used to assess the quality of a specific (but arbitrary) allocation for a given problem: the difference $\mathrm{VF}_{\vartheta^{\mathrm{HSP}}} - \mathrm{LB}_{\vartheta^*}$ for a given problem is referred to the estimated distribution so that the likelihood that the specific allocation is in fact a VF-best allocation can be assessed. We present theory that addresses estimation using continuous, mixed, and discrete distributions, and we demonstrate this theory on 64 different data sets. In addition, we present a number of examples that illustrate the use of the fitted distribution, and we discuss an application of the distribution

as a stopping rule for a heuristic search strategy. Finally, we summarize the lessons learned in this work, detailing unresolved issues and the strengths and weaknesses of the approach.

One area of further research is the use of the performance to the lower bound metric as a means of identifying the difficulty of a designated DRC job shop scheduling problem. Another area of interest arises from the probabilistic analysis: the method presented makes use of enumeration to generate the data set of PLB(ϑ^{VFB}) values that is then characterized using a theoretical probability distribution. However, this approach is not viable when the DRC job shop scheduling problems are much larger than those considered in this article. Because the size of the allocation search space grows exponentially with an increase in either the number of machine groups or the number of machines, it is impractical to obtain the necessary data set of PLB(ϑ^{VFB}) values. A revised method that is computationally tractable in problems of realistic size and complexity is needed, and one such method is presented in Lobo et al. (2013c).

Appendix

Plots of Empirical and Fitted Distributions of **PLB**(ϑ^{VFB}) *for All DRC Job Shop Scheduling Problems*

The following figures correspond to the distribution fitting for the experimental design. For each designated DRC job shop scheduling problem, $Q = 500$ simulation replications were generated. The data set of PLB(ϑ^{VFB}) values was constructed as described previously. More than 10 % of the PLB(ϑ^{VFB}) values were equal to zero for each of the following cases: (a) the symmetric job shop with 90 % staffing and a due-date range of 200 through 2600; (b) the asymmetric job shop with 70 % staffing; (c) the asymmetric job shop with 80 % staffing and a due-date range of 2600 and 3000; and (d) the asymmetric job shop with 90 % staffing and a due-date range of 2600 and 3000. In these cases, a conditional p.d.f. was fitted to the data set composed of the nonzero PLB(ϑ^{VFB}) values. Less than 50 nonzero PLB(ϑ^{VFB}) values were observed in each of the following cases: (i) the asymmetric job shop with 80 % staffing and a due-date range of 200 through 2200; and (ii) the asymmetric job shop with 90 % staffing and a due-date range of 200 through 2200. In cases (i) and (ii), we simply used the observed values in each data set to define a discrete probability distribution. The "Best Fit" for each of the 64 different job shop type, due-date range, and staffing level combinations, as determined using the methodology outlined previously, are given in Tables 13.2 and 13.3.

The p-value given on each graph is the p-value reported for the chi-square goodness-of-fit test by the Stat::Fit software (Geer Mountain Software Corp. 2001). On the set of graphs corresponding to a single designated DRC job shop scheduling problem, the number of "Datapoints" equals Q' if the data set of the form (13.4) for that designated DRC job shop scheduling problem was fitted using a continuous distribution, and equals $Q'(1 - p_0)$ if the data set of the form (13.4) for that designated problem was fitted using a mixed distribution.

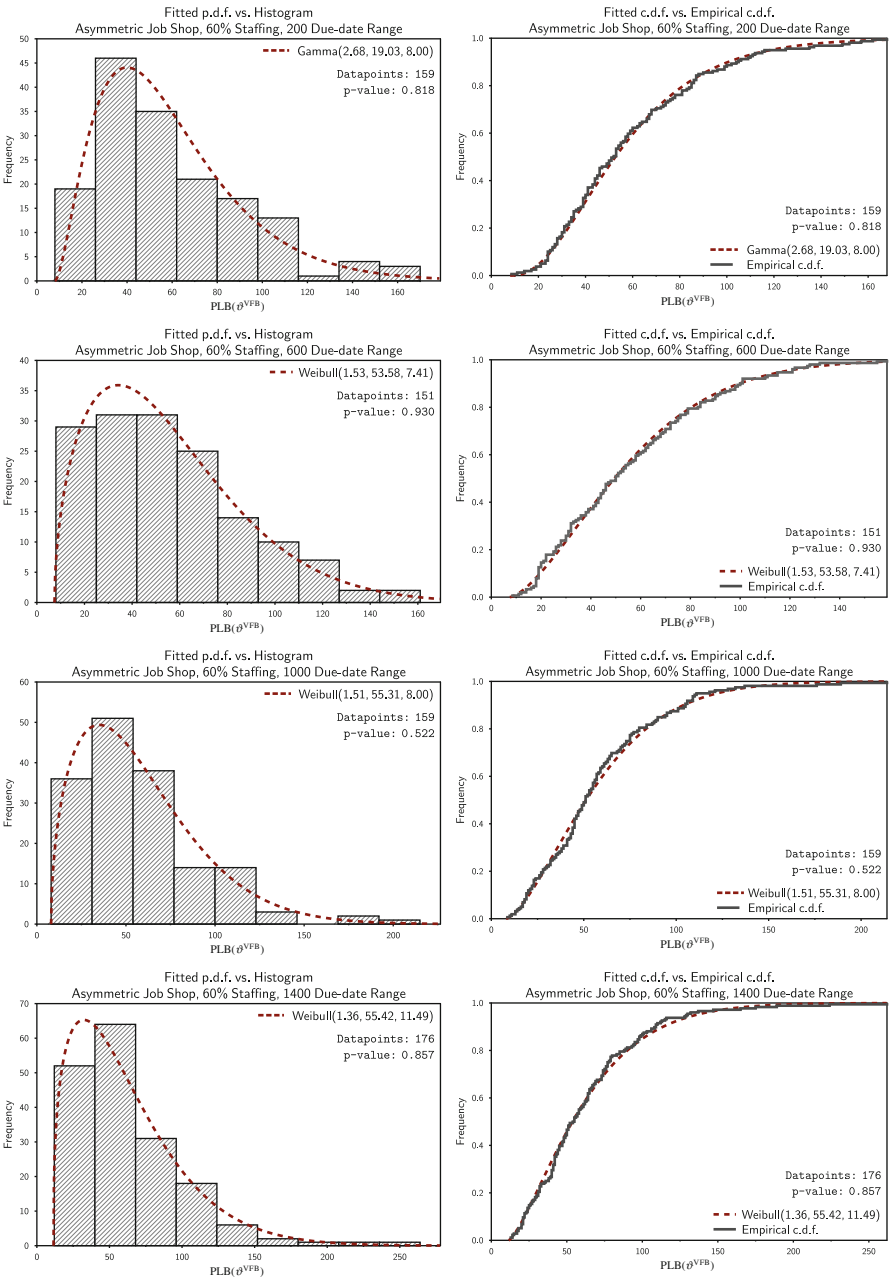


Fig. 13.3 Probability distribution fitting, asymmetric job shop, 60 % staffing

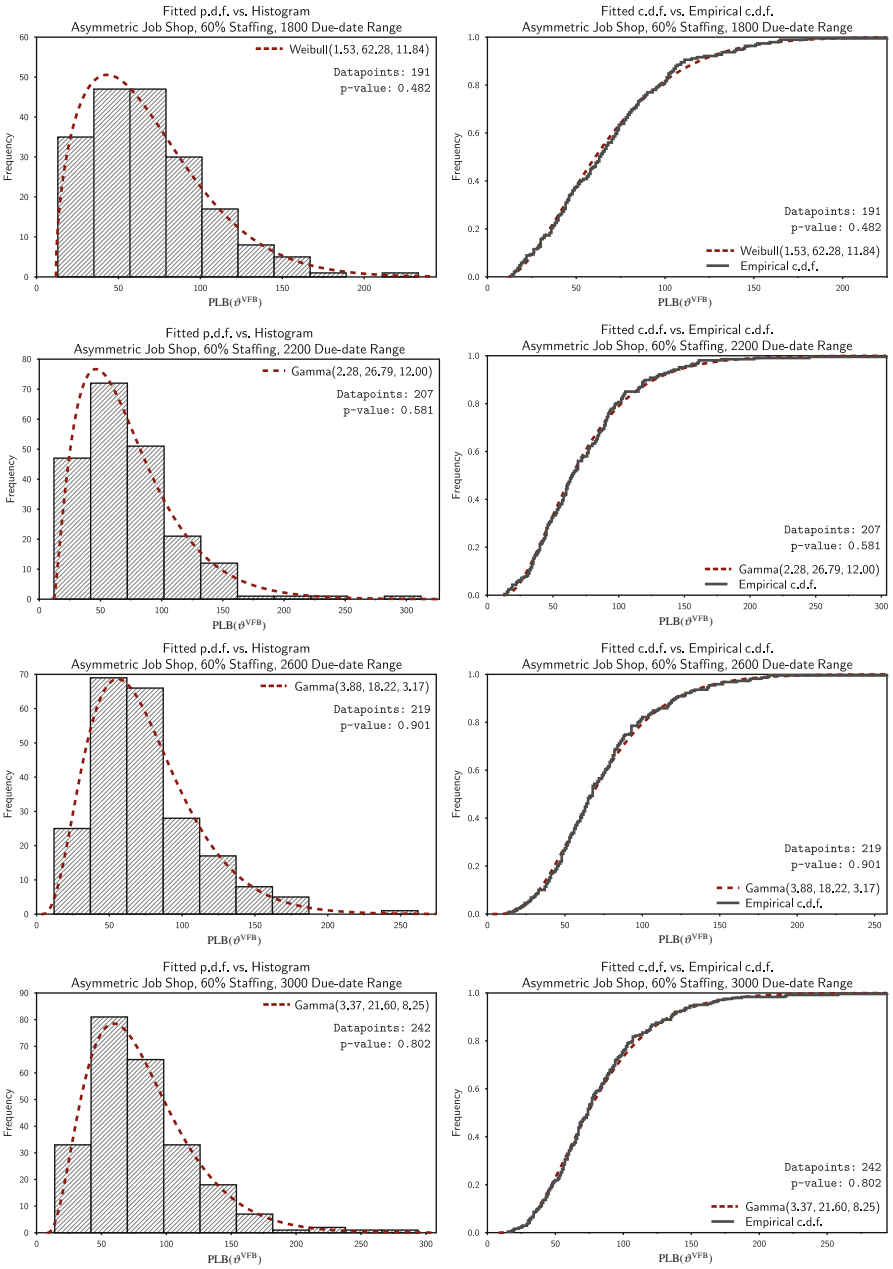


Fig. 13.4 Probability distribution fitting, asymmetric job shop, 60 % staffing, contd.

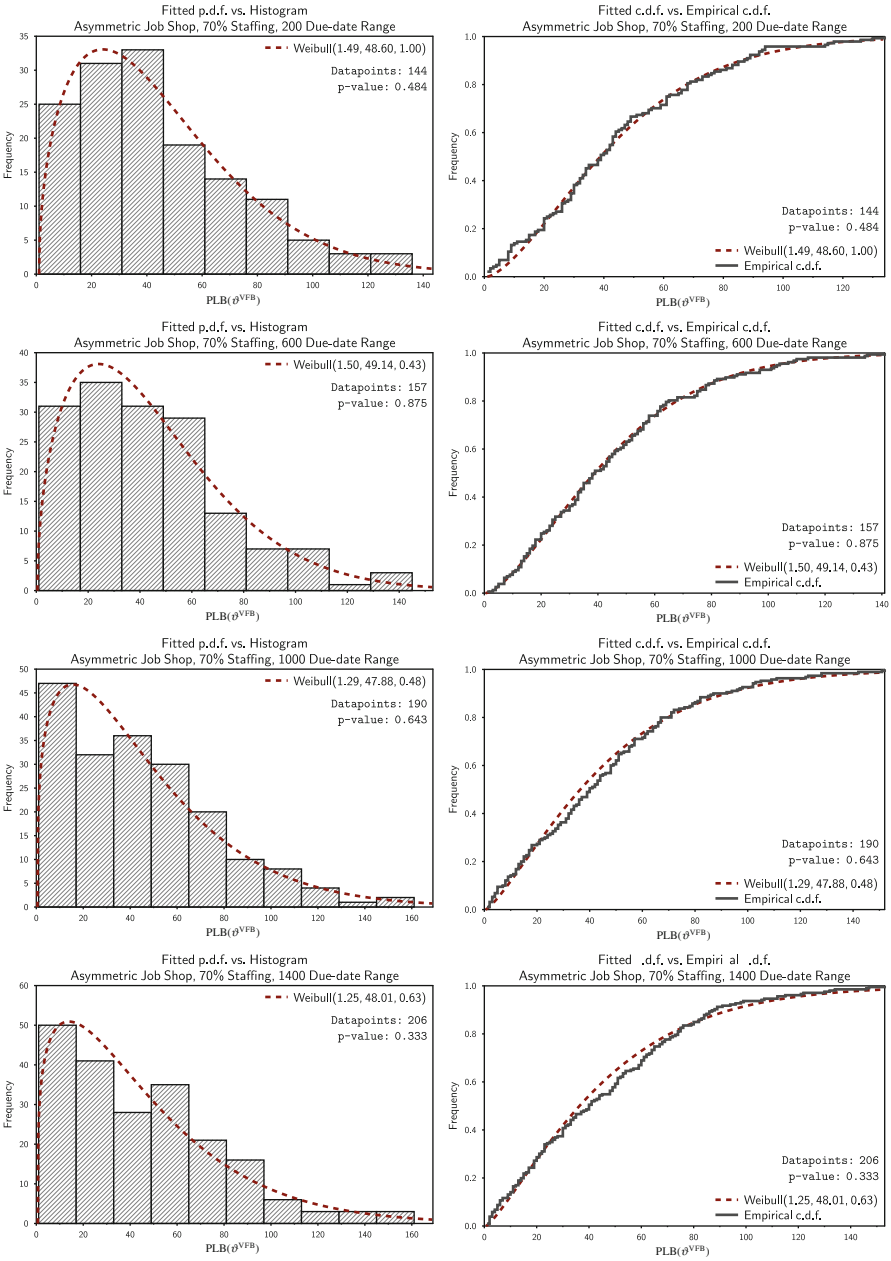


Fig. 13.5 Probability distribution fitting, asymmetric job shop, 70 % staffing

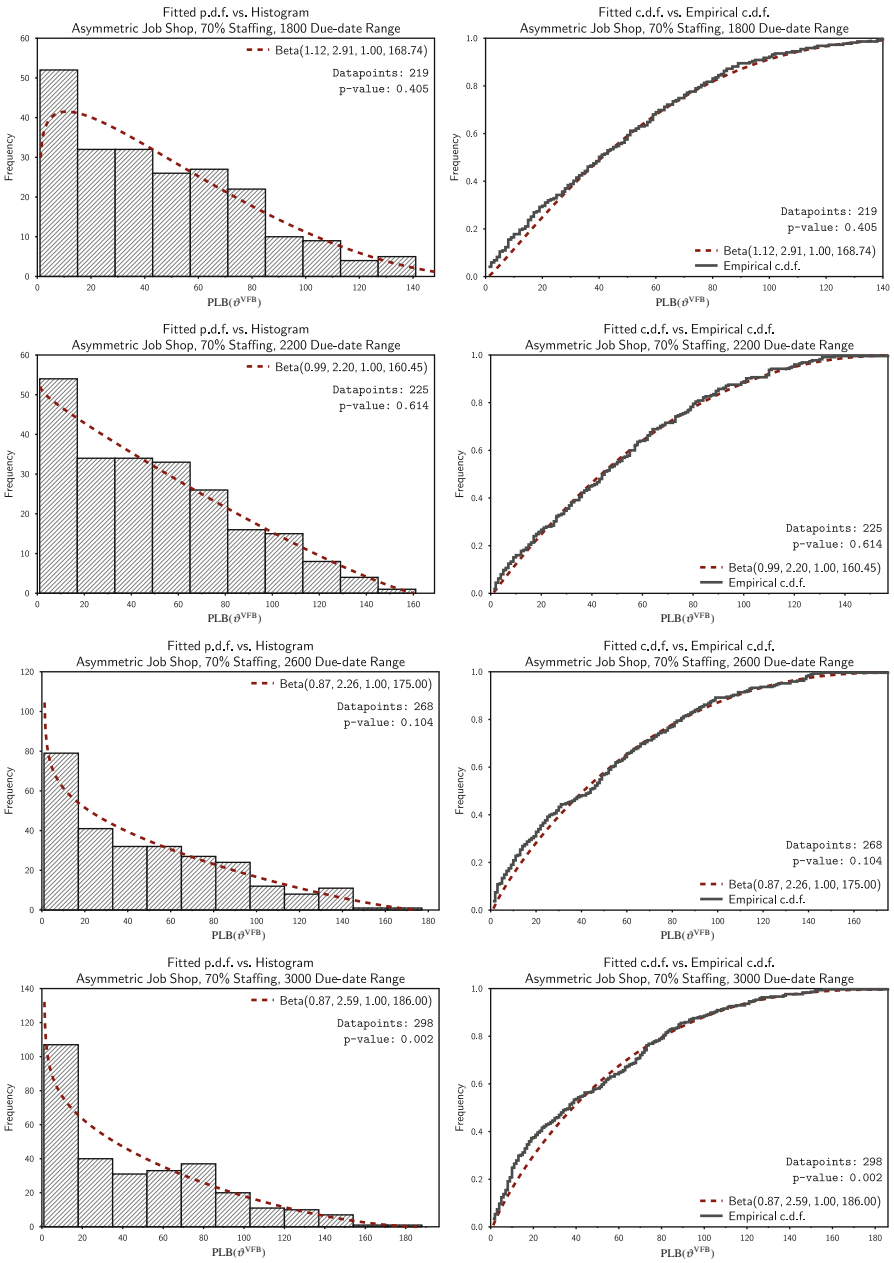


Fig. 13.6 Probability distribution fitting, asymmetric job shop, 70 % staffing, contd.

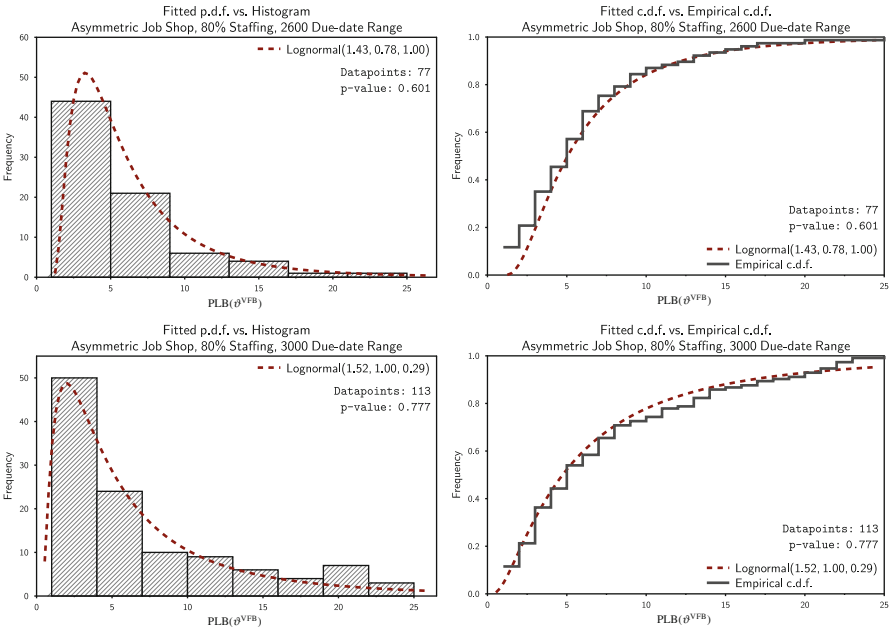


Fig. 13.7 Probability distribution fitting, asymmetric job shop, 80 % staffing

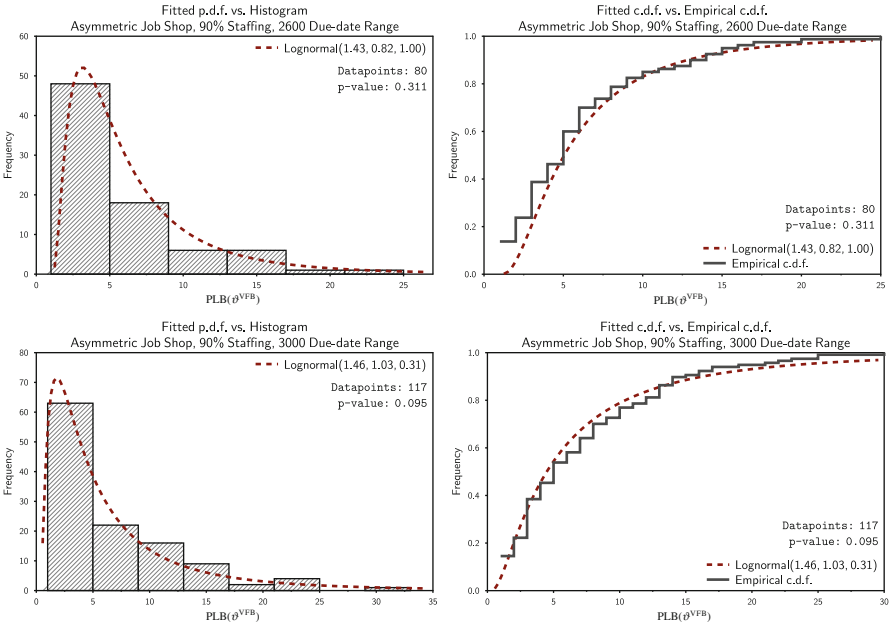


Fig. 13.8 Probability distribution fitting, asymmetric job shop, 90 % staffing

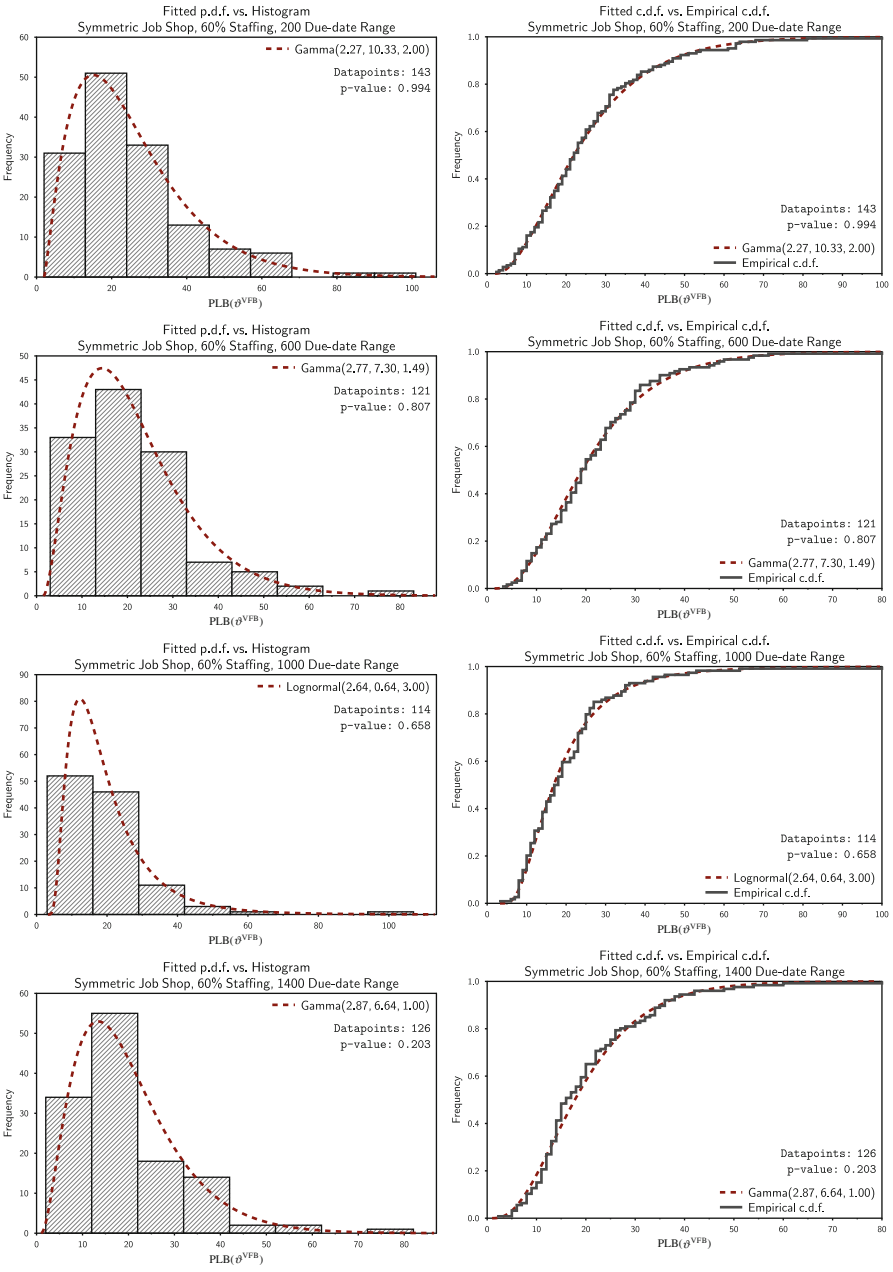


Fig. 13.9 Probability distribution fitting, symmetric job shop, 60 % staffing

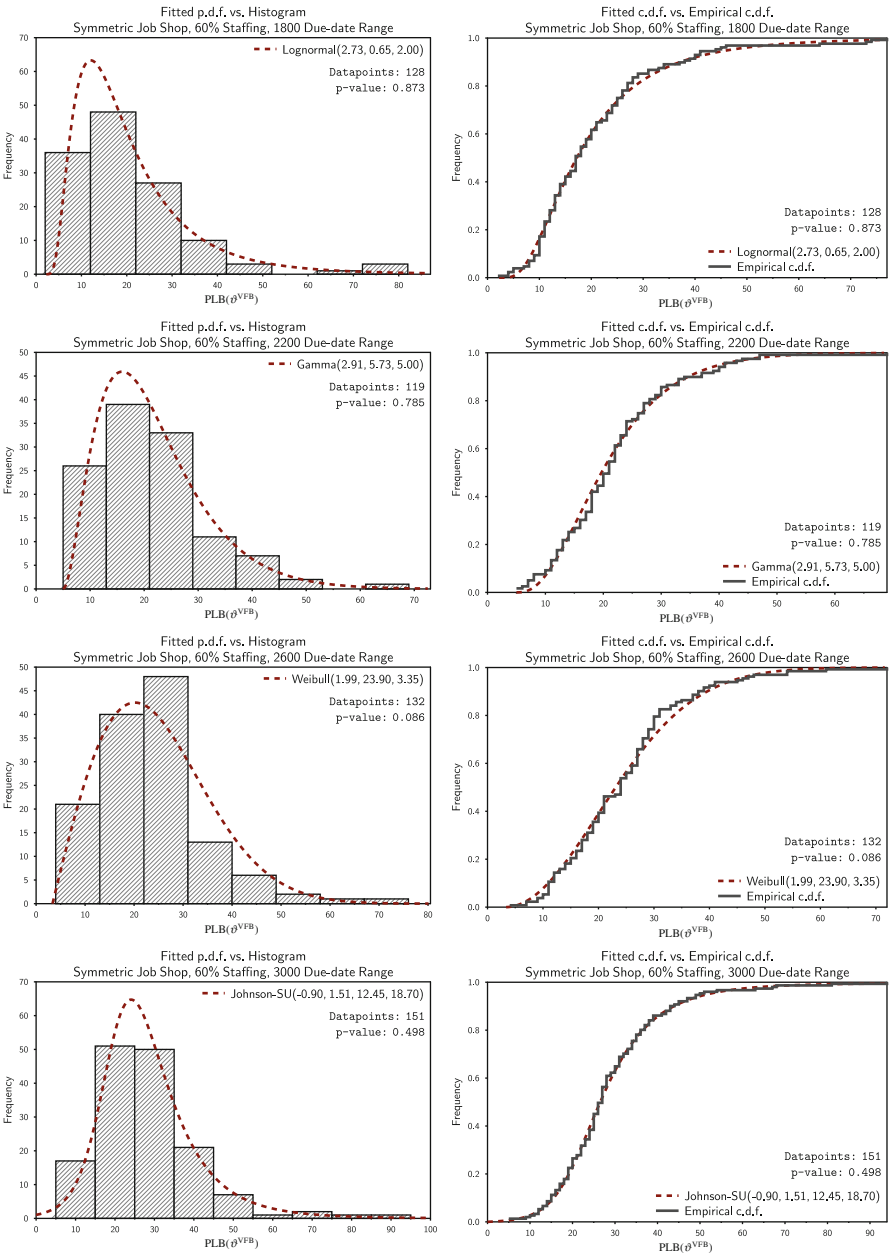


Fig. 13.10 Probability distribution fitting, symmetric job shop, 60 % staffing, contd.

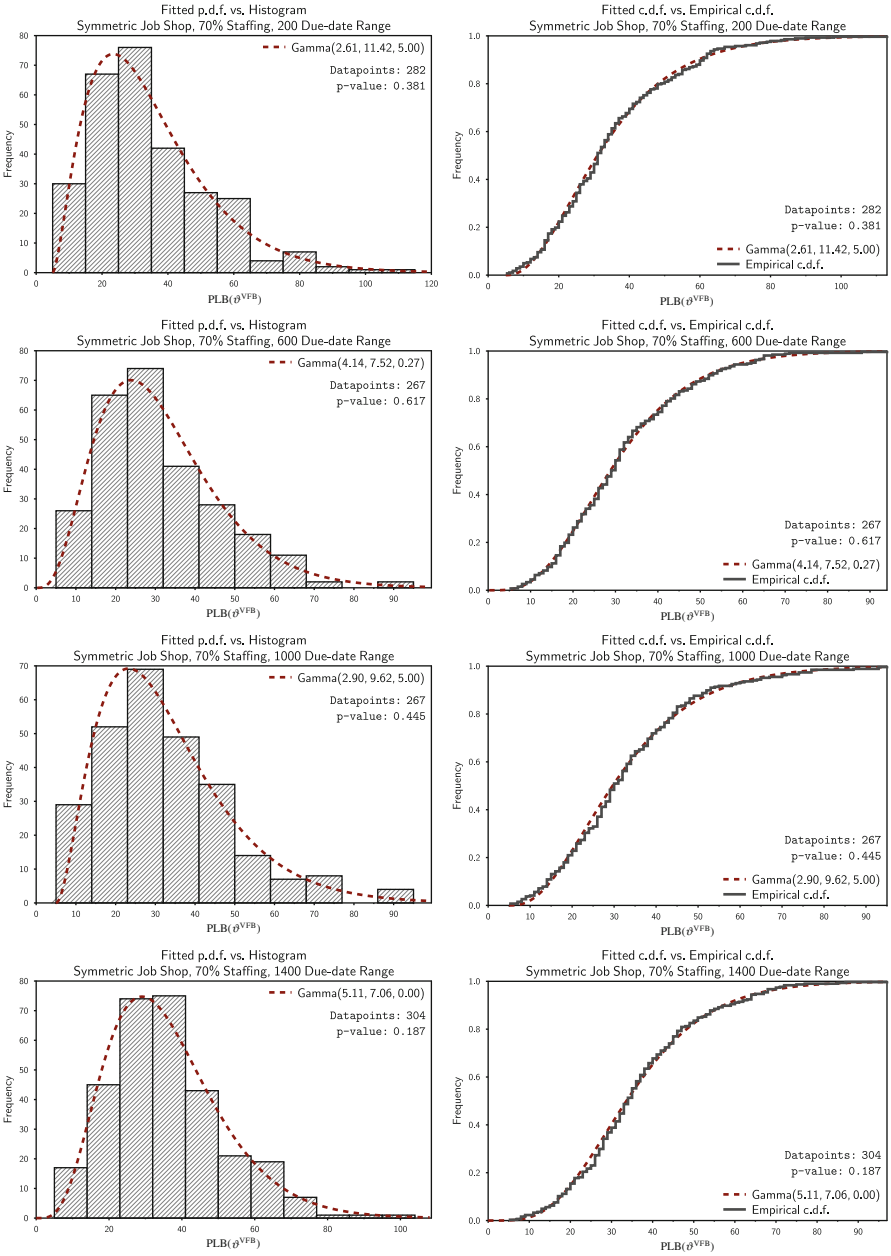


Fig. 13.11 Probability distribution fitting, symmetric job shop, 70 % staffing

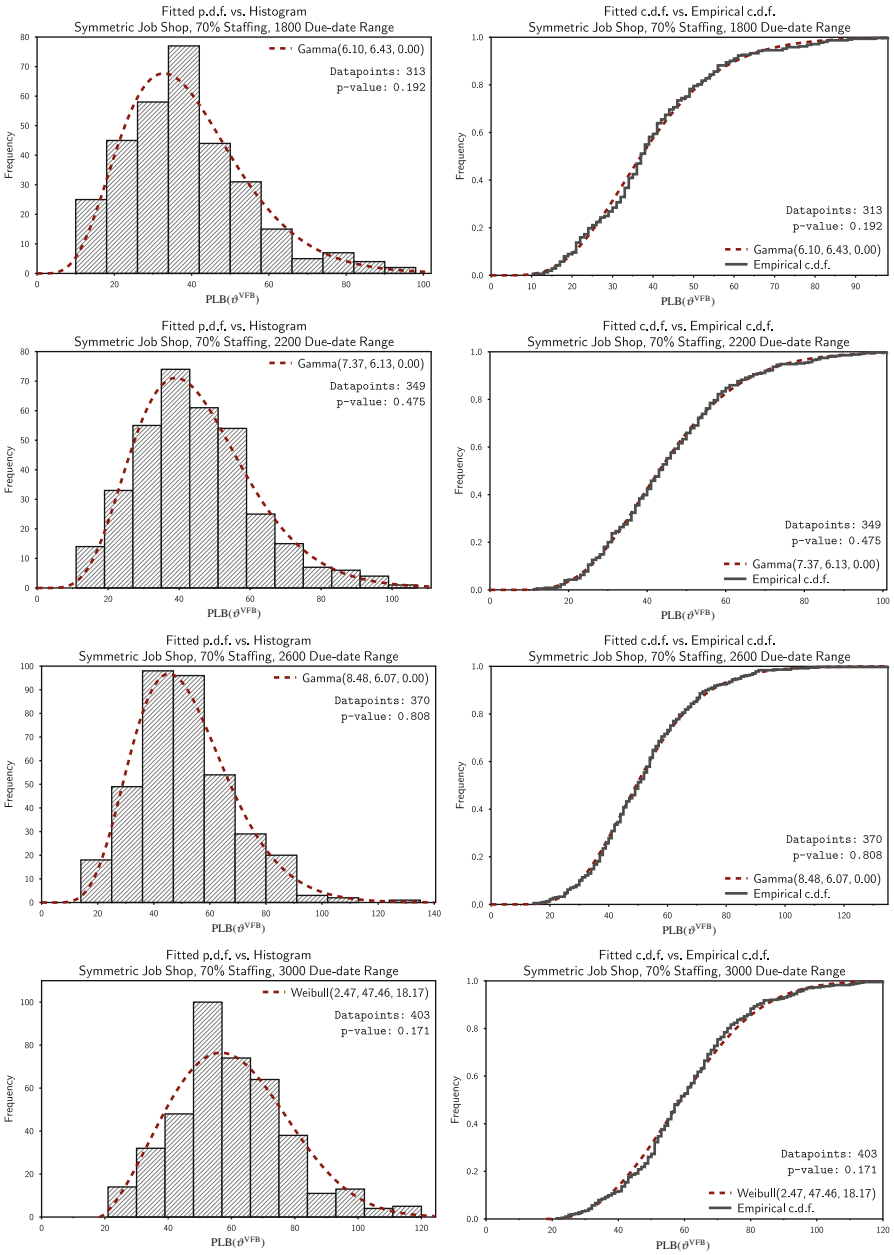


Fig. 13.12 Probability distribution fitting, symmetric job shop, 70 % staffing, contd.

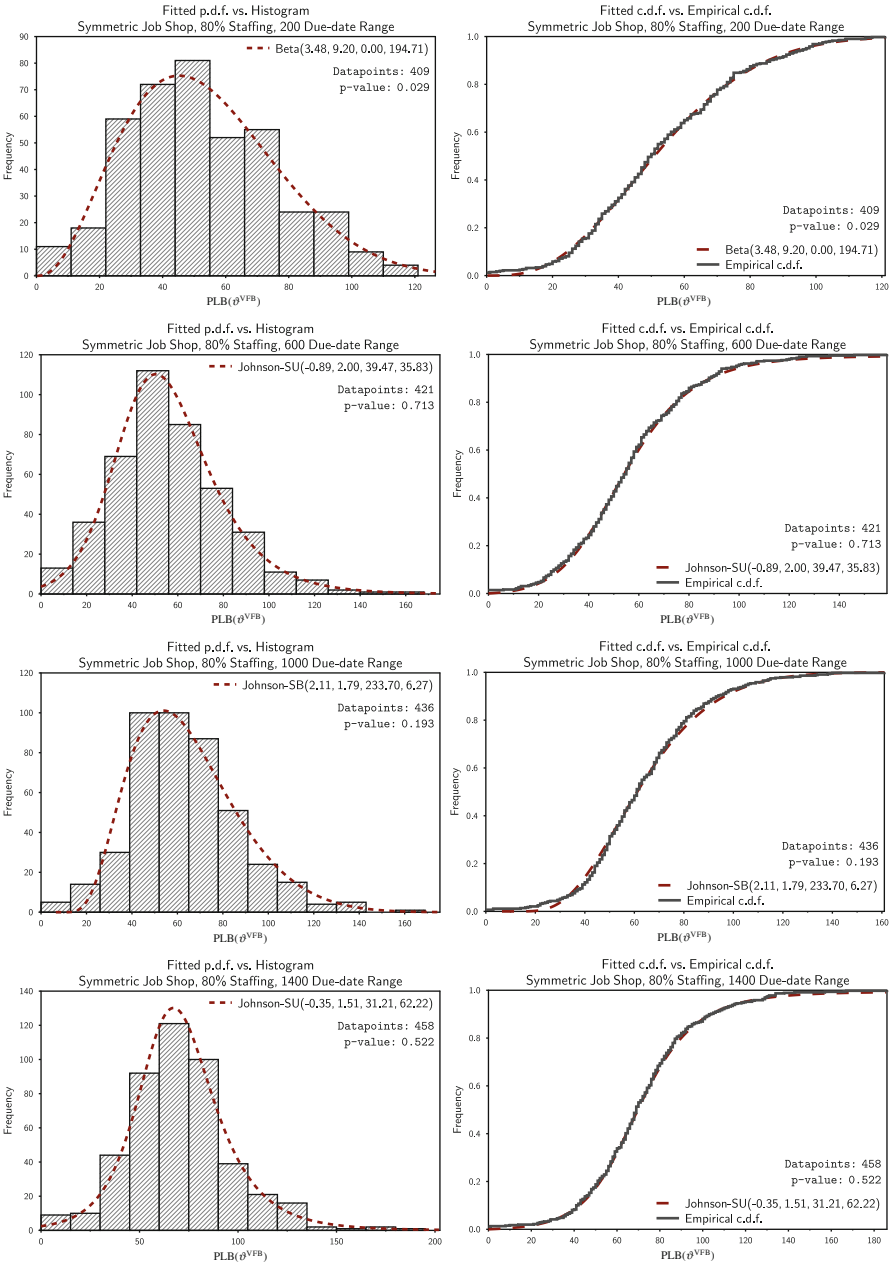


Fig. 13.13 Probability distribution fitting, symmetric job shop, 80 % staffing

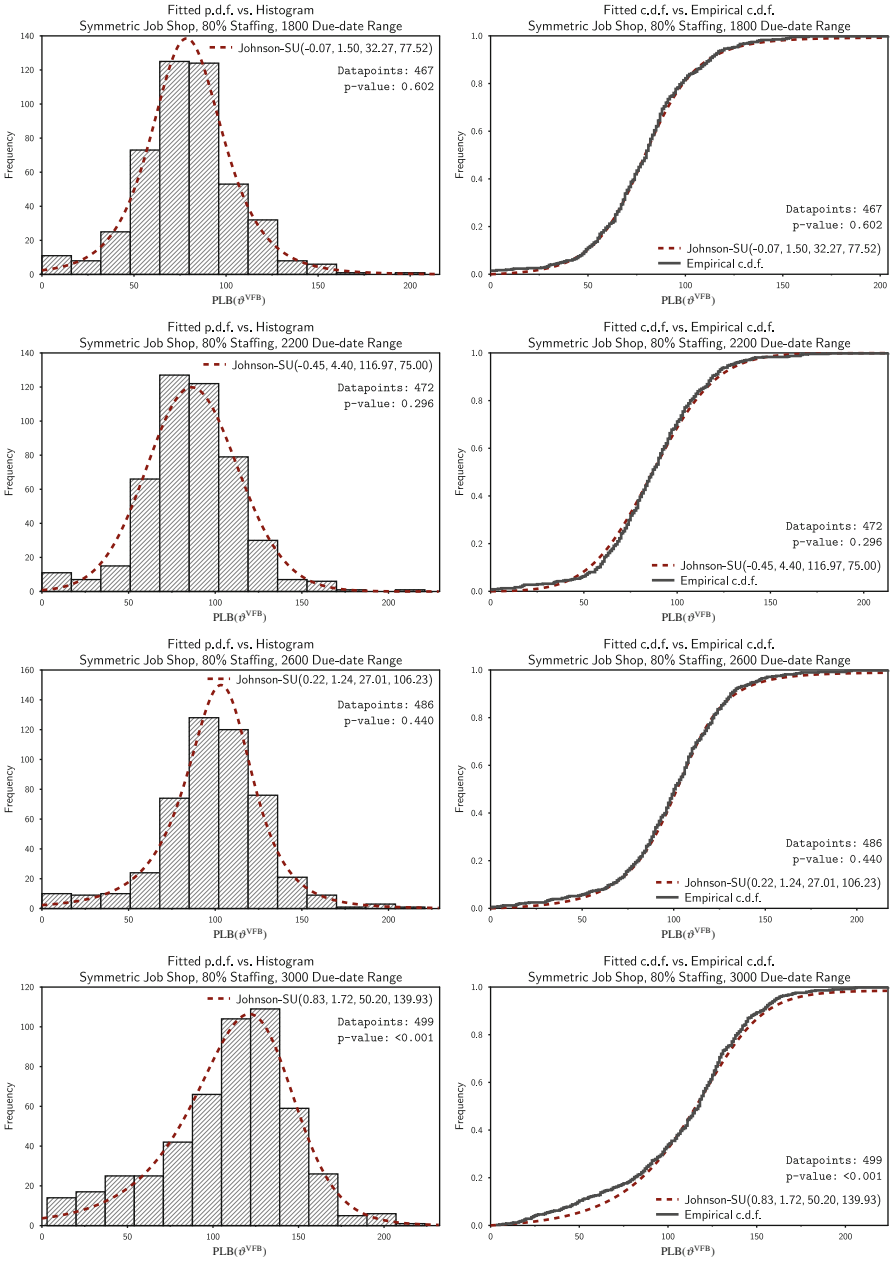


Fig. 13.14 Probability distribution fitting, symmetric job shop, 80 % staffing, contd.

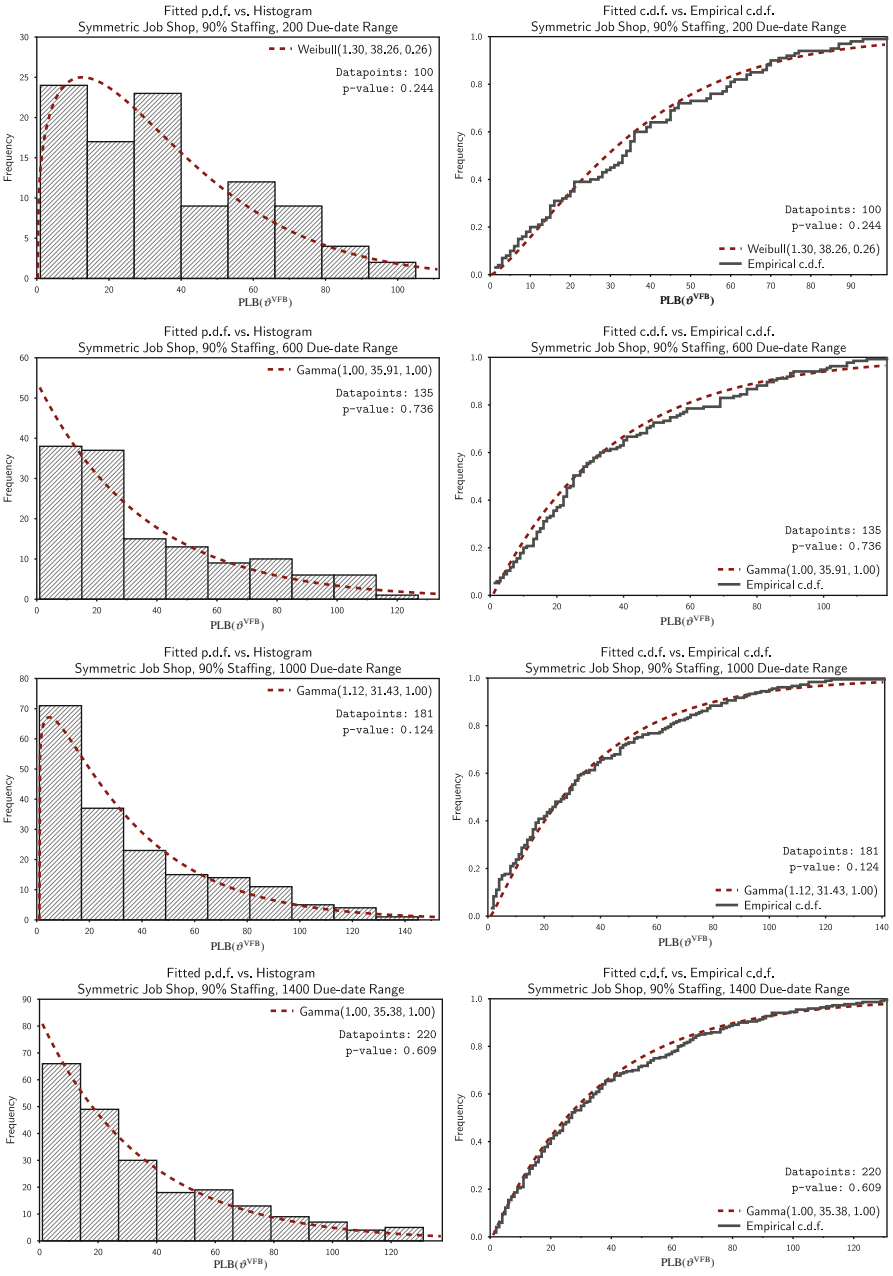


Fig. 13.15 Probability distribution fitting, symmetric job shop, 90 % staffing

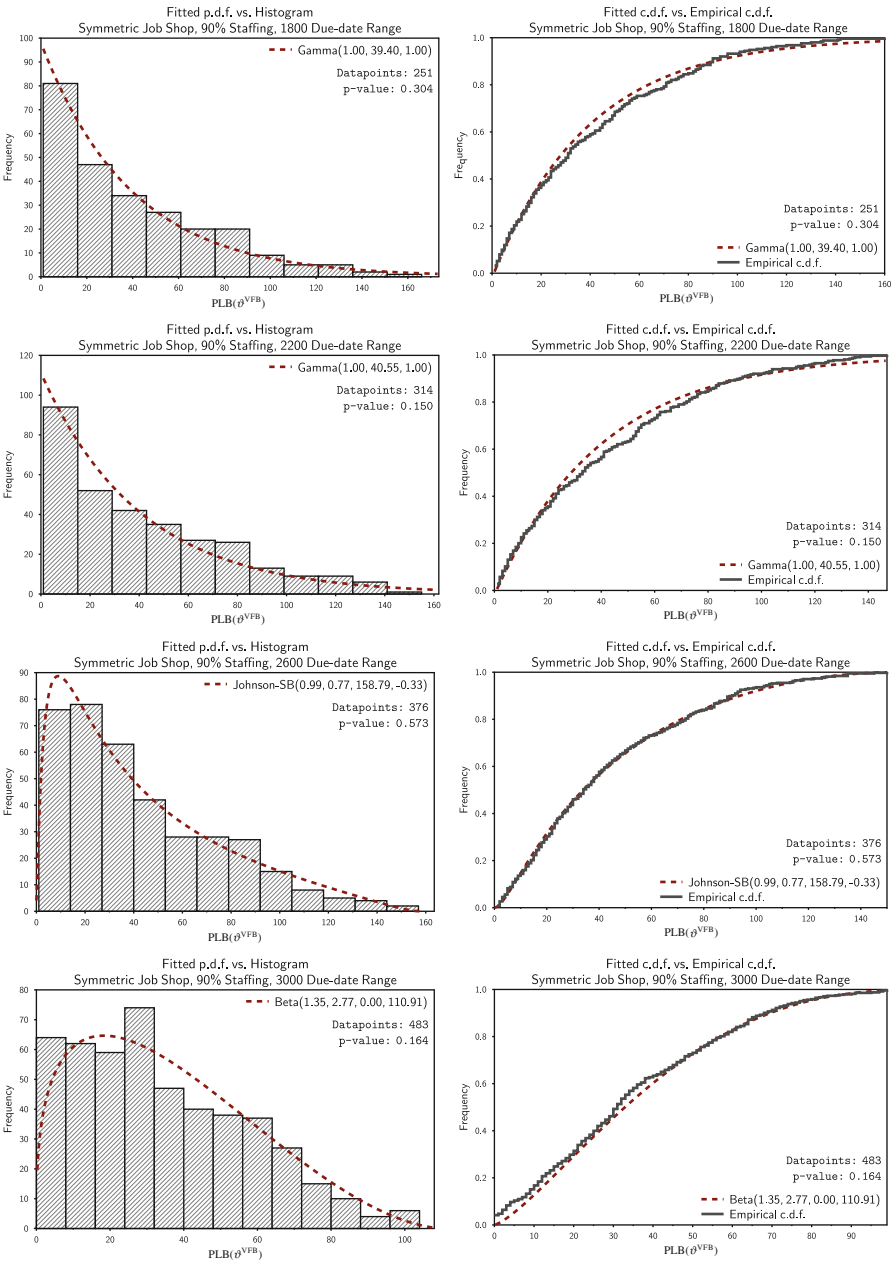


Fig. 13.16 Probability distribution fitting, symmetric job shop, 90 % staffing, contd.

References

Adams, J., Balas, E., & Zawack, D. (1988). The shifting bottleneck procedure for job shop scheduling. *Management Science, 34*(3), 391–401.

Bickel, P. J., & Doksum, K. A.. (2007). *Mathematical Statistics: Basic Ideas and Selected Topics (2nd edn.)*. Upper Saddle River, Pearson Prentice Hall.

Coles, S. (2001). *An Introduction to Statistical Modeling of Extreme Values*. Springer, London.

Derigs, U. (1985). Using confidence limits for the global optimum in combinatorial optimization. *Operations Research, 33*(5), 1024–1049.

Felan, J., Fry, T., & Philipoom, P. (1993). Labour flexibility and staffing levels in a dual-resource constrained job shop. *International Journal of Production Research, 31*(10), 2487–2506.

Gargeya, V., Deane, R. (1996). Scheduling research in multiple resource constrained job shops: A review and critique. *International Journal of Production Research*, *34*(8), 2077–2097.

Geer Mountain Software Corp. (2001). *Stat::Fit, Version 2*. South Kent, Geer Mountain Software Corp.

Golden, B. L., & Alt, F. B. (1979). Interval estimation of a global optimum for large combinatorial problems. *Naval Research Logistics Quarterly, 26*(1), 69–77.

Hodgson, T., Cormier, D., Weintraub, A., & Zozom, A. (1998). Satisfying due dates in large job shops. *Management Science, 44*(10), 1442–1446.

Hodgson, T., King, R., Thoney, K., Stanislaw, N., Weintraub, A., & Zozom, A. (2000). On satisfying due-dates in large job shops: Idle time insertion. *I*IE Transactions, *32,* 177–180.

Hodgson, T., Melendez, B., Thoney, K., & Trainor, T. (2004). The deployment scheduling analysis tool (DSAT). *Mathematical and Computer Modelling, 39*(6-8), 905–924.

Hottenstein, M., & Bowman, S. (1998). Cross-training and worker flexibility: A review of DRC system research. *The Journal of High Technology Management Research, 9*(2), 157–174.

Jaber, M., & Neumann, W. (2010). Modelling worker fatigue and recovery in dual-resource constrained systems. *Computers & Industrial Engineering, 59*(1), 75–84.

Kelton, W. D., Sadowski, R. P., & Swets, N. B.. (2010). *Simulation with Arena 4th edn*. New York, McGraw-Hill.

Kher, H. (2000). Examination of flexibility acquisition policies in dual resource constrained job shops with simultaneous worker learning and forgetting effects. *The Journal of the Operational Research Society, 51*(5), 592–601.

Kher, H., Malhotra, M., Philipoom, P., & Fry, T. (1999). Modeling simultaneous worker learning and forgetting in dual resource constrained systems. *European Journal of Operational Research, 115*(1), 158–172.

Kuhl, M., Ivy, J., Lada, E., Steiger, N., Wagner, M., & Wilson, J. (2010). Univariate input models for stochastic simulation. *Journal of Simulation, 4,* 81–97.

Law, A. M (2007). *Simulation Modeling and Analysis*. McGraw-Hill, New York, 4th edn.

Law, A. M. (2011). How the ExpertFit distribution-fitting software can make your simulation models more valid. In *Proceedings of the 2011 Winter Simulation Conference*, S. Jain, R. R. Creasy, J. Himmelspach, K. P. White, & M. Fu (eds.) *Institute of Electrical and Electronics Engineers* (pp. 63–69). New Jersey, Piscataway.

Leadbetter, M. R., Lindgren, G., & Rootzén, H. (1983). *Extremes and Related Properties of Random Sequences and Processes*. New York, Springer-Verlag.

Lenstra, J., & Rinnooy Kan, A. (1979). Computational complexity of discrete optimization problems. In *Discrete Optimization I Proceedings of the Advanced Research Institute on Discrete Optimization and Systems Applications of the Systems Science Panel of NATO and of the Discrete Optimization Symposium*, P. Hammer, E. Johnson, and B. Korte, eds., Elsevier, vol. 4 of *Annals of Discrete Mathematics*. 121–140.

Lobo, B., Hodgson, T., King, R., Thoney, K., & Wilson, J. (2013a). An effective lower bound on $L_{\max}$ in a worker-constrained job shop. *Computers & Operations Research, 40*(1), 328–343.

Lobo, B., Hodgson, T., King, R., Thoney, K., & Wilson, J. (2013b). Allocating job-shop manpower to minimize $L_{\max}$ in a job shop: Optimality criteria, search heuristics, and probabilistic quality metrics. *Computers & Operations Research*, **40**(10), 2569–2584.

Lobo, B., Wilson, J., Thoney, K., Hodgson, T., & King, R. (2013c). A practical method for evaluating worker-allocations in large-scale dual resource constrained job shops. http://people.engr.ncsu.edu/bjlobo/papers/loboetal_paper4.pdf.

Malhotra, M., Fry, T., Kher, H., & Donohue, J. (1993). The impact of learning and labor attrition on worker flexibility in dual resource constrained job shops. *Decision Sciences*, *24*(3), 641–664.

Nelson, R. (1967). Labor and machine limited production systems. *Management Science, 13*(9), 648–671.

Park, S., & Miller K (1988) Random number generators: good ones are hard to find. *Communications of the ACM, 31*(10), 1192–1201.

Pinedo, M. (2012). *Scheduling: Theory, Algorithms, and Systems, (4th edn).* Springer

Schultz, S., Hodgson, T., King, R., & Thoney, K. (2004). Minimizing $L_{\max}$ for large-scale, job-shop scheduling problems. *International Journal of Production Research, 42*(23), 4893–4907.

Scott, D. (1979). On optimal and data-based histograms. *Biometrika, 66*(3), 605–610.

Thoney, K., Hodgson, T., King, R., Taner, M., & Wilson, A. (2002). Satisfying due-dates in large multi-factory supply chains. *IIE Transactions, 34*(9), 803–811.

Treleven, M. (1989). A review of the dual resource constrained system research. *IIE Transactions, 21*(3), 279–287.

Treleven M, & Elvers D. (1985). An investigation of labor assignment rules in a dual-constrained job shop. *Journal of Operations Management, 6*(1), 51–68.

Weintraub, A., Cormier, D., Hodgson, T., King, R., Wilson, J., & Zozom, A. (1999). Scheduling with alternatives: a link between process planning and scheduling. *IIE Transactions, 31*(11), 1093–1102.

Wilson, A. D., King, R. E., & Wilson, J. R. (2004). Case study on statistically estimating minimum makespan for flow line scheduling problems. *E*uropean Journal of Operational Research, *155,* 439–454.

Zozom, A., Hodgson, T., King, R., Weintraub, A., & Cormier, D. (2003). Integrated job release and shop-floor scheduling to minimize WIP and meet due-dates. *International Journal of Production Research, 41*(1), 31–45.

Chapter 14
Mine Planning Above and Below Ground: Generating a Set of Pareto-Optimal Schedules Considering Risk and Return

Carson McFadden and Candace A. Yano

Introduction

Both long-term planning and more detailed scheduling of extraction at mines are difficult problems. Broadly speaking, mine planning usually involves determining the three-dimensional volumes of earth to be extracted, and mine scheduling involves the timing and methods of extraction at a finer level of detail. Over the past five decades, researchers have developed increasingly more sophisticated techniques to aid in decision-making, but thus far, researchers have not developed techniques to support decision-making when both above- and underground mining are involved, and when uncertainty exists about various factors such as the grade (density) of ore, ore prices, construction costs, etc.

The goal of this chapter is to develop a procedure to identify Pareto-optimal solutions with two goals: expected net present value (NPV) and a risk measure such as standard deviation, value-at-risk, or probability of meeting a profit target. Such a procedure will allow the decision-maker to be better-informed about the tradeoff between expected profit and risk considerations without the need to search over the weights or penalties that are typically used in multiobjective models involving a tradeoff between risk and return.

We address this problem in the context of a mine planning problem in which the firm can extract material either above- or underground, or both. We describe our problem setting in more detail in the next section. Before doing so, we first provide a review of relevant literature.

C. A. Yano (✉)
Haas School of Business and IEOR Department,
The University of California - Berkeley, Berkeley, California, 94720, USA
e-mail: yano@ieor.berkeley.edu

C. McFadden
Watarah Advisors, Toronto, Ontario, M5C 2V6, Canada
e-mail: carson.mcfadden@gmail.com

P. S. Pulat et al. (eds.), *Essays in Production, Project Planning and Scheduling,*
International Series in Operations Research & Management Science 200,
DOI 10.1007/978-1-4614-9056-2_14,

Literature Review

Much research has been done on mine planning and scheduling. The vast majority of this research has been based on the assumptions that both metal or mineral grades or density as well as prices are deterministic. Furthermore, among the literature that takes a deterministic approach, a substantial majority focuses exclusively on either above-ground (surface) or underground mining; very few papers consider both. We refer the reader to Osanloo et al. (2007) and Newman et al. (2010) for recent surveys of this literature.

A limited amount of research has been done that considers excavation both above and below ground, and to the best of our knowledge, all of this research is based on deterministic models. For examples of articles on this topic, see Stacey and Terbrugge (2000); Visser and Ding (2007); Epstein et al. (2010), and Newman, Yano and Rubio (2013).

It is well known that, *a priori*, there is uncertainty in both the grade of ore in a given location and the price that can be obtained from selling it at some future time when it is extracted; other factors such as capital, excavation or processing costs are also uncertain. Golamnejad et al. (2006) provide a nice overview of the sources of uncertainty. Two common methods have been used for capturing grade uncertainty in the mining literature: kriging, a geostatistical method (see Chiles and Delfiner 2012) which often utilizes sequential Gaussian simulation (Deutch and Journel 1998), and conditional simulation (Monte Carlo simulation to generate images of *in situ* orebody grades with three-dimension spatial correlation; cf. Dimitrakoupolos 1998).

A great majority of the research on mine planning under uncertainty focuses on the uncertainty of the ore grade, or equivalently, the ore yield, from extraction of specified physically-defined volumes of earth. Much of this research is described in the survey articles by Osanloo et al. (2007) and Newman et al. (2010). To the best of our knowledge, all of this research addresses surface mining. Typically, the problem is posed as one of when to mine each *production block* (a three-dimensional volume of earth) to maximize NPV subject to constraints on such things as capacity for processing the extracted material, satisfying production targets, precedence constraints, etc., with uncertainty incorporated in various ways. Here, we summarize the various strategies that have been employed.

Many researchers have used a technique which is called "conditional simulation" in the mining literature; it is essentially a scenario-based approach in which potential scenarios are generated via simulation and a solution is somehow constructed based on solutions for these scenarios. Early methods (see, for example, Ravenscroft 1992, Denby and Schofield 1995 and most of the references in Dimitrakoupolos 2011) have developed either *ad hoc* methods with embedded optimization or metaheuristics to identify heuristic solutions. As an example, some methods involve generating a number of scenarios, finding the optimal or an approximate solution for each (now deterministic) scenario, then using the set of solutions derived from solving these problems as the basis for choosing or constructing a solution. As one example of such an approach, Dimitrakopoulos et al. (2007) generate a number of conditionally simulated orebodies, and for each such orebody, solve the block scheduling problem

as if the orebody characteristics were deterministic. They then evaluate some of the resulting solutions with respect to the distribution of return on investment plus other operational considerations.

It is only fairly recently that researchers have proposed approaches based on formal optimization modeling and solution techniques. Golamnejad et al. (2006) propose a chance-constrained approach for block scheduling with the original intent of imposing a chance constraint in each period. Each chance constraint specifies that the average grade of blocks mined in that period should exceed a specified threshold with a specified probability. Their final formulation, however, has an objective like that of the capital asset pricing model (CAPM), i.e., expected profit less a constant times the standard deviation of aggregate profit. Under the premise that many blocks will be mined in each period, the authors appeal to the central limit theorem and utilize normality assumptions when applying the CAPM model. (In our model, the extraction of a stratum takes one or more periods and only one stratum is mined at a time, so multiple strata are not completed in the same period.) They propose, but do not test genetic algorithms as a solution approach. Kumral (2010) seeks to address a similar problem, but eventually solves a version of the problem in which two types of penalties are imposed: penalties for deviations of the profit from its expected value, and penalties for deviations of actual capacity requirements from prespecified targets for both mining (extraction) and processing. By varying the values of the penalties, Kumral is able to generate a spectrum of solutions that vary with respect to the mean and standard deviation of net present value and deviations from capacity limits.

Boland et al. (2008) develop a sophisticated multiperiod stochastic binary optimization model with the goal of maximizing expected net present value. Among other features, the model handles endogenous uncertainty, i.e., uncertainty that is a function of the decisions thus far. For example, once some blocks are extracted, the ore grade of exposed blocks can be determined with much more accuracy than when they were covered. The authors allow for decisions based on two types of time lags: (i) decisions that can be made with no implementation time lag, such as deciding which material to send to the processing plant; and (ii) decisions that can be implemented with a positive time lag, such as which blocks to extract (which cannot be changed instantaneously). To keep the size of the problem manageable, they aggregate blocks and define the decisions in terms of these aggregated blocks. They use a scenario-based approach but include nonanticipativity constraints to ensure that the same decision applies to all relevant scenarios if it would not have been possible to distinguish between scenarios at the point when the applicable decision would be made. They characterize properties of the optimal solution, which then allows them to reformulate the problem and devise efficient algorithms to find near-optimal solutions.

De Lara et al. (2013) pose the block sequencing problem as a dynamic program and offer conceptual frameworks for including on-line (updated) information, adaptive strategies that can utilize this information and decision criteria that could be applied in the stochastic version of the problem. In addition to expected net present value, with or without a chance constraint on profit, they suggest two different maximin criteria. This recent working paper does not report on implementation of any of the proposed approaches.

Several authors (e.g., Lemelin et al. 2006, Abdel-Sabour and Poulin 2010, and Martinez 2006) have developed real-options-based approaches for estimating the value of a mine. Generally speaking, these methods involve allowing contingent decisions regarding such things as mining mode, temporary suspension, or abandonment in response to observations of factors that were uncertain at the outset but become known as time progresses. These methods typically are based on the assumption that the material under consideration for extraction and the sequence in which it is to be extracted are specified in advance, so the result is an expected profit or NPV of profit for a specific initial plan, but allowing for short-term changes in execution. The approaches do not seek to determine the best areas to extract or the associated optimal plan or schedule.

We highlight the fact that virtually all of the articles mentioned above that consider uncertainty deal with surface mining, and in particular block sequencing. In a typical formulation for optimizing block sequencing, many blocks can be extracted within the duration of a basic time period in a typical optimization formulation. Thus, even when ore grades are correlated in blocks that are near each other, there still may be some pooling of risk across multiple blocks because the grades are not perfectly correlated. In the mine planning and scheduling problem that we address, each stratum may take several time periods to extract because a stratum represents a much larger volume of earth. Thus, there is no opportunity for the pooling of risk within a time period. Furthermore, we consider both surface mining and underground mining, a context with a high degree of uncertainty because the cost of underground infrastructure is very significant but must be incurred before any underground extraction takes place, while the ore grade (or yield) deep underground may be much more uncertain than in the areas closer to the surface. Thus, there is much more inherent uncertainty in our problem and fewer ways to mitigate it.

The remainder of this chapter is organized as follows. In Sect. 2, we provide a description of our problem. In Sect. 3, we describe our solution framework and methodology, and we present a numerical example that illustrates the approach. We conclude the chapter in Sect. 4 with a discussion of ways in which our approach can be implemented, as well as future research directions.

Problem Description

We seek to identify a portion of the Pareto frontier for the objectives of expected net present value and one or more risk measures selected by the decision-maker. We accomplish this by finding the extraction plans that yield the k highest expected net present values and evaluate each of these plans with respect to the selected risk measures. In choosing this strategy, we are implicitly assuming that the decision-maker is willing to sacrifice some, but not too much, expected NPV in exchange for risk reduction. By choosing k large enough, one can generate a wide range of solutions.

In the remainder of this section, we briefly describe the deterministic problem of maximizing the NPV. We then address the problem of finding the solutions with the k highest expected net present values and the calculation of their associated risk measures in Sect. 3.

In the mining industry, planners traditionally think in terms of production blocks. Here, we consider the problem at a slightly higher level of aggregation, and instead use *strata*, a collection of production blocks that generally form a horizontal layer. Each stratum may have a different height, but in practice, they tend to be similar. We assume that each stratum must be extracted entirely or not at all. The mine planner must also choose how each stratum is to be mined, and often has choices of *cutoff grade* (roughly defined, the minimum ore concentration that will be sent to the processing plant to recover ore) and may have the choice of mining speed. (The available methods of extraction vary and are situation-dependent.) We call each option of how to mine a stratum (e.g., combination of cutoff grade and mining speed) a *mode*. In some cases, there are costs for changing from one mode to another, and/or constraints specifying that all extraction underground must be done using the same mode. The latter are often a consequence of the need to match underground infrastructure capabilities with the rate of extraction.

In a typical mine, extraction starts on the surface and a number of strata may be extracted, but due to the need to maintain stability of the sides of the mining pit, it is necessary to extract progressively more volume as one extracts deeper strata on the surface. (Imagine extracting material to form a larger and larger cone.) The additional extracted volume with each progressively deeper stratum often yields more *waste* (material that is not sent to the processing plant due to the low ore concentration), so eventually it becomes more economical to mine underground. Although underground extraction may be slower than surface extraction, it can often be targeted toward areas of higher ore density, and generally reduces the amount of waste material. Before underground mining can begin, however, it is necessary to build underground infrastructure, whose physical form depends upon the type of ore being mined. For base metals such as copper, the underground infrastructure generally includes a shaft with elevators for both employees and ore haulage, and various supporting equipment. This infrastructure is expensive and associated costs are incurred within a short time frame, so mining firms have a tendency to delay the transition to underground mining past the optimal time. We restrict consideration to one set of underground infrastructure. We are aware that a few mines have sunk two or three shafts to progressively deeper strata, but in these cases, the shafts were sunk decades apart and the more recent shafts were built after new mining technology became available, enabling deeper extraction. Geotechnical considerations often limit the duration when both surface mining and construction of underground infrastructure are occurring simultaneously. Any limits of this type can be accommodated within our framework.

To be concrete, in the remainder of this paper, we assume that the method of *block caving* (see Hustrulid and Bullock 2001) is used underground. In block caving, the shaft is sunk to the deepest stratum to be mined and extraction progresses upward toward the surface. Rock is blasted and it falls to the bottom of the underground

cave through vertical shafts called *drawpoints*. "Good" material is transported to the surface and any material regarded as waste can usually be left behind.

We assume that a single surface pit is under consideration along with some underground extraction below it in a single cave. Typically, the depth of the underground cave is restricted due to geotechnical reasons; we can account for such restrictions in our solution approach. Mines rarely shut down unless the price at which they can sell ore falls below their costs. As most mines sell primarily to satisfy long-term contracts at fixed prices and it is difficult and expensive to stop and restart mining due to the need to relocate personnel and to provide security for the mine during any closures, temporary suspensions of operations are infrequent. For these reasons, we assume that the mine operates continuously (i.e., there is no idle time during the extraction of a stratum or between strata). We also assume that ore is sold in the period in which it is extracted; when ore is sold primarily under long-term contracts at fixed prices, there is little incentive to hold inventory. Also, for many types of ore, space considerations limit how much inventory can be held.

In summary, for the single-solution deterministic optimization problem, we seek a mine plan that specifies (i) which strata are to be extracted on the surface and the choice of mining mode for each, (ii) the depth (stratum) to which underground infrastructure should be installed, which also defines the bottom-most stratum extracted underground, and (iii) the shallowest stratum to be mined underground and the choice of mining mode for each stratum extracted underground. The goal is to maximize NPV. For more details on various aspects of the deterministic problem including the formulation as a longest path network (a network representation of a dynamic programming problem), see Newman, Yano and Rubio (2013).

Solution Methodology

Our approach to the problem couples a method for finding the k best solutions in terms of expected NPV and a method for calculating a user-specified risk measure. By identifying the k best solutions from the standpoint of expected NPV and calculating the selected risk measure(s) for each, we are able to construct a portion of the Pareto-optimal frontier. Our approach was motivated by the fact that mine planning problems—due to their combinatorial nature—have an enormous number of feasible solutions. Our expectation is that, among the many solutions, there are many that are near-optimal in terms of expected NPV. If we can identify these solutions, then we can easily compute risk-related performance metrics for them, even in the presence of spatial correlation of ore grades and/or time-correlated ore prices, or other risk-related complications.

In the next subsection, we present our approach for generating the k best solutions in terms of expected NPV. In the following subsection, we discuss an approach for estimating the distribution of the NPV, from which many different risk measures can be derived.

Finding the k Solutions with the Highest Expected NPVs

Our approach is based on a longest-path network formulation of the problem, which is a convenient method for finding the optimal mine plan (maximizing net present value) in a deterministic setting where both above- and underground extraction are involved. The key decisions are which strata to extract above ground, how deep to construct the underground infrastructure (which defines the starting point of underground extraction), and how far up (toward the surface) to continue the underground extraction. If different mining modes are available, then associated decisions must also be made. The fact that underground extraction proceeds upward instead of downward creates a complication: the network representation of the longest-path problem must, somehow, carry information about the deepest stratum extracted above ground because underground extraction cannot proceed past that point. This complication significantly expands the size of the network. A second complication is that the profit obtained from a stratum depends upon when it is extracted (due to the effects of discounting), so these effects must be incorporated by representing time explicitly in the network.

Our problem is more complicated than the standard longest-path problem as we need to find the k solutions with the highest expected NPVs. We adapt the algorithm of Yen (1971), which identifies the k longest paths in a loopless network. Due to the structure of our problem, with the underground mining proceeding upward and the need to properly account for discounting, we cannot use a straightforward network representation, but require special "aggregate" strata in the network that represent different subsolutions corresponding to a set of contiguous strata mined underground. We next present a formulation of our problem as a longest path network using a standard representation without aggregate strata. Later, we introduce the special aggregate strata that facilitate the process of finding the k longest paths.

Each node is defined by the state of the mine and the time period. In addition to indices for real strata, we also introduce indices for *pseudostrata*, indicating the development of underground infrastructure down to a specified stratum depth. The real strata are numbered $1, \ldots, n$ with stratum 1 being at the surface and stratum n being the deepest stratum considered for extraction. The state of the mine is defined by the triple (L_s, L_u, m), where L_s denotes the deepest stratum extracted on the surface, L_u denotes the shallowest stratum mined underground and m represents the mode utilized for the most recent stratum extracted. (Note that this state definition does not fully characterize the physical state of the mine, as we would also need to know the deepest stratum mined underground. However, this state definition is sufficient for future decision-making.) We use the following additional notation:
s: index of strata; $s = 1, \ldots, n, n+1, \ldots, 2n$, where stratum $i+n$ indicates a pseudostratum denoting construction of underground infrastructure down to the physical stratum i, $i = 1, \ldots, n$.
t: index of time periods, $t = 1, \ldots, T$.
$\mathcal{S}(L_s, L_u)$: set of strata that are feasible immediate successors if the system state is (L_s, L_u).

$\pi(s,m,t)$: discounted profit from extracting stratum $s, s \in \{1,\ldots,n\}$ by mode m starting at time t, or the cost (negative profit) of developing underground infrastructure starting in period t to support extraction beginning at pseudostratum s by mode m for $s \in \{n+1,\ldots,2n\}$.

Given an (L_s, L_u) pair, the most recent location of activity is:

$$\sigma = \begin{cases} L_s & \text{if } L_u = 0 \\ L_u & \text{otherwise.} \end{cases} \tag{14.1}$$

Observe that the most recent location of activity can be inferred from the state definition. We assume that underground extraction is performed via block caving which requires starting at the deepest point of the cave and working upward. Therefore, L_s is needed in the state definition so that block caving does not continue past the bottom of the pit created by surface mining.

The network consists of source and sink nodes, and a node for each four-tuple (L_s, L_u, m, t) arranged systematically. (One simple arrangement consists of one column of nodes for each (L_s, L_u) pair, ordered lexicographically.) Within each such column are all relevant (L_s, L_u, m, t) nodes. A directed arc connects node (L_s^1, L_u^1, m^1, t^1) to node (L_s^2, L_u^2, m^2, t^2) if the transition from state (L_s^1, L_u^1) to (L_s^2, L_u^2) is feasible via extraction of a single stratum in $\mathcal{S}(L_s^1, L_u^1)$ and the times and modes are compatible. Typically, we have either $L_s^1 = L_s^2$ or $L_u^1 = L_u^2$. Let $y(s,m,t) = 1$ if stratum s is extracted via mode m starting at time period t and 0 otherwise. Each stratum can be extracted at most once and using at most one mode; these constraints are implicit in the construction of the network, the definition of $\mathcal{S}$ (which partially defines which arcs exist) and the fact that we have a longest-path problem on a directed network.

For compatible m and t values, the arc between nodes (L_s^1, L_u^1, m^1, t^1) and (L_s^2, L_u^2, m^2, t^2) has a "length" $\pi(L_s^1, m^1, t^1)$ if $L_s^1 \neq L_s^2$ or $\pi(L_u^1, m^1, t^1)$ if $L_u^1 \neq L_u^2$, which is the discounted profit from mining stratum L_s^1 (respectively, L_u^1) starting at time period t^1 via mode m^1. For pseudostrata representing underground setups, this represents the discounted cost of constructing the underground infrastructure for excavation starting at stratum L_u^2. All of these values can be computed from the problem data. The time duration corresponding to the arcs emanating from each underground setup node is the net delay between the termination of surface mining and the beginning of underground mining due to construction of the underground infrastructure. If the underground infrastructure can be constructed during the last few periods of surface mining and underground mining can start immediately after surface mining is complete, then the time duration would be zero. If an arc originates at a node with mode m_1 and terminates at a node with mode $m_2 \neq m_1$, costs of switching modes (as applicable) can be included, and if modes m_1 and m_2 are incompatible, then no arc exists between the nodes.

Figure 14.1 illustrates a network for a simple case with three strata and only one mining mode for each stratum. (The mode index, m, is omitted for simplicity.) In the network, S denotes the source and T the terminus or sink, and we seek the longest path between s and t. In this example, strata 1 and 3 take one period to extract either above- or underground, and stratum 2 takes two periods to extract,

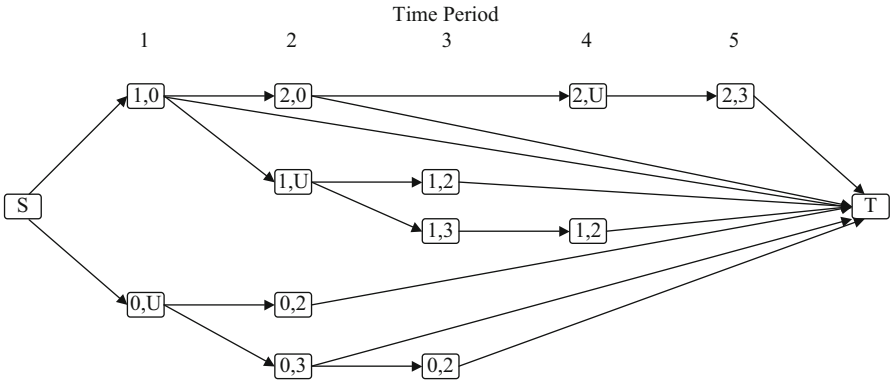


Fig. 14.1 Standard longest path network for example problem

either above- or underground. We also assume that stratum 1, if extracted, must be extracted on the surface, and stratum 3, if extracted, must be extracted underground. For simplicity, we assume that construction of underground infrastructure takes one period, incurs the same cost irrespective of depth, and no extraction can take place while the underground infrastructure is being built. With these simplifications, we need only one pseudostratum for the underground infrastructure, which we label as U. In the figure, due to space limitations, time is indexed on the horizontal axis but in a formal network representation, it would be included in the node definition. If a node appears in a column corresponding to time t, then the corresponding activity starts at the beginning of period t. Corresponding to each arc going into a node (but not shown in the figure) is the net present value associated with the corresponding activity. The ability to account for and distinguish timing is critical here because the discounted cash flows depend upon when each stratum is extracted and when the underground infrastructure is built.

The aggregate strata for underground mining are defined by an underground mining plan, including the set of strata to be mined underground, and where applicable, the mode for each. More specifically, each aggregate stratum represents the *optimal* extraction plan for a contiguous set of strata that can be mined underground. Figure 14.2 illustrates how the network shown in Fig.14.1 changes with the introduction of aggregate strata. We define the aggregate strata as follows:
A1: underground infrastructure built plus stratum 2 alone extracted underground
A2: underground infrastructure built plus stratum 3 alone extracted underground
A3: underground infrastructure built and strata 3 and 2 extracted underground

Because the aggregate strata define the full underground plan, we no longer need to carry L_u in the definition of the nodes in the network. Note that we can also consolidate the construction of the underground infrastructure into the aggregate strata. Because the timing of events within each underground mine plan is known, discounting effects can be handled in an "offline" calculation of the discounted profit for each aggregate stratum. The network with aggregate strata for our example is shown in Fig. 14.2. (Again, the net present value associated with each activity is not shown in the figure.)

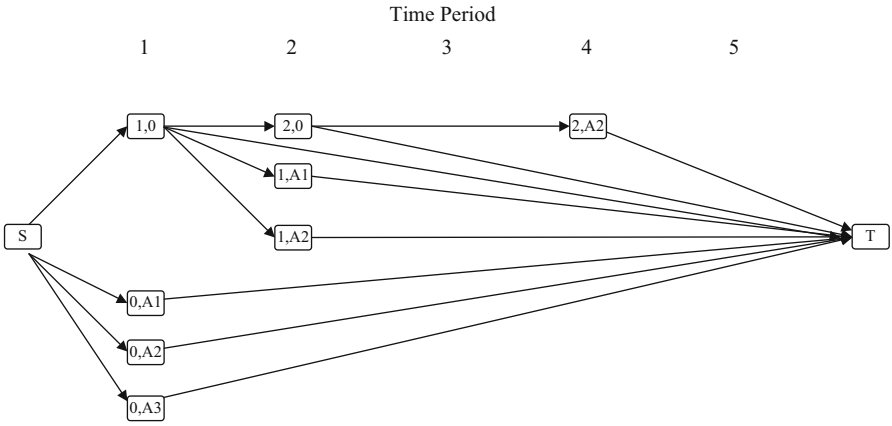


Fig. 14.2 Longest path network with aggregate strata for example problem

The network representation with aggregate strata allows us to easily handle limits on the depth of the underground cave (often called "height of draw") by excluding any underground extraction options that exceed relevant limits. Particularly for large mining projects where it would be technically infeasible to perform all of the extraction exclusively above or exclusively below ground, the use of aggregate nodes significantly reduces the size of the network. Notice that the network in Fig. 14.2 is simpler than that in Fig. 14.1. In this example, because there is only one mode for each activity, the number of paths from S to T is the same in Figs. 14.1 and 14.2, but more typically, the number of paths in the network with aggregate strata will be far smaller. Suppose that we had a choice of cutoff grades for underground mining. Then the standard longest path diagram would show all possible options, but the longest path network with aggregate strata would retain the same structure as shown in Fig. 14.2 because each aggregate node represents only the optimized underground plan for a combination of strata that can be mined underground.

Yen's algorithm was designed for finding the k shortest (or longest) paths in a standard acyclic network. It can be applied in a straightforward fashion to our longest path network with aggregate strata to find the longest path, but it needs to be adapted to deal with the aggregate strata when searching for the kth longest path, $k > 1$. Yen's algorithm iteratively identifies, for $j = 2, \ldots k$, the jth longest path essentially by removing the distinctive portions of the first longest through $(j - 1)$ st longest paths from contention. If, in the course of finding the k longest paths, we find that the jth longest path includes a node associated with an aggregate stratum, then an adjustment of Yen's procedure is needed. Recall that initially, each aggregate stratum in our network represents the *optimal* plan for a set of strata that can be mined underground. As such, the second best plan for that same set of strata could constitute part of second-best overall plan (i.e., the second longest path). So, in general, whenever the jth longest path includes a node associated with an aggregate stratum, before we search for the $(j + 1)$st longest path, we need to replace the

information associated with that aggregate stratum with information on the next best solution for excavating the corresponding set of strata. At each iteration, this process only needs to be performed for at most a single node associated with an aggregate stratum, and Yen's algorithm can be used to find the next best solution for the associated subnetwork.

We note that there are alternate ways to represent the longest path network and alternate ways to find the k longest paths, but our approach utilizing aggregate strata for the underground portion of the mining helps to compartmentalize and simplify the changes in the network that need to be implemented during the iterations of Yen's algorithm.

Calculating Risk Measures

In practice, a decision-maker may be concerned about a variety of different risk measures. Once the set of k solutions with the highest expected NPVs has been identified, it would be straightforward to (numerically) calculate any desired risk measure if the relevant random variables are not too highly linked. Here, we present an example to show that analytical approximations are possible even when some correlation exists among the random variables. The tractability is a consequence of performing the evaluation *for a given solution* rather than performing the evaluation while trying to optimize or otherwise construct a solution. This is one of the main motivations for our approach, and allows the decision-maker to choose almost any risk measure. Even if analytic approximations are unworkable, Monte Carlo simulation could be used, although the computational effort would be more extensive.

We illustrate our approach using a simple example with five strata. There is only one cutoff grade. There is one speed for underground excavation and two speeds above ground. For simplicity, we assume that the ore prices are represented as a time series of lognormal distributions and that the ore yields for each plan are also represented as a time series of lognormal distributions. (The time series of ore distributions depends upon the mining plan, including the sequence of strata to be extracted and the mode by which each is extracted.) Historically, ore prices have been autocorrelated over time, and it is well-established that ore density tends to be spatially correlated. To capture these two effects in a very approximate way, we assume that the *product* of ore price and ore quantity is positively autocorrelated over time. Using hypothetical parameters and accounting for the assumed autocorrelation, we computed the 95 % value-at-risk (VaR), i.e., the 5th percentile of profit, for the 10 longest paths in the network. They are shown in Table 14.1 along with the corresponding expected profit.

In this example, there is considerable disparity among the top ten solutions in terms of expected profit because the problem itself is small. (There are only a few dozen paths, even if one considers unrealistic options such as extracting stratum 1 underground.) The best solution in terms of expected profit (Path 1) has a 95 %+ chance of generating a positive profit, but this may or may not be acceptable to the firm. If the decision-maker wants greater assurance of a positive profit, he or she

Table 14.1 Solutions with highest expected profit and corresponding 95 % value-at-risk

Path	Expected Profit	VaR
1	2200	6
2	2060	−72
3	2060	−72
4	2000	312
5	1980	−146
6	1920	−151
7	1910	−340
8	1860	235
9	1860	234
10	1340	274

might choose Path 4, whose expected profit is 10 % smaller than the maximum but offers a 95 % chance of returning a profit of 312 or higher. Observe also that even among solutions with quite similar expected profit (e.g., Paths 6–9), the 95 % VaR varies widely (between -151 and 235) because the paths themselves represent quite distinct mining plans.

Here, we have presented a simple example for illustration. In practice, the number of strata varies depending upon the location and dispersion of the ore under consideration for extraction and the granularity with which the firm would like to model the extraction decisions. Typically, the number of strata would be several to many dozen. For large problems, we have found that the most challenging aspects are data and memory management in identifying the k longest paths. This difficulty is exacerbated when underground extraction proceeds upward rather than downward because of the need to ensure that extraction does not proceed beyond the deepest stratum mined above ground. (The network representations are much simpler when all extraction proceeds downward.) Some of the data management challenges can be overcome via advanced techniques from computer science, and the challenges related to computer memory will diminish with advancements in the associated technology.

Conclusions

Mine planning involves many uncertain factors, such as ore grade and yield, ore prices, and the costs of necessary inputs, such as labor and energy. Mining firms need to choose a plan that takes into account both expected net present value and various risk measures. We propose a systematic method to identify a portion of the Pareto frontier of solutions. Our approach couples a method for finding the k longest paths in a network, where the path lengths are the net present values of various mining plans, with methods for evaluating the k longest paths with respect to managerially-selected risk measures. Our method allows a decision-maker to determine exactly how much NPV is sacrificed for each unit of risk reduction, and enables the decision-maker to consider multiple risk measures simultaneously, if desired.

We began this study with the hope of considering multiple types of uncertainty, but soon discovered how little fundamental statistical theory exists to represent the

distribution of profit in an easy-to-use form for a realistic instance with characteristics such as the following: (i) time series of ore prices with a specified volatility and autocorrelation; (ii) matrix of distributions of underground infrastructure costs (by stratum depth and time period); and (iii) matrix of distributions of ore yield (by stratum and extraction mode), with some spatial correlation. As decision-makers in mining and other capital-intensive industries increasingly wish to consider risk in making long-term plans, it will be valuable to be able to derive approximate distributions of economic performance measures (e.g., profit) starting with primitives that are represented using common distributions. Further development along these lines will make approaches like ours much more versatile.

More broadly, further research is needed to develop other methods that allow the decision-maker to consider tradeoffs between risk and return in an explicit fashion, and to expand upon our approach by including information updating and contingency options.

Acknowledgment This research was undertaken while the first author was a graduate student at the University of California, Berkeley. The first author gratefully acknowledges support from a fellowship from the National Sciences and Engineering Research Council of Canada.

References

Abdel Sabour, S.A., & Poulin, R. 2010. Mine expansion decisions under uncertainty. *International Journal of Mining, Reclamation and Environment*, *24*(4), 340–349.

Boland, N., Dumitrescu, I.,& Froyland, G.. 2008. *A multistage stochastic programming approach to open pit mine production scheduling with uncertain geology. Working paper, School of Mathematics and Physical Sciences*, Australia: University of Newcastle

Chen, J., Gu, D., & Li, J. 2003. Optimization principle of combined surface and underground mining and its applications. *Journal of the Central South University of Technology (China)*, *10*(3), 222–225.

Chicoisne, R., Espinosa, M., Goycoolea, M., Moreno, E., & Rubio,E. 2011. A new algorithm for the open-pit mine scheduling problem. *Operations Research*, *60*(1), 4–17.

Chiles, J.-P., & Delfiner, P. 2012. *Geostatistics: Modeling spatial uncertainty, Second edition* New York: Wiley Interscience.

Denby, B., & Schofield, D. 1995. Inclusion of risk assessment in open-pit design and scheduling. *Transactions of the Institution of Mining and Metallurgy. Section A: Mining Industry*. *104*, A67–A71.

de Lara, M., Morales, N., & Beeker N. 2013. *Adaptive strategies for the open-pit mine optimal scheduling problem. Working paper, Centre d'Enseignement et de Recherche en Mathematiques et Calcul Scientifique*. France: Universite Paris-Est.

Deutsch, C. V., & Journel, A.G. 1998. *GSLIB: Geostatistical software library and user's guide, Second edition*. New York: Oxford University Press.

Dimitrakoupoulos, R. 1998. Conditional simulation algorithms for modeling orebody uncertainty in open pit mines. *International Journal of Surface Mining, Reclamation and Environment*, *12*, 173–179.

Dimitrakoupoulos, R., Martinez, L., & Ramazan S. 2007. Maximum upside / minimum downside approach to the traditional optimization of open pit mine designs. *Journal of Mining Science*, *43* (1), 73–82.

Dimitrakoupolos, R. 2011. Strategic mine planning under uncertainty. *Journal of Mining Science*, *47*(2), 138–150.

Epstein, R., Goic, M., Weintraub, A., Catalan, J., Santibanez, P., Urrutia, R., Cancino, R., Gaete, S., Aguayo, A.,& Caro, F. 2010. Optimizing long-term production plans in underground and open pit copper mines. *Operations Research 60*(1), 4–17.

Golamnejad, J., Osanloo, M., & Karimi, B. 2006. A chance-constrained programming approach for open pit long-term production scheduling in stochastic environments. The Journal of the South African Institute of Mining and Metallurgy 106, 105–114.

Hershberger, J., Maxel, M., & Suri, S. 2007. Finding the *k* shortest simple paths: A new algorithm and its implementation. *ACM Transactions on Algorithms*, *3*(4), 1–9.

Hustrulid, W.A. & Bullock, R.L. (eds). 2001. *Underground mining methods: Engineering fundamentals and international case studies*. Littleton, Colorado: Society for Mining, Metallurgy and Exploration.

Kumral, M. 2010. Robust stochastic mine production scheduling. *Engineering Optimization*, *42*(6), 567–579.

Lemelin, B., Abdel Sabour, S.A., & Poulin, R. 2006. Valuing Mine 2 at Raglan using real options. *International Journal of Mining, Reclamation and Environment*, *20*(1), 46–56.

Martinez, L.A. 2006. Strategic coal mining planning project using an integrated real options model approach. *Proceedings of the Bowen Basin Symposium*, McKay, Queensland, Australia, October 6–8, 2010.

Newman, A.M., Rubio, E., Caro, R., Weintraub A., & Eurek, K. 2010. A review of operations research in mine planning. *Interfaces*, *40*(3), 222–246.

Newman, A.M., Yano, C.A., & Rubio, R. 2013. Mining above and below ground: Timing the transition. *IIE Transactions 45* (8), 865–882.

Osanloo, M., Golamnejad, J., & Karimi B. 2007. Long-term open pit mine production planning: a review of models and algorithms. *International Journal of Mining, Reclamation and Environment*, *22*(1), 1–33.

Ravenscroft, P.J. 1992. Risk analysis for mine scheduling by conditional simulation. *Transactions of the Institution of Mining and Metallurgy Section A: Mining Industry 101*, A104–A108.

Sarin, R., & West-Hansen, J. 2005. The long-term mine production scheduling problem. *IIE Transactions*, *37*(2), 109–121.

Stacey, T.R., & Terbrugge,P.J. 2000. Open pit to underground–transition and interaction. *Proceedings of the MassMin 2000 Conference*, Brisbane, Queensland, Australia, 29 October–2 November 2000, 97–104

Topuz, E., & Duan, C. 1989. A survey of operations research applications in the mining industry. *CIM Bulletin*, *82*(925), 48–50.

Visser, W.F., & Ding, B.2007. Optimization of the transition from open pit to underground mining. *Proceedings of the International Mining Symposia 2007*, Aachen University, Aachen, Germany, 30–31 May 2007, 131–148

Yen, J.Y. 1971. Finding the *k* shortest loopless paths in a network. *Management Science*, *17*(11), 712–716.

Chapter 15
Multiple-Lot Lot Streaming in a Two-stage Assembly System

Liming Yao and Subhash C. Sarin

Introduction

Lot streaming is the process of using transfer batches (sublots) to move completed portions of a production lot to downstream machines so that their operations can be undertaken in an overlapping fashion. Kalir and Sarin (2000) have shown potential benefits of lot streaming in the flow shop environment for the three commonly used performance measures: makespan, mean flow time and average work-in-process. In this chapter, we address a problem of streaming multiple lots in a two-stage assembly system to minimize the makespan. We designate this problem as a multiple-lot, two-stage assembly, lot streaming problem (ML-TSALP). The presence of multiple lots requires simultaneous determination of sublot sizes and the sequence in which to process the lots.

The configuration of the assembly system that we consider is illustrated in Fig. 15.1. The first stage of this system consists of multiple, parallel machines. Each of these machines produces a particular subassembly type for each production lot. These subassemblies are then assembled into final products at the second stage. The example in Fig. 15.1 consists of three subassembly machines at Stage 1 and two production lots of 50 and 40 items each. A lot-detached setup (that can be performed before the arrival of the lot) is incurred on every machine at both stages. For example, these values are assumed to be 40, 40, and 60 units for lot 1 on the subassembly machines, and 10 units on the assembly machine. Similar numbers for lot 2 are 30, 20, and 30 units for the setups on the subassembly machines, and 60 units on the assembly machine. The processing times for lot 1 are 1.5, 1, and 1 units per item on the subassembly machines and 1 unit on the assembly machine. For lot 2, the processing times are 1, 2, and 0.5 units per item on the subassembly machines and 2 units on the assembly machine.

We depict three schedules (designated Schedule 1, Schedule 2 and Schedule 3) in Fig. 15.1. Figure 15.1a presents Schedule 1, in which the processing of lot 1 precedes

S. C. Sarin (✉) · L. Yao
Grado Department of Industrial and Systems Engineering,
Virginia Tech, Blacksburg, VA 24061, USA
e-mail: sarins@vt.edu

P. S. Pulat et al. (eds.), *Essays in Production, Project Planning and Scheduling,*
International Series in Operations Research & Management Science 200,
DOI 10.1007/978-1-4614-9056-2_15, © Springer Science+Business Media New York 2014

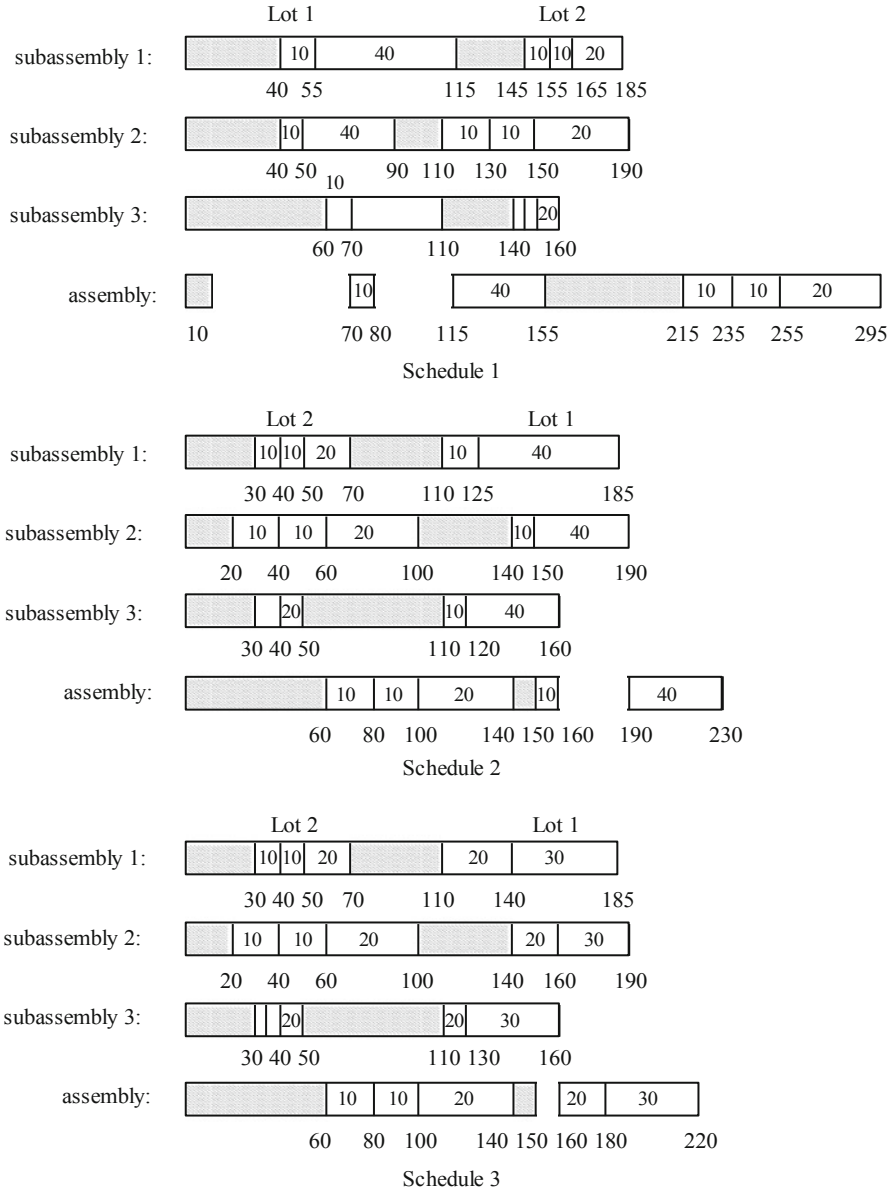


Fig. 15.1 Example depicting streaming of multiple lots in a two-stage assembly system

that of lot 2. The sublot sizes used for lot 1 are 10 and 40, and those for lot 2 are 10, 10, and 20. Schedule 2 is shown in Fig. 15.1b in which lot 2 is processed before lot 1, while the sublot sizes used for both the lots are the same as those in Schedule 1. Due to this change in sequence in the processing of the lots, makespan decreases

from 295 to 230. Note that if we change the sublot sizes for lot 1 to 20 and 30 while keeping the sequence to be lot 2 followed by lot 1 (Schedule 3 in Fig. 15.1c), the makespan decreases further from 230 to 220. Our aim is to determine the sequence of the lots and sublot sizes for each lot so as to minimize the makespan.

The ML-TSALP has not been addressed in the literature. Lee et al. (1993), Hariri and Potts (1997), and Sun et al. (2003) have addressed the scheduling problem for the above two-stage assembly system without considering lot streaming. Lee et al. (1993) have studied a three-machine assembly scheduling problem with two subassembly machines at Stage 1 and an assembly machine at Stage 2. They have shown that their problem is strongly NP-hard and have identified special cases of the problem that are solvable in polynomial time. They also present several heuristics along with their respective worst-case performance bounds. Hariri and Potts (1997) have extended the problem to consider an arbitrary number of subassembly machines at Stage 1, and they have developed a branch-and-bound algorithm for its solution. Sun et al. (2003) have considered a three-machine assembly scheduling problem and have presented several heuristics to address the worst-case scenarios presented in the literature. Recently, Sarin et al. (2011) have presented polynomial-time algorithms to determine the optimal number of continuous and integer-sized sublots, given a maximum number of sublots, for the single-lot TSALP. Lot streaming in the presence of multiple lots has also been addressed for two-machine flow shop, which is a special case of the two-stage assembly system considered here. For this problem, Baker (1995) and Cetinkaya and Kayaligil (1992) have shown that unit-sized sublots are optimal for the problem with no setup time, and have solved the resulting sequencing problem using a modification of Johnson's algorithm (Johnson 1954). For the case with lot-detached setup times and sublot transfer times, Vickson (1995) has shown that the sublot sizing and lot sequencing problems are independent. They derive optimal sublot sizes and then solve the lot sequencing problem using Johnson's algorithm. For the case with lot-detached setup and removal times, Cetinkaya (1994) has also shown that the sublot sizing problem and sequencing problems are independent, and furthermore, the optimal sublot sizes are geometric. A sequence of lots is determined using a modification of Johnson's algorithm based on run-in and run-out times. Sriskandarajah and Wagneur (1999) have addressed the multiple-lot, lot streaming problem in a no-wait two-machine flow shop, and they have proved that the sublot sizing and lot sequencing problems are independent in this case as well. The optimal continuous sublot sizes are geometric and the optimal lot sequence can be obtained using an algorithm proposed by Gilmore and Gomory (1964). Kalir and Sarin (2003) have considered a problem with sublot-attached setups. They have presented solution procedures for two cases: equal and unequal sublot sizes for all the lots. For the equal sublot size case, their algorithm iterates over all possible values of the size of a sublot, and it sequences the lots using a modified Johnson's algorithm. For the unequal sublot size case, they have proposed a two-phase procedure in which the construction phase determines the sequence using a modified Johnson's algorithm and the improvement phase reoptimizes sublot sizes based on the sequence obtained. The iteration continues until no improvement can be made.

This chapter is organized as follows. In the section "Problem Description and Basic Properties," we describe the ML-TSALSP and present its useful structural properties. A mixed integer programming formulation for this problem is presented in the following section. In the section "A Branch-and-Bound-based Methodology for the ML-TSALSP," we propose a branch-and-bound procedure, which includes the development of various lower and upper bounds and dominance rules that help in curtailing nodes of the branch-and-bound tree. Results of our computational experimentation demonstrate the efficacy of the proposed branch-and-bound algorithm. Finally, concluding remarks are made in last section.

Problem Description and Basic Properties

The ML-TSALSP can formally be described as follows. There are N production lots to be processed in a two-stage assembly system. Each lot j consists of U_j items. There is a set Ω of M subassembly machines at the first stage and an assembly machine at the second stage. The per-unit processing time for the items of a lot can vary over the machines, and is designated by p_{jk} for lot j on machine k. The unit processing time is different for different lots on the assembly machine, and is designated by p_{jA} for lot j. Lot-detached setup times are incurred before each lot j starts its processing on subassembly machine $k, k \in \Omega$, and the assembly machine A, and are denoted by $t_{jk}, k \in \Omega$, and t_{jA}, respectively. We assume that all the machines (those at Stage 1 and Stage 2) use the same number of sublots n_j, for lot $j, j = 1, ..., N$, which are given. We make the following assumptions: (1) all machines are available at time zero; (2) sublot sizes are continuous; (3) the processing of a sublot on machine A can be started only after a sufficient number of its components have finished processing at the first stage; (4) sublot intermingling is not allowed, that is, once a machine starts processing a lot, it has to finish all items (and hence all sublots) of that lot before beginning to process the next lot; and (5) each assembly of job j requires one subassembly of each kind. In case, an assembly of job j requires α_k subassemblies from subassembly machine k, without loss of generality, we can assume $p'_{jk} \equiv \alpha_k p_{jk}$. The objective is to determine: (1) the sequence in which to process production lots and (2) sublot sizes for each lot j for processing on the subassembly and assembly machines so as to minimize the makespan, that is, the completion time of the last sublot of the last lot on the assembly machine A. We use the following notation:

Parameters

N—Number of production lots.
M—Number of subassembly machines.
U_j—Number of items in lot $j, j = 1, ..., N$.
n_j—Number of sublots of lot $j, j = 1, ..., N$.
t_{jk}—Detached setup time of lot j on subassembly machine $k, k = 1, ..., M$.
t_{jA}—Detached setup time of lot $j$on the assembly machine.
p_{jk}—Unit processing time of lot j on subassembly machine $k, k = 1, ..., M$.
p_{jA}—Unit processing time of lot j on the assembly machine.

Variables

x_{ijk} = 1, if lot j is sequenced in position i of the sequence on subassembly machine k; and 0, otherwise.
x_{ijA} = 1, if lot j is sequenced in position i of the sequence on the assembly machine; and 0, otherwise.
s_{juk} = Size of sublot u of lot j on subassembly machine k.
s_{juA} = Size of sublot u of lot j on the assembly machine.
ξ_{ik} = Completion time of a lot on subassembly machine k if it is sequenced in position i.
ξ_{iA} = Completion time of a lot on the assembly machine if it is sequenced in position i.

Before presenting a mathematical formulation for the ML-TSALSP, we first mention the following properties, which help in curtailing the type of sequence and sublot sizes that we need to consider.

Property 1. *There exists an optimal schedule in which the sequences of the lots are the same on all the machines.*
Let μ_k be a sequence of lots on subassembly machine k, and μ_A be a different sequence on assembly machine A. It is easy to see that we can alter the sequence μ_k on each subassembly machine k to conform to μ_A without worsening the makespan. Consequently, we can drop the subscripts "k" and "A" from the notation of the sequencing variables, x_{ijk} and x_{ijA}.

Property 2. *For a given sequence of lots, there exist optimal sublot sizes for each lot such that each lot's completion time is minimized.*
Clearly, there exists no idle time in between the processing of production lots on each subassembly machine because, otherwise, the makespan can potentially be improved by a local left shift. Therefore, the minimization of makespan for the ML-TSALSP can be considered as the minimization of total idle time on the assembly machine A. Hence, for a given sequence of lots, the minimization of total idle time on the assembly machine A is equivalent to minimizing the completion time of each lot in the sequence.

Property 3. *There exists an optimal schedule in which each lot is split into consistent sublots for processing on subassembly and assembly machines.*
By Property 2, the problem of minimizing the completion time of a lot is equivalent to a two-stage, single-lot, lot-detached setup time makespan minimization lot streaming problem. Sarin et al. (2011) have shown the optimality of consistent sublots for this problem. Because of Property 3, we can drop the subscripts "k" and "A" from the notation of sublot sizes, namely s_{iek} and s_{ieA}.

Observation Property 3 also establishes the fact that for a given sequence of lots, the optimality conditions for the two-stage, single-lot lot streaming problem, will be valid for the ML-TSALSP as well.

A Mixed Integer Programming Formulation

Our model formulation for the ML-TSALSP is as follows:

ML-TSALSP-M:

$$\text{Minimize} \quad \xi_{NA} \tag{15.1a}$$

$$\text{Subject to} \quad \sum_{i=1}^{N} x_{ij} = 1, \quad \forall j = 1, ..., N, \tag{15.1b}$$

$$\sum_{j=1}^{N} x_{ij} = 1, \quad \forall i = 1, ..., N, \tag{15.1c}$$

$$\xi_{ik} \geq \xi_{i-1,k} + \sum_{j=1}^{N} (t_{jk} + p_{jk} U_j) x_{ij}, \quad \forall i = ..., N, k = 1, ..., M, \tag{15.1d}$$

$$\xi_{iA} \geq \xi_{i-1,A} + \sum_{j=1}^{N} \left(t_{jA} + p_{jA} U_j\right) x_{ij}, \quad \forall i = 1..., N, \tag{15.1e}$$

$$\xi_{iA} + (1 - x_{ij}) \left((p_{jk} + p_{jA}) U_j\right) \geq \xi_{i-1,k} + t_{jk} x_{ij} + p_{jk} \sum_{u=1}^{e} s_{ju} + \rho_{jA} \sum_{u=e}^{n_j} s_{ju},$$

$$\forall i, j = 1....N, e = 1, ..., n_j, \quad k = 1, ..., M, \tag{15.1f}$$

$$\sum_{u=1}^{n_j} s_{ju} = U_j, \quad \forall j = 1, ..., N, \tag{15.1g}$$

$$s_{ju} \geq 0, \quad \forall = 1, ..., N, u = 1, ...n_j, \tag{15.1h}$$

$$x_{ij} \in \{0,1\}, \quad \forall i, j = 1, ..., N. \tag{15.1i}$$

Constraints (15.1b) and (15.1c) are assignment constraints that ensure that each lot is assigned to a position and each position is allocated to only one lot, respectively, in a permutation of lots. Constraints (15.1d) and (15.1e) ensure that a subassembly machine at Stage 1 and the assembly machine at Stage 2, respectively, can process only a single production lot at-a-time, and also they capture the respective completion times of the lots on the subassembly and assembly machines. Constraints (15.1f) assert that the completion time of each lot *j,* if assigned to position *i* on the assembly machine, can be no less than the completion time of each of its sublots on assembly machine *A*. Specifically, if $x_{ij} = 1$, we have

$$\xi_{iA} \geq \xi_{i-1,k} + t_{jk} + p_{jk} \sum_{u=1}^{e} s_{ju} + p_{jA} \sum_{u=e}^{n_j} s_{ju}, \quad \forall e = 1, ..., n_j, i, j = 1, ..., N,$$

$$k = 1, ..., m,$$

which requires the completion time of lot j in position i on assembly machine A to be greater than or equal to the total time required to complete every sublot of lot j, taking into consideration its processing on every subassembly machines $k, k = 1, ..., M$. In case $x_{ij} = 0$, we have

$$\xi_{iA} + \left(p_{jk} + p_{jA}\right) U_j \geq \xi_{i-1,k} + p_{jk} \sum_{u=1}^{e} s_{ju} + p_{jA} \sum_{u=c}^{n_j} s_{ju}.$$

Since $(p_{jk} + p_{jA})U_j \geq p_{jk} \sum_{u=1}^{e} s_{ju} + p_{jA} \sum_{u=e}^{n_j} s_{ju,} \ \forall \ e = 1, ..., n_j$, the above constraint is redundant. Constraints (15.1g) ensure that the sum of the sublot sizes of lot j is equal to the number of items in lot j. Constraints (15.1h) represent the nonnegativity of sublot sizes, and constraints (15.1i) represent the binary restriction on the assignment variables.

A Branch-and-Bound-based Methodology for the ML-TSALSP

In this section, we propose a branch-and-bound-based methodology for the ML-TSALSP. First, we present a mathematical expression for the makespan that is utilized in the sequel.

Expression for Makespan

The expression for the makespan of a given sequence of lots relies on the properties stated in the section "Problem Description and Basic Properties." Given a feasible Schedule π, *let* $\pi(i)$, $i = 1, ..., N$, denote the lot located at its ith position. Then, the makespan can be expressed by

$$M(\pi) = \max \left\{ \max_{1 \leq k \leq M} \left\{ \max_{1 \leq i \leq N} \left\{ \max_{1 \leq e \leq n_{\pi(i)}} \left\{ \psi_{iek}(\pi) \right\} \right\} \right\}, \sum_{j=1}^{N} \left(t_{jA} + p_{jA} U_j\right) \right\}, \quad (15.2)$$

where $\psi_{iek}(\pi)$ is the completion time of sublot e of the lot in position i of Schedule π on machine k, and is given by

$$\psi_{iek}(\pi) = \sum_{v=1}^{i-1} \left(t_{\pi(v)k} + p_{\pi(v)k} U_{\pi(v)}\right) + \left(t_{\pi(i)k} + \sum_{w=1}^{e} p_{\pi(i)k} s_{\pi(i)w} + \sum_{w=e}^{n_{\pi(i)}} p_{\pi(i)A} s_{\pi(i)w} \right) + \sum_{v=i+1}^{N} \left(t_{\pi(v)A} + p_{\pi(v)A} U_{\pi(v)}\right) \quad (15.3)$$

From the definition of makespan, we have the following inequality:

$$M(\pi) \geq \psi_{iek}(\pi), \forall k = 1, ..., M, e = 1, ..., n_{\pi(i)}, i = 1, ..., N. \tag{15.4}$$

A critical sublot is defined to be a sublot for which the equality holds in (15.4) for some machine k,while a critical lot is a lot to which the critical sublot belongs. Note that if the equality holds for some i, e, and k, we define that lot, sequenced in position i, and its sublot e to be critical with respect to machine k. Also, if a lot is critical, all its sublots will be critical based on the criticality of sublots in the single-lot, two-stage assembly lot streaming problem (see Sarin et al. (2011)). For instance, if a production lot in position c of Schedule π is critical with respect to machine k, we have

$$\begin{aligned} M(\pi) = \psi_{cek}^{(\pi)} = & \sum_{v=1}^{c-1} \left(t_{\pi(v)k} + p_{\pi(v)k} U_{\pi(v)}\right) \\ & + \left(t_{\pi(c)k} + \sum_{u=1}^{e} p_{\pi(c)k} s_{\pi(c)u} + \sum_{u=e}^{n_{\pi(c)}} p_{\pi(c)A} s_{\pi(c)u} \right) \\ & + \sum_{v=c+1}^{N} \left(t_{\pi(v)A} + p_{\pi(v)A} U_{\pi(v)}\right), \forall e = 1, ..., n_{\pi(c)} \end{aligned} \tag{15.5}$$

Also, note that when the makespan $M(\pi) = \sum_{j=1}^{N} \left(t_{jA} + p_{jA} U_j\right)$, there will be no critical sublots and critical lots.

Let τ_i denote a partial sequence containing a set of lots that has been scheduled up to position i in the permutation, and τ_i' denote the set of lots yet to be scheduled. In the branch-and-bound tree (see Fig. 15.2), a node at level i represents a subproblem in which a partial sequence τ_i has been fixed, and the remaining sequence needs to be determined among the lots in set τ_i' in order to minimize the makespan. We denote such a subproblem by $P_{\tau_i}^{i}$.

Let $C_k(\tau_i)$ and $C_A(\tau_i)$ denote the completion times on the subassembly machine k and the assembly machine A, respectively, for the lot in position i of the partial sequence τ_i. At a level i of the branch-and-bound tree, each node corresponds to a partial permutation in which the lots have been sequenced in the first i positions. If π is a complete sequence built from a τ_i, then the makespan of such a sequence is given by

$$M(\pi) = \max \left\{ \max_{1 \leq k \leq M} \left\{ \max_{i+1 \leq q \leq N} \left\{ \max_{1 \leq e \leq n_{\pi(q)}} \left\{ \psi_{qek}(\pi) \right\} \right\} \right\}, C_A(\tau_i) + \sum_{j \in \tau'_i} \left(t_{jA} + p_{jA} U_j\right) \right\}, \tag{15.6}$$

where ψ_{qek} for any position q after i is obtained as follows:

$$\psi_{qek}(\pi) = C_k(\tau_i) + \sum_{v=i+1}^{q-1} \left(t_{\pi(v)k} + p_{\pi(q)k} U_{\pi(v)}\right)$$

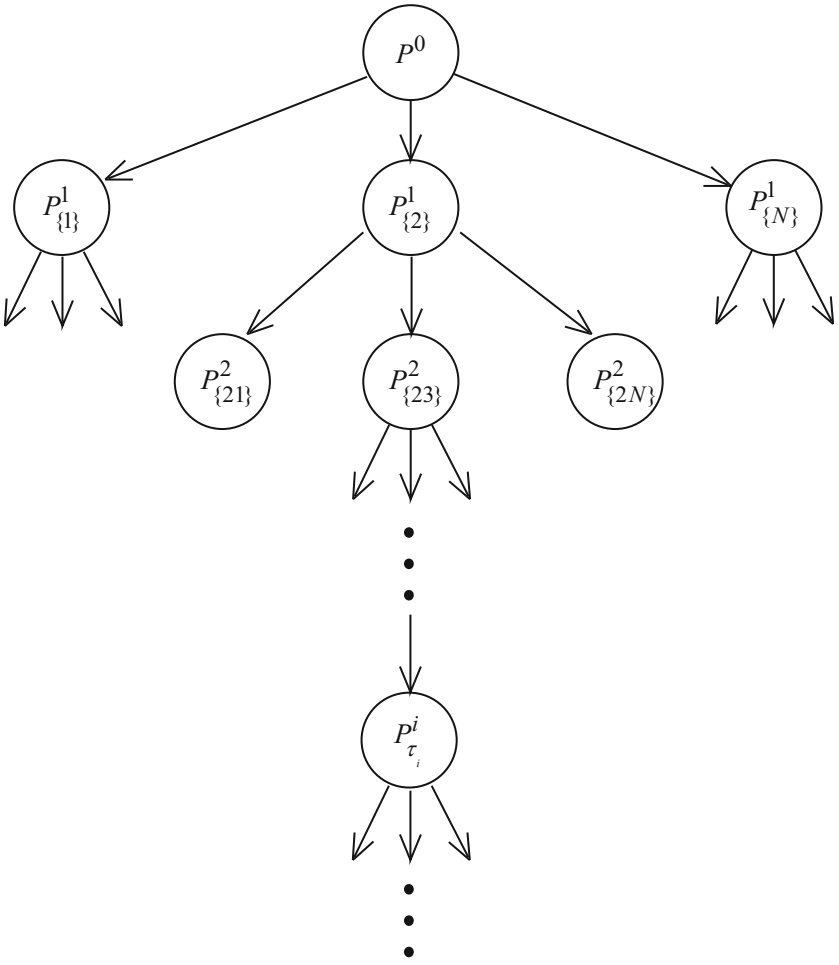


Fig. 15.2 The branch-and-bound tree for the ML-TSALSP

$$+\left(t_{\pi(q)k}+\sum_{w=1}^{e}p_{\pi(q)k}s_{\pi(q)w}+\sum_{w=e}^{n_{\pi(q)}}p_{\pi(q)A}s_{\pi(q)w}\right)$$
$$+\sum_{v=q+1}^{N}\left(t_{\pi(v)A}+p_{\pi(v)A}U_{\pi(v)}\right) \tag{15.7}$$

Note that $M(\pi)\geq \psi_{qek}(\pi), \forall k=1,...,M, e=1,...,\pi(q), q=1,...,N$. In case the equality in the above expressions holds for a lot in position c with respect to some machine k, then that lot $\pi(c)$ and its sublots are critical.

Determination of Lower Bounds

Let $C_A^0(\tau_i, j)$ be the completion time of lot j on machine A if it is sequenced in position $i+1$, assuming that lot j is processed immediately on machine A after the largest among its completion times on subassembly machines. Similarly, $C_A^0(\pi\backslash\{j\}, j)$ is the completion time of lot j if it is sequenced in the last position assuming that lot j is processed immediately on machine A after the largest among its completion times on subassembly machines. We have the following lower bounds:

$$\text{Lower Bound 1}: LB^1(\tau_i) = C_A(\tau_i) + \sum_{j \varepsilon \tau'_2} (t_{jA} + p_{jA}U_j). \tag{15.8}$$

$$\text{Lower Bound 2}: LB^2(\tau_i) = \min_{j \varepsilon \tau'_1} \left\{ C_A^0(\tau_i, j) + \sum_{\substack{q \varepsilon \tau'_1, \\ q \neq j}} (t_{qA} + p_{qA}U_q) \right\}. \tag{15.9}$$

$$\text{Lower Bound 3}: LB^3(\tau_i) = \min_{j \varepsilon \tau'_1} \left\{ C_A^0(\pi \setminus \{j\}, j) \right\}. \tag{15.10}$$

Note that $LB^1(\tau_i)$, $LB^2(\tau_i)$and $LB^3(\tau_i)$are machine-based lower bounds. Next, we present a lower bound based on a relaxed version of the original problem. As before, let τ_i' be the set of remaining lots corresponding to τ_i, and $\pi_k^*(\tau'_i)$ denote an optimal sequence of the lots in set τ_i' for a two-machine system consisting of the subassembly machine k and the assembly machine A. Let $C_k(\pi_k^*(\tau'_i))$ denote the optimal makespan obtained for this relaxed problem. We have the following lower bound:

$$\text{Lower Bound 4}: LB^4(\tau_i) = \max_{1 \leq k \leq M} \left\{ C_k(\tau_i) + C_k(\pi_k^*(\tau'_i)) \right\}. \tag{15.11}$$

Next, we present the modified Johnson's algorithm for a two-machine problem containing machine k and machine A to obtain $\pi_k^*(\tau'_i)$ and $C_k(\pi_k^*(\tau'_i))$. But, first, we state the following property (see Cetinkaya (1994).

Property 4. *The optimal sublot sizes for a lot $j \in \tau_i'$ belonging to the optimal sequence $\pi_k^*(\tau'_i)$, can be obtained by solving the single-lot lot streaming problem for a two-machine flow shop containing machine k and machine A, irrespective of the lot sequence.*

As a result of this property, the sublot sizes s_{juk}^* for each lot j in solution $\pi_k^*(\tau'_i)$ are geometric, and can be obtained as follows:

$$s_{juk}^* = \begin{cases} \dfrac{(q_k)^{u-1} - (q_{jk})^u}{1 - (q_{jk})^{n_j}}, \text{ if } q_{jk} \neq 1, \\ \dfrac{U}{n_j}, \text{otherwise}, \forall k = 1, ..., M_i, u = 1, ..., n_j, j \in \tau', \end{cases} \tag{15.12}$$

where $q_{jk} = \frac{p_{jA}}{p_{jk}}$. With the sublot sizes determined, we then calculate run-in and run-out times for each lot as follows:

$$RI_{jk} = \max\left\{0, t_{jk} + p_{jk}s^*_{j1k} - t_{jA}\right\}, \quad \forall \mathrm{j} \in \tau_i', k = 1, ..., M \tag{15.13}$$

$$RO_{jA} = \max\left\{p_{jA}s^*_{jnjk}, (t_{jA} - t_{jk}) + U_j(p_{jA} - p_{jk})\right\}, \quad \forall \mathrm{j} \in \tau_i', k = 1, ..., M \tag{15.14}$$

The modified Johnson's rule based on the concept of run-in and run-out times, which is similar to the algorithm proposed in Cetinkaya (1994), is as follows:

Proposition 1. *In a two-machine flow shop with machines k and A, lot u precedes lot v in solution* $\pi^*_k(\tau'_i)$ *if the following is true*

$$\min\{RI_{uk}, RO_{vA}\} \leq \min\{RI_{vk}, RO_{uA}\}, \quad \forall u,v \in \tau_i'. \tag{15.15}$$

Determination of Upper Bounds

Given a partial sequence τ_i, an upper bound can be computed by

$$UB(\tau_i) = \min_{1 \leq k \leq M}\left\{C_k(\tau_i + \pi^*_k(\tau'_i))\right\}. \tag{15.16}$$

Note that $\tau_i + \pi^*_k(\tau'_i)$ constitutes a complete, feasible sequence for each k and $C(\tau_i + \pi^*_k(\tau'_i))$ can be obtained by solving a linear programming model for the permutation sequence $\tau_i + \pi^*_k(\tau'_i)$.

Development of Dominance Rules

In this section, we present some dominance rules that reduce the number of branches generated for a branch-and-bound-tree-based method.

From Properties 2 and 3, we have that if a permutation of lots is given, then the problem can be decomposed into N single-lot, two-stage assembly lot streaming subproblems (Sarin et al. 2011). Therefore, the optimality conditions developed for that problem hold for the ML-TSALSP. Given a partial permutation of τ_i and set τ_i' containing the remaining lots, we consider the problem of assigning a lot to position $i+1$. As noted earlier, for any lot $j \in \tau'$, the criticality of a sublot of j is associated with a subassembly machine. A subassembly machine may or may not have a critical sublot associated with it. If there exists a critical sublot for a machine, we call it a pattern-switching sublot and designate that machine as k_d. When more than one sublots are critical for a machine, then the last of these sublots is a pattern-switching sublot. We denote the pattern-switching sublot of lot j for machine k by ρ_{jk}. This is illustrated in Fig. 15.3. Let σ_{jk_d} represent cumulative sublot sizes of lot j beyond which machine k_d is not critical. Let $D(\tau_i, j) = \{k_1, k_2, ..., k_r\}$ be a set of dominant machines of

lot j, each of which has a pattern-switching sublot associated with it; and let $W(\tau_i, j) = \{\rho_{jk_1}, \rho_{jk_2}, ..., \rho_{jk_r}\}$ be the sequence of the pattern-switching sublots of lot j. Suppose all of the subassembly machines are ordered in their nondecreasing values of unit processing time p_{jk} for lot j, and reindexed from 1 to M. We have the following optimality condition for the sublots of lot j (see Yao 2008) and Sarin et al. (2011).

Property 5. *Given a partial permutation τ_i, there exists an optimal sublot-size solution for a lot $j \in \tau'$ in position $i+1$ in which the following conditions hold:*

$$\sum_{u=1}^{n_j} s_{ju} = U_j, \tag{15.17}$$

$$\begin{aligned} &\sigma_{jk_{d-1}} < \sum_{u=1}^{e} s_{ju} \le \sigma_{jk_d}, \forall k_{d-1}, k_d \in D(\tau_i, j), \\ &\forall \rho_{jk_{d-1}}, \rho_{jk_d} \in W(\tau_i, j), e = \rho_{jk_{d-1}} + 1, ..., \rho_{jk_d}, \end{aligned} \tag{15.18}$$

$$\begin{aligned} &\left(C_{k_{d-1}}(\tau_i) + t_{jk_{d-1}}\right) + p_{jk_{d-1}} \sum_{u=1}^{\rho_{jk_{d-1}}} s_{ju} + p_{jA} s_{j p_{jk_{d-1}}} \\ &\quad = \left(C_{k_d}(\tau_i) + t_{k_d}\right) + p_{jk_d} \sum_{u=1}^{\rho_{jk_{d-1}}+1} s_{ju}, \\ &\forall k_{d-1}, k_d \in D(\tau_i, j), \rho_{jk_{d-1}}, \rho_{jk_{d-1}} \in W(\tau_i, j), \end{aligned} \tag{15.19}$$

$$s_{j,u+1} = s_{ju} q_{jk_d}, \forall k_d \in D(\tau_i, j), \rho_{jk_d} \in W(\tau_i, j), u = \rho_{jk_{d-1}} + 1, ..., \rho_{jk_d} - 1. \tag{15.20}$$

Note that we treat $(C_{i-1,k_{d-1}} + t_{jk_{d-1}})$ and $(C_{i-1,k_d} + t_{jk_d})$ as the lot-detached setup time for lot j to be scheduled in position $i+1$. Similarly, we also have the following property:

Property 6. *Given a partial permutation τ_i, the above conditions (15.18), (15.19) and (15.20) lead to the following inequality for lot j:*

$$\begin{aligned} &q_{jk_d} s_{j\rho_{jk_{d-1}}} \le s_{j\rho_{jk_{d-1}}+1} \le q_{jk_{d-1}} s_{j\rho_{jk_{d-1}}}, \\ &\forall k_{d-1}, k_d \in D(\tau_i j), \forall \rho_{jk_{d-1}}, \rho_{jk_d} \in W(\tau_i, j). \end{aligned} \tag{15.21}$$

Properties for the First and the Last Sublots

Consider a subproblem P_j^k, which is a two-machine flow shop lot streaming problem for lot j corresponding to subassembly machine k and assembly machine A, in the absence of lot-detached setups. Problem P_j^k can be easily solved and its optimal sublot sizes are geometric in nature as shown by Trietsch (1987) and Potts and Baker (1989). Hence, for problem P_j^k, the sizes of the first and last sublots for a given lot j

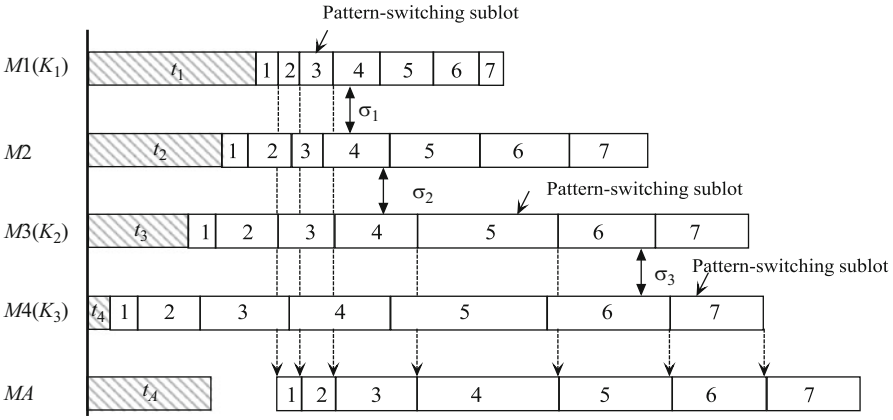


Fig. 15.3 Illustration of pattern-switching sublots

are as follows:

$$s_{j1}^{k} = \begin{cases} \dfrac{1-q_{jk}}{1-(q_{jk})^{n_j}} U_j, \text{ if } q_k \neq 1, \\ \dfrac{U_j}{n_j}, \text{otherwise}, \forall k = 1, ..., M, j = 1, ..., N, \end{cases} \tag{15.22}$$

and,

$$s_{jn_j}^{k} = \begin{cases} \dfrac{(q_{jk})^{n_j-1} - (q_{jk})^{n_j-1}}{1-(q_{jk})^{n_j-1}} U_j, \text{ if } q_k \neq 1, \\ \dfrac{U_j}{n_j}, \text{otherwise}, \forall k = 1, ..., M, j = 1, ..., N. \end{cases} \tag{15.23}$$

Without loss of generality, suppose all the subassembly machines are ordered in nondecreasing order of their unit processing times $p_{jk}, k = 1, ..., M$, for lot j, and reindexed from 1 to M. We have

$$s_{j1}^{1} \leq s_{j1}^{2} \leq ... \leq s_{j1}^{k} \leq ... \leq s_{j1}^{M}, \text{and} \tag{15.24}$$

$$s_{jn_j}^{1} \geq s_{jn_j}^{2} \geq ... \geq s_{jn_j}^{k} \geq ... \geq s_{jn_j}^{M}, \tag{15.25}$$

since $q_{j1} \geq q_{j2} \geq ... \geq q_{jM}$ (see (15.22) and (15.23)). Let $lb_{j1}, ub_{j1}, lb_{jn_j}$ and ub_{jn_j} be defined as follows:

$$lb_{j1} = \min_{1 \leq k \leq M} \left\{ s_{j1}^{k} \right\} = s_{j1}^{1}, \tag{15.26}$$

$$ub_{j1} = \max_{1 \leq k \leq M} \left\{ s_{j1}^{k} \right\} = s_{j1}^{M}, \tag{15.27}$$

$$lb_{jn_j} = \min_{1 \leq k \leq M} \left\{ s_{jn_j}^{k} \right\} = s_{jn_j}^{M}, \tag{15.28}$$

$$ub_{jn_j} = \max_{1 \leq k \leq M} \left\{ s_{jn_j}^{k} \right\} = s_{jn_j}^{1}. \tag{15.29}$$

Next, we show that the $lb_{j1}, ub_{j1}, lb_{jn_j}, ub_{jn_j}$ values defined above constitute lower and upper bounds on the optimal-sized first and last sublots s_{j1} and s_{jn_j}, respectively, of lot j.

Proposition 2. *Given a partial permutation τ_i, if lot j is assigned to position $i+1$, then there exists an optimal solution for lot j for which the first sublot size s_{j1} and last sublot size s_{jn_j} satisfy the following inequalities:*

$$lb_{j1} \le s_{j1} \le ub_{j1}, \text{ and } lb_{jn_j} \le s_{jn_j} \le ub_{jn_j}, \forall j = 1, ..., N \tag{15.30}$$

Proof: We prove this result by contradiction. Our argument relies on the fact that if the alleged limits for sublot sizes are not met, then the sum of the sublot sizes is either less than or greater than u_j. Let $Q_j^1 = \left\{s_{j1}^1, ..., s_{jn_j}^1\right\}$ and $Q_j^M = \left\{s_{j1}^M, ..., s_{jn_j}^M\right\}$ be two optimal solutions for relaxed problems P_j^1 and P_j^M, respectively. We have the following geometric relationships:

$$s_{ju+1}^1 = q_{j1} s_{ju}^1, \quad \forall u = 1, ..., n_j - 1, \tag{15.31}$$

$$s_{ju+1}^M = q_{jM} s_{ju}^M, \quad \forall u = 1, ..., n_j - 1, \tag{15.32}$$

Let solution Q_j^* be an optimal sublot-size solution $\left\{s_{j1}^*, ..., s_{jn_j}^*\right\}$ for lot j in position $i+1$. Let $D(\tau_i)$ be the set of r dominant machines, each of which has a pattern-switching sublot associated with it; and let $W(\tau, j) = \left\{\rho_{jk_1}, \rho_{jk_2}, ..., \rho_{jk_r}\right\}$ be a sequence of pattern-switching sublots for lot j. By the order of subassembly machines in the nondecreasing order of unit processing time, we have the following relationship among the processing time ratios:

$$q_{jM} \le q_{jk_r} \le ... \le q_{jk_1} \le q_{j1}. \tag{15.33}$$

We have the following four cases: (1) $(s_{j1}^* < s_{j1}^1 (= lb_{j1})$; (2) $s_{j1}^* > s_{j1}^M (= ub_{j1})$; (3) $s_{jn_j}^* < s_{jn_j}^M (= lb_{jn_j})$; and (4) $s_{jn_j}^* > s_{jn_j}^1 (= ub_{jn_j})$.

We analyze each of these cases next.

Case (1): $s_{j1}^* < s_{j1}^1 (= lb_{j1})$.

The geometric relationships among the sublots from 1 to ρ_{jk_1} in both Q^* and Q^1, give $s_{ju+1}^* = q_{jk_1} s_{ju}^*$ and $s_{ju+1}^1 = q_{j1} s_{ju}^1$, for $u \in [1, \rho_{jk_1} - 1]$. By the assumption $s_{j1}^* < s_{j1}^1$ and $q_{jk_1} \le q_{j1}$ (see (15.33)), we have $s_{ju}^* < s_{ju}^1$ for $u \in [1, \rho_{jk_1}]$. For sublot $\rho_{jk_1} + 1$ in Q_j^* and Q_j^1, we have $s_{j\rho_{jk_1}+1}^* \le q_{jk_1} s_{j\rho_{jk_1}}^*$ by (15.21) and $s_{j\rho_{jk_1}+1}^1 = q_{j_1} s_{j\rho_{jk_1}}^1$ by (15.31), respectively. By the fact that $s_{j\rho_{j_{k_1}}}^* < s_{j\rho_{j_{k_1}}}^1$ and $q_{jk_1} \le q_{j1}$, we have $s_{j\rho_{jk_1}+1}^* < s_{j\rho_{jk_1}+1}^1$. Hence, we have $s_{ju}^* < s_{ju}^1$, for sublot u, $\forall u \in [1, \rho_{jk_1} + 1]$. We can use similar arguments to show $s_{ju}^* < s_{ju}^1$ for a sublot u in ranges $[\rho_{jk_1} + 1, \rho_{jk_2} + 1], [\rho_{jk_2} + 1, \rho_{jk_3} + 1], ..., [\rho_{jk_{r-1}} + 1, \rho_{jk_r}]$, sequentially. This leads to $\sum_{u=1}^{n_j} s_{ju}^* < U_j$,which contradicts the feasibility of Q_j^*. □

Case (2): $s^*_{j1} > s^M_{j1}(= ub_{j1})$.
The geometric relationships for sublots from 1 to ρ_{jk_1} in both Q^* and Q^M, yield $s^*_{ju+1} = q_{jk_1}s^*_{ju}$ and $s^M_{ju+1} = q_{jM}s^M_{ju}$, for $u \in [1, \rho_{jk_1} - 1]$. By the assumptions that $s^*_{j1} > s^M_{j1}$ and $q_{jk_1} \geq q_{jM}$ (see (15.33)), we have $s^*_{ju} > s^M_{ju}$, for $u \in [1, \rho_{jk_1}]$. For sublot $\rho_{jk_1} + 1$ in Q^*_j and Q^M_j, we have $s^*_{j\rho_{jk_1}+1} \geq q_{jk_2}s^*_{j\rho_{jk_1}}$ by (15.21) and $s^M_{j\rho_{jk_1}+1} = q_{jM}s^M_{j\rho_{jk_1}}$ by (15.32), respectively. By the fact that $s^*_{j\rho_{jk_1}} > s^M_{j\rho_{jk_1}}$ and $q_{jk_2} \geq q_{jM}$, we have $s^*_{j\rho_{jk_1}+1} > s^M_{j\rho_{jk_1}+1}$. Hence, we have $s^*_{ju} > s^M_{ju}$, for sublot $u, \forall u \in [1, \rho_{jk_1} + 1]$. We can use similar arguments to show $s^*_{ju} > s^M_{ju}$, for a sublot u in ranges $[\rho_{jk_1} + 1, \rho_{jk_2} + 1], [\rho_{jk_2} + 1, \rho_{jk_3} + 1], ..., [\rho_{jk_{r-1}} + 1, \rho_{jk_r}]$, sequentially. This leads to $\sum_{u=1}^{n_j} s^*_{ju} > U_j$, which contradicts the feasibility of Q^*_j.

Case (3): $s^*_{jn_j} < s^M_{jn_j}(= lb_{jn_j})$.
The geometric relationships for sublots from $\rho_{jk_{r-1}} + 1$ to $\rho_{jk_r}(= n_j)$ in both Q^* and Q^M, afford $s^*_{ju} = \frac{s^*_{ju+1}}{q_{jk_r}}$ and $s^M_{ju} = \frac{s^M_{ju+1}}{q_{jM}}$, for $u \in [\rho_{jk_{r-1}} + 1, \rho_{jk_r} - 1]$. By the assumption $s^*_{jn_j} < s^M_{jn_j}$ and $q_{jk_r} \geq q_{jM}$ (see (15.33)), we have $s^*_{ju} < s^M_{ju}$, for $u \in [\rho_{jk_{r-1}} + 1, \rho_{jk_r}]$. For sublot $\rho_{jk_r-1_r}$ in Q^*_j and Q^M_j, we have $s^*_{j\rho_{jk_{r-1}}} \leq \frac{s^*_{j\rho_{jk_{r-1}}+1}}{q_{jk_r}}$ by (15.21) and $s^M_{j\rho_{jk_{r-1}}} = \frac{s^M_{j\rho_{jk_{r-1}}+1}}{q_{jM}}$ by (15.31), respectively. By the fact that $s^*_{j\rho_{jk_r-1}+1} < s^M_{j\rho_{jk_{r-1}}+1}$ and $q_{jk_r} \geq q_{jM}$, we have $s^*_{j\rho_{jk_{r-1}}} < s^M_{j\rho_{jk_{r-1}}}$. Hence, we have $s^*_{ju} < s^M_{ju}$ for sublot u, $\forall u \in [\rho_{jk_{r-1}}, \rho_{jk_r}]$. We can use similar arguments to show $s^*_{ju} < s^M_{ju}$ for a sublot u in ranges $[1, \rho_{jk_1}], [\rho_{jk_1}, \rho_{jk_2}], ..., [\rho_{jk_{r-2}}, \rho_{jk_{r-1}}]$, sequentially, in the reverse order. This leads to $\sum_{u=1}^{n_j} s^*_{ju} < U_j$, which contradicts the feasibility of Q^*_j.

Case (4): $s^*_{jn_j} > s^1_{jn_j}(= ub_{jn_j})$.
The geometric relationships for sublots from $\rho_{jk_{r-1}} + 1$ to $\rho_{jk_r}(= n_j)$ in both Q^* and Q^1, yield $s^*_{ju} = \frac{s^*_{ju+1}}{q_{jk_r}}$ and $s^1_{ju} = \frac{s^1_{ju+1}}{q_{j1}}$, for $u \in [\rho_{jk_{r-1}} + 1, \rho_{jk_r} - 1]$. By the assumption $s^*_{jn_j} > s^1_{jn_j}$ and $q_{jk_r} \leq q_{j1}$ (see (15.33)), we have $s^*_{ju} > s^1_{ju}$, for $u \in [\rho_{jk_{r-1}} + 1, \rho_{jk_r}]$. For sublot ρ_{jk_r-1} in Q^*_j and Q^1_j, we have $s^*_{j\rho_{jk_{r-1}}} \geq \frac{s^*_{j\rho_{jk_{r-1}}+1}}{q_{jk_{r-1}}}$ by (15.21) and $s^1_{j\rho_{jk_{r-1}}} = \frac{s^1_{j\rho_{jk_{r-1}}+1}}{q_{j1}}$ by (15.31), respectively. By the fact that $s^*_{j\rho_{jkr_{r-1}}+1} > s^M_{j\rho_{jk_{r-1}}+1}$ and $q_{jk_r} - 1 \leq q_{j1}$, we have $s^*_{j\rho_{jk_{r-1}}} > s^1_{j\rho_{jk_{r-1}}}$. Hence, we have $s^*_{ju} > s^1_{ju}$, for sublot u, $\forall u \in [\rho_{jk_{r-1}}, \rho_{jk_r}]$. We can use similar arguments to show $s^*_{ju} > s^1_{ju}$ for a sublot u in ranges $[1, \rho_{jk_1}], [\rho_{jk_1}, \rho_{jk_2}], ..., [\rho_{jk_{r-2}}, \rho_{jk_{r-1}}]$, sequentially, in the reverse order. This leads to $\sum_{u=1}^{n_j} s^*_{ju} > U_j$, which contradicts the feasibility of Q^*_j. □

Dominance Rules (DR)

Proposition 3. (DR1) Let τ_i and $\hat{\tau}_i$ be two partial solutions up to position i. If $\hat{\tau}_i = \tau_i$, and their corresponding completion times on the assembly machine A are such that $C_A(\tau_i) \leq C_A(\hat{\tau}_i)$, then there exists an optimal solution schedule which does not start with $\hat{\tau}_i$.

Proof: Since $C_A(\tau_i) \leq C_A(\hat{\tau}_i)$, and the same set of lots is contained in τ_i and $\hat{\tau}_i$, the completion of τ_i can never be worse than that of $\hat{\tau}_i$. □

Proposition 4. (DR2) *Given completion times* $C_k(\tau_i)$ and $C_A(\tau_i)$ *of* τ_i, *if there exists a lot* $f \in \tau_i'$ *such that*

$$t_{fA} + p_{fA}U_f - (t_{fk} + p_{fk}U_f) \geq \max_{j \in \tau'_1 - \{f\}} \left\{ p_{jk}(ub_{j1} - lb_{j1}) \right\}, \forall k = 1, ..., M, \tag{15.34}$$

and

$$C_k(\tau_i) + t_{fk} + p_{fk}ub_{f1} + p_{fA}U_f + \sum_{j \in \tau'_1 - \{f\}} (t_{jA} + p_{jA}U_j) \leq LB(\tau_i), \forall k = 1, ..., M, \tag{15.35}$$

then there exists an optimal schedule in which lot f is sequenced in position i + 1.

Proof: We prove this result also by contradiction. Suppose there exists an optimal solution π^* in which lot f is sequenced in position f', where $f' > i + 1$. Let π be a solution obtained by moving lot f from position f' to position $i + 1$. For π, the makespan can be represented by

$$\begin{aligned} M(\pi) = \max \Bigg\{ & C_k(\tau_i) + \sum_{j=i+1}^{c-1} \left(t_{\pi(j)k} + p_{\pi(j)k}U_{\pi(j)}\right) \\ & + \left(t_{\pi(c)k} + p_{\pi(c)k}s_{\pi(c)1} + p_{\pi(c)A}U_{\pi(c)}\right) \\ & + \sum_{j=c+1}^{N} \left(t_{\pi(j)A} + p_{\pi(j)A}U_{\pi(j)}\right), C_A(\tau_i) + \sum_{j=i+1}^{N} \left(t_{\pi(j)A} + p_{\pi(j)A}U_{\pi(j)}\right) \Bigg\} \end{aligned} \tag{15.36}$$

for some $c (c = i + 1, ..., N)$ and $k (k = 1, ..., M)$. For the position c, we have the following cases: (1) c does not exist from $i+1$ to N, (2) $c = i+1$, (3) $i+2 \leq c \leq f'$, and (4) $c \geq f' + 1$. We consider each of these cases next.

Case (1): c does not exist from $i + 1$ to N.
For this case, the makespan can be represented by

$$M(\pi) = C_A(\tau_i) + \sum_{j=i+1}^{N} (t_{\pi(j)A} + p_{\pi(j)A}U_{\pi(j)}). \tag{15.37}$$

Then π is an optimal solution because it is equal to $LB^1(\tau_i)$ in (15.8).

Case (2): $c = i + 1$.
Since sublot f occupies position $i+1$ in π, it is a critical sublot. We have

$$M(\pi) = C_k(\tau_i) + t_{fk} + p_{fk}s_{f1} + p_{fA}U_f + \sum_{j=i+2}^{N} \left(t_{\pi(j)A} + p_{\pi(j)A}U_{\pi(j)}\right)$$

$$\leq C_k(\tau_i) + t_{fk} + p_{fk}ub_{f1} + p_{fA}U_f + \sum_{j=i+2}^{N} \left(t_{\pi(j)A} + p_{\pi(j)A}U_{\pi(j)}\right).$$

In view of (15.35), we have $M(\pi) \leq LB(\tau_i)$, which indicates that π is an optimal solution.

Case (3): $i+2 \leq c \leq f'$.
Based on (15.2) and (15.4), for solution π^*, we have

$$\begin{aligned} M(\pi^*) \geq C_k(\tau_i) + \sum_{j=i+1}^{c-2} \left(t_{\pi^*(j)k} + p_{\pi^*(j)k} U_{\pi^*(j)}\right) \\ + \left(t_{\pi^*(c-1)k} + p_{\pi^*(c-1)k} s^*_{\pi^*(c-1)1} + p_{\pi^*(c-1)A} U_{\pi^*(c-1)}\right) \\ + \sum_{j=c}^{N} \left(t_{\pi^*(j)A} p_{\pi^*(j)A} U_{\pi^*(j)}\right). \end{aligned} \tag{15.38}$$

Since f is in position $i+1$ in π, and it is a position $f' \geq c$ in π^*, and the lots in position $i+1$ to $c-2$ in π^* are identical to those in positions $i+2$ to $c-1$ in π, we have

$$\sum_{j=i+1}^{c-2} \left(t_{\pi^*(j)k} + p_{\pi^*(j)k} U_{\pi^*(j)}\right) = \sum_{j=i+1}^{c-1} \left(t_{\pi(j)k} + p_{\pi(j)k} U_{\pi(j)}\right) - \left(t_{fk} + p_{fk} U_f\right). \tag{15.39}$$

Furthermore, for the lots in position c to N in π^* and lots in $c+1$ to N in π, we have

$$\sum_{j=c}^{N} \left(t_{\pi^*(j)A} + p_{\pi^*(j)A} U_{\pi^*(j)}\right) = \sum_{j=c+1}^{N} \left(t_{\pi(j)A} + p_{\pi(j)A} U_{\pi(j)}\right) + \left(t_{fA} + p_{fA} U_f\right). \tag{15.40}$$

Note that $\pi^*(c-1) = \pi(c)$, which results in,

$$t_{\pi^*(c-1)k} + p_{\pi^*(c-1)A} U_{\pi^*(c-1)} = t_{\pi(c)k} + p_{\pi(c)A} U_{\pi(c)}. \tag{15.41}$$

By substituting (15.39), (15.40), and (15.41) into (15.38), we have

$$\begin{aligned} M(\pi^*) \geq C_k(\tau_i) + \sum_{j=i+1}^{c-1} \left(t_{\pi(j)k} + p_{\pi(j)k} U_{\pi(j)}\right) - \left(t_{fk} + p_{fk} U_f\right) \\ + \left(t_{\pi(c)k} + p_{\pi(c)k} s^*_{\pi^*(c-1)1} + p_{\pi(c)A} U_{\pi(c)}\right) + \sum_{j=c+1}^{N} \left(t_{\pi(j)A} + p_{\pi(j)A} U_{\pi(j)}\right) \\ + \left(t_{fA} + p_{fA} U_f\right) \geq C_k(\tau_i) + \sum_{j=i+1}^{c-1} \left(t_{\pi(j)k} + p_{\pi(j)k} U_{\pi(j)}\right) \end{aligned}$$

$$+ (t_{\pi(c)k} + p_{\pi(c)A}U_{\pi(c)}) + \sum_{j=c+1}^{N} \left(t_{\pi(j)A} + p_{\pi(j)A}U_{\pi(j)}\right) + \left(t_{fk} + p_{fk}U_f\right)$$

$$- \left(t_{fA} + p_{fA}U_f\right) + p_{\pi(c)A}lb_{\pi^*(c-1)1}. \tag{15.42}$$

As $lb_{\pi^*(c-1)1} = lb_{\pi(c)1}$ because $\pi^*(c-1) = \pi(c)$ (as noted above), and using (15.34), we have

$$M(\pi^*) \geq C_k(\tau_i) + \sum_{j=i+1}^{c-1} \left(t_{\pi(j)k} + p_{\pi(j)k}U_{\pi(j)}\right)$$

$$+ \left(t_{\pi(c)k} + p_{\pi(c)k}ub_{\pi(c)1} + p_{\pi(c)A}U_{\pi(c)}\right) + \sum_{j=c+1}^{N} \left(t_{\pi(j)A} + p_{\pi(j)A}U_{\pi(j)}\right).$$

In view of (15.36), we have $M(\pi) \leq M(\pi^*)$, which indicates that solution π is at least as good as π^*.

Case (4): $c \geq f' + 1$.
In this case, the movement of lot f from position $i+1$ to position f' can only create a chance for idle time to increase before position c on machine A. Therefore, we have $M(\pi^*) \geq M(\pi)$, which again indicates that solution π is at least as good as π^*. □

Proposition 5. (DR3) *Given machine availability times* $C_k(\tau_i), k = 1, ..., M$ and $C_A(\tau_i)$, *if there exists a lot f such that*

$$t_{fk} - t_{fA} + p_{fk}U_f - p_{fA}U_f \geq \max_{j \in \tau'_1 - \{f\}} \left\{p_{jA}(ub_{jn_j} - lb_{jn_j})\right\}, \quad \forall k = 1, ..., M, \tag{15.43}$$

and

$$C_k(\tau_i) + \sum_{j \in \tau'_1} (t_{jk} + p_{jk}U_j) + p_{fA}ub_{fn_f} \leq LB(\tau_i), \quad \forall k = 1, ..., M, \tag{15.44}$$

then there exists an optimal schedule in which lot f is sequenced last.
Proof: This result can also be shown by contradiction. Suppose there exists an optimal solution π^* in which lot f is sequenced in position f', where $f' < N$. Let π be a solution obtained by moving lot f from position f' to the last position N. For π, the makespan can be represented by

$$M(\pi) = \max \Bigg\{ C_k(\tau_i) + \sum_{j=i+1}^{c-1} \left(t_{\pi(j)k} + p_{\pi(j)k}U_{\pi(j)}\right)$$

$$+ \left(t_{\pi(c)k} + p_{\pi(c)k}U_{\pi(c)} + p_{\pi(c)A}s_{\pi(c)n_{\pi(c)}}\right) \tag{15.45}$$

$$+ \sum_{j=c+1}^{N} \left(t_{\pi(j)A} + p_{\pi(j)A}U_{\pi(j)}\right), C_A(\tau_i) + \sum_{j=i+1}^{N} \left(t_{\pi(j)A} + p_{\pi(j)A}U_{\pi(j)}\right) \Bigg\},$$

for some c $(c = i+1, ..., N)$ and $k (k = 1, ..., M)$. We have the following cases: (1) c does not exist from $i+1$ to N, (2) $c = N$, (3) $f' \leq c \leq N-1$, and (4) $c \leq f' - 1$. We consider each of these cases next.

Case (1): c does not exist from $i+1$ to N.
For this case, the makespan can be represented by

$$M(\pi) = C_A(\tau_i) + \sum_{j=i+1}^{N} \left(t_{\pi(j)A} + p_{\pi(j)A} U_{\pi(j)} \right).$$

Then π is an optimal solution because the makespan is equal to $LB^1(\tau_i)$.

Case (2): $c = N$.
The sublot f in position c is critical. We have

$$\begin{aligned} M(\pi) &= C_k(\tau_i) + \sum_{j=i+1}^{N-1} \left(t_{\pi(j)k} + p_{\pi(j)k} U_{\pi(j)} \right) + \left(t_{fk} + p_{fk} U_f + p_{fA} s_{fn_f} \right) \\ &= C_k(\tau_i) + \sum_{j=i+1}^{N} \left(t_{\pi(j)k} + p_{\pi(j),k} U_{\pi(j)} \right) + \left(p_{fA} s_{fn_f} \right) \\ &\le C_k(\tau_i) + \sum_{j \in \tau'} \left(t_{\pi(j)k} + p_{\pi(j)k} U_{\pi(j)} \right) + \left(p_{fA} ub_{fn_f} \right). \end{aligned}$$

In view of (15.44), we have $M(\pi) \le LB(\tau_i)$, which indicates that π is an optimal solution.

Case (3): $f' \le c \le N-1$.
Based on (15.2) and (15.4), for solution π^*, we have

$$\begin{aligned} M(\pi^*) \ge C_k(\tau_i) &+ \sum_{j=i+1}^{c} \left(\tau_{\pi^*(j)k} + p_{\pi^*(j)k} U_{\pi^*(j)} \right) \\ &+ \left(t_{\pi^*(c+1)k} + p_{\pi^*(c+1)k} U_{\pi^*(c+1)} + p_{\pi^*(c+1)A} s^*_{\pi^*(c+1)n_{\pi^*(c+1)}} \right) \\ &+ \sum_{j=c+2}^{N} \left(t_{\pi^*(j)A} + p_{\pi^*(j)A} U_{\pi^*(j)} \right). \end{aligned} \tag{15.46}$$

Since f is in position N in π, and it is in position $f' \le c$ in π^*, and the lots in positions $c+2$ to N in π^* are identical to those in positions $c+1$ to $N-1$ in π, we have

$$\sum_{j=c+2}^{N} \left(t_{\pi^*(j)A} + p_{\pi^*(j)A} U_{\pi^*(j)} \right) = \sum_{j=c+1}^{N} \left(t_{\pi(j)A} + p_{\pi(j)A} U_{\pi(j)} \right) - \left(t_{fA} + p_{fA} U_f \right). \tag{15.47}$$

Furthermore, for the lots in positions $i+1$ to c in π^* and in positions $i+1$ to $c-1$ in π, we have

$$\sum_{j=i+1}^{c} \left(t_{\pi^*(j)k} + p_{\pi^*(j)k} U_{\pi^*(j)} \right) = \sum_{j=i+1}^{c-1} \left(t_{\pi(j)k} + p_{\pi(j)k} U_{\pi(j)} \right) + \left(t_{fk} + p_{fk} U_f \right). \tag{15.48}$$

Note that $\pi^*(c+1) = \pi(c)$, which results in,

$$t_{\pi^*(c+1),k} + p_{\pi^*(c+1),k} U_{\pi^*(c+1)} = t_{\pi(c),k} + p_{\pi(c),k} U_{\pi(c)}. \tag{15.49}$$

By substituting (15.47), (15.48), and (15.49) into (15.46), we have

$$\begin{aligned} M(\pi^*) \geq C_k(\tau_i) &+ \sum_{j=i+1}^{c-1} \left(t_{\pi(j)k} + p_{\pi(j)k} U_{\pi(j)}\right) - \left(t_{fk} + p_{fk} U_f\right) \\ &+ \left(t_{\pi(c)k} + p_{\pi(c)k} U_{\pi(c)+P_{\pi(c)A}} s^*_{\pi^*(c+1)n_{\pi^*(c+1)}}\right) \\ &+ \sum_{j=c+1}^{N} \left(t_{\pi(j)A} + p_{\pi(j)A} U_{\pi(j)}\right) - \left(t_{fA} + p_{fA} U_f\right) \geq C_k(\tau_i) \\ &+ \sum_{j=i+1}^{c-1} \left(t_{\pi(j)k} + p_{\pi(j)k} U_{\pi(j)}\right) + \left(t_{\pi(c)k} + p_{\pi(c)k} U_{\pi(c)}\right) \\ &+ \sum_{j=c+1}^{N} \left(t_{\pi(j)A} + p_{\pi(j)A} U_{\pi(j)}\right) \\ &+ \left(t_{fk} + p_{fk} U_f\right) - \left(t_{fA} + p_{fA} U_f\right) + p_{\pi(c)A} lb_{\pi^*(c+1)n_{\pi^*(c+1)}}. \end{aligned}$$

As $lb_{\pi^*(c+1)n_{\pi^*(c+1)}} = lb_{\pi(c)n_{\pi(c)}}$ because $\pi^*(c+1) = \pi(c)$ (as noted above), and using (15.43), we have

$$\begin{aligned} M\left(\pi^*\right) \geq C_k\left(\tau_i\right) &+ \sum_{j=i+1}^{c-1} \left(t_{\pi(j)k} + p_{\pi(j)k} U_{\pi(j)}\right) \\ &+ \left(t_{\pi(c)k} + p_{\pi(c)k} U_{\pi(c)} + p_{\pi(c)A} ub_{\pi(c)n_{\pi(c)}}\right) + \sum_{j=c+1}^{N} \left(t_{\pi(j)A} + p_{\pi(j)A} U_{\pi(j)}\right). \end{aligned}$$

In view of (15.45), we have $M(\pi^*) \geq M(\pi)$, which indicates that solution π is at least as good as π^*.

Case (4): $c \leq f' - 1$.

In this case, the movement of lot f from position N to position f' can only create a chance for idle time to increase after position c on machine A. Therefore, we have $M(\pi^*) \geq M(\pi)$, implying that solution π is at least as good as π^*. □

Proposition 6. (DR4) *Given machine availability times* $C_k(\tau_i), k = 1, ..., M$ and $C_A(\tau_i)$, *if there exist two lots f and g such that*

$$t_{fA} - t_{fk} + p_{fA} U_f - p_{fk} U_f \geq \max_{j \in \tau'_1 - \{f\}} \left\{p_{jk}(ub_{j1} - lb_{j1})\right\}, \quad \forall k = 1, ..., M, \tag{15.50}$$

and

$$C_k\left(\tau_i\right) + t_{fk} + p_{fk} ub_{f1} - t_{fA} \leq \max\left\{C_k\left(\tau_i\right) + t_{gk} + p_{gk} lb_{g1} - t_{gA}, C_A\left(\tau_i\right)\right\}, \quad \forall k = 1, ..., M, \tag{15.51}$$

then there exists an optimal schedule in which lot g is not sequenced in position i + 1.

Proof: Again, we use contradiction to prove this result. Suppose that π^* is an optimal solution in which lot g is sequenced in position $i+1$ and lot f is sequenced in position f', where $f' \geq i+2$. Let π be obtained from π^* by moving lot f from position f' to position $i+1$. The makespan for π can be represented by

$$\begin{aligned} M(\pi) = \max \Bigg\{ & C_k(\tau_i) + \sum_{j=i+1}^{c-1} \left(t_{\pi(j)k} + p_{\pi(j)k} U_{\pi(j)}\right) \\ & + \left(t_{\pi(c)k} + p_{\pi(c)k} s_{\pi(c)1} + p_{\pi(c)A} U_{\pi(c)}\right) \\ & + \sum_{j=c+1}^{N} \left(t_{\pi(j)A} + p_{\pi(j)A} U_{\pi(j)}\right),\ C_A(\tau_i) + \sum_{j=i+1}^{N} \left(t_{\pi(j)A} + p_{\pi(j)A} U_{\pi(j)}\right) \Bigg\}. \end{aligned} \tag{15.52}$$

□

If $M(\pi) = C_A(\tau_i) + \sum_{j=i+1}^{N} \left(t_{\pi(j)A} + p_{\pi(j)A} U_{\pi}(j)\right)$, then π is an optimal solution because it is equal to $LB^1(\tau)$. Otherwise, we have the following cases: (1) $c = i+1$, (2) $c = i+2$, (3) $f' \geq c \geq i+3$, and (4) $c \geq f'+1$. We consider each of these cases next.

Case (1): $c = i+1$.
The sublot f at position c is critical, and the makespan for π can be rewritten as

$$\begin{aligned} M(\pi) &= C_k(\tau_i) + t_{fk} + p_{fk} s_{f1} + p_{fA} U_f + \sum_{j=i+2}^{N} \left(t_{\pi(j)A} + p_{\pi(j)A} U_{\pi(j)}\right), \\ &= C_k(\tau_i) + t_{fk} + p_{fk} s_{f1} - t_{fA} + \sum_{j=i+1}^{N} \left(t_{\pi(j)A} + p_{\pi(j)A} U_{\pi(j)}\right). \end{aligned}$$

For solution π^*, we have

$$\begin{aligned} M(\pi^*) &\geq \max \Bigg\{ C_k(\tau_i) + t_{gk} + p_{gk} s^*_{g1} + p_{gA} U_g + \sum_{j=i+1}^{N} \left(t_{\pi^*(j)A} + p_{\pi^*(j)A} U_{\pi^*(j)}\right), \\ &\qquad C_A(\tau_i) + \sum_{j=i+1}^{N} \left(t_{\pi^*(j)A} + p_{\pi^*(j)A} U_{\pi^*(j)}\right) \Bigg\} \\ &= \max \left\{ C_k(\tau_i) + t_{gk} + p_{gk} s^*_{g1} + p_{gA} U_g - t_{gA} - p_{gA} U_g,\ \ C_A(\tau_i) \right\} \\ &\qquad + \sum_{j=i+1}^{N} \left(t_{\pi(j)A} + p_{\pi(j)A} U_{\pi(j)}\right) \\ &= \max \left\{ C_k(\tau_i) + t_{gk} + p_{gk} s^*_{g1} - t_{gA},\ \ C_A(\tau_i) \right\} + \sum_{j=i+1}^{N} \left(t_{\pi(j)A} + p_{\pi(j)A} U_{\pi(j)}\right) \\ &\geq \max \left\{ C_k(\tau_i) + t_{gk} + p_{gk} lb_{g1} - t_{gA},\ \ C_A(\tau_i) \right\} + \sum_{j=i+1}^{N} \left(t_{\pi(j)A} + p_{\pi(j)A} U_{\pi(j)}\right). \end{aligned}$$

In view of (15.51), we have $M(\pi^*) \geq M(\pi)$, which indicates that solution π is at least as good as π^*.

Case (2): $c = i + 2$.
The makespan for π can be rewritten as

$$M(\pi) = C_k(\tau_i) + t_{fk} + p_{fk}U_f + t_{gk} + p_{gk}s_{g1} + p_{gA}U_g + \sum_{j=i+3}^{N} \left(t_{\pi(j)A} + p_{\pi(j)A}U_{\pi(j)}\right)$$

$$\begin{aligned}
&= C_k(\tau_i) + t_{fk} - t_{fA} + t_{gk} - t_{gA} + (p_{fk} - p_{fA})U_f \\
&\quad + p_{gk}s_{g1} + \sum_{j=i+1}^{N} \left(t_{\pi(j)A} + p_{\pi(j)A}U_{\pi(j)}\right) \\
&\quad \le C_k(\tau_i) + t_{fk} - t_{fA} + t_{gk} - t_{gA} \\
&\quad + (p_{fk} - p_{fA})U_f + p_{gk}ub_{g1} + \sum_{j=i+1}^{N} \left(t_{\pi(j)A} + p_{\pi(j)A}U_{\pi(j)}\right).
\end{aligned}$$

Based on (15.50), we have

$$M(\pi) \le C_k(\tau_i) + t_{gk} + p_{gk}lb_{g1} - t_{gA} + \sum_{j=i+1}^{N} \left(t_{\pi(j)A} + p_{\pi(j)A}U_{\pi(j)}\right).$$

For solution π^*,

$$\begin{aligned}
M(\pi^*) &\ge C_k(\tau_i) + t_{gk} + p_{gk}s^*_{g1} + p_{gA}U_g + \sum_{j=i+2}^{N} \left(t_{\pi^*(j)A} + p_{\pi^*(j)A}U_{\pi^*(j)}\right) \\
&\ge C_k(\tau_i) + t_{gk} + p_{gk}lb_{g1} - t_{gA} + \sum_{j=i+1}^{N} \left(t_{\pi^*(j)A} + p_{\pi^*(j)A}U_{\pi^*(j)}\right).
\end{aligned} \tag{15.53}$$

This leads to $M(\pi^*) \ge M(\pi)$, which indicates that solution π is at least as good as π^*.

Case (3): $f' \ge c \ge i + 3$.
Based on (15.2) and (15.4), for solution π^*, we have

$$\begin{aligned}
M(\pi^*) \ge C_k(\tau_i) &+ \sum_{j=i+1}^{c-2} \left(t_{\pi^*(j)k} + p_{\pi^*(j)k}U_{\pi^*(j)}\right) \\
&+ \left(t_{\pi^*(c-1)k} + p_{\pi^*(c-1)k}s^*_{\pi^*(c-1)1} + p_{\pi^*(c-1)A}U_{\pi^*(c-1)}\right) \\
&+ \sum_{j=c}^{N} \left(t_{\pi^*(j)A} + p_{\pi^*(j)A}U_{\pi^*(j)}\right).
\end{aligned} \tag{15.54}$$

Since f is in position $i+1$ in π, and it is in a position $f' \geq c$, and the lots in positions $i+1$ to $c-2$ in π^* are identical to those in positions $i+2$ to $c-1$, we have

$$\sum_{j=i+1}^{c-2} \left(t_{\pi^*(j)k} + p_{\pi^*(j)k} U_{\pi^*(j)}\right) = \sum_{j=i+1}^{c-1} \left(t_{\pi(j)k} + p_{\pi(j)k} U_{\pi(j)}\right) - \left(t_{fk} + p_{fk} U_f\right). \tag{15.55}$$

Furthermore, for the lots in positions c to N in π^* and $c+1$ to N in π, we have

$$\sum_{j=c}^{N} \left(t_{\pi^*(j)A} + p_{\pi^*(j)A} U_{\pi^*(j)}\right) = \sum_{j=c+1}^{N} \left(t_{\pi(j)A} + p_{\pi(j)A} U_{\pi(j)}\right) + \left(t_{fA} + p_{fA} U_f\right). \tag{15.56}$$

Note that $\pi^*(c-1) = \pi(c)$, which results in

$$t_{\pi^*(c-1)k} + p_{\pi^*(c-1)A} U_{\pi^*(c-1)} = t_{\pi(c)k} + p_{\pi(c)A} U_{\pi(c).} \tag{15.57}$$

By substituting (15.56), (15.57), and (15.58) into (15.55), we have

$$\begin{aligned} M(\pi^*) \geq C_k(\tau_i) &+ \sum_{j=i+1}^{c-1} \left(t_{\pi(j)k} + p_{\pi(j)k} U_{\pi(j)}\right) - \left(t_{fk} + p_{fk} U_f\right) \\ &+ \left(t_{\pi(c)k} + p_{\pi(c)k} s^*_{\pi^*(c-1)1} + p_{\pi(c)A} U_{\pi(c)}\right) \\ &+ \sum_{j=c+1}^{N} \left(t_{\pi(j)A} + p_{\pi(j),A} U_{\pi(j)}\right) + \left(t_{fA} + p_{fA} U_f\right) \geq C_k(\tau_i) \\ &+ \sum_{j=i+1}^{c-1} \left(t_{\pi(j)k} + p_{\pi(j)k} U_{\pi(j)}\right) + \left(t_{\pi(c)k} + p_{\pi(c)A} U_{\pi(c)}\right) \\ &+ \sum_{j=c+1}^{N} \left(t_{\pi(j)A} + p_{\pi(j)A} U_{\pi(j)}\right) + \left(t_{fk} + p_{fk} U_f\right) - \left(t_{fA} + p_{fA} U_f\right) \\ &+ p_{\pi(c)A} lb_{\pi^*(c-1)1}. \end{aligned} \tag{15.58}$$

As $lb_{\pi^*(c-1)1} = lb_{\pi(c)1}$ because $\pi^*(c-1) = \pi(c)$ (as noted above), using (15.50), we have

$$\begin{aligned} M(\pi^*) \geq C_k(\tau_i) &+ \sum_{j=i+1}^{c-1} \left(t_{\pi(j)k} + p_{\pi(j)k} U_{\pi(j)}\right) \\ &+ \left(t_{\pi(c)k} + p_{\pi(c)k} ub_{\pi(c)1} + p_{\pi(c)A} U_{\pi(c)}\right) + \sum_{j=c+1}^{N} \left(t_{\pi(j)A} + p_{\pi(j)A} U_{\pi(j)}\right). \end{aligned}$$

In view of (15.52), we have $M(\pi) \leq M(\pi^*)$, which indicates that solution π is at least as good as π^*.

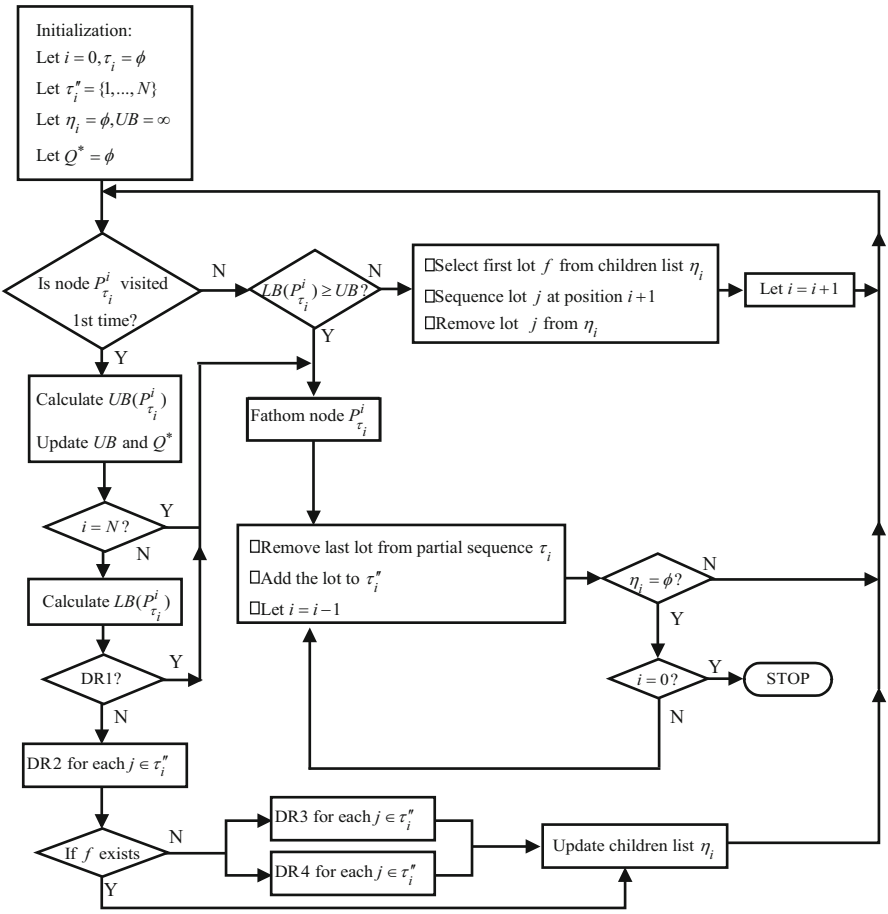


Fig. 15.4 Flowchart for the proposed branch-and-bound method

Case (4): $c \geq f' + 1$.
In this case, the movement of lot f from position $i+1$ to position f' can only create a chance for idle time to increase before position c on machine A. Therefore, we have $M(\pi^*) \geq M(\pi)$, which again indicates that solution π is at least as good as π^*. □

Branch-and-Bound-based Algorithm

A flowchart of the proposed branch-and-bound algorithm (ML-TSALSP-BB) is depicted in Fig. 15.4. The lower and upper bounds and the dominance rules developed above appropriately applied as shown. We use the depth-first branching method. At a node (a partial sequence τ_i) dominance rule DR1 is first applied to determine whether

or not to fathom that node. If the node is not fathomed, the proposed lower bounds are calculated and compared with the best incumbent objective value to further determine whether to fathom the current node. If the node is still active, we apply (1) dominance rule DR2 to determine whether to fix an appropriate lot at position $i + 1$, (2) dominance rule DR3 to determine whether to fix a lot at the last position, and hence, to eliminate further consideration of that lot in other positions, and (3) dominance rule DR4 to determine whether or not to eliminate a lot to be sequenced in position $i + 1$. If the current node is fathomed, we backtrack to continue branching. The upper bound of the objective value is updated once a better incumbent solution has been found.

Computational Experimentation

In this section, we present results of our numerical experimentation conducted to study the computational effectiveness of the mixed integer formulation (TSA-MSLP-M) and the proposed branch-and-bound method ML-TSALSP-BB, which was coded in C#. All the runs were made on an Intel Xeon 3.6 GHz computer. CPLEX Solver (version 10.1) was used for solving the sublot-sizing subproblem at each node of ML-TSALSP-BB and for the direct solution of ML-TSALSP-M.

Performance of ML-TSALSP-BB

First, we present results of our computational experimentation to demonstrate the effectiveness of the proposed method. In particular, our aim is to test the efficacy of using various dominance rules (i.e., DR1, DR2, DR3, and DR4). Note that if inequalities (15.34), (15.43), and (15.50) hold, then the dominance rules, DR2, DR5, and DR4, respectively, are likely to be more effective. Therefore, we generated different combinations of the per-unit processing times on the assembly and subassembly machines to enable application of various combinations of inequalities (15.34), (15.43), and (15.50). Three problem sets were generated using uniform distributions for the number of lots (N), number of subassembly machines (M), number of items in a lot $j(U_j)$, number of sublots of a lot $j(n_j)$, and unit processing time for the items of lot j on subassembly machine k (p_{jk}) and the assembly machine (p_{jA}), as shown in Table 15.1.

For each combination of N and M, 20 problem instances were generated randomly by using the uniform distributions shown in Table 15.1. Note that the three problem sets differ due to different ranges of values for p_{jA} in relation to that for p_{jk}. In Set 1, the average value of p_{jA} is less than the average of p_{jk}. They are the same in Set 2, while in Set 3, the average value of p_{jA} is greater than that of p_{jk}. Also, for all problem sets, the same uniform distributions were used to generate values of U_j, n_j and p_{jk}. Moreover, to clearly determine the impact of the inequalities (15.34), (15.43), and

Table 15.1 Sets of Problem Instances

Problem set	N	M	U_j	n_j	p_{jk}	p_{jA}
1	(20, 50, 100)	(3, 6, 9)	$U(10, 100)$	$U(1, 10)$	$U(50, 100)$	$U(25, 75)$
2	(20, 50, 100)	(3, 6, 9)	$U(10, 100)$	$U(1, 10)$	$U(50, 100)$	$U(50, 100)$
3	(20, 50, 100)	(3, 6, 9)	$U(10, 100)$	$U(1, 10)$	$U(50, 100)$	$U(75, 125)$

(15.50), the lot-detached setup times at all the machines were set to zero. We used a central processing unit (CPU) time limit of 500 s.

The results are depicted in Table 15.2. For each problem instance, five combinations of dominance rules, namely, DR1, DR1 + DR2, DR1 + DR3, DR1 + DR4, and all DRs were tested for their performance. For each combination, information on four criteria, namely, average CPU time (ACT) in seconds, average number of nodes (ANN) explored before the algorithm stops, number of problem instances solved at the root node (NSR), and number of unsolved problems (NU) were gathered to evaluate relative performance.

Referring to Table 15.2, the following can be observed:

1. For each combination of dominance rules, the computational effort, generally, has an upward trend against the number of lots and number of subassembly machines.
2. Many problem instances are solved right at the root node due to the tightness of the lower and upper bounds used. Note that Problem Set 1 has the highest number of instances that were solved at the root node, followed by Problem Set 2 and then Problem Set 3.
3. Comparing the results presented in columns 5, 6, and 7 with those in Column 4, dominance rules DR2, DR3, and DR4 are more effective when applied to instances of Problem Set 3, which is indicated by a greater reduction in the average CPU time (ACT) required and the number of nodes generated than those for Problem Sets 1 and 2. This reduction is mixed for instances of Problem Set 2 due to identical average values of p_{jk} and p_{jA}; while it is the least for Problem Set 1.
4. For instances in Problem Set 3, DR2 is generally more effective than the other two dominance rules (DR3 and DR4).
5. When the number of subassembly machines increases, the effectiveness of dominance rules decreases as observed from the ACT and ANN values for instances of Problem Set 2. This is also as expected since an increment in number of subassembly machines decreases the possibility for inequalities (15.34), (15.43), and (15.50) to hold true.
6. No significant extra computational time is incurred because of the use of dominance rules in contrast to the savings achieved by reduction in nodes generated (see Problem Sets 1 and 2). Consequently, it is fairly efficient to use all dominance rules (note the values depicted in the last column of Table 15.2 in relation to those shown in the others).

Table 15.2 Computational result of ML-TSALSP-BB

			DR1				DR1+DR2				DR1+DR3				DR1+DR4				All DRs			
Set	N	M	ACT	ANN	NSR	NU	ACT	ANN	NSR	NU	ACT	ANN	NSR	NU	ACT	ANN	NSR	NU	ACT	ANN	NSR	NU
1	20	3	25.42	617	19	1	25.41	617	19	1	25.33	617	19	1	25.51	617	19	1	25.41	617	19	1
		6	50.86	620	18	2	50.86	616	18	2	50.86	617	18	2	50.86	626	18	2	50.85	626	18	2
		9	101.25	708	16	4	101.21	743	16	4	101.21	749	16	4	101.19	753	16	4	101.24	738	16	4
	50	3	51.05	1097	18	2	51.07	1075	18	2	51.08	1059	18	2	50.98	1087	18	2	51.06	1045	18	2
		6	52.2	406	18	2	52.33	431	18	2	52.18	439	18	2	52.19	443	18	2	52.19	429	18	2
		9	3.85	1	20	0	4.25	1	20	0	4.26	1	20	0	4.08	1	20	0	4.04	1	20	0
	100	3	27.05	410	19	1	26.98	415	19	1	26.97	415	19	1	26.97	415	19	1	27.05	361	19	1
		6	29.31	89	19	1	29.34	88	19	1	29.35	90	19	1	29.33	89	19	1	29.36	83	19	1
		9	56.93	11	18	2	56.98	11	18	2	57.31	11	18	2	56.97	11	18	2	56.79	11	18	2
2	20	3	143.77	1108	12	5	141.82	1131	12	5	66.86	600	12	2	141.8	1132	12	5	66.73	601	12	2
		6	194.18	573	9	6	194.29	630	9	6	184.58	604	9	6	191.94	573	9	6	188.39	545	9	6
		9	182.37	923	11	7	181.72	965	11	7	182.21	947	11	7	181.8	924	11	7	182.39	926	11	7
	50	3	77.82	134	12	1	76.92	135	12	1	25.16	24	12	0	61.57	111	12	1	25.84	24	12	0
		6	98.72	211	15	2	98.93	212	15	2	98.6	52	15	2	98.3	238	15	2	98.22	52	15	2
		9	55.59	18	17	1	55.62	18	17	1	55.7	18	17	1	55.65	18	17	1	55.6	18	17	1

Table 15.2 (continued)

			DR1				DR1+DR2				DR1+DR3				DR1+DR4				All DRs			
Set	*N*	*M*	ACT	ANN	NSR	NU	ACT	ANN	NSR	NU	ACT	ANN	NSR	NU	ACT	ANN	NSR	NU	ACT	ANN	NSR	NU
	100	3	123.3	62	11	2	123.35	62	11	2	90.77	45	11	0	101.42	51	11	1	90.75	45	11	0
		6	79	19	17	3	78.86	18	17	3	79.39	19	17	3	79.15	18	17	3	79.22	18	17	3
		9	81.64	13	17	3	81.85	13	17	3	81.81	13	17	3	81.85	13	17	3	81.87	13	17	3
3	20	3	6.93	16	6	0	2.79	7	7	0	4.84	11	6	0	4.59	11	6	0	2.22	5	6	0
		6	17.56	19	3	0	13.89	16	3	0	12.68	14	3	0	13.5	15	3	0	11.05	13	3	0
		9	35.18	24	2	0	32.34	22	2	0	22.89	16	4	0	30.13	21	2	0	21.85	15	3	0
	50	3	48.31	44	3	0	2.19	2	3	0	31.22	29	3	0	22.79	21	3	0	2.16	2	3	0
		6	113.07	49	1	0	77.65	34	1	0	93.21	41	1	0	75.18	33	1	0	66.71	30	1	0
		9	206.69	54	0	0	158.35	44	1	0	200.39	54	0	0	149.77	42	0	0	168.97	44	1	0
	100	3	205.1	92	2	0	15.8	8	2	0	143.69	57	2	0	80.47	31	3	0	12.46	6	2	0
		6	476.03	102	0	2	330.41	65	0	13	394.79	85	0	1	303.18	61	0	1	295.1	58	0	1
		9	504.67	71	0	20	456.28	58	0	18	478.41	64	1	19	475.68	64	0	10	431.07	54	1	17

Comparison of ML-TSALSP-BB and Direct Solution of ML-TSALSP-M

Next, we compare the computational effort required for implementing the ML-TSALSP-BB method with that for the direct solution of ML-TSALSP-M by CPLEX 10.1 (implemented with default settings). The data set used for this experimentation is shown in Table 15.3. It is identical to the data set presented in Table 15.1 except for the values of N. We also used a time limit of 500 s.

The results are presented in Table 15.4. We compare the performance of the two methods with respect to the following criteria: Average Gap at root node (AGR), ACT, ANN, NSR, and NU. The instances of Problem Set 6 require the least computational effort for both ML-TSALSP-M and ML-TSALSP-BB, while the instances of Problem Set 5 require the largest computational effort for every criterion listed. Table 15.4 also reveals that ML-TSALSP-BB, generally, requires much less computational effort to solve the same instances than that required by the direct solution of ML-TSALSP-M with respect to all criteria. The ML-TSALSP-BB method not only requires less CPU time and number of nodes explored, it also solves more problem instances at the root node and leaves fewer instances unsolved within the prespecified time limit of 500 s. This is also indicated by their AGR values. Therefore, the ML-TSALSP-BB method outperforms the direct solution of the ML-TSALSP-M formulation by CPLEX 10.1.

Performance of ML-TSALSP-BB when Applied to Large-sized Problem Instances

In this section, we present computational results regarding the performance of the ML-TSALSP-BB method on large problem instances involving 300–1,000 lots. The rest of data is the same as shown in Table 15.3. We designate these as Problem Sets 7, 8, and 9. Also, the CPU time limit used is 500 s.

The results are presented in Table 5 for AGR, ACT, ANN, NSR, and NU. The proposed branch-and-bound method is able to solve large problem instances within a reasonable time. Specifically: (1) the AGR values, the average gap at root node, are very small for all problem instances, (2) a large portion of instances in Problem Set 7 and Set 8 are solved at root node, and (3) Problem Set 9 has the fewest number of instances solved at the root node and the largest number of unsolved problem instances within the time limit of 500 s. Note that this behavior is identical to that observed in Table 15.2 for $N = 20$, 50, and 100. A greater number of unsolved problems in Table 15.5, especially for the instances in Problem Set 9, is because of an upper limit on CPU time (500 s) used. Also, in lieu of our observation for instances of Problem Set 3 in Table 15.2, it is expected that the proposed dominance rules will be effective for instances of Problem Set 9, beyond the root node. Consequently, the ML-TSALSP-BB method, with proposed bounding procedure and dominance properties, is an effective approach for solving large-sized problem instances.

Table 15.3 Sets of Problem Instances

Problem set	N	M	U_j	n_j	p_{jk}	p_{jA}
4	(5, 15, 25)	(3, 6, 9)	$U(10, 100)$	$U(1, 10)$	$U(50, 100)$	$U(25, 75)$
5	(5, 15, 25)	(3, 6, 9)	$U(10, 100)$	$U(1, 10)$	$U(50, 100)$	$U(50, 100)$
6	(5, 15, 25)	(3, 6, 9)	$U(10, 100)$	$U(1, 10)$	$U(50, 100)$	$U(75, 125)$

Table 15.4 Comparison of direct solution of ML-TSALSP-M by CPLEX 10.1 with ML-TSALSP-BB

			ML-TSALSP-M					ML-TSALSP-BB				
Set	N	M	AGR (%)	ACT	ANN	NSR	NU	AGR (%)	ACT	ANN	NSR	NU
4	5	3	11.38	1.87	12	1	0	0.01	0.45	11	18	0
		6	11.44	3.31	11	0	0	0.001	1.45	18	15	0
		9	11.43	5.34	12	0	0	0.001	0.58	3	19	0
	15	3	7.31	6.07	80	0	0	0.001	0.29	1	20	0
		6	7.12	41.19	2482	0	1	0.001	75.51	899	16	3
		9	7.39	168.87	4062	0	4	0.001	50.91	432	18	2
	25	3	4.31	284.21	1015	0	3	0.001	25.45	664	19	1
		6	5.12	483.04	397	0	18	0.001	25.99	346	19	1
		9	4.39	500.67	35	0	20	0.001	51.52	402	18	2
5	5	3	6.34	0.19	20	0	0	0.61	0.32	4	11	0
		6	6.94	0.36	28	0	0	0.76	1.14	8	9	0
		9	7.65	0.53	28	0	0	0.46	1.66	10	12	0
	15	3	3.61	445.24	30526	0	17	1.26	76.42	1158	10	3
		6	4.90	476.97	19162	0	19	0.81	167.52	848	7	6
		9	5.75	437.67	12196	0	17	0.49	178.96	719	11	7
	25	3	2.15	477.61	11670	0	18	0.40	33.49	213	9	1
		6	3.32	496.95	3305	0	19	0.32	186.69	771	10	5
		9	2.78	500.57	579	0	20	0.30	84.89	315	15	3
6	5	3	2.67	0.17	14	0	0	0.62	0.2	2	10	0
		6	5.29	0.21	22	0	0	0.66	0.71	4	8	0
		9	6.11	0.31	28	0	0	0.60	1.13	5	10	0
	15	3	0.16	87.06	12621	0	3	0.08	1.07	3	7	0
		6	0.20	182.89	11261	0	4	0.14	5.48	9	3	0
		9	0.26	274.84	7624	0	9	0.14	9.49	9	3	0
	25	3	0.08	173.07	6362	0	5	0.05	1.88	4	4	0
		6	0.10	379.4	4368	0	13	0.07	11.6	11	3	0
		9	0.11	440.04	2542	0	16	0.07	34.91	22	2	0

Concluding Remarks

In this chapter, we have discussed a multiple-lot, lot streaming problem for a two-stage assembly system. This system consists of M parallel machines at Stage 1 and a single assembly machine at Stage 2. Such a system for the processing of the lots has been considered in the literature. However, we include a new feature pertaining to the streaming of the lots over the stages. This adds another level of complexity to the problem. For a given number of sublots of a lot, we show that the

Table 15.5 Performance of ML-TSALSP-BB for large problem instances.

Set	N	M	AGR (%)	ACT	ANN	NSR	NU	N	M	AGR (%)	ACT	ANN	NSR	NU
7	300	3	0	83.95	1	20	0	700	3	0	19.82	1	20	0
		6	0.001	41.92	2	19	1		6	0	42.01	1	20	0
		9	0.001	76.25	2	18	2		9	0.001	84.47	1	19	1
	400	3	0	11.08	1	20	0	880	3	0	22.43	1	20	0
		6	0	23.50	1	20	0		6	0	47.54	1	20	0
		9	0.001	64.65	1	19	1		9	0	71.27	1	20	0
	500	3	0.001	14.05	1	19	1	900	3	0.001	47.06	1	19	1
		6	0	54.35	1	20	0		6	0.001	71.11	1	19	1
		9	0.001	71.09	1	19	1		9	0.001	99.92	1	19	1
	600	3	0	16.79	1	20	0	1000	3	0.001	48.69	1	19	1
		6	0	35.59	1	20	0		6	0.001	53.07	1	19	1
		9	0.001	105.81	1	18	2		9	0.001	85.41	1	19	1
8	300	3	0.001	149.35	14	14	5	700	3	0.001	167.58	8	14	6
		6	0.001	143.49	6	15	5		6	0.001	137.89	3	16	4
		9	0.001	104.76	3	17	3		9	0	67.21	1	20	0
	400	3	0.001	139.24	9	14	5	800	3	0.001	200.34	7	13	7
		6	0.001	78.30	2	18	2		6	0.001	155.73	2	16	4
		9	0.001	90.59	2	18	2		9	0.001	114.66	1	19	1
	500	3	0.001	139.18	8	15	5	900	3	0.002	252.97	7	11	9
		6	0.001	105.19	3	17	3		6	0.001	184.09	1	18	2
		9	0	47.97	1	20	0		9	0.001	144.18	1	18	2
	600	3	0.002	212.61	12	12	8	1000	3	0.001	179.24	4	14	6
		6	0.001	84.60	2	18	2		6	0.001	141.13	1	17	3
		9	0.001	129.22	2	17	3		9	0.001	137.74	1	19	1
9	300	3	0.002	13.79	1	2	0	700	3	0.001	43.47	2	0	3
		6	0.002	199.09	12	0	7		6	0.001	142.84	2	0	2
		9	0.003	448.62	14	0	17		9	0.001	428.75	2	0	9
	400	3	0.001	237.54	1	2	0	800	3	0.001	53.91	2	0	0
		6	0.002	146.24	5	0	3		6	0.001	153.71	2	0	2
		9	0.002	455.61	10	0	10		9	0.001	420.18	4	0	13
	500	3	0.001	32.53	1	1	0	900	3	0.001	60.79	2	0	0
		6	0.001	133.31	3	0	3		6	0.001	167.76	2	0	1
		9	0.002	417.31	8	0	8		9	0.001	429.45	4	0	8
	600	3	0.001	37.23	2	0	0	1000	3	0.001	61.31	2	0	0
		6	0.001	167.65	4	0	14		6	0.001	166.57	2	0	2
		9	0.001	445.36	7	0	9		9	0.001	484.25	4	0	13

sublot sizes are consistent. We determine lower and upper bounds on the sizes of the first and last sublots of a lot. We have also derived dominance properties for the sequencing of the lots. A branch-and-bound method is developed for the solution of our problem that relies on effective lower bounds (on the makespan values), which are also established. Results of a detailed computational investigation on the performance of the proposed branch-and-bound method reveal its efficacy for solving both small- and large-sized problem instances. Our proposed method outperforms the direct solution of a mathematical model of the ML-TSALSP by CPLEX 10.1.

References

Baker, K. R. (1995). Lot streaming in the two-machine flow shop with setup times. *Annals of Operations Research, 57,* 1–11.

Cetinkaya, F. C. (1994). Lot streaming in a two-stage flow shop with set-up, processing and removing times separated. *Journal of Operations Research Society, 45*(12), 1445–1455.

Cetinkaya, F. C., & Kayaligil, M. S. (1992). Unit-sized transfer batch scheduling with setup times. *Computers and Industrial Engineering, 22*(2), 177–182.

Gilmore, P. C., & Gomory, R. E. (1964). Sequencing a one state-variable machine: A solvable case of the traveling salesman problem. *Operations Research, 12,* 665–679.

Hariri, A. M. A., & Potts, C. N. (1997). A branch- and bound algorithm for the two-stage assembly scheduling problem. *European Journal of Operational Research, 103,* 547–556.

Johnson, S. M. (1954). Optimal two- and three-stage production schedule with setup times included. *Naval Research Logistics, 1*(1), 61–68.

Kalir, A., & Sarin, S. C. (2000). Evaluation of potential benefits of lot streaming in flow-shop systems. *International Journal of Production Economics, 66,* 131–142.

Kalir, A., & Sarin, S. C. (2003). Constructing near optimal schedules for the flow shop lot streaming problem with sublot-attached setups. *Journal of Combinatorial Optimization, 7,* 23–44.

Lee, C.-Y., Cheng, T. C. E., & Lin, B. M. (1993). Minimizing the makespan in the 3-machine assembly-type flow shop scheduling problem. *Management Science, 39*(5), 616–625.

Potts, C. N., & Baker, K. R. (1989). Flow shop scheduling with lot streaming. *Operation Research Letters, 8,* 297–303.

Sarin, S. C., Yao, L., & Trietsch, D. (2011). Single-batch, lot streaming in a two-stage assembly system. *Journal of Planning and Scheduling, 1,* 90–108.

Sriskandarajah, C., & Wagneur, E. (1999). Lot streaming and scheduling multiple products in 2-machine no-wait flow shops. *IIE Transactions, 31,* 695–707.

Sun, X., Morizawa, K., & Nagasawa, H. (2003). Powerful heuristics to minimize makespan in fixed, 3-machine, assembly-type flowshop scheduling. *European Journal of Operational Research, 146,* 498–516.

Trietsch, D. (1987). Optimal transfer lots for batch manufacturing. *Manuscript presented at the ORSA/TIMS Conference*.

Vickson, R. G. (1995). Optimal lot streaming for multiple products in a two-machine flow shop. *European Journal of Operations Research, 85,* 556–575.

Yao, L. (2008) Modeling analysis and solution approaches for some optimization problems: High multiplicity asymmetric traveling salesman, primary pharmaceutical manufacturing scheduling, and lot streaming in an assembly system," Doctoral Dissertation.

SALAH E. ELMAGHRABY, PhD

University Professor Emeritus
Department of Industrial & Systems Engineering
and the Graduate Program in Operations Research
North Carolina State University, Raleigh NC, 27695-7913
Telephone (919)515-7077
FAX (919)515-5281
e-mail: "elmaghra@ncsu.edu"

Education

B.Sc.	Mechanical Engineering	Cairo University	1948
M.Sc.	Industrial Engineering	Ohio State University	1955
Ph.D.	Industrial Engineering	Cornell University	1958

Personal Data

Born:	21 October, 1927
Married:	to Amina Ishac
Has three daughters:	Wedad Jasmine, Karima Noor, and Leila.
Home address:	3604 Ranlo Drive, Raleigh, NC 27612, USA.

Employment

July 2010 to Present	University Professor Emeritus, NCSU
July 1967 to June 2010	University Professor, N.C. State University, Raleigh, N.C.
January 1970 to June 1990	Director, Graduate Program in Operations Research
July 1971 to June 1975	Associate Head and Graduate Administrator, Industrial Engineering Dept., N.C. State University.
July 1962 to July 1967	Associate Professor, Dept. of Adm. Sci., Yale University, New Haven, Connecticut.
Spring Semester, 1967	Visiting Associate Professor, School of Industrial Engineering, Cornell University, Ithaca, N.Y.
June 1958 to July 1962	Research Leader, Western Electric Company, Engineering Research Center, Princeton, New Jersey.
Sept. 1955 to June 1958	Research Assistant, Cornell University, Ithaca, N.Y.

P. S. Pulat et al. (eds.), *Essays in Production, Project Planning and Scheduling,*
International Series in Operations Research & Management Science 200,
DOI 10.1007/978-1-4614-9056-2, © Springer Science+Business Media New York 2014

October 1949 to June 1954	Inspecting Engineer, Foreign Inspection Office of the Egyptian State Railways in London, Brussels and Budapest.
Jan. 1949 to October 1949	Teaching Assistant, School of Engineering, Cairo, Egypt.
Aug. 1948 to Jan. 1949	Plant Manager, The Coca Cola Bottling Co., Cairo, Egypt

Foreign Experience of Professional Nature

- 5 years in Europe—firsthand experience with the majority of Western European industry: England, France, Belgium, Holland, Denmark, Norway, Luxembourg, Federal German Republic, Austria, Italy, and the Hungarian heavy industries.
- Visiting Professor, European Institute for Advanced Studies in Management, in Brussels, and the Katholic University of Leuven, in Leuven, Belgium, Academic year 1974–1975.
- Visiting Lecturer, Department of Production Engineering, Alexandria University, Egypt, December, 1976.
- Principal Scientist, Kuwait Institute for Scientific Research, Kuwait, Academic years 1981–1983, on leave from NCSU.
- Visiting Professor, The Thomson Chair Professor of Production Management, Claude Bernard University of Lyon I, Lyon, France, May–June 1991 and May–June 1992, June–July 1995.
- Visiting Professor, Department of Systems Engineering, Nagoya Institute of Technology, Nagoya, Japan, December 1997 to March 1998.
- Visiting Professor, CREGI, Faculte Universitaire Catholique de Mons (FUCAM), Summers of 2000, 2001, 2002, 2003.
- Visiting Professor, University of Paris, SUPMECA, May 2004 and June 2006.
- Chair Professor, Tsinghua University, Beijing, PRC, Spring semester 2005 (January through June).
- Visiting Professor, University de Minho, Guimaraes campus, Portugal, summer 2006 and 2008.
- Visiting Professor, Université d'Artois at Bethune, France, June 2008.
- Visiting Professor, **National Chiao Tung University (NCTU)**, Taiwan, Oct.-Nov., 2008.

Guest Lecturer

- **Belgium**: The Center of Operations Research and Econometrics; Université Catholique de Louvain; Katholieke Universiteit Leuven; Faculté Universitaire de Mons.

- **Canada**: The Univ. of Toronto, Toronto; the University of Montreal and the Polytechnique University, Montreal; Laval University, Quebec City.
- **China:** Guest lecturer at the Academy of Science, the Department of Mathematical Sciences at Tsinghua University, The National Academy of Mathematical Sciences in Beijing; Zhejiang University in Hangzhou, both in PRC.
- **Egypt:** The American University of Cairo.
- **France**: The University of Lille; The University of Grenoble, University of Lyon 1 (Claude Bernard), Lyon, the ISMCM of the University of Paris, and the Université de Valenciennes et du Hainaut Cambrésis, Valenciennes.
- **Germany**: The University of Bonn; The Free University of Berlin; The Technical University of Aachen; the University of Karlsruhe, Karlsruhe
- **Holland**: The Mathematische Centrum, Amsterdam
- **Japan**: Nagoya Institute of Technology, Nagoya; Osaka Institute Of Technology, Osaka; Aoyama Gakuin University, Tokyo; Ashikaga Institute Of Technology, Ashikaga, Kyoto Institute of Technology, Kyoto
- **Portugal**: The Technical University of Lisbon, Lisbon; the University of Minho, Guimaraes
- **Saudi Arabia**: The University of Petroleum and Minerals, Dhahran
- **South Africa**: The University of Johannesburg
- **Sweden**: The Tech. University of Linköping
- **Turkey**: Bilkent University, Ankara
- **UAE**: The University of the United Arab Emirates, Dubai
- **UK**: The Lucas Center for Engineering Production, Birmingham; Brunel University, London
- **US**: Over 30 universities in the US, the latest of which is the Louisiana State Univ. Chancellor's Distinguished Lecture Series, March 1999.

Membership in Learned and Professional Societies

IIE	The Institute of Industrial Engineers
NSPE	National Society of Professional Engineers
INFORMS	Operations Research Society of America & The Institute of Management Science
POMS	Production and Operations Management Society

Membership in Honorary Societies

Alpha Pi Mu	(Industrial Engineering)
Phi Kappa Phi	(Science)
Sigma Xi	(Scientific Research)
Tau Beta Pi	(Engineering)
Sigma Iota Rho	(International)

Special Honors and Awards

- Student Competition of ILS10 has been named after him; Casablanca, Morocco April 14–17, 2010.
- Elected Fellow, INFORMS, October 2004
- Recipient of the Frank and Lillian Gilbreth Award, IIE, 2003
- Recipient of the Alexander Quarles Holladay Medal for Excellence, NCSU, 2000
- Recipient of Honorary Doctorate, University Claude Bernard Lyon I, Lyon, France, October 1998.
- Invited as Visiting Professor, Faculté Universitaire Catholique de Mons, Belgium, the summers of (1999 through 2003).
- Invited as The Thomson Chair Professor of Production Management, Claude Bernard University of Lyon I, Lyon, France, the summers of 1991, 1992, 1995.
- Recipient of The Kuwait Foundation for the Advancement of Science Distinguished Award, May 1990.
- Recipient of The R. J. Reynolds Distinguished Award in Research and Education, College of Engineering, NCSU, 1987.
- Elected Fellow of the Institute of Industrial Engineers, 1986.
- Operations Research Division Award, IIE, 1980.
- Awarded the David F. Baker Distinguished Research Award, IIE, 1970.
- Recipient of the First Prize, National Center for Education & Research in Equip. Policy, 1958.
- Recipient of the First Prize, Morse Chain Co. Competition Award, Ithaca, NY, 1957.

Research Grants and Awards (Principal Investigator)

- The Academy of Applied Science (Summer 1980), US$ 2,500
- Alcatel Network Systems (August 1991–July 1993), US$ 230,000 (co-PI)
- The Army Research Office (1972–1978), approximately US$ 130,000
- The Army Research Office (1979–1985) approximately US$ 97,000
- The Army Research Office (1986–1989), US$ 280,000
- Bell Northern Research (1990–1992), US$ 127,000 (co-PI)
- National Aeronautics and Space Administration (1969–1970), US$ 15,000
- Department of Correction (1975–1977), US$ 73,000
- National Science Foundation (1964–1974), approximately US$ 250,000
- National Science Foundation (1978–1981), approximately US$ 102,000
- Northern Telecom (January 1991–December 1991), US$ 18,000
- The Office of Naval Research (1971–1974), approximately US$ 75,000
- Tultex Inc., Martinsville, VA, (Jan. 1992–Dec. 1992), US$ 23,000 (co-PI)
- IBM Corp, RTP, NC (Aug. 1994–May 1995), US$ 43,000
- Glen Raven Mills, Burnsville, NC (Jan. 1991–Dec. 1996), US$ 95,000

- Northern Telecom, Research Triangle Park, NC. (Jan. 1997–Aug. 1997), US$ 40,000
- Glen Raven Mills, Burnsville, NC (Sept. 1992–Dec. 1997), US$ 120,000
- North Carolina Sea Grant (Jan.2001–June 2001), US$ 8,518
- ABB, Centennial Campus (May 22, 2001–July 31, 2001), $7,770, and August 2002 through July 3, US$ 41,310.
- SAS, Cary N. C., Fall 2005, US$ 34,000.
- Hamlin Sheet Metal Co., Benson, NC, 2007–2010, US$ 55,000.

Professional Activities

- Keynote Speaker, INFORMS 2011, Charlotte, NC.
- Keynote Speaker, ILS2010, Casablanca, Morocco, April 14–17, 2010.
- Co-Chairman of Students Research Competition, IESM07 Conference, Beijing, PRC, May 29-June 1, 2007. Also Member of the Conference Scientific Committee.
- Founder and Editor-in-Chief, Jour. Oper. and Logistics (JOL), I4E2 Society, 2005 to present.
- Co-Chairman of Students Research Competition, Information Systems, Logistics and Supply Chain (ILS) Conference, Lyon, France, May 15–18, 2005.
- Regional Editor (The Americas), Intern'l Jour. Production Economics (IJPE), Elsevier, Aug. 1995–2000.
- Co-Founder and Council Member, PMS Intern'l Workshops, 1988–1998. Keynote speaker, PMS98, Istanbul, Turkey, July 7, 1998.
- Vice President for education, PMI Research Triangle Chapter, 1991
- Program co-Chairman, POM-91, November 1991
- Member, Board of Advisors, POMS, 1990-present
- Program Chairman, Manufacturing International '90, Atlanta, Georgia, March 1990
- Member, Advisory Committee of Computer and Mathematics in Engineering Design, National Science Foundation, Washington, D.C., 1964–1966
- Member, Information Systems Committee, ASEE, 1970–1973
- Director of Research in Production and Inventory Control, Institute of Industrial Engineers, 1961–1962.
- Research Chairman for the Operations Research Division, AIIE, 1967
- Member, AIIE Membership Board of Review, 1976–1979
- ORSA Lectureship Series, 1972–1973, 1974–1976, 1978–1979
- Associate Editor, Management Science, Theory & Applications, 1968–1976
- Abstractor for International Abstracts in OR, the official publication of IFORS, 1970–1973
- Program Chairman, ORSA National Meeting, Miami, Florida, November 10–12, 1969

- Member, Advisory Board, Encyclopedia of Computer Science and Technology, M. Dekker Publishers, 1973–1978.
- Associate Editor, The Transactions of Industrial Engineering, 1969–1972
- Senior Editor, IE Transactions, 1972–1974
- Department Editor, Production Planning/Scheduling/Control, IIE Transactions, 1976–1980 and again 1985–1988. Department Editor, Feature Applications, IE Transactions, 1989–1992.
- Associate Editor, Opsearch (Indian Journal on Operations Research), 1982-present Systems Area Program Chairman, ORSA National Meeting, Atlantic City, N.J., November, 1972
- Member, Publications Committee, ORSA, 1974–1978
- Regional Counselor, Omega Rho, 1975–1976 (Honor Society for Operations Research)
- Referee for: Army Research Office, National Science Foundation, The Research Council of Canada, The Research Council of Belgium, ASME, Euro. J. Oper. Res., IE Trans., J. Engr. & Appl. Sci., Management Sci., Nav. Res. Logistics, Omega, Oper. Res., POMS, Intern'l J. Prod. Res., Annals of Operations Research.
- Member, Board of Advisors, POMS, 1990-present
- Program Chairman, Manufacturing International '90, Atlanta, Georgia, March 1990
- Member, Advisory Committee of Computer and Mathematics in Engineering Design, National Science Foundation, Washington, D.C., 1964–1966
- Member, Information Systems Committee, ASEE, 1970–1973
- Director of Research in Production and Inventory Control, Institute of Industrial Engineers, 1961–1962.
- Research Chairman for the Operations Research Division, AIIE, 1967
- Member, AIIE Membership Board of Review, 1976–1979
- ORSA Lectureship Series, 1972–1973, 1974–1976, 1978–1979
- Abstractor for International Abstracts in OR, the official publication of IFORS, 1970–1973
- Program Chairman, ORSA National Meeting, Miami, Florida, November 10–12, 1969
- Member, Advisory Board, Encyclopedia of Computer Science and Technology, M. Dekker Publishers, 1973–1978.
- Associate Editor, The Transactions of Industrial Engineering, 1969–1972
- Senior Editor, IE Transactions, 1972–1974

University/College Service

- Senator, N.C. State University Faculty Senate, 1972–1974, 1999–2002.
- Chairman, Graduate Studies Committee, School of Engineering, 1985–1986.
- Member, Administrative Board of the Graduate School, 1988–1994.
- Member, Advisory Council of the Graduate School, 1991–1994.

- Member, Council of University Professors.
- President, NCSU Chapter of Sigma Xi, 1998.
- Chairman of over twenty ad-hoc committees in the Oper. Res. Graduate Program, the Ind. Eng. Department, the College of Engineering, and the University at large, and member of over thirty other ad hoc committees in the University, college, department of Ind. Eng. and Graduate Program of Operations Research.

Consulting Experience

ABB, Centennial Campus (Raleigh NC)
Cyanamid Co. (Wallingford, Conn.)
Ford Foundation (Cairo, Egypt)
Glen Raven Mills (Burnsville, N.C.)
IBM Co. (Research Triangle Park, N.C.)
Logistics Management Institute, (McLean, VA)
Messier-Dowty (Bidos, France)
Nello-Teer Co. (Durham, N.C.)
Olivetti Co. (Ivrea, Italy)
The Kharafi Construction Co. (Abu Dhabi and Kuwait)
The Kuwait Institute for Scientific Research (Kuwait)
The Kuwait Foundation for the Advancement of Science (Kuwait)
The Kuwait University, College of Engineering
Texasgulf Co. (Raleigh, N.C.)
The TOKTEN program of the UNDP, Egypt
Tultex, Inc., Martinsville, VA
The Western Electric, Co., Research Center (Princeton, N.J.)

Biographical Listings

Who's Who in the South and Southwest
Who's Who in America
Men and Women of Distinction
International Who's Who of Intellectuals
Men of Achievement
American Men and Women of Science
Who's Who in Engineering
Global Register's Who's Who

PhD Dissertations and Master's Theses Supervised

Doctor of Philosophy Degree

Name	Title	Year Graduated
Ramachandra, Girish	Optimal resource allocation in Activity Networks	Dec., 2006
Matta, Marie E.	An empirical and theoretical study of outpatient	May, 2004
Karnoub, Razek E.	Scheduling problems employing simulation and genetic algorithm methodologies (co-chair, Duke University, Durham NC) An exact bidirectional approach to the resource constrained project scheduling problem	May, 2002
Stephanie R. Earnshaw	The location/allocation of field representatives and trainers to sites (co-chair)	May, 2000
Soewandi, Hanijanto	Sequencing jobs on two- and three-stage hybrid flowshop to minimize makespan	May, 1998
Zhang-Lo, Shuzhi	ATM topological design and network modification	December, 1996
Michael, David J.	The optimal representation of activity networks as directed acyclic graphs	May 1991
Ferrell, William G.	Systems dynamics in quality assurance	May 1989
Pulat, P. Simin	Maximum flow problem for generalized networks	May 1984
Dodin, Bajis M.	On the completion time of stochastic PERT networks	May 1982
Salem, Adel M.	Optimal time reduction in activity networks under convex cost functions	May 1980
Sarin, Subhash C.	Project planning under constrained resources	May 1978
Elimam, Abdelghani A.	Makespan minimization on identical parallel machines	August 1978
McGinnis, Leon F.	Approximate and exact solution procedures for a class of facilities location problems (co-chair)	May 1975
Rihani, Fouad Akil	A model for selecting the optimum number, size and location of highway maintenance yards (co-chair)	May 1974
Modi, Jamshed A.	The use of solution generating systems for knapsack problems with extensions to general integer linear programming	August 1973
Mallik, Arup K.	The scheduling of a single processor under deterministic demand for several products	May 1972
Arisawa, Sanji	Solution of the hub-wheel scheduling problem in transportation networks	May 1972
Wig, Monmahan	On the stock cutting problem	May 1969

Master of Science Degree

Name	Title	Year Graduated
Rajneesh Rajneesh	Scheduling precedence-constrained jobs on two resources to minimize the total weighted completion time	August 2012
Adam J. Rudolph	An algorithm for determining the optimal resource allocation in stochastic activity networks	May, 2008
Clayton D. Morgan	Meta-heuristics for resource allocation in activity networks	Dec., 2006
Ramachandra, Girish	Scheduling precedence related jobs on identical parallel processors	May, 2002
Lightner, Constance	Analysis of linear programming models applied to an emergency vehicle location problem	August, 1997
Taner, Mehmet R.	A study of the mean and variance of project duration in a probabilistic activity network (co-chair)	May, 1998
Thoney, Kristin A.	The two-machine stochastic flowshop sequencing problem	August 1997
Baxter, Elizabeth J.	Simulation of a software engineering process	May 1993
Reed, Stephanie J.	The sequencing of picking stations in a garment warehouse	May 1993
Schellenberger, Keith W.	Sequencing of parallel processors	May 1993
Hurchalla, David A.	An optimal algorithm for solving the pick-up and delivery problem with time constraints	May 1991
Singh, Major	The modeling of large scale software development	May 1991
Johnson, Jerry W.	Economic manufacturing quantity with inspection for processes under the influence of learning	May 1986
Dillery, D. Scott	On the optimal partitioning of a seasonal distribution	May 1985
Venable, Charles J.	Capacity loading and scheduling in a testing facility	May 1985
Colby, Anthony H.	On the complete reduction of acyclic networks	May 1984
Nishimura, Morio	Scheduling flow shops of the ordered and semi-ordered types	May 1980
Samara, Hanan	The dynamic economic lot size model with fixed batch size and sequence dependent change over costs	May 1979
Dillehunt, Susan L.	The optimization of a machine planting system	May 1978
Allen, Jerry W., Jr.	Sequencing on a single processor with a common deadline	May 1978
Pulat, P. Simin	Optimal project compression with multiple node due dates	May 1977
Bazzi, Muna	A simulation study of the faculty population in a university	May 1977
Dodin, Bajis M.	A branch-and-bound algorithm for scheduling n products on a single processor under deterministic demand	May 1975
Worley, Jerry S.	The application of legendre polynomials and spline functions to dynamic programming	May 1974
Dix, Lynn P.	Scheduling lot size production with constant demand	May 1973
Park, Sung H.	A branch-and-bound method for the problem and group theoretic aspects of an integer programming formulation	May 1971

Supervised 9 Master of Operations Research and Master of Industrial Engineering (non-thesis). Currently supervising 3 Master and 1 doctoral candidate in the USA, and co-advisor of 2 doctoral candidates in Portugal, 1 doctoral candidate in France.

Member of 40 + other Master and doctoral Advisory Committees.

"Member of the Jury" of 6 doctoral students in Europe and Canada. They are, arranged chronologically:
Jamal Ouenniche, Laval University, Canada, 1998 (External Examiner)
Fouad Riane, Faculte Catholique de Mons, Belgium, 1998 (Rapporteur)
Omar Moursli, Universite Catholique de Louvain, Belgium, 1999 (Rapporteur)
Trond Jorgensen, Norges-Tecknisk Naturvitenskapelige Universitet, Trondheim, Norway, 1999 (External Examiner)
Anabela Tereso, Universidade do Minho, Braga, Portugal, 2002 (co-Chair)
Hamid Allaoui, Faculte Catholique de Mons, Belgium, 2004 (co-Chair)
Emilie Grandgirard, University Claude Bernard Lyon 1, Lyon, France, November 2007 (External Examiner)

"External Reader" for 4 doctoral theses. They are:
N. R. Achutan, Indian Statistical Inst., Calcutta, India, 1980
V. R. Prasad, Indian Statistical Inst., New Delhi, India, 1982
Kanda, Indian Inst. of Tech., Delhi, India, 1985
V., Suresh, Indian Inst. of Tech., Madras, India, 1994.

Scientific Publications—Books

- **Production Capacity: Its Bases, Functions, and Measurement,** Chapter I-4 in Handbook of Production Planning, K. Kempf, P. Keskinocak, and R. Uzsoy, eds, (2011).
- **A Multi-Model Approach for Production Planning and scheduling in an Industrial Discrete-Continuous Environment**, Chaper II-19 in Handbook of Production Planning, K. Kempf, P. Keskinocak, and R. Uzsoy eds, (2011), co-authored with A. Artiba, D. Duvivier and V. Dhaevers.
- **Operations Research,** chapter in Encyclopedia of Physical Science and Technology, 3rd ed., R. A. Meyers, ed., (2001). Co-authored with S-C. Fang.
- **Activity Nets: PERT/CPM and Their Extensions**, Section 15.5 in Handbook of Discrete and Combinatorial Mathematics, K. H. Rosen, ed., (1998).
- **The Planning and Scheduling of Production Systems: Methodologies and Applications**; Chapman & Hall, (1997), co-editor with A. Artiba, 367 pages. This is a book of original readings on the subject.
- **Handbook of Operations Research**, co-editor with J. J. Moder, Reinhold Van-Nostrand Publishers, Vol. 1, January, (1978); Vol. 2, April (1978). This handbook was translated into Russian.

- **Advances in Project Scheduling,** (1989), Chapter I, Part III, R. Slowinski, J. Weglarz (Eds.). "The estimation of some network parameters in the PERT model of activity networks: Review and critique." Elsevier, Amsterdam, pp. 371–432.
- **Operations Research,** Chapter in the Encyclopedia of Physical Science and Technology (1992); co-authored with S-C Fang.
- **Activity Networks: Project Planning and Control by Network Methods**, John Wiley & Sons, 443 pages, (1977). This book was translated into Japanese.
- **Allocation Models**, contributed to the Encyclopedia of Computer Science, (1974), 371–382; co-authored with M. El-Kammash.
- **Scheduling Theory and Its Applications**, Editor, Proceedings of Symposium, Springer-Verlag, (1973).
- **Operations Research**, Chapter 3 of Section 10 in Industrial Engineering Handbook, McGraw-Hill, 3rd Ed., (1971), H. B. Maynard, Ed.
- **Some Network Models in Management Science**, Springer-Verlag Lecture Series in Operations Research, No. 29, June, (1970).
- **The Design of Production Systems**, Reinhold Publishing Company, New York City, 481 pages, May, (1966). This book was translated into Rumanian.

Unpublished Manuscripts:

- **Dynamic Programming: Models and Applications**; Lecture notes in manuscript form, to be submitted for publication.
- **Risk and Uncertainty in Activity Networks,** Lecture notes in manuscript form, to be submitted for publication.

Scientific Publications—Published Papers (or Accepted for Publication)

1. "A Note on Production Scheduling by the Use of Transportation Method", Letter to the Editor., *Oper. Res. 5* (1957), 565–566.
2. "Design of In-Process Storage Facilities", *J. Ind. Eng. 8* (1957); co-authored with Eugene Richman.
3. "Probabilistic Considerations in Equipment Replacement Studies", *The Engineering Economist, 4*, Summer 1958. This paper was awarded first place at the graduate level in the 1958 contest conducted by the National Center of Education and Research in Equipment Policy.
4. "A Single-Sample Multiple-Decision Procedure for Selecting the Multinomial Event Which Has the Highest Probability", *The Ann. of Math. Stat. 38* (1959); co-authored with R. E. Bechhofer and N. Morse.
5. "An Approach to Linear Programming Under Uncertainty", *Oper. Res. 7* (1959), 208–216.

6. "Allocation Under Uncertainty When the Demand has a Continuous Distribution Function", *Management Sci. 6* (1960), 270–294.
7. "On the Feedback Approach to Industrial Systems Design", *Management Sci.: Models and Techniques,* Vol. 1, Pergamon Press, N.Y., (1960), 149–167. Paper presented to the 6th International TIMS Meeting in Paris, September 1959.
8. "Research in Computerized Production Control Systems", *presented to the 13th Annual Conference, The Institute of Industrial Engineers, and printed in the Proceedings*, (1962), 269–279.
9. "A Note on the Problem of 'Explosion' and 'Netting' in the Planning of Material Requirements", *Oper. Res. 11* (1963), 530–535.
10. "On the Control of Production in Small Job Shops", *J. Ind. Eng. 14* (1963), 186–196; co-authored with R. T. Cole.
11. "On the Dynamic Programming Approach to the 'Caterer' Problem", *J. Math. Anal. And Appl. 8* (1964), 202–217.
12. "An Algebra for the Analysis of Generalized Activity Networks", *Management. Sci. 10* (1964), 494–514.
13. "A Dynamic Model for the Optimal Loading of Linear Multi-Unit Shops", *Management Tech. 4* (1964), 47–58; co-authored with A. S. Ginsberg.
14. "Sensitivity Analysis of Multi-Terminal Flow Networks", *Oper. Res. 12* (1964), 680–688.
15. "On the Relationship Between the Cut-Tree and the Fundamental Cut-Set of Multi-Terminal Flow Networks", *Jour. Franklin Inst. 278* (1964), 262–266.
16. "An Operational System for the Smoothing of Batch Type Production", *Management Sci., Series B. 12* (1966), B433-B449; co-authored with J. Jeske and R. O'Malley.
17. "On the Generalized Activity Networks", *J. Ind. Eng. 17* (1966), 621–631, Special issue on Research in Industrial Engineering.
18. "On the Expected Duration of PERT Type Networks", *Management Sci., Series A, 13* (1967), 229–306.
19. "The Determination of Optimal Activity Duration in Project Scheduling", *J. Ind. Eng.19* (1968), 48–51.
20. "The One Machine Sequencing Problem with Delay Costs", *J. Ind. Eng. 19* (1968), 105–108.
21. "The Sequencing of 'Related' Jobs", *Nav. Res. Log. Quart. 15* (1968), 23–32.
22. "The Role of Modeling in IE Design", *J. Ind. Eng. 19* (1968), 292- 305. Paper presented to the June 1967 Annual Meeting, ASEE, East Lansing, Michigan.
23. "The Sequencing of n Jobs on m Parallel Processors with Extensions to the Scarce Resources Problem of Activity Networks"; presented at the *Inaugural Conference, the Scientific Computation Center, Cairo, Egypt, December 17–20,* (1969)*; and appeared in the Proceedings* of the Conference, 230–255.
24. "The Machine Sequencing Problem: Review and Extensions", *Nav. Res. Log. Quart. 15* (1968), 205–232. Invited paper presented at the Symposium on Production Sequencing and Control, Stevens Institute of Technology, December 1967.

25. "A Loading Problem in Process Type Production", *Oper. Res., 16* (1968); also published in Cuadernos de Estadistica Aplicado e Investigacion Operative, Vol. VI, Fasc. 3, Ano. (1969), 902–914.
26. "The Concept of State in Discrete Dynamic Programming", (1970), *J Math. Anal. & Appl. 29(3)*: 523–557.
27. "The Scheduling of Lots on a Single Facility", *IIE Trans. 2* (1970), 203–213; co-authored with A. Mallik and H. L. W. Nuttle.
28. "Theory of Network Models and Management Science, *Part I,*" (1970), *Management Sci. 17(1)*: 1–34.
29. "Theory of Network Models and Management Science, *Part II,*" (1970), *Management Sci. 17(2)*: B54–B71.
30. "A Graph theoretic interpretation of the sufficiency conditions for contiguous binary switching (CBS)-rule," (1971), *Nav. Res. Log. Quart. 18(3)*: 339–344.
31. "Hyperbolic Programming with a Single Constraint and Upper-Bounded Variables," (1972), *Management Sci. 19(1)*: 42–45; with S. Arisawa.
32. "On the Sequencing of n Jobs on One Machine to Minimize the Number of Jobs Late," (1972), Letter to the Editor, *Management Sci. 18(7)*: 389.
33. "Optimal Time-Cost Trade-Offs in GERT Networks," (1972), *Management Sci 18(11)*: 589–599; with S. Arisawa.
34. "Optimization of Batch Ordering Under Deterministic Variable Demand," (1972), *Management Sci. 18(9)*: 508–517; with V. Y. Bawle.
35. "On the Scheduling of Jobs on a Number of Identical Machines," (1974), *IIE Trans. 6*: 1–13; with S. Park.
36. "The Scheduling of a Multi-Product Facility," (1973), *Proceedings of the Symposium of Theory of Scheduling and Its Applications*, Springer-Verlag, November, pp. 244–277; with A. K. Mallik.
37. "Sequencing jobs on a single machine to minimize total weighted tardiness when all jobs have same due date," (1975), *Oper. Res. 23*: B371, Suppl.2; with J. W. Allen and H. L. W. Nuttle.
38. "Branch-and-Bound Revisited: A Survey of Basic Concepts and Their Applications in Scheduling", *Modern Trends in Logistics Research, Chapter 8, the MIT Press*, (1976), 133–205, W. H. Marlow, Ed; co-authored with A. N. Elshafei.
39. "The 'Hub' and 'Wheel' Scheduling Problem. Part I: The 'Hub' Scheduling Problem: The Myopic Case. Part II: The 'Hub' Operation Scheduling Problem (HOSP): Multi-Period and Infinite Horizon, and the Wheel Operations Scheduling Problem (WOSP)", *Transportation Sci. 11* (1977), Part I: 24–146, Part II: 147–165; co-authored with S. Arisawa.
40. "An Extended Basic Period Approach to the Economic Lot Scheduling Problem (ELSP)," *Proceedings, the 4th International Conference on Production Research*, Tokyo, Japan, August 27–30, (1977); Proceedings of 4th ICPR.
41. "The Economic Lot Scheduling Problem (ELSP): Review and Extensions," (1978), *Management Sci. 24(6)*: 587–598.
42. "Activity Networks: Their Uses and Misuses in Project Planning and Control," Presented at International Conference on Systems Modeling in Developing

Countries, May 8–11, (1978), Asian Inst. of Tech., Bangkok, Thailand. Appeared as *Chapter 4 in Systems Models for Decision Making, Asian Inst. of Tech*., May (1978).
43. "Optimal Project Compression with Due-Dated Events," (1979), *Nav Res Log Quart 26(2)*: 331–348; with P. Simin Pulat. Presented at the ORSA/TIMS Meeting, November 7–9, (1977), Atlanta, GA.
44. "Some OR Approaches to Automobile Gear Train Design," (1979), *Math. Prog Study 11(Oct)*: 150–175, with S. Metwalli and C. F. Zorowski.
45. "Knapsack-Based Approaches to the Makespan Problem on Multiple Processors," (1980), *IIE Trans. 12(1)*: 87–96; with A. Elimam.
46. "On the Measurement of Complexity in Activity Networks," (1980), *EJOR 5(4)*: 223–234; with Willy S. Herroelen.
47. "Design Parameters of a Mounted Tree Planter for Optimum Productivity", *Proceedings of the Winter Conference of ASAE*; (1978) co-authored with Susan Dillehunt, W. Hafley, A. Hassan and S. Metwalli.
48. "Recent Advances in Activity Networks". Presented at the International Conference on Industrial Systems Engineering and Management in Developing Countries; Bangkok, Thailand, November 3–6, (1980), and appeared in the *Proceedings of the ISEMDC.*
49. "A Note on EMQ Under Learning and Forgetting," (1981), *AIIE Trans 13(1)*: 86–90, with Sven B. Axsäter.
50. "Optimal Project Compression Under Convex Functions, I and II", *Appl. Management Sci., Vol. 2*, (1982), 1–39, co-authored with A. Salem.
51. "Batch Production over Finite Horizon with Sequence-Dependent Setup Cost", *Proceedings of the First International Conference on Current Advances in Mechanical Design and Production*, Cairo University, Egypt, December 27–29, (1979); co-authored with J. Wijngaard.
52. "An Approach to the Control of Research and Development Projects", *Scand. J. Mat. Adm. 9* (1984), 26–57.
53. "Bounds on the Performance of a Heuristic to Schedule Precedence-Related Jobs on Parallel Machines," (1984), *IJPR 22(1),*17–30; with S. Sarin.
54. "Composite Mix Design in Production of Asbestos/Cement Pipes," (1984), *Appl Math. Modelling 8(6)*: 425–432; with A. Elimam.
55. "Optimal Linear Approximation in Project Compression", (1984), *IIE Trans 16(4)*: 339–347; with A. M. Salem.
56. "Some Recent Advances in Activity Networks", *Proceedings of ARO Workshop on Analytical and Computational Issues in Logistics R & D,* George Washington University, May 7–9, (1984).
57. "Approximating the Criticality Indexes of the Activities in PERT Networks", (1985), *Management Sci. 31(2)*: 207–223; with B. Dodin.
58. "Industrial Diagnostics Research: An Approach," (1985), *IJPR 23(4)*: 675–689; with A. Elimam.
59. "Comments on a DP Model for the Optimal Inspection Strategy", (1986), *IIE Trans. 18(1),* 104–108.

60. "On the Reduction Method for Integer Linear Programs, II," (1985), *Discr. Appl. Math. 12(3),* 241–260; with A. A. Elimam.
61. "Optimal Partitioning of a Seasonal Distribution," *Proceedings, Symposium on Systems Analysis in Forest Resources*, December 9–11 (1985), Georgia Center for Continuing Education, University of Georgia, Athens, GA, co-authored with H. A. Devine, D. S. Dillery, and J. E. deSteiguer.
62. "The Estimation of Some Network Parameters in the PERT Model of Activity Networks: Review and Critique," *Chapter 1, Part III in Advances in Project Scheduling,* R. Słowinski and J. Weglarz, eds., Elsevier (1989).
63. "Research needs and challenges in application of computer and information sciences for industrial engineers," (1989), *IIE Trans 21(1);* with S. Y. Nof and G. Salvendy.
64. "The Knapsack Problem with Generalized Upper Bounds (KnPGUB)," (1989), *EJOR 38(2),* 242–254.
65. "On Heuristics and Their Performance Evaluation for Dynamic Lot Sizing," (1989), *Opsearch 26,* 1–10.
66. "Quality Assurance and Stage Dynamics in Multi-Stage Manufacturing, Part I," (1990), *Int'l J Prod Res 28(5),* 853–877; with W. G. Ferrell.
67. "Quality Assurance and Stage Dynamics in Multi-Stage Manufacturing, Part II," (1990), *Int'l J Prod Res 28(6),* 1083–1097; with W. G. Ferrell.
68. "Documentation of BIDNET: Project Bidding for CPM and PERT Activity Networks," appeared in abstracted form in *Euro. J. Operl. Res. 41* (1989), 122–123, co-authored with D. Michael.
69. "Optimal Control of the Southern Pine Beetle (SPB) Infestation," (1990), *Appl Math Modeling 4(3), 155*–164 (1989).
70. "On Project Representation and Activity Floats," (1990), *Arab J. Sci.& Eng. 15*(4B), 626–637, with J. Kamburowski.
71. "Project Bidding Under Deterministic and Probabilistic Activity Durations," (1990), *Euro. J. Oper'l Res. 49(1)*, 14–34.
72. "The Scheduling of Activities to Maximize the Net Present Value of Projects," (1990), *Euro. J. Oper'l Res. 49(1)*; 35–49; with W. S. Herroelen.
73. "Economic Manufacturing Quantities Under Conditions of Learning and Forgetting (LaF)," *J. Prod. Planning & Control 1* (1990); 196–208.
74. "Manufacturing Capacity and Its Measurement," (1991), *Comp. Oper. Res. 18(7),* 515–627.
75. "The Analysis of Activity Networks Under Generalized Precedence Relations", *Management Sci. 38(9)*, 1245–1263, (1992); co-authored with J. Kamburowski.
76. "System Modeling: Petri Nets and Activity Nets in Juxtaposition," *Proc. IEEE, SMC'92,* Chicago, IL, Oct. 18–21, vol. 2, 853–860, (1992).
77. "Resource allocation via dynamic programming in Activity Networks," (1993), *Euro. J. Oper'l Res. 64(2)*, 199–215. Presented at the Second International Workshop on Project Management and Scheduling, Compiègne, France, June 20–23, 1990.

78. "An approach to the modeling and analysis of software production process," *Int'l. Trans. Oper'l Res. 5*, 389–394; with E. I. Baxter, and M. A. Vouk (1994).
79. "Activity Nets: A Guided Tour Through Some Recent Developments," (1995), *Euro. J. Oper'l. Res. 82(3)*, 383–408.
80. "Optimal Capacity Allocation: A Case of Constrained Markov Programming," *Proc.Int'l Conf. Ind. Eng. and Prod. Management*, Marrakesh, April 4–7, (1995); Vol. 1, 595–606. Co-authored with Erik Perkasa.
81. "DAGEN: A generator of testsets for project activity nets," (1996), *Euro. J. Oper'l Res. 90(2)*, 376–382; with A. Agrawal and W. Herroelen.
82. "Optimal procedures for the discrete time cost trade-off problem in project networks," (1996), *Euro. J. Oper'l Res. 88(1),* 50–68.
83. "Production Control in Flexible Flowshops: An Example from Textile Manufacturing," (1996), Chapter 6 in *The Planning and Scheduling of Production Systems*: Methodologies and Applications, Elmaghraby and A. Artiba, eds, Chapman & Hall; co-authored with Razek Karnoub,
84. "On the expected completion time of diffusion activity networks (DiAN)", (1997), in *Managing and Modeling Complex Projects*, 47–67, T. M. Williams, ed., Kluwer; with M. K. Agrawal.
85. "Call admission control schemes and ATM network topological design," (1997), *Euro. J. Oper'l Res. 111(2),* 393–404; with Shuzhi Zhang-Lo, B. A. Makrucki and G. L. Bilbro.
86. "Production scheduling/rescheduling in flexible manufacturing," (1997), *Int. J. Prod. Res. 35,* 281–309; with Jain, A. K.
87. "A hybrid three-stage flowshop problem: Efficient heuristic to minimize makespan," (1998), *Euro. J. Oper'l Res. 109(2),* 321–329; with F. Riane and A. Artiba.
88. "On the sensitivity of project variability to activity mean duration," (1999), *IJPE 62(3),* 219–232, with Y. Fathi and M. R. Taner.
89. "Optimal start times under stochastic activity durations," (2000), *IJPE 64(1–3)*, 153–164 with A. A. Ferreira and L. V. Tavares. Paper presented at IEPM, Lyon, France, October 20–24, 1998.
90. "The two-machine stochastic flowshop problem: The case of arbitrary distributions," (1997), *IIE Trans. 31(5),* 467–477; with K. A. Thoney.
91. "An optimal assembly mode of multi-type printed circuit boards," (1999), *Comp. & Ind. Eng. 36(2)*, 451–471; with K. Ohno and Z. H. Jin.
92. "Simple heuristics for the two machine openshop problem with blocking," (2000), *J. Chinese Inst. of Ind. Eng. 17,* 537–547, joint with M-J. Yao and H. Soewandi.
93. "On criticality and sensitivity in activity networks," (2000) *Euro. J. Oper'l Res. 127(2),* 220–238.
94. "On computing the distribution function of the sum of independent random variables," (2001), *Comp. & Oper. Res. 28(5),* 473–483; with M. K. Agrawal.
95. "Chance-constrained programming in activity networks: A critical evaluation," (2001) *Euro. J. Oper'l. Res. 131(2),* 440–458; with H. Soewandi and M. J. Yao.

96. "The economic lot scheduling problem under power-of-two policy," (2001) *Comp.& Math. Appl. 41,*1379–1393.
97. "Sequencing three stage flexible flowshops with identical machines to minimize makespan," (2001), *IIE Trans.33(11),* 985–993; with H. Soewandi.
98. "On the optimal release time of jobs with random processing times, with extensions to other criteria," (2001), *IJPE 74(1–3),* 103–113.
99. "The economic lot scheduling problem under power-of-two policy," (2001), *Comp.& Math. Appl. 41(10–11),* 1379–1393; with M-J Yao.
100. "Adaptive Resource Allocation in Multimodal Activity Networks," (2001) *IJPE 92,* 1–10, with A. P. Tereso and M. M. T. Araújo.
101. "Scheduling hybrid flowshops in printed circuit board assembly lines," (2002), *POM 11 (2),* 216–230. Co-authored with Z. H. Jin, K. Ohno, and T. Ito.
102. "Sequencing a hybrid two-stage flowshop with dedicated machines," (2002) *IJPR.40(17),* 4353–4380 Co-authored with A. Artiba and F. Riane.
103. "Risk analysis in activity networks: Resource allocation and budget estimation," (2000), *Invited Plenary Session Talk, PMS011,* Istanbul, Turkey, and appearing in the *Proceedings* of the conference.
104. "On the Feasibility Testing of the Economic Lot Scheduling Problem Using the Extended Basic Period Approach," *J. Chinese Inst. Ind. Eng. 13,* 13–20 (2002). Co-authored with M-J. Yao and I-C. Chen.
105. "Note on the paper 'Resource-constrained project management using enhanced 'theory of constraints' by Wei et al.", *Inter'l J. Project Management 21* (2003), 301–305. With W. S. Herroelen and R. Leus.
106. "Sequencing on two-stage hybrid flowshops with uniform machines to minimize makespan," (2003), *IIE Trans.35(5),* 467–478. Co-authored with H. Soewandi.
107. "On the feasibility testing of the economic lot sizing problem using the embedded basic period approach," (2003), *J. Chinese Inst. Ind. Eng. 20 (5),* 435–448.
108. "Sequencing precedence related jobs on parallel machines to minimize the weighted completion time," *presented at ISS'02, Hamanako, Japan,* June 4–6, 2002, and appearing in extended abstract form in the *Proceedings* of the conference. Co-authored with Girish Ramachandra. *International Journal of Production Economics, 100,(1),*(2006), 44–58.
109. "Adaptive Resource Allocation in Multimodal Activity Networks", (2004) *International Journal of Production Economics, 92, (1)*, 1–10.
110. "On the fallacy of averages in project risk management," *European Journal of Operational Research;* (2005), 165(2), 307–313.
111. "Scheduling of a two-machine flowshop with availability constraints on the first machine," (2006), *Intl J. Prod. Econ. 99,* 16–27; with H. Allaoui, A. Artiba, and F. Riane.
112. "Resource allocation in activity networks under deterministic conditions: A geometric programming approach," (2007) *J. Operations and Logistics 1(2)*; pp. II.1–II.22; with Clay D. Morgan.

113. "Resource allocation in activity networks under stochastic conditions: A geometric programming-sample path optimization approach," (2007) Tijdschrift voor Economie en Management (*Journal of Economics and Management*), Vol. LII, 3, Katheolike Universität Leuven (2007), 367–389; with Clay D. Morgan.
114. "Optimal resource allocation in stochastic activity networks via the Electromagnetism Approach: A platform implementation in Java," with Anabela P. Tereso, Rui A. Novais, M. Madalena T. Araujo, (2008), *Control & Cybernetics 38(3),* 745–782.
115. "Polynomial Time Algorithms for Two Special Classes of the Proportionate Multi-Processor Open Shop," *European Journal of Operational Research, 201(3)*, (2010), 720–728. Co-authored with Marie Matta.
116. "On The Approximation of Arbitrary Distributions by Phase-Type Distributions," presented at INCOM 09, Moscow, and submitted for publication, co-authored with R. Benmansour, A. Artiba, and H. Allaoui.
117. "The relevance of the 'alphorn of uncertainty' to the financial management of project under Uncertainty," submitted for publication, co-authored with J. Zhang.
118. "On Consistency and Feasibility of Generalized Precedence Relations (GPRs) in Activity Networks," submitted for publication, co-authored with J. Zhang.
119. "In defense of Activity Networks," submitted for publication.
120. "Project Compression Under Generalized Precedence Relations (GPR's)," submitted for publication, co-authored with H. R. Tareghian and J. Qi.

Book Reviews

"Principles of Operations Research" by Harvey M. Wagner, Prentice-Hall, 1969; appeared in the *American Scientist*, March, 1970.

"Societal Systems" by John N. Warfield, John Wiley & Sons, 1976; *Interfaces*, Vol. 9, No. 3, May, 1979.

"The Art and Theory of Dynamic Programming" by S. E. Dreyfus and A. M. Law, Academic Press, 1977; *Interfaces,* Vol. 9, No. 3, 1979.

"On the Control of Complete Industrial Organizations" by Joan E. van Aken; H. E. Stenfert Kroese B. V., The Netherlands, 1978; *Euro. J. Oper. Res*., 1979.

"Network Flow Programming" by Paul Jensen and J. Wesley Barnes, John Wiley, 1980, appeared in *Interfaces,* 12, 1982; 99–100.

"Production and Inventory Management" by Arnoldo C. Hax and Dan Candea, Prentice-Hall, 1983; *Operations Management Review*, Vol. 2, No. 3, Spring 1984.

"Managing Quality" by David A. Garvin, The Free Press, 1988; *IIE Trans*., 21, 191–192, 1989.

"Quality Engineering Using Robust Design" by Madhav S. Phadke, Prentice-Hall, *IE Journal*, June 1990, 88–89.

"Stochastic Project Networks," by Klaus Neumann, Springer-Verlag, *Zeitschrift fur Operations Research*, 1992.

"Modeling the Supply Chains," by J. F. Shapiro, Duxbury Press, 2001, appeared in *The Eng. Economist.*

"Mathematical Programming and Financial Objectives for Scheduling Projects," by Alf Kimms, Kluwer's International Series, 2001. Appeared in *Scheduling.*

Scientific Publications—Unpublished Papers and Conference Presentations

1. "A generalization in the calculation of equipment reliability". Technical Report No. 29, Signal Corps Tube Analysis Program, Cornell University, Ithaca, N.Y.; November 15, 1956.
2. "On the replacement policy of electron tubes". Technical Report No. 30, Signal Corps Tube Analysis Program, Cornell University, Ithaca, N.Y.; November 15, 1956.
3. "Smooth production patterns: A linear programming formulation". Paper presented to the Regional Conference of APICS; April, 1963.
4. "An algorithm for the solution of the 'Zero-One' problem of integer linear programming". Research Memorandum, Yale University; 1963.
5. "On the treatment of stock cutting problems as Diophantine programs". OR Report No. 61, N.C. State University, May 11, 1970; with N. K. Wig. Presented at the 7th International Symposium on Mathematical Programming, The Hague, Holland; September 14–18; 1970.
6. "An empirical investigation in the structure of optimal batch ordering policies". OR Report No. 73, N.C. State University, Raleigh; with V. Bawle; July 26, 1971.
7. "Automation of the design-manufacturing process". Invited paper delivered at the International Design Automation Conference of ASME, Toronto, Canada; September 8–10, 1971.
8. "Some recent developments in aggregate production planning and scheduling—A Bibliography", OR Report No. 85, January 3, 1973. Presented at 42nd ORSA Meeting, Atlantic City, N.J.; November 8–10, 1972.
9. "Scheduling a single facility under constant demand and fixed production rate". OR Report No. 87, October 10, 1975 (revised); with Lynn P. Dix. Presented at the Mathematische Centrum, Amsterdam, Holland; November, 1974.
10. "The scheduling of jobs on parallel processors: A survey and annotated bibliography".OR Report No. 97; with A. N. Elshafei. Paper presented at the Logistics Research Conference, George Washington University; May 8–10, 1974.
11. "On scheduling *n* products on a single facility under constant demand and production rates". OR Report No. 104, with Bajis Dodin, June, 1975. Presented to the ORSA/TIMS Scheduling Conference, Orlando, Florida; February 4–6, 1976.
12. "A Q-GERTS simulation study of the faculty population in a university". OR Report No. 114, with Muna Bazzi; June, 1977.

13. "The multiple processor makespan problem: A branch-and-bound approach". OR Report No. 137, with A. A. Elimam; September, 1978.
14. "Scheduling of line digraphs on a single processor". OR Report No. 145, with Subhash C. Sarin; June, 1979.
15. "A note on optimal Chebychev approximations in mathematical programming". OR Report No. 156; June, 1980.
16. "Irreducibility of acyclic digraphs". OR Report No. 154, with B. Dodin, 1979. Revised November; 1981.
17. "Procedures for heuristic scheduling under limited resources in activity networks". OR Report No. 178; with Z. M. Naman, June 1981.
18. "Approximating criticality indices in PERT networks: Summary results". OR Report No. 174; with B. Dodin, April 1981.
19. "On the complete reduction of directed acyclic graphs". OR Report No. 197; with A. H. Colby, May 1984.
20. "Maximum flow in generalized networks". OR Report No. 202; with P. S. Pulat, June 1984.
21. "The Taguchi approach to quality control and enhancement: A Primer". OR Report No. 213; with Y. Fathi and W. G. Ferrell, November 1986.
22. "The Tramp Ship scheduling problem". OR Report No. 262; with David Hurchalla, January 1992.
23. "The sequencing of orders on picking stations: An empirical investigation of the Christofides heuristic". OR Report No. 274; with Stephanie J. Reed and Thomas T. Honeycutt, June 1993.
24. "Sequencing on multiple processors: An alternative approach". OR Report No. 273; with Alain Guinet and Keith W. Schellenberger, August 1993.
25. "An optimal procedure for the discrete time-cost trade-off problem in project networks"; with Erik Demeulemeester and Willy S. Herroelen, 1993.
26. "Approaching the ELSP via simulated annealing". OR Report No. 301; with Mani K. Agrawal, March 1995.
27. "Control net bounds on the expected makespan of the two-machine flowshop problem". OR Report No. 306, June 1995.
28. "Comments on activity networks generation". OR Report No. 314; with Mani K. Agrawal, March 1996.
29. "ATM network topological design: Heuristic and lower Bound". OR Report No. 318, March 1996. Paper presented at 4th International Conference on Telecommunication Systems Modeling and Analysis, Nashville, TN; with S. Zhang and G. L. Bilbro, March 21–24, 1996.
30. "ATM network topological design and network modification"; with S. Zhang-Lo and G. L. Bilbro, March 1997.
31. "Adaptive Resource Allocation in Multimodal Activity Networks". *Paper presented at Optimization 2001*, Aveiro, Portugal. Extended abstract published in the conference *Proceedings*; with Tereso, A. P. and Araújo, M. M., July 22–25, (2001).
32. "Sequencing to minimize the weighted completion time subject to 'dedicated' resources and arbitrary precedence". *Paper presented at PMS'02*, Valencia, Spain,

and appearing in extended abstract form in the *Proceedings* of the conference; with S. Balisetti, April 3–5, (2002).

33. "Experimental Results of an Adaptive Resource Allocation Technique to Stochastic Multimodal Projects". *Paper presented at the International Conference on Industrial Engineering and Production Management (IEPM'03)*, Porto, Portugal. Full paper was published in the *Proceedings*; with Tereso, A. P. and Araújo, M. M., May 26–28, (2003).
34. "Basic Approximations to an Adaptive Resource Allocation Technique to Stochastic Multimodal Projects". *Paper presented at EURO/INFORMS Joint International Meeting, Istanbul, Turkey*. Extended abstract published in the conference *Proceedings*; with Tereso, A. P. and Araújo, M. M., July 6–10, (2003).
35. "A global optimum search scheme for the economic lot scheduling problem under power-of-two policy." *Paper presented at the Fourth International Conference on Operations and Quantitative Management (ICOQM-IV)*; with M-J. Yao, January 2–5, (2003).
36. "On the transformation of non-series/parallel graphs into series/parallel graphs through augmentation". *Research Report, NCSU*; with Ramachandra, G., April (2004).
37. "The Optimal Resource Allocation in Stochastic Activity Networks via the Electromagnetism Approach". *Paper presented at Ninth International Workshop on Project Management and Scheduling (PMS'04),* Nancy, France. Extended abstract published in the proceedings; with Tereso, A. P. and Araújo, M. M., April 26–28, (2004).
38. "Project Management: multiple resources allocation". *Paper presented at the International Conference on Engineering Optimization (EngOpt2008)*, Rio de Janeiro, Brazil; with Tereso, A. P., Araújo, M. M. and Moutinho, R., June 1–5, (2008).
39. "Project Management: Multiple Resources Allocation," *Paper presented at IESM09*, Montreal, Canada, and appearing in extended abstract form in the conference *Proceedings*; with Tereso, A. P., Arajo, M. M. and Moutinho, R., May 13–15, (2009).
40. "The Optimal Resource Allocation in Stochastic Activity Networks via Continuous Time Markov Chains," *Paper presented at IESM09*, Montreal, Canada, May 13–15, and appearing in extended abstract form in the conference *Proceedings*; with Adam J. Rudolph, (2009).
41. "Approximation of continuous distribution via the Generalized Erlang Distribution," *Paper presented at INCOM09,* Moscow, and appearing in extended abstract form in the conference *Proceedings*; with Benmansour, R., Artiba, A., and Allaoui, H.; (2009).
42. "Quantity-Oriented Resource Allocation Strategy on Multiple Resources Projects under Stochastic Conditions", *International Conference on Industrial Engineering and Systems Management (IESM' 2009),* Montreal—Canada; with Tereso, A. P., Araújo, M. M. and Moutinho, R., May 13–15 (2009).

43. "Duration-Oriented Resource Allocation Strategy on Multiple Resources Projects under Stochastic Conditions", (2009) *International Conference on Industrial Engineering and Systems Management (IESM' 2009)*, Montreal—Canada; with Tereso, A. P., Araújo, M. M. and Moutinho, R., May 13–15 (2009).
44. "Optimal resource allocation in activity networks: I. The deterministic case," *Research Report, NCSU*; with Ramachandra, G.; (2012).
45. "Optimal resource allocation in activity networks: II. The stochastic case," *Research Report, NCSU*; with Ramachandra, G.; (2012).
46. "On The Optimal Resource Allocation in Projects Considering the Time Value of Money", *Third International Conference on Information Systems, Logistics and Supply Chain (ILS 2010)*, Casablanca—Morocco, April 13–16. Co-authored with Anabela Tereso, Duarte Barreiro, and Madalena Araújo,
47. "Optimization of run-Time mapping on heterogeneous architectures CPU/FPGA," *Paper presented at MOSIM* (2012), 9th International Conference on Modeling, Optimization & Simulation: Performance, Interoperability and Safety for Sustainable Development, 6–8 June, 2012 IMS laboratory, University of Bordeaux, France. Co-authored with Suissi, O., BenAttietalla, R., and Artiba, A.
48. "The Relevance of the "Alphorn of Uncertainty" to the Financial Management of Project Under Uncertainty;" *Research Report, NCSU*; with Jingwen Zhang, (2012).
49. "On Consistency and Feasibility of Generalized Precedence Relations (GPR's) in Activity Networks"; *Research Report, NCSU*; with Jingwen Zhang, (2012).

Personal Data

Place and Date of Birth:	Fayoum, Egypt, October 21, 1927
Marital Status:	Married, three children
Address:	3604 Ranlo Drive, Raleigh, NC 27612 (Tel: (919) 787–0855)
e-mail:	elmaghra@eos.ncsu.edu

Languages

Arabic (fluently, mother tongue)
English (fluently)
French (conversational and reading)
German (a smattering).

Index

P. S. Pulat et al. (eds.), *Essays in Production, Project Planning and Scheduling,*
International Series in Operations Research & Management Science 200,
DOI 10.1007/978-1-4614-9056-2, © Springer Science+Business Media New York 2014